

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

**Lecture - 07**  
**Basics of IoT Networking- Part- III**

So, we continue with our discussions about the different protocols that are used for communication and networking of internet of things.

(Refer Slide Time: 00:34)



So, the next protocol that we are going to cover is the CoAP protocol and the full form of which is Constrained Application Protocol.

(Refer Slide Time: 00:37)

**Introduction**

- ✓ CoAP – **Constrained Application Protocol**.
- ✓ **Web transfer protocol** for use with constrained nodes and networks.
- ✓ **Designed for Machine to Machine (M2M)** applications such as smart energy and building automation.
- ✓ Based on **Request-Response model** between end-points
- ✓ Client-Server interaction is **asynchronous over a datagram oriented transport protocol** such as UDP

Source: Z. Shelby, K. Hartke, C. Bormann, "The Constrained Application Protocol (CoAP)", Internet Engineering Task Force (IETF), Standards Track, 2014

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

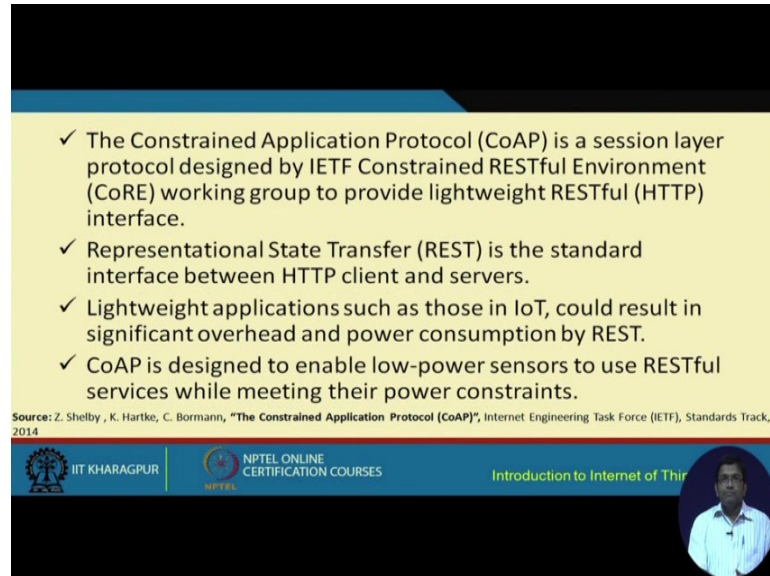
This protocol is particularly used for web transfer and by web transfer I mean very similar to the HTTP, but web transfer in the context of constrained networks resource, constrained networks with nodes which are constrained with respect to different resources, such as limited energy or power supply, limited computational resources, limited communication resource, limited bandwidth environment and so on.

So, CoAP is sort of like an HTTP equivalent that can be used in the context of IoT and the other thing that we have to understand is CoAP is strictly speaking a session layer protocol. However, we can also contribute as an application layer protocol as well. So, in IoT particularly when we taking to consideration the different applications involving machine to machine communication for example, smart energy, smart environment you know building automation and this kind of applications CoAP comes out to be very much useful.

CoAP is based on a request response model. So, basically you know it will very clear shortly about how CoAP works. So, at this point you know you have to understand that there are two endpoints, the source and the destination and a request is sent and a response is received back from the in the other end point and that means, the destination. So, this is how CoAP works. So, there is sort of like a client server, kind of interaction that goes on. So, there is the datagram that is sent from one endpoint to another and that

basically, it is an asynchronous kind of communication and it also works on top of in the transport protocol UDP.


(Refer Slide Time: 02:55)



- ✓ The Constrained Application Protocol (CoAP) is a session layer protocol designed by IETF Constrained RESTful Environment (CoRE) working group to provide lightweight RESTful (HTTP) interface.
- ✓ Representational State Transfer (REST) is the standard interface between HTTP client and servers.
- ✓ Lightweight applications such as those in IoT, could result in significant overhead and power consumption by REST.
- ✓ CoAP is designed to enable low-power sensors to use RESTful services while meeting their power constraints.

Source: Z. Shelby, K. Hartke, C. Bormann, "The Constrained Application Protocol (CoAP)", Internet Engineering Task Force (IETF), Standards Track, 2014.

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



So, basically CoAP, we have to keep in mind that CoAP works on top of UDP. So, this particular protocol is based on IETF RESTful Environment Working Group. So, there is architecture, restful architecture and these people who have to post the restful architecture, they have proposed the CoAP protocol. So, it is basically used sort of like a lightweight equivalent of the HTTP and it is a standard REST. So, let us go back to the rest protocol first.

So, REST is a standard interface between the HTTP clients and servers, but this REST protocol is useful where there is no resource limitation because you know this is quite resource hungry kind of protocol which consumes lot of resource. REST basically is not good for constrained environments like IoT.

So, CoAP is sort of like a protocol which is a lightweight equivalent of the rest architecture and rest protocol and it helps to communicate with under low power constraints.

(Refer Slide Time: 04:19)

- ✓ Built over UDP, instead of TCP (which is commonly used with HTTP) and has a light mechanism to provide reliability.
- ✓ CoAP architecture is divided into two main sub-layers:
  - Messaging
  - Request/response.
- ✓ The messaging sub-layer is responsible for reliability and duplication of messages, while the request/response sub-layer is responsible for communication.
- ✓ CoAP has four messaging modes:
  - Confirmable
  - Non-confirmable
  - Piggyback
  - Separate

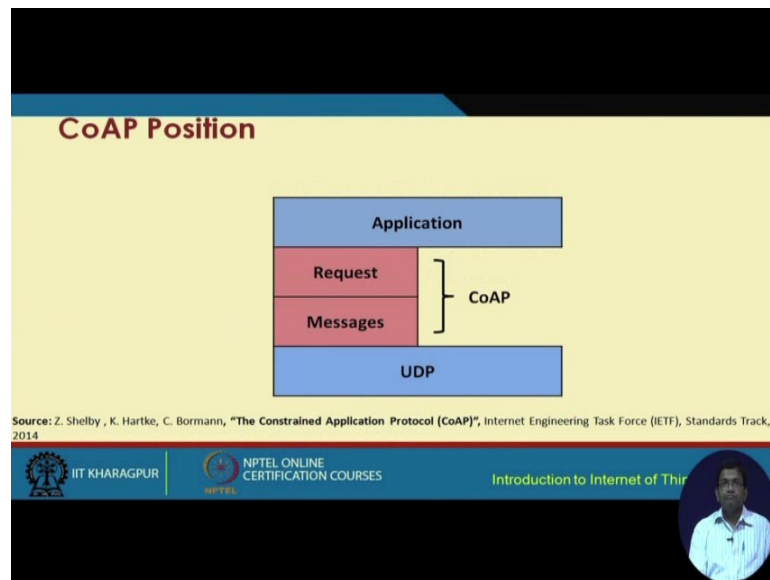
Source: V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. Alonso-Zarate, "A survey on application layer protocols for the internet of things," Transaction on IoT and Cloud Computing, vol. 3, no. 1, pp. 11-17, 2015

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

So, this is how CoAP works. So, as I was telling you that it is a session layer protocol or even we can think of it as an application layer protocol. So, it basically works on top of the transport layer. So, session layer or application layer are on top of the transport layer and the transport layer protocol that is used in the context of CoAP is that UDP. So, CoAP basically has two main sub-layers, one is the messaging sub-layer and the other one is the request response sub-layer and I will show you pictorially how it looks like shortly.

So, in summary actually at a high level, we can think of the messaging sub-layer to be responsible for functionality, such as reliability and duplication of avoidance of duplication of messages while the request response sub-layer is responsible for the communication, exact communication that is going to take place, the request being sent and the response being received. So, these are two main sub-layers that are there in the CoAP protocol or the CoAP architecture more specifically. So, there are different messaging modes for CoAP, one is the confirmable mode, the second is the non-confirmable mode, the third is the piggyback mode and the fourth is the separate.

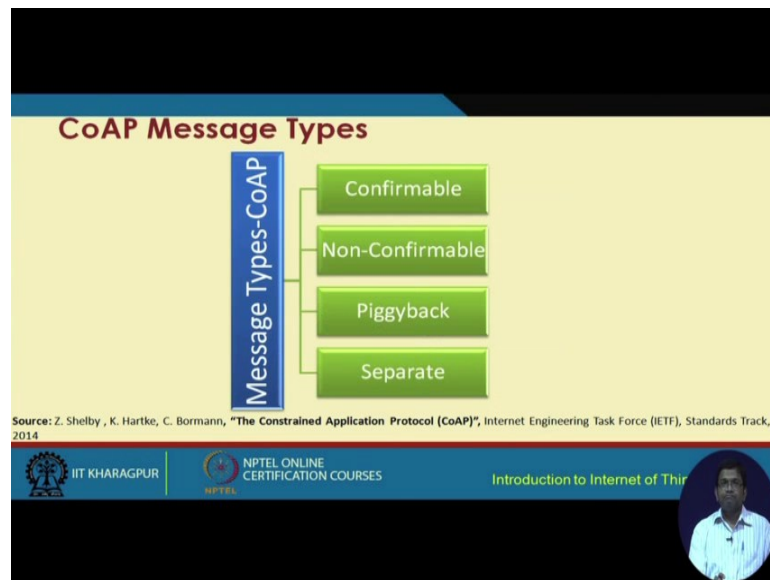
(Refer Slide Time: 05:41)



So, as I was telling you in the protocol stack in terms of the layered architecture, CoAP is a protocol of the session layer. Some can also think of it you know in some cases we try to avoid the session layer. So, in that case, we can think of CoAP to be merged with the application layer, but if session layer is considered, it is a protocol of the session layer. So, session layer means that it lies between the transport layer and the application layer.

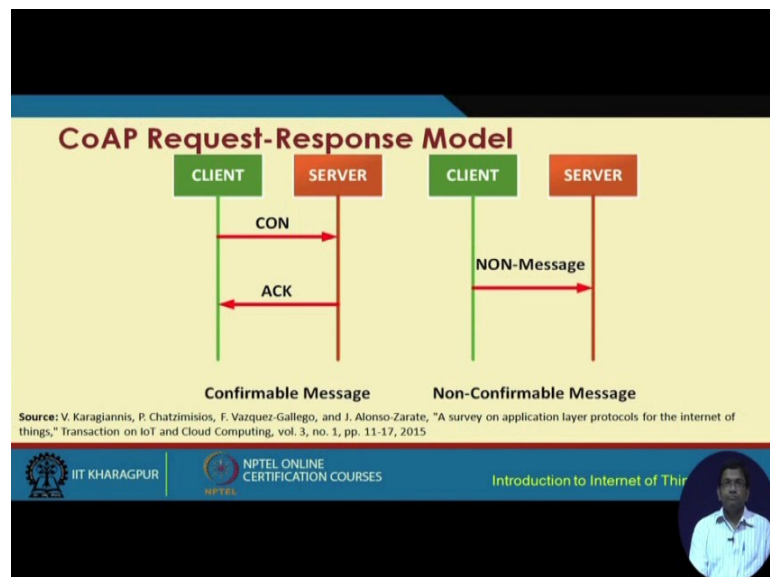
So, at the transport layer, we have the UDP protocol and different applications being run in the application layer and CoAP basically sits in between. So, we have two sub-layers, one is the request response and the other one is the messages. Messages is mostly concerned about the reliability in sharing, reliability of the network, reliability in communication whereas, request response is more to do with the exact communication in terms of sending a request and getting a response back.

(Refer Slide Time: 06:49)



So, this is what I was mentioning earlier. So, we have different types of messages that are used in CoAP. The first one is the confirmable message, the second is the non-confirmable message, the third is the piggyback message and the fourth is the separate message.

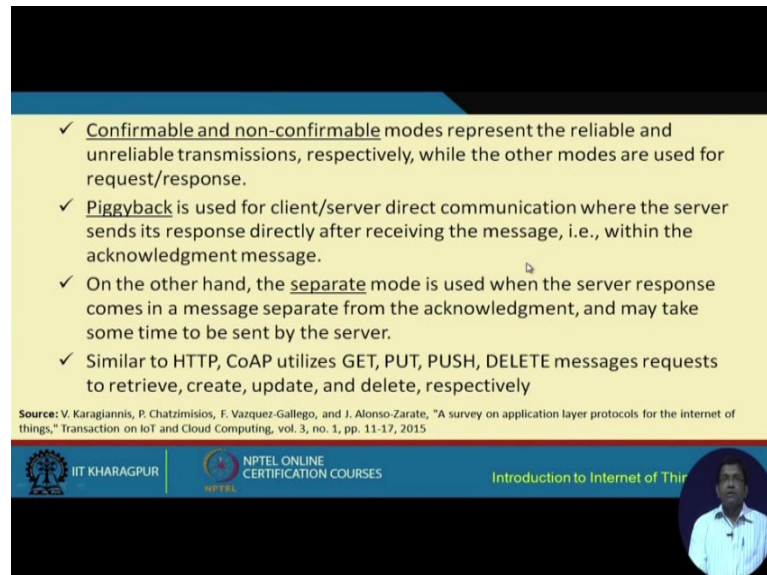
(Refer Slide Time: 07:11)



So, basically when we look at CoAP confirmable message, this is how it works. So, we have CoAP is basically a connection between the client and the server in a resource

constraint environment. So, what happens is a message is sent and an acknowledgement is received in the case of a confirmable message.

(Refer Slide Time: 07:48)



✓ Confirmable and non-confirmable modes represent the reliable and unreliable transmissions, respectively, while the other modes are used for request/response.


✓ Piggyback is used for client/server direct communication where the server sends its response directly after receiving the message, i.e., within the acknowledgment message.

✓ On the other hand, the separate mode is used when the server response comes in a message separate from the acknowledgment, and may take some time to be sent by the server.

✓ Similar to HTTP, CoAP utilizes GET, PUT, PUSH, DELETE messages requests to retrieve, create, update, and delete, respectively

Source: V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. Alonso-Zarate, "A survey on application layer protocols for the internet of things," Transaction on IoT and Cloud Computing, vol. 3, no. 1, pp. 11-17, 2015

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



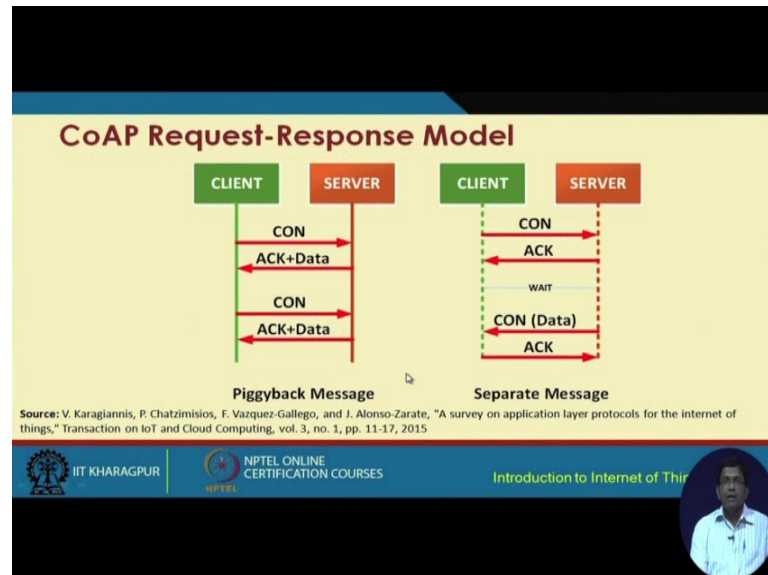
So, this message basically you know it gets an acknowledgement back. So, it is a confirmable message and then, for non-confirmable message, there is no acknowledgement from the server and then, we have the piggyback message which is used for a client server direct communication where the server sends its response directly after receiving the message. So, that basically you know what happens is along with the acknowledgement message, the data is also sent, the response is also sent in the case of piggyback messages.

And in the case of the separate mode, it is used when the server response comes in a message separate from the acknowledgement and that basically may take some time to be sent to the server, this particular message might you know because it is coming separate from the acknowledgement. May be acknowledgement might be received and the message might be received after a rewind back.

So, no sorry find thereafter I am sorry and so, similar to HTTP code basically utilizes different functionalities, such as get functionality, get message, put message, push message, delete message etcetera. So, basically get is for retrieval of some data, put is for creation. So, you want to put some data or some message into the repository, so in that

case push or put the server. So, in that case that put message is used and then, we have the update for that push is used and the delete message is for deletion purpose.

(Refer Slide Time: 09:30)

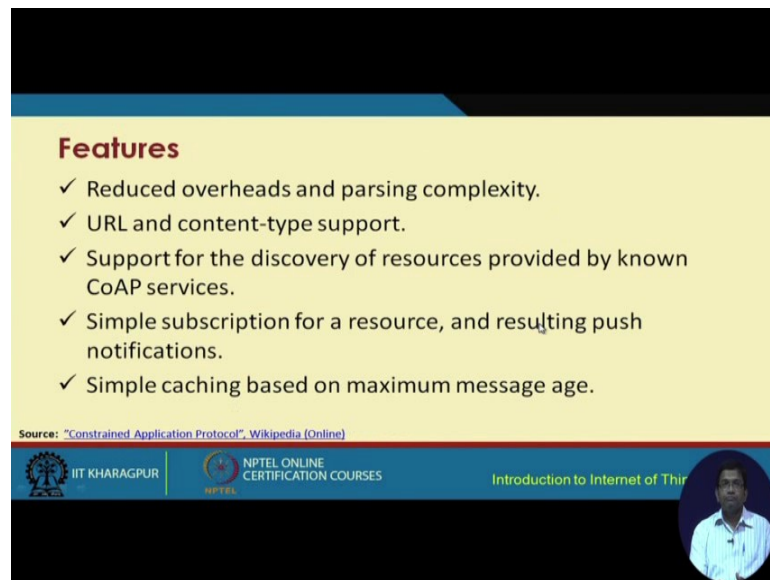


So, we have already looked at how the confirmable and the non-confirmable message request response looks like. So, let us now look at pictorially how the piggyback message request response model looks like. So, here basically as we can see first a message is sent in piggyback in contrast to the previous two models. That means, the confirmable and the non-confirmable models, rather the confirmable model. So, what we have vary as we can see the data is basically piggybacked along with the acknowledgement message. So, this is how the piggyback message request response model functions.

Separate message, we have a message being sent and acknowledgement being received and there is a wait period after which the data is going to be sent separately from the server to the client and corresponding to that the client is going to send an acknowledgement back to the server.



(Refer Slide Time: 10:52)




**Features**

- ✓ Reduced overheads and parsing complexity.
- ✓ URL and content-type support.
- ✓ Support for the discovery of resources provided by known CoAP services.
- ✓ Simple subscription for a resource, and resulting push notifications.
- ✓ Simple caching based on maximum message age.

Source: "Constrained Application Protocol", Wikipedia (Online)

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



So, this is how separate messages look like so these basically CoAP as a whole and these different message types, they together help to induce the overhead and the parsing complexity of the network. So, there are different types of discovery of resources that are supported by CoAP and we are going to go through them little bit further.

(Refer Slide Time: 11:23)



**XMPP**

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



So, but so we now start with the XMPP protocol which is the next protocol to be discussed.

(Refer Slide Time: 11:34)

### Introduction

- ✓ **XMPP – Extensible Messaging and Presence Protocol.**
- ✓ A communication protocol for **message-oriented middleware** based on XML (Extensible Markup Language).
- ✓ Real-time exchange of structured data.
- ✓ It is an open standard protocol.

Source: "XMPP", Wikipedia (Online)

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

The full form of XMPP is Extensible Messaging and Presence Protocol. So, it is a message oriented middleware that is based on XML, whereas XML is particularly used for unstructured data. XMPP is useful for real time exchange of structured data and it is an open standard protocol.

(Refer Slide Time: 12:03)

- ✓ XMPP uses a **client-server architecture.**
- ✓ As the model is **decentralized**, no central server is required.
- ✓ XMPP provides for the discovery of services residing locally or across a network, and the **availability information** of these services.
- ✓ Well-suited for cloud computing where virtual machines, networks, and firewalls would otherwise present obstacles to alternative service discovery and presence-based solutions.
- ✓ Open means to support machine-to-machine or peer-to-peer communications across a diverse set of networks.

Source: "XMPP", Wikipedia (Online)

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

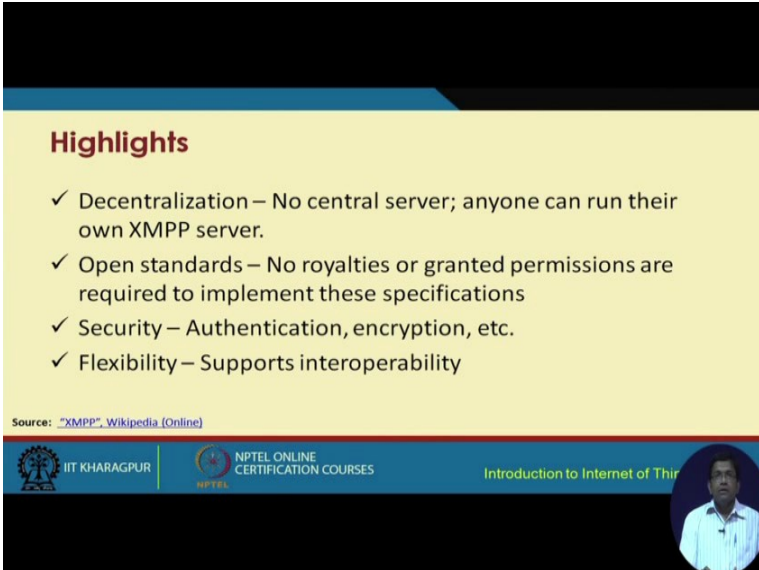
So, XMPP uses a client server architecture, it uses a decentralized model meaning that there is no server that is involved in the message transfer and it provides facilities for

discovery of messages which are residing locally or globally across the network and the availability information of these services.

So, as we can now basically think about it, so it is well suited for cloud computing environments, where virtual machines networks and firewalls are involved and would otherwise present obstacles to the alternative service discovery and message based solutions. So, you know think of it this way that with the help of XMPP, we can do things very similar to like pin protocol. So, in the case of pin, basically when we have the involvement of firewalls etcetera, so pin cannot be used as such, right.

So, in this particular case, in the case of XMPP, it basically removes all these constraints, these barriers for having the discovery of the services and if it is the discovery of services locally, then it is no problem, but if it is across the network and there is a firewall in between, then XMPP can still work.

(Refer Slide Time: 13:34)



**Highlights**

- ✓ Decentralization – No central server; anyone can run their own XMPP server.
- ✓ Open standards – No royalties or granted permissions are required to implement these specifications
- ✓ Security – Authentication, encryption, etc.
- ✓ Flexibility – Supports interoperability

Source: "XMPP", Wikipedia [Online]

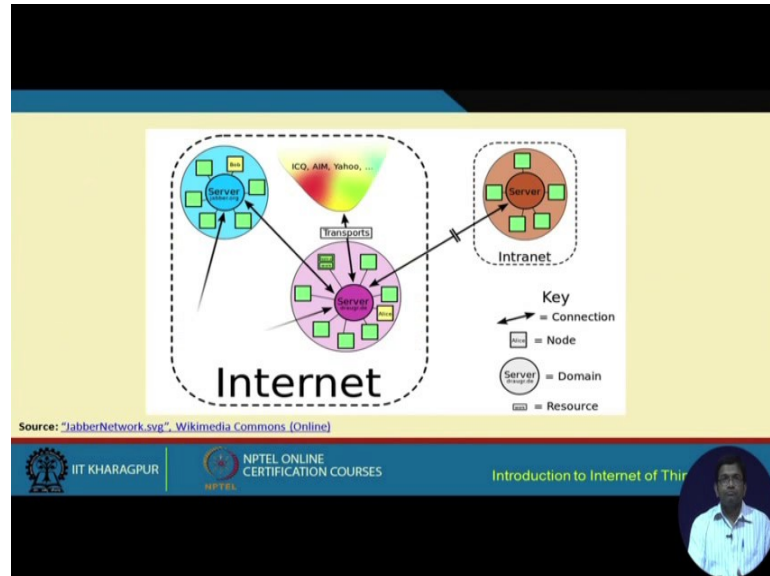
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Introduction to Internet of Things

The slide features a yellow background for the highlights section. At the bottom, there is a blue banner with logos for IIT Khharagpur and NPTEL, along with the course title 'Introduction to Internet of Things'. A small circular inset in the bottom right corner shows a man in a white shirt.

So, some of these highlights of the XMPP protocol, it is based on the concept of decentralization where there is no central server and then, you know everybody can run the XMPP server theoretically and it is based on open standard. So, there is no involvement of royalties or granting permissions to implement the XMPP specifications, different security features that the standard ones, such as authentication, encryption, etcetera, can be implemented using XMPP on top of XMPP rather and XMPP also offers

flexibility in terms of supporting interoperability between different systems, different devices, different protocols, and so on.

(Refer Slide Time: 14:26)



So, consequently I was giving you the analogy with the traditional pin protocol that is used for internet and here we are trying to have something similar, but you know it is bit different in this particular manner. So, it is now if you look at this particular figure what we see is with the help of XMPP, not only it is possible to communicate with other servers like in the case of the traditional internet, but also with other messaging platform such as ICQ, AIM, Yahoo and so on. So, this is also possible. So, not only that this is possible, but additionally it is also possible to communicate with other intranets, other intranets.

(Refer Slide Time: 12:25)

**Core XMPP Technologies**

- Core**
  - information about the core XMPP technologies for XML streaming
- Jingle**
  - multimedia signalling for voice, video, file transfer
- Multi-user Chat**
  - flexible, multi-party communication
- PubSub**
  - alerts and notifications for data syndication
- BOSH**
  - HTTP binding for XMPP

Source: "XMPP: Technology Overview", XMPP.org (Online)

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

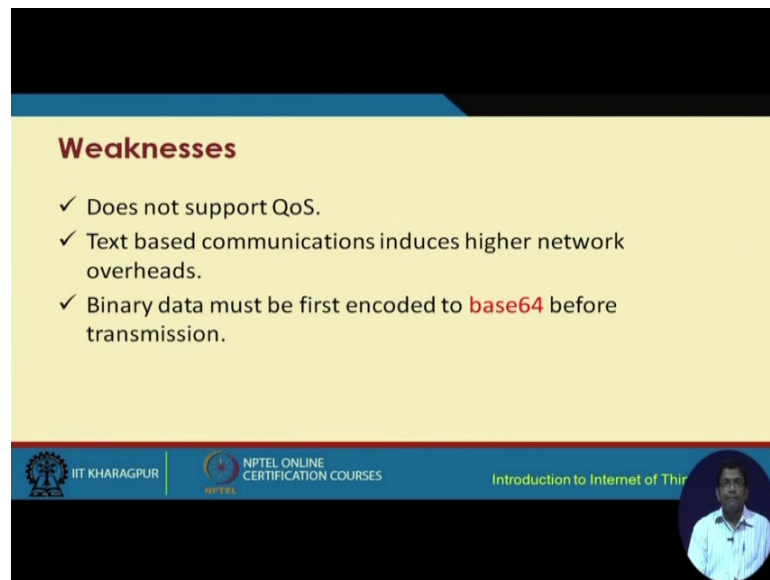
NPTEL

Introduction to Internet of Things

So, XMPP basically helps in doing this. There are few core XMPP technologies, one is the core technology which provides information about the core XMPP technologies for XML streaming, then we have jingle which is used for multimedia signaling with the help of voice you know wherever there is multimedia resources, such as voice, video, file transfer etcetera, it can help in signaling jingle multi-user chat.

It is a flexible technology which can be used for multi-party communication. Pub sub is Publish Subscriber. Publish subscribe model and publish basically alerts, this pub sub model basically alerts and notifies for data syndication and the BOSH technology. It is used for HTTP, binding for XMPP wherever there is required meant for HTTP binding when using XMPP, this can be used.


(Refer Slide Time: 16:23)



**Weaknesses**

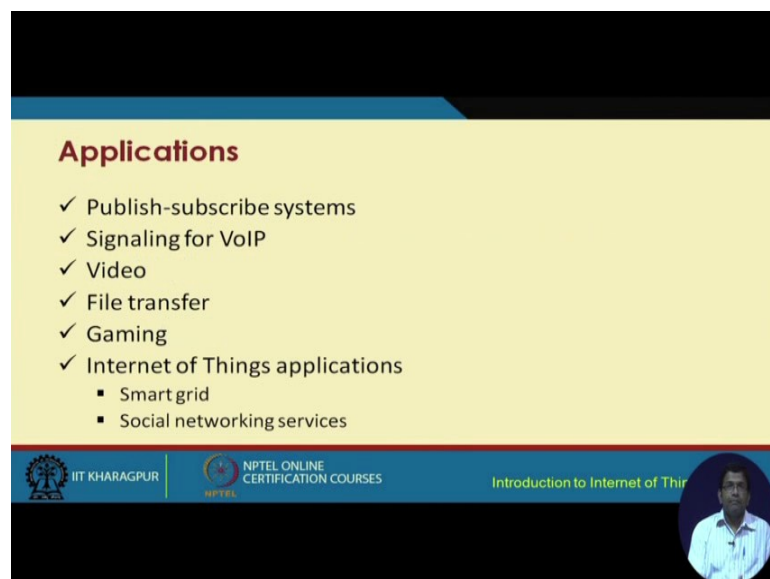
- ✓ Does not support QoS.
- ✓ Text based communications induces higher network overheads.
- ✓ Binary data must be first encoded to **base64** before transmission.

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



There are different weaknesses as well of XMPP protocol. It does not support QoS, text based communication including you know Higher Network Overheads are involved in the use of XMPP. So, it is not good for text based communication.


(Refer Slide Time: 16:56)



**Applications**

- ✓ Publish-subscribe systems
- ✓ Signaling for VoIP
- ✓ Video
- ✓ File transfer
- ✓ Gaming
- ✓ Internet of Things applications
  - Smart grid
  - Social networking services

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



Binary data must be first encoded to base 64 before it can be transmitted. The different applications that use XMPP publish subscribe systems, pub sub systems, then signaling for voice video file transfer, gaming applications, IoT applications such as smart, grid, social networking and so on.

So, with this we have come to the end of two order protocols. We have discussed that two order protocols. So, XMPP is a protocol that is very useful. So, we have discussed about the CoAP protocol first with the session layer protocol which is useful for use in a similar kind of platform, where rest is required for communication between the client and the server. So, CoAP protocol and then, we have discussed about the XMPP protocol.

Thank you.