Introduction to Internet of Things

Prof. Sudip Misra
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

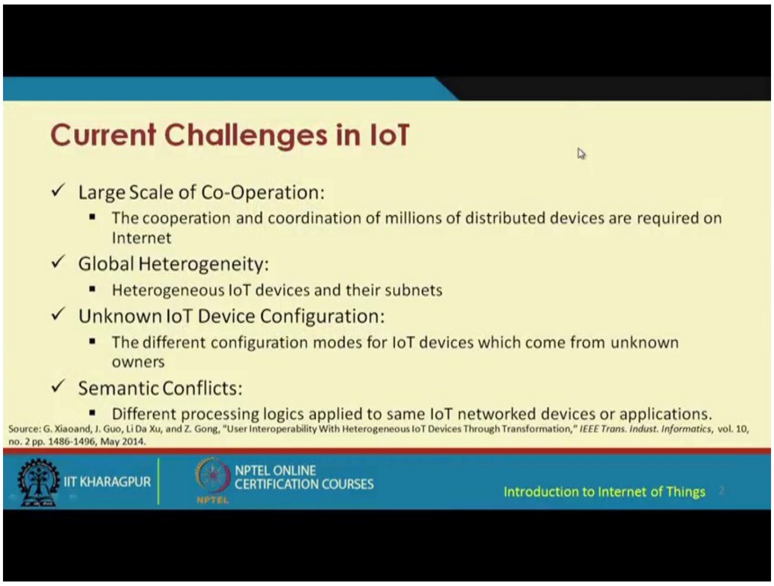
Lecture - 21
Interoperability in Internet of Things

This topic is on interoperability in internet of things. Internet of things uses different types of devices. These devices are made by different vendors following different specifications; there is no one standard for IOT. So, consequently what happens is for different things the different IoT devices are made by different vendors following different specifications. Again these different devices by different vendors they follow different protocols not necessarily that they all follow the same protocol. Even the kind of users their user profiles these can also be different. So, there is so much of diversity that is inherent to these systems IoT systems and that is why it is very important to address this particular issue.

In internet of things one of the core problems or issues that has been studied quite extensively is heterogeneity of devices, protocols, user groups and many other heterogeneity aspects in from different other angles. So, this has been studied quite extensively. And one of the requirements to handle this heterogeneity issue is basically to have some kind of interoperability interoperability between these different heterogeneous aspects. Interoperability means what that let us say that one particular device is following a particular protocol; another device follows another protocol. So, how do they talk to each other they speak the different language they speak language. So, they do not speak the same language. So, how do they talk to each other, this is one aspect.

Similarly, at different physical levels, different specifications, different devices how do they talk to each other, they all have been made in different ways because there is no one standard that has been followed in developing these systems. So, when you want to build a singular IoT system comprising of all these different heterogeneous objects, devices, protocols, standards etcetera you need to have some kind of handshaking. And that handshaking is where these protocols have been devised which can be some kind of a middleman a middleware rather which can help these two different diverse groups to be able to talk to each other.

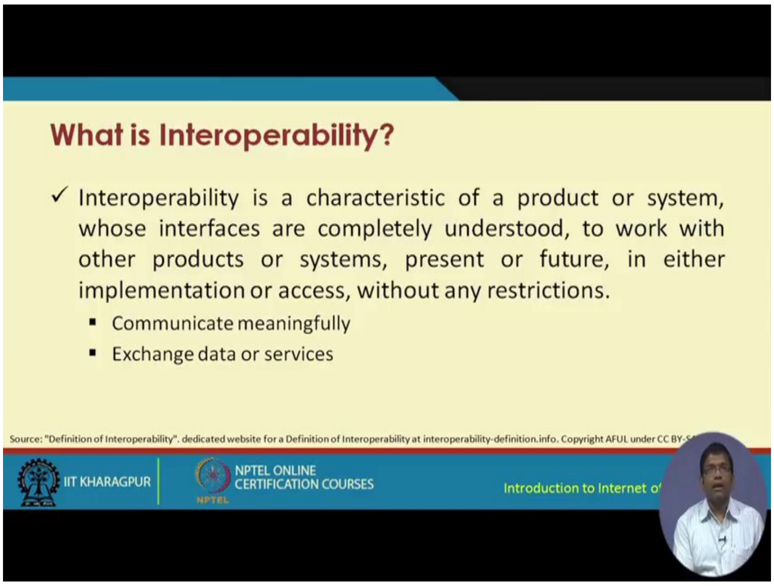
(Refer Slide Time: 03:30)



So, let us try to understand the interoperable interoperability issue in further detail. So, when we talk about IoT we are talking about large scale networks. Large scale networks requiring the use of large number of different devices millions and billions of devices, these devices are distributed all across in the internet or all across in the world. And what is required is to have some kind of cooperation between the different devices, some kind of coordination mechanism to be enforced between these different devices to be able to talk to each other. So, this is one issue.

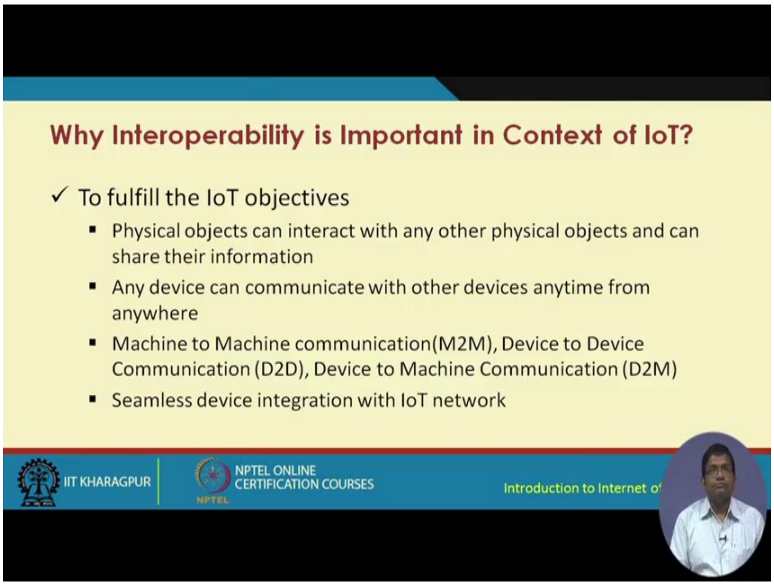
Second issue is that the devices as I was telling you before they all have been designed with different specifications heterogeneous in all respects in the physical device level, in the protocol level, user level, so in all different aspects. So, heterogeneous IoT devices and their subnets is a challenge that has to be worked on in the context of internet of things. Another very typical concern that is specific of IoT is the device configuration. Typically these IoT devices their configuration is unknown, unknown all across. The different configuration modes for IoT devices come from unknown owners and that is you know inherently that brings in lot of complexity and that has to be handled. Another very interesting complexity is how do you handle handle differences in semantics how do you handle differences in semantics not only so there could be conflicting semantics as well. So, different processing logics are applied to the same IoT network devices or applications by different developers, different user groups and so on. So, how do you handle these differences this conflicts in the semantics.

(Refer Slide Time: 05:40)



So, what is interoperability? Interoperability is a characteristic of a product or system whose interfaces are completely understood to work with other products or systems present or future in either implementation or access without any restrictions. I think this is quite understood in the backdrop of what we have discussed so far. So, what is required is to have all these different units devices protocols etcetera, etcetera that I was mentioning before to be able to communicate meaningfully. So, that there is exchange of data, exchange of services and to the user it should the interoperability has to be handled in such a manner that to the user the user should feel that he or she is getting access to the services of the IoT system in a seamless manner. The user should not have to get into how these are implemented what is the translation that is going so on and so forth.

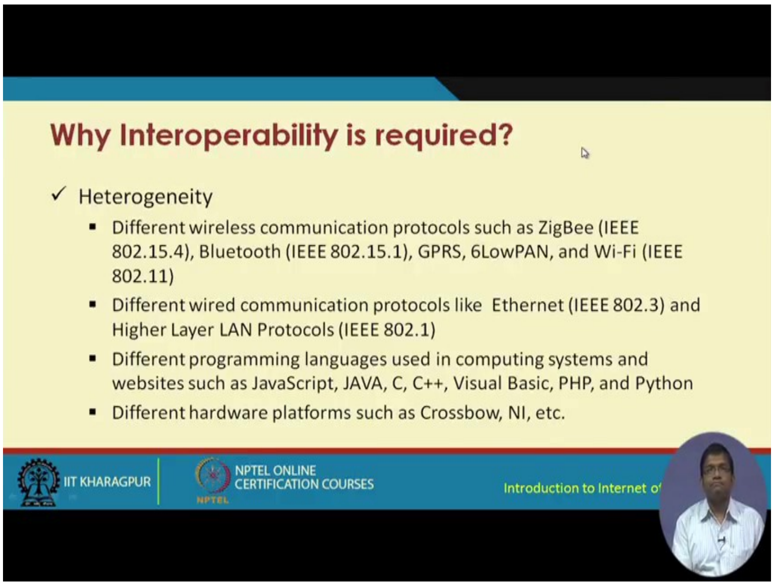
(Refer Slide Time: 06:53)



So, what why interoperability is important in the context of IoT because it is required to fulfill different IoT objectives, seamless device is what I was mentioning, physical objects would have to interact with other physical objects for sharing information, any device can communicate with any other device at any time and at anywhere. So, devices communicating anytime, anywhere and any kind of device that means, anything, anytime anywhere communication, right. So, if we have to have anything, anytime, anywhere communication then obviously we need to handle this problem, so that is the reason why heterogeneity and interoperability are the core issues of IoT that has to be addressed before large scale IoTs are made functional.

Issues such as machine to machine communication, device to device communication. So, machine to machine communication we have already gone through. And then we have the device to device communication. In device to device communication it is basically a standard in the LTE advanced and so where basically one mobile phone talk talking directly to another mobile phone you know that is addressed. And then you have the device to machine communication. So, we have M2M, D2D and D2M communication. And there has to be seamless device integration all across in the IoT network.

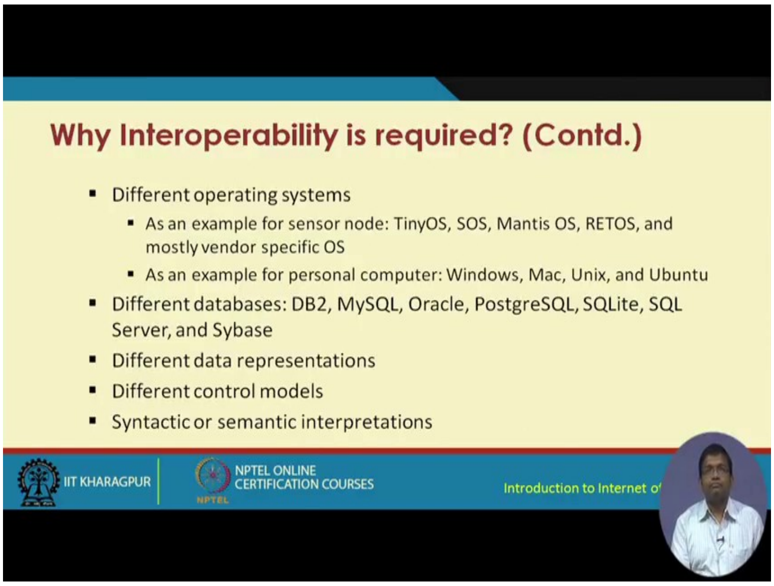
(Refer Slide Time: 08:28)



Heterogeneity - different communication protocols ZigBee following IEEE802.15.4 which we have discussed before. Bluetooth following 802.15.1, GPRS, 6LowPAN, Wi-Fi which follows 802.11 standard. So, all these different types of standards all different types of communication protocols handshaking with each other, communicating with each other. Different wired communication protocols such as 802.3 and 802.1 talking to each other that is required because otherwise we cannot have this seamless you know anytime, anywhere, any device connectivity that is not going to be possible. So, so much of heterogeneity is going to be there all across.

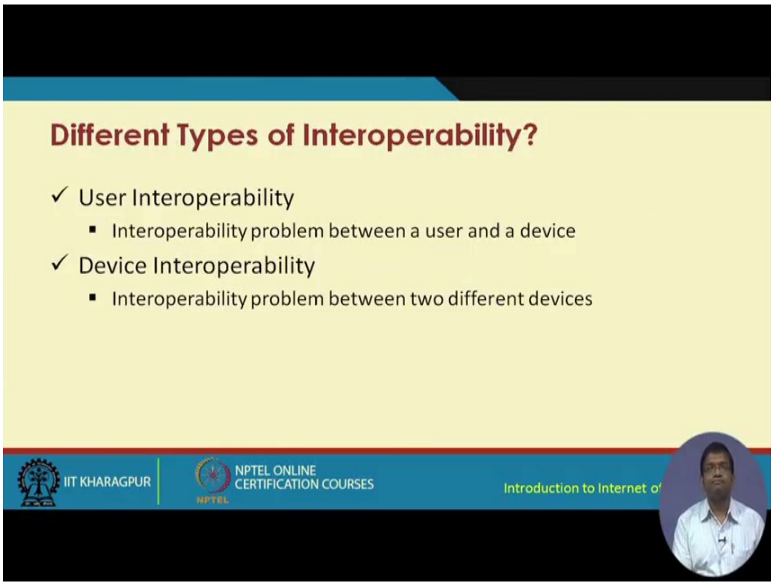
Then different programming languages are used in different computing systems for example, JavaScript in one, Java in another, C, C++, Visual Basic, PHP, python, so many different programming languages platforms of different kinds are used. Again there are different hardware platforms as well not just programming platforms in the hardware platforms such as crossbow based products talking to national instrument products talking to Cisco products and so on. So, all these hardware platforms can also be varying they you know different types of heterogeneous hardware platforms. So, interoperability is very much required in this kind of backdrop.

(Refer Slide Time: 10:06)



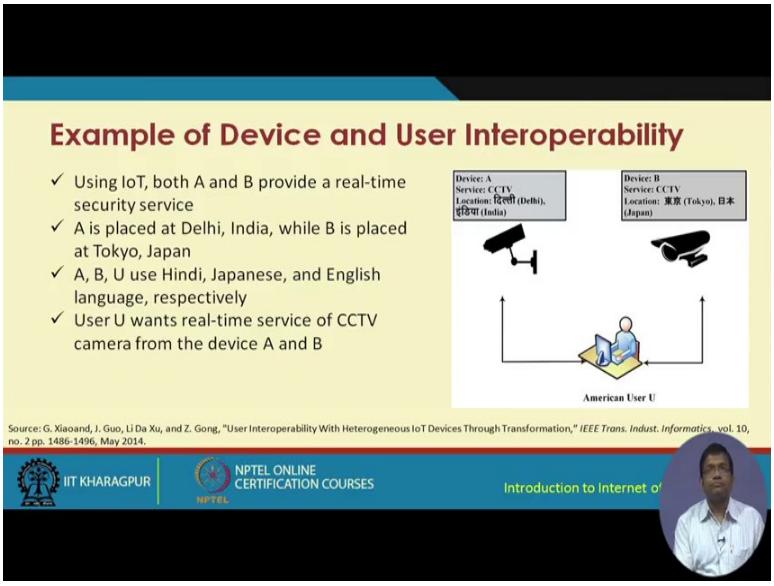
Further when we talk about operating systems for a sensor node some sensor nodes have implemented TinyOS, some SOS, some Mantis OS, some RETOS, and some could be even implementing some vendor specific operating system. And from a user point of view some PCs, for example, might be implementing might be using the Windows OS some MAC OS, some Unix, some Linux, some Ubuntu. So, all these different types of operating system from the device or sensor node point of view as well as the user point of view then different databases - DB 2, MySQL, Oracle, PostgreSQL, SQLite, SQL server, Sybase all these different, different different databases are implemented all across. Different data representations as well the way the data is represented very hard to handle this kind of heterogeneity then different control models, different syntaxes, semantics and so on. So, these have to be handled.

(Refer Slide Time: 11:19)



So, what is required is to have different types of interoperability, and these interopretability issues of different types have to be handled in different ways. User interoperability is between a user and a device. Device interoperability is between two different devices.

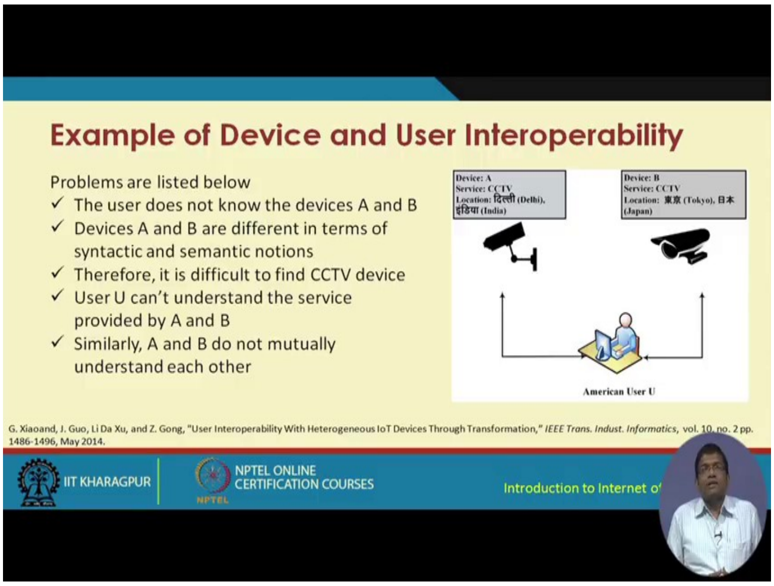
(Refer Slide Time: 11:47)



So, the issues are different user and device, device and device, so how do you handle, right. Now let us take an example to understand why user interoperability is so important. Let us say that there are two devices, one is located a CCTV for example, in this figure which is installed in Delhi. And as you can see that it is also the location it is written in different language. The other device which is another CCTV is located in Tokyo and you know it is its specification sets, not specifications, but you know the device level semantics are handled in Japanese language. And then there is an IoT user who is based in America US.

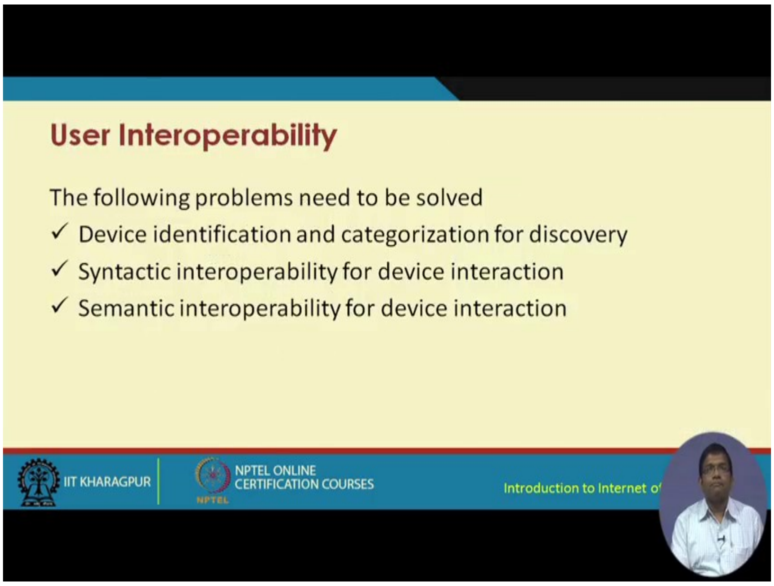
So, this American user has to operate both of these devices remotely from America and this is what IoT does, this is what IoT does that remotely how you can monitor different devices you know the end you know which could be end not only that different devices, but these devices might be themselves located quite far apart. So, using IoT both A and B provide real time security service in this particular example A is placed at Delhi, whereas B is placed in Tokyo. So, A, B and U India, Japan, America, they all use different languages Hindi, Japanese, English respectively. So, user U in America wants real-time service of CCTV camera from the devices A and B.

(Refer Slide Time: 13:45)



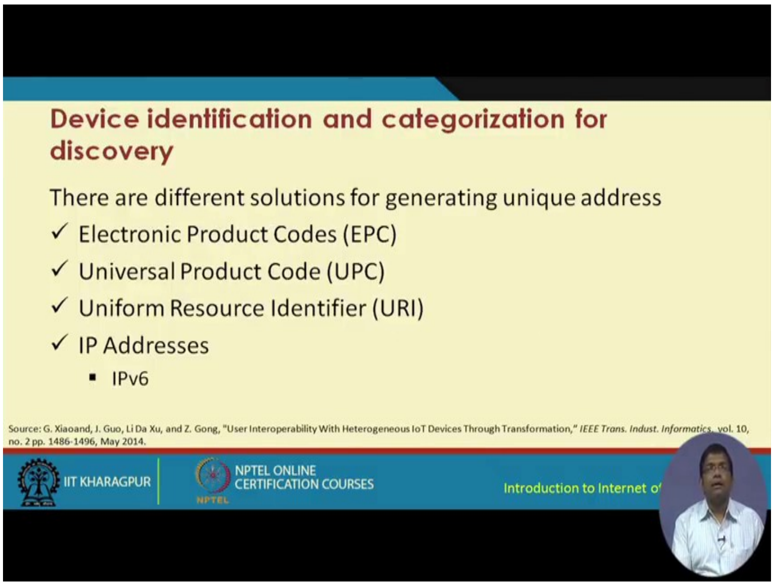
So, what are the problems that are going to happen? Number one the user does not know the devices A and B user does not know the devices A and B. Number two problem devices A and B are different in terms of syntaxes and semantics and this is this is what I was telling you just a minute back. Therefore, it is difficult to find the CCTV device user you cannot understand the service provided by A and B because of the language difference, and similarly A and B do not mutually understand each other for the same reason. So, you see that differences in syntaxes, differences in semantics, differences differences in the user specifications, all these are bringing in lot of complexity for a simple basic problem that is code to IoT.

(Refer Slide Time: 14:43)



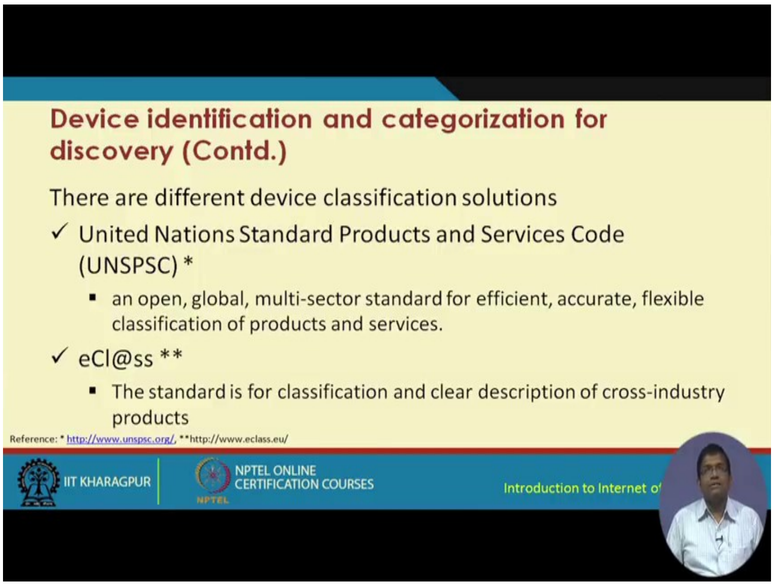
The following problems need to be solved. Number one the device has to be identified and it has to be categorized. So, device identification and categorization for discovery because for discovery you need to have need to identify the device and you need to categorize the device as what type of device whether it is a CCTV device or some other type of device. And you have to also identify the device and there is language issues between these different locations, syntactic interoperability for device integration and semantic interpretability for device interaction. So, these are the different differences in the device level, syntactic level and semantic level that have to be handled by the user. And you see that when we are talking about user interoperability, we have these different device you know different users some user you know using a different syntax, another user using another syntax or semantics or language and the corresponding knowledge basis are also different the devices are also different. So, how do these two users talk to one another this is what user interoperability tries to address.

(Refer Slide Time: 16:02)



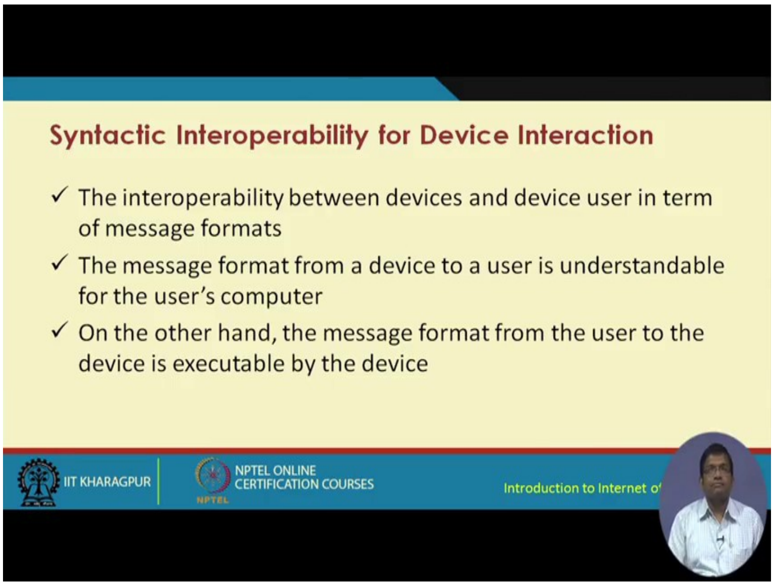
So, for device identification categorization for discovery, there are different solutions for generating unique addresses. So, these includes the EPC - electronic product code, universal product code – UPC, universal resource identifier – URI, and IPv6 the traditional IPv6 based addressing as well.

(Refer Slide Time: 16:28)



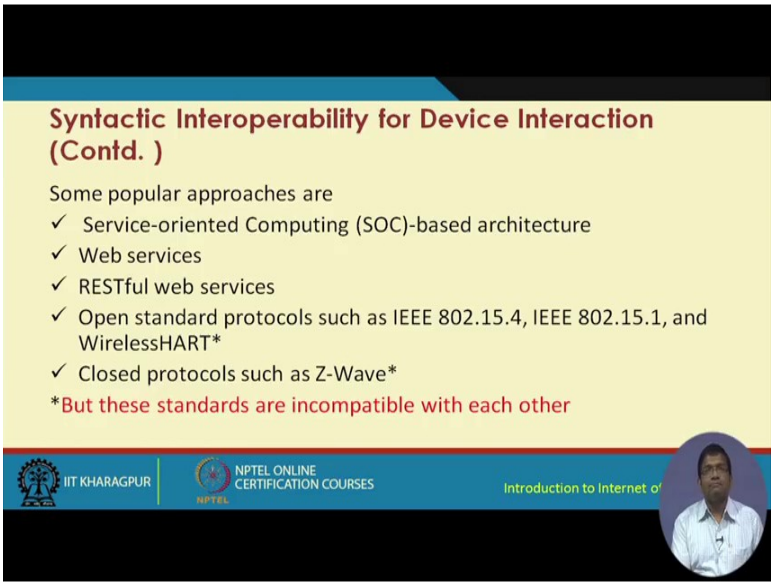
So, there are different device classification solutions and there are two of them that are quite popular one is the UNSPSC which is united nations standards standard products and services code which is an open global multi sector standard for efficient accurate flexible classification of products and services. And the other one is the eCl@ss which is the standard for classification and clear description of cross industry products.

(Refer Slide Time: 17:01)



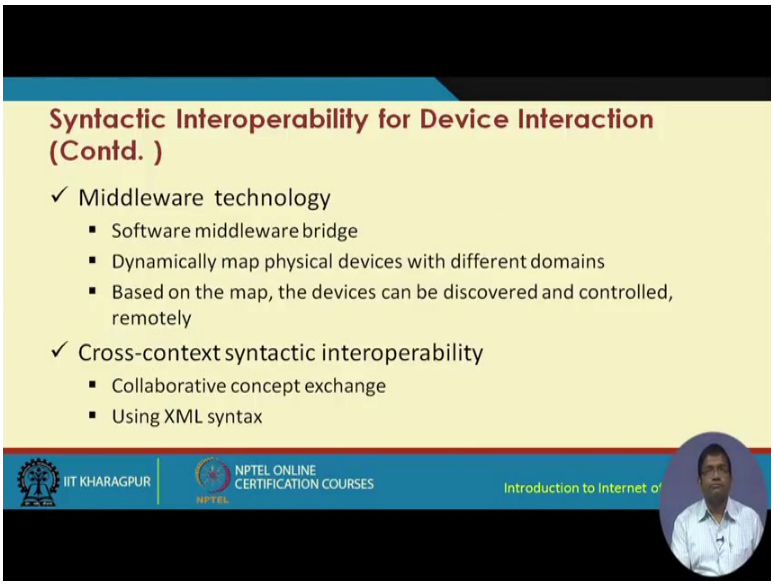
Next comes the syntactic interoperability for device interaction the interoperability between devices and device user in terms of message formats is what is the concern of this type of interoperability. The message format from a device to a user is understandable for the user's computer. On the other hand the message format from the user to the device is executable by the device.

(Refer Slide Time: 17:32)



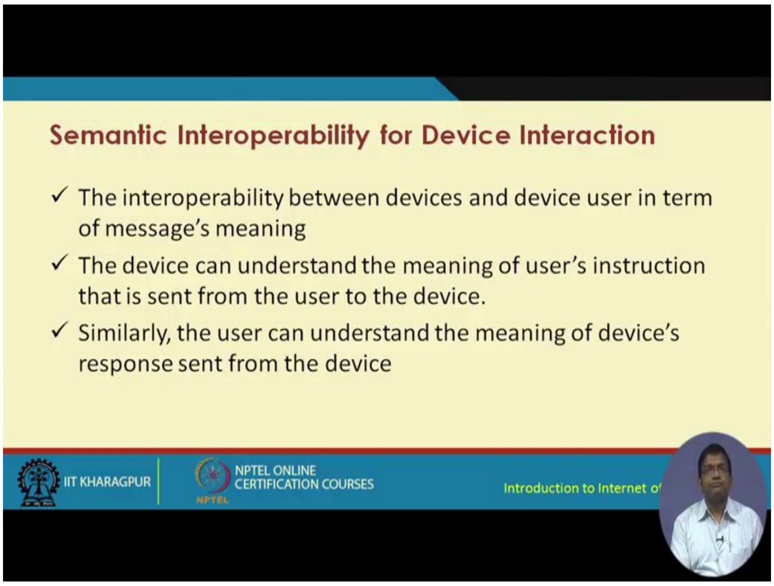
Some popular approaches are service oriented computing SOC based architecture. Then web services using web services RESTful web services, which is quite popular for IoT. So, rest architecture, and the RESTful architecture and the corresponding RESTful web services. Open following some open standard protocols such as 802.15.4, 802.15.1, WirelessHART and then following some closed protocols such as Z-wave. But we have to keep in mind that Z-wave is proprietary and this is a you know proprietary closed protocol not a standard protocol and that is why you know Z-wave devices will talk with Z-wave devices, whereas you know those which follow the WirelessHART you know they can talk to each other.

(Refer Slide Time: 18:31)



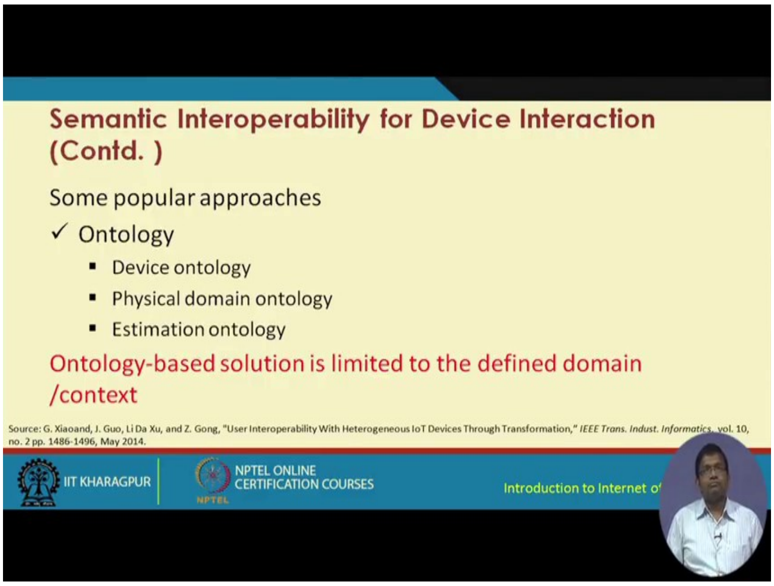
Middleware technology, so this middleware technology is sort of like a software middleware bridge, which dynamically maps the physical devices with the different domains and based on the map the devices can be discovered and controlled remotely. Then we have the cross context syntactic interoperability, which concerns collaborative concept exchange and using XML syntax.

(Refer Slide Time: 18:59)



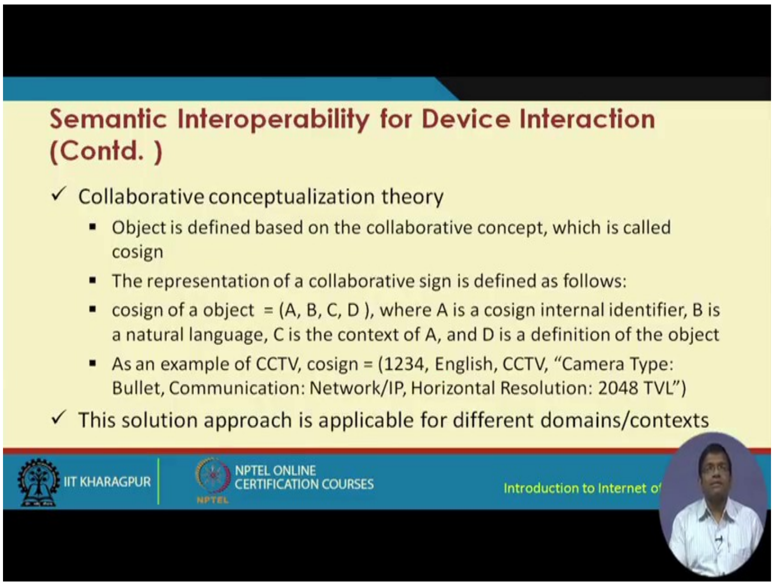
Now, semantic interoperability for device interaction; and here we are talking about the semantics, semantics and the exchange of the semantics. So, the messages that are sent between these different devices whether they are understood by the respective party if not there has to be some middleware in between which has to make it happen. So, the device can understand the meaning of users instructions that is sent from the user to the device, similarly the user can understand the meaning of devices response sent from the device.

(Refer Slide Time: 19:41)



So, some of the popular approaches are ontology based approach number one which again can be classified as device ontology, physical domain ontology and estimation based ontology. Device so, ontology means what some kind of a knowledge base. So, device ontology basically is you know the knowledge base about devices. So, this is required, you know so if two devices have to talk to each other, so the ontology the corresponding knowledge basis of these different devices have to be formed. Physical domain in ontology I do not need to elaborate this further, so you know so knowledge base about the physical domains of operation. And estimation ontology is about you know based on the previous data estimating what is going to happen, so that sort of rule, rule base has to be maintained in some kind of a knowledge base. So, this is estimation ontology. So, ontology based solution is limited to the defined domain or context.

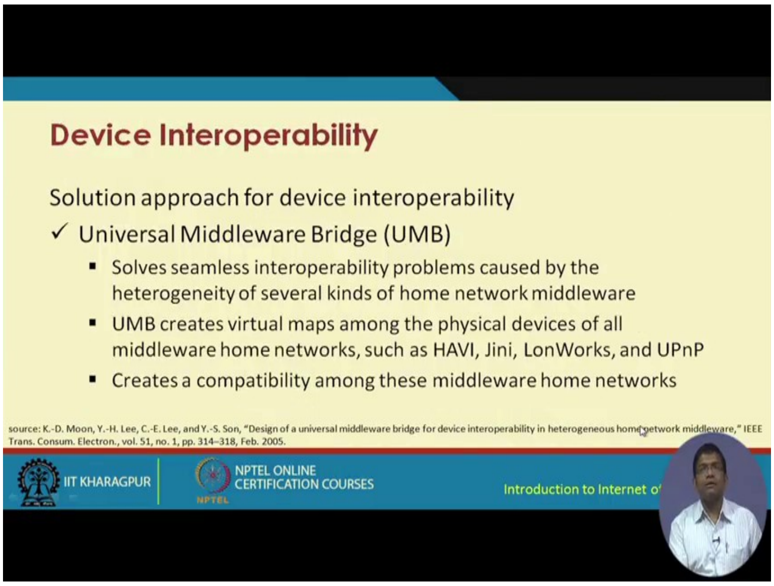
(Refer Slide Time: 20:44)



So that is the reason why there is one theory which is known as the collaborative conceptualization theory that was proposed. So, the object is defined based on this particular collaborative concept which is also known as the cosign concept, cosign is the short form of collaborative sign. So, cosign of an object is defined as a quadruple A, B, C, D where A is the cosign internal identifier, B is the natural language, C is the context of A, and D is the definition of the object.

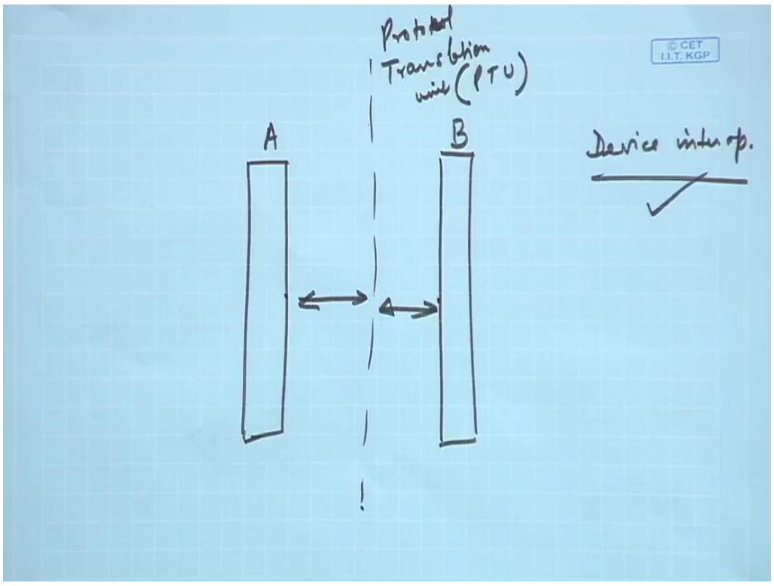
So, in our CCTV example the cosign of the CCTV is equal to 1234 which is some kind of an identifier for this CCTV object, English is the natural language that is used, C is the context of A that means, here it is CCTV the CCTV is the context and D is the definition of the object. So, here we have camera type bullet camera type equal to bullet communication equal to network or IP and horizontal resolution equal 2048 TVL. So, this is basically the definition of the object with respect to this particular specification. So, this solution approach is applicable for different domains and contexts.

(Refer Slide Time: 22:08)



Now, about device interoperability let us try to understand before we go any further how this is going to work.

(Refer Slide Time: 22:28)

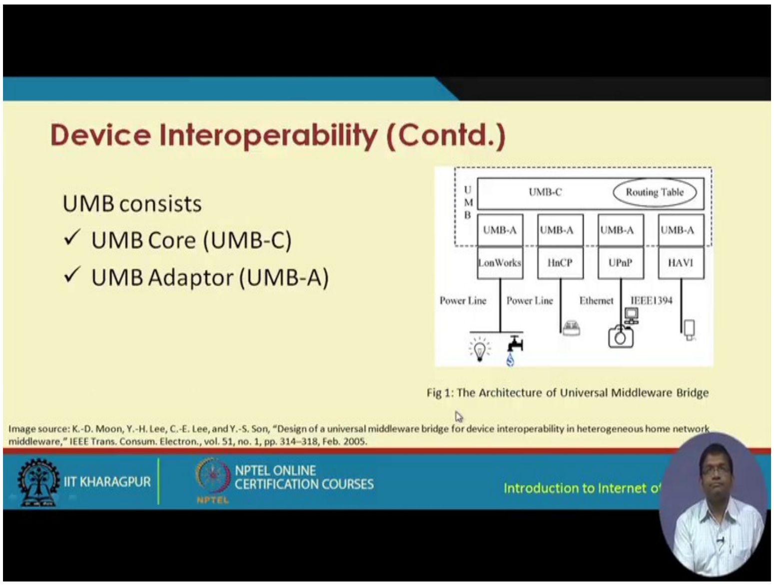


Let us say that even before we start with the device interoperability, let us say that we have you know two devices: device A and device B. And in our example let us assume that A and B as such do not follow the same specific same protocol, they did not have the same specifications at the physical layer or you know the other layers. So, no common protocol is available across all these different layers. So, how do they communicate how do they communicate. So, this is the problem of device interoperability. So, how do they communicate.

So, let us assume initially that we have some kind of a middleware some kind of a middleware, which will understand the language of A and the language of B. So, it will understand the language of A as well as the language of B. So, this will help this one to be able to translate what A is saying A wants to send, he wants to communicate, and similarly what B is say. So, this sort of approach not only can be used for device interoperability, but also other forms of interoperability as well. So, this one basically becomes a translation device translation unit. So, if we are talking about two different protocols we can call it as a protocol translation unit - PTU. So, this protocol translation unit will translate the protocols separate protocols or the languages that are followed by both, ok. So, this is an approach that can be adopted and can be extended for similar other situations as well.

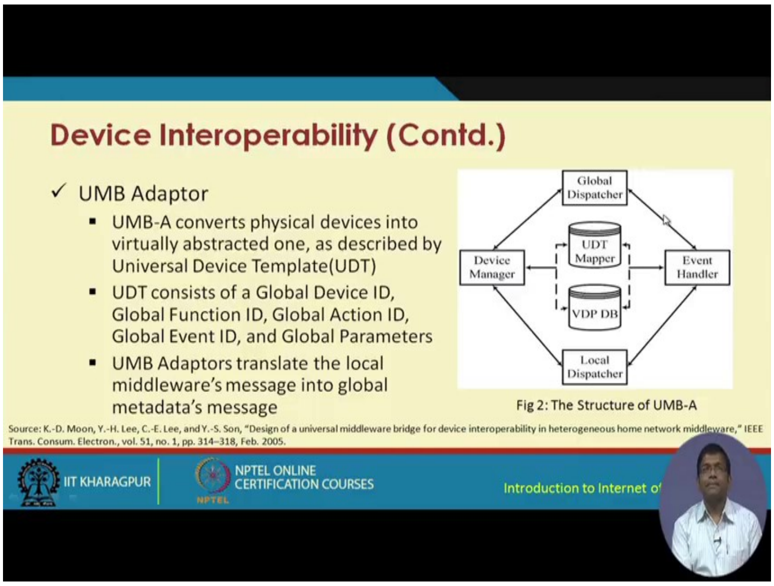
So, we talked about in device interoperability we talked about some kind of a universal middleware bridge which solves seamless interoperability problems caused by heterogeneity of several kinds of home network middleware. So, this bridge is basically it is a middleware that creates a virtual map among the physical devices of all middleware home networks such as HAVI, Jinl, LonWorks, UPnP and so on. And it creates a compatibility among these middleware home networks. So, it is basically some middleware based solution this middleware will act as an agent for this kind of translation or handshaking between two different heterogeneous devices.

(Refer Slide Time: 25:42)



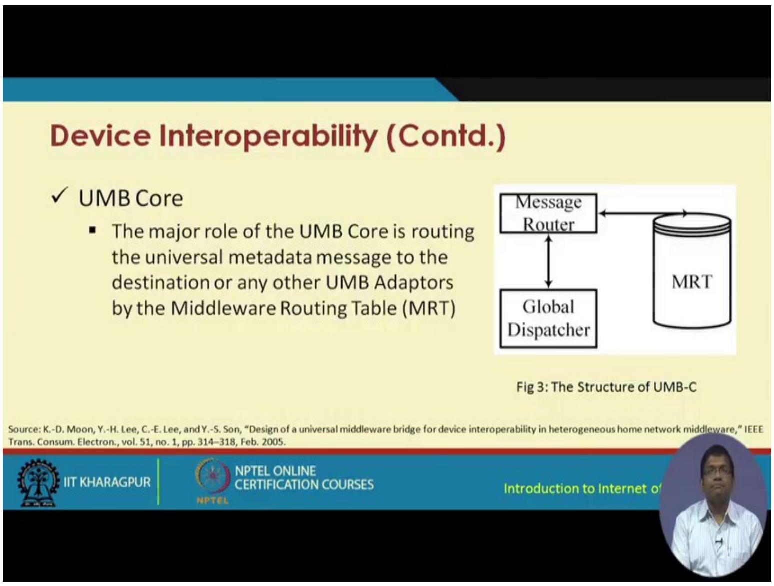
So, in device interoperability we have two parts the UMB core and the UMB adaptor. So, this is let us say that physically these are the devices. So, physically these are the devices, we have a lamp, we have an water tap, we have a camera, we have a computer and so on. So, these are the physical devices and they have to be internetworked together. And let us say that they have they are completely heterogeneous, they do not you know talk to each other as such. So, this is what is going to happen, this is what is going to happen. We are going to create a UMB socket we are going to create a UMB socket with different adaptors like this, which will fit into these ones which are the abstractions of these physical devices. So, these adaptors would fit to these abstractions fit to the abstractions of these devices and that is how this communication is going to take place. So, this is the function of the adaptor. Then what is this UMB-C the code does what it does is it basically helps in communication dissemination of the data between these different adaptors using with the help of Routing tables.

(Refer Slide Time: 27:15)



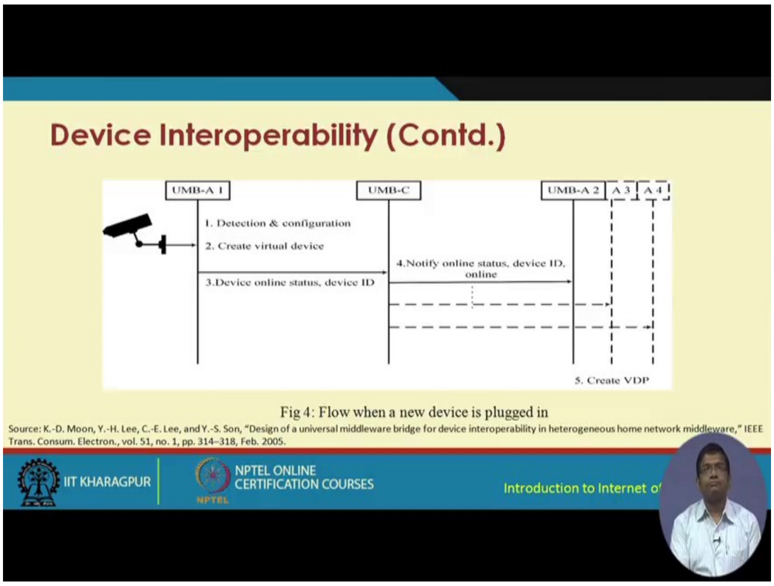
So, the adaptor basically converts the physical devices into some virtual abstractions and this is done with the help of something known as the universal device template. So, this universal device template or the UDT basically helps in coming up with the abstraction the virtual abstractions and that is stored in some kind of a database. So, this UDT mapper will come up with these virtual abstractions which are stored in the database and this UDT basically consists of the global device identifier, global function identifier, global action identifier, global event identifier and global parameters and the UMB adaptors translates the local middleware's message into the global metadata message.

(Refer Slide Time: 28:04)



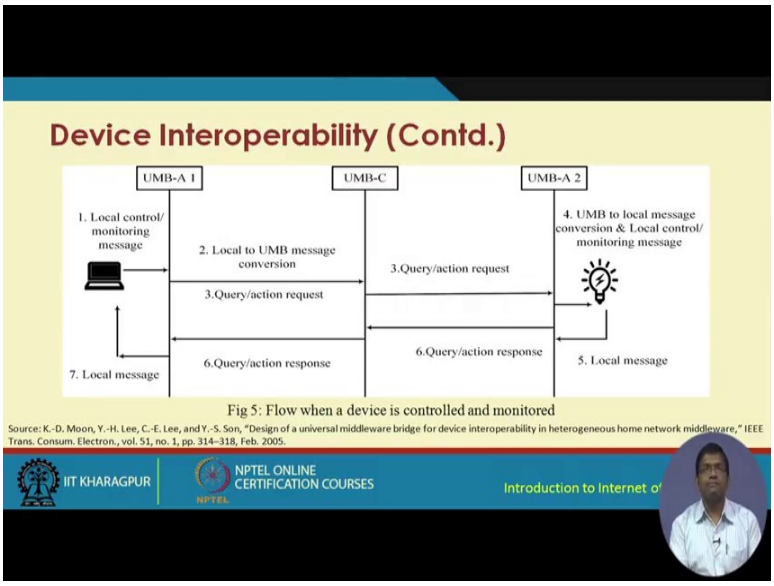
Now comes the core which is on top of the adaptor in that architecture. So, the core basically has the role of routing this is the main the most important role of the core. So, it uses something known as a middleware routing table; and from that from the middleware routing table with the help of the data that is residing in it, it creates the message router and from the message router we get the global dispatcher that information is basically used at the core to send the data between these different devices.

(Refer Slide Time: 28:47)



So, let us consider some few basic scenarios we have this particular device connected to this adaptor it detects and in first it has to detect and it has to configure then it creates some virtual device. So, then the device online status the device ID etcetera are sent across from the adaptor to the core and then from the core this message is sent to another adaptor notifying the online status the device ID and ID and whether it is online or not and it is sent to another adaptor, third adaptor, fourth adaptor and so on. So, this is how communication using UMB-A and UMB-C takes place between two heterogeneous devices following different configurations.

(Refer Slide Time: 29:54)



Now, let us look at a scenario when the device is controlled and monitored. So, we have UMB-A 1, then UMB-C and UMB-A 2. So, this, the first message let us say that these two would have to be talking to each other. So, this one has to control this and this has to be monitored right. So, the local control or monitoring message is first sent from the device to the adaptor A 1, then the local to UMB message conversion takes place then it is sent to the UMB-C. Then from here a query or action request is sent to the second adaptor and then the UMB is used to local message conversion for local message conversion and local control or monitoring message. Then this local message is sent to this particular adaptor from the device then a response is sent back another response is sent back and the local message is delivered to this particular device.

So, we have thus come to the end of how the communication using adaptors happen in the context of interoperability in internet of things. So, we have seen that in this interoperability architecture, we have two types of components one is the adaptor which basically is a virtual or software abstraction of the actual physical device. And this you know this abstracted virtual or softwarized device talks to other similar kind of abstracted devices with the help of this particular core the UMB-C. And this is how the communication takes place across different heterogeneous devices in internet of things.

Thank you.