Design for Internet of Things
Prof. T V Prabhakar
Department of Electronic Systems Engineering
Indian Institute of Science-Bengaluru

Lecture - 51
BLE Mesh Technology

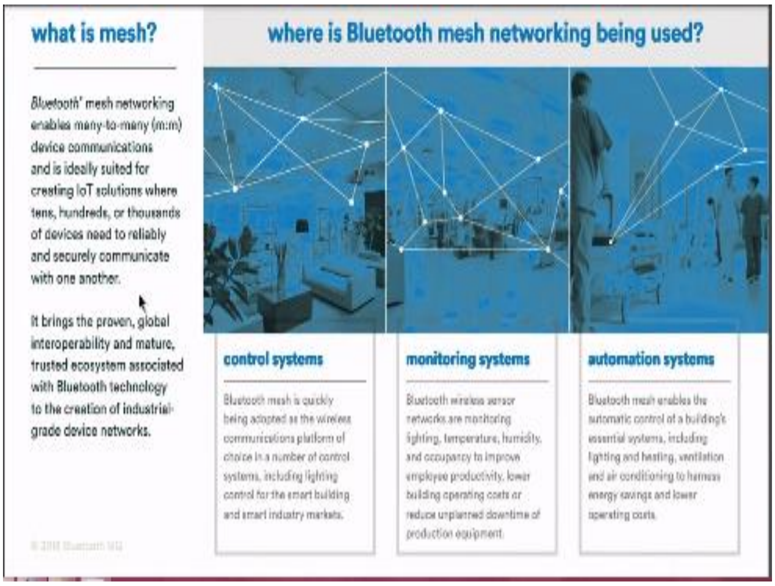
Folks in this same module, let us also look at what is happening in the currently in the industrial world and it is not low power wide area networks, it is neither low power area systems, but it is somewhere in the middle. You want to build let us say a very reliable wireless network for industrial applications, okay. That is the idea. And how do you do that?

A fast upcoming standard is the mesh networking standard for Bluetooth low energy. So BLE mesh has picked up quite a bit. And since in our lab, we have been doing some work in that space, I thought it is important to show you a small demo, excite you into building future networks which are based on mesh systems ideally suited for inside indoor applications.

If you want to do indoor localization, you can use mesh networks, you can use these mesh systems. You want to do industrial monitoring of machines in a reliable way under highly noisy conditions, RF noise conditions. Lot of equipment emitting RF radiation can disturb packets which are flowing on the wireless link, right.

So all of that essentially means you need reliable communication to you know in order to make your actions. Either you take an action, or you want to close the loop or you want to do a control action and so on. So that essentially brings you to very exciting protocol.

(Refer Slide Time: 02:09)



what is mesh?

Bluetooth® mesh networking enables many-to-many (m:m) device communications and is ideally suited for creating IoT solutions where tens, hundreds, or thousands of devices need to reliably and securely communicate with one another.

It brings the proven, global interoperability and mature, trusted ecosystem associated with Bluetooth technology to the creation of industrial-grade device networks.

where is Bluetooth mesh networking being used?

control systems

Bluetooth mesh is quickly being adopted as the wireless communications platform of choice in a number of control systems, including lighting control for the smart building and smart industry markets.

monitoring systems

Bluetooth wireless sensor networks are monitoring lighting, temperature, humidity, and occupancy to improve employee productivity, lower building operating costs or reduce unplanned downtime of production equipment.

automation systems

Bluetooth mesh enables the automatic control of a building's essential systems, including lighting and heating, ventilation and air conditioning to harness energy savings and lower operating costs.

Here what we did was we downloaded an article to give you an overview of where we are with respect to the Bluetooth mesh networking place. You can see it is for many-to-many, ideally suited for creating IoT solutions. And you can have thousands of devices that need to be reliably and securely are communicating with one another.

One good thing that will happen if you do mesh based networking with Bluetooth is the human holds a phone in the hand. And this human holding the phone in the hand allows you to get data also on the phone. That is really amazing, right? You have an industrial network, and it is all connected with the Bluetooth mesh. And then I go with my mobile phone, which also has Bluetooth.

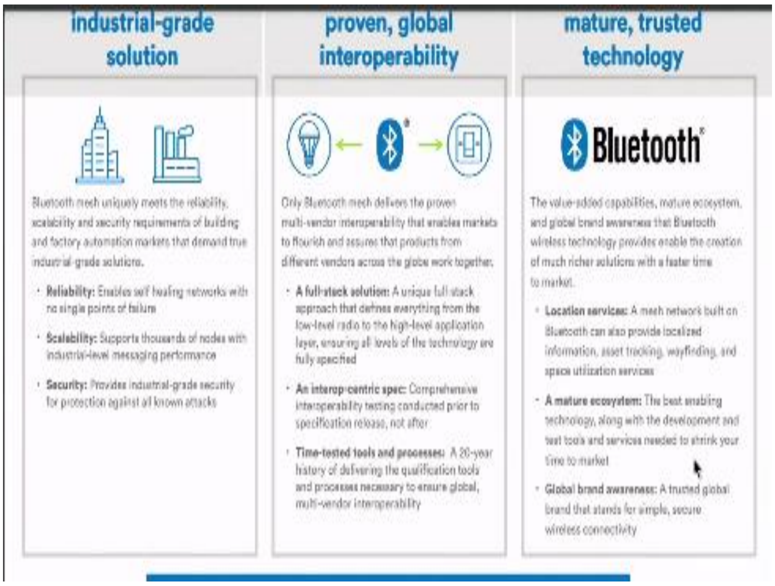
I can enter that mesh and I can start getting data, right? Well, you may say that that is a secure network, how can you just enter and get the data from it. Yeah, there are ways by which you can make your mobile phone secure and part of that mobile network. That is where the whole game is. You have to look at how to provision my phone in the pocket to be part of that mesh network so that I can securely receive data.

What do you mean provisioning my phone? It simply means some encryption key or some security related data will also have to sit in my phone so that it is all part of that mesh and my phone becomes part of that mesh and I can start seeing the data. So essentially all that is supported in the mesh networking paradigm. And essentially, there are some three major applications which they show here.

One is for control systems to set up Bluetooth mesh quickly. And essentially it could include lighting control for smart building, smart industry markets and so on. You can also do monitoring, simple monitoring. It could be for lighting, temperature, humidity and occupancy and all that. Recall what we discussed about the blinds, window blind examples, which we did in the beginning.

All those window blind examples can actually be realized with Bluetooth mesh systems. Then of course automation. It allows you to, you can build very compelling applications for automation. It could include ventilation, air conditioning, to harness energy savings and lower operating cost and so on.

(Refer Slide Time: 04:31)



industrial-grade solution

Bluetooth mesh uniquely meets the reliability, scalability and security requirements of building and factory automation markets that demand true industrial-grade solutions.

- **Reliability:** Enables self healing networks with no single points of failure
- **Scalability:** Supports thousands of nodes with industrial-level messaging performance
- **Security:** Provides industrial-grade security for protection against all known attacks

proven, global interoperability

Only Bluetooth mesh delivers the proven multi-vendor interoperability that enables markets to flourish and assures that products from different vendors across the globe work together.

- **A full-stack solution:** A unique full stack approach that defines everything from the low-level radio to the high-level application layer, ensuring all levels of the technology are fully specified
- **An interop-centric spec:** Comprehensive interoperability testing conducted prior to specification release, not after
- **Time-tested tools and processes:** A 20-year history of delivering the qualification tools and processes necessary to ensure global, multi-vendor interoperability

mature, trusted technology

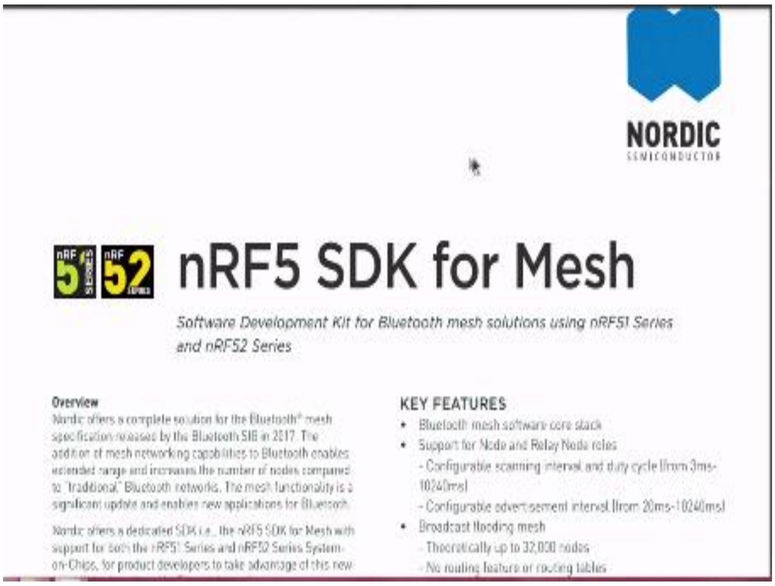
The value-added capabilities, mature ecosystem, and global brand awareness that Bluetooth wireless technology provides enable the creation of much richer solutions with a faster time to market.

- **Location services:** A mesh network built on Bluetooth can also provide localized information, asset tracking, wayfinding, and space utilization services
- **A mature ecosystem:** The best enabling technology, along with the development and test tools and services needed to shrink your time to market
- **Global brand awareness:** A trusted global brand that stands for simple, secure wireless connectivity

So all of that essentially means that the stack has the Bluetooth stack has become quite extensive enough to support this mesh network, providing you reliability, scalability and without any compromise on the security of the system. Now the Bluetooth mesh is interoperable. That means any vendor can participate in this mesh. And devices can be heterogeneous, need not be coming from the same vendor.

Because there is indeed a full stack solution and there is a standard for implementation. So interoperability between vendors is possible. And there are good enough tools for you to understand the whole process by which this multi-vendor interoperability is working successfully or not. As I mentioned to you, mesh you can use for location services. You can be part of monitoring and so on and so many other monitoring and other applications and so on. So that is an important thing.

(Refer Slide Time: 05:33)



NORDIC
SEMICONDUCTOR

nRF5 SDK for Mesh

Software Development Kit for Bluetooth mesh solutions using nRF51 Series and nRF52 Series

Overview
Nordic offers a complete solution for the Bluetooth® mesh specification released by the Bluetooth SIG in 2017. The addition of mesh networking capabilities to Bluetooth enables extended range and increases the number of nodes compared to "traditional" Bluetooth networks. The mesh functionality is a significant update and enables new applications for Bluetooth.

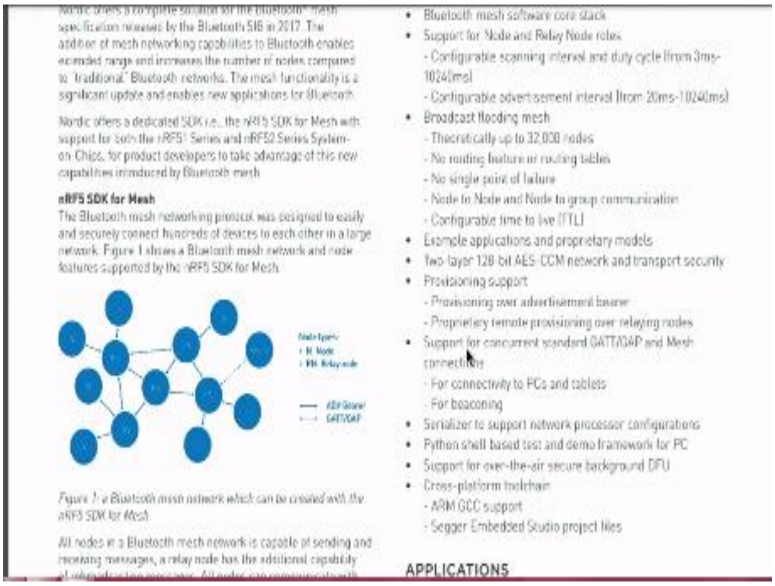
Nordic offers a dedicated SDK i.e., the nRF5 SDK for Mesh with support for both the nRF51 Series and nRF52 Series System-on-Chips, for product developers to take advantage of this new

KEY FEATURES

- Bluetooth mesh software core stack
- Support for Node and Relay Node roles
  - Configurable scanning interval and duty cycle (from 3ms-10240ms)
  - Configurable advertisement interval (from 20ms-10240ms)
- Broadcast flooding mesh
  - Theoretically up to 32,000 nodes
  - No routing feature or routing tables

Now nRF Nordic has actually provided an SDK for mesh technology. But before we go into the detail as I mentioned to you, we must look up the nRF SDK for Nordic has actually offered a beautiful startup kit on their 52 series components that parts which are available, microcontroller parts and here are some of the features of it.

(Refer Slide Time: 06:01)



Nordic offers a complete solution for the Bluetooth® mesh specification released by the Bluetooth SIG in 2017. The addition of mesh networking capabilities to Bluetooth enables extended range and increases the number of nodes compared to "traditional" Bluetooth networks. The mesh functionality is a significant update and enables new applications for Bluetooth.

Nordic offers a dedicated SDK i.e., the nRF5 SDK for Mesh with support for both the nRF51 Series and nRF52 Series System-on-Chips, for product developers to take advantage of this new capabilities introduced by Bluetooth mesh.

nRF5 SDK for Mesh
The Bluetooth mesh networking protocol was designed to easily and securely connect hundreds of devices to each other in a large network. Figure 1 shows a Bluetooth mesh network and node features supported by the nRF5 SDK for Mesh.

Figure 1: a Bluetooth mesh network which can be created with the nRF5 SDK for Mesh

All nodes in a Bluetooth mesh network is capable of sending and receiving messages, a relay node has the additional capability

- Bluetooth mesh software core stack
- Support for Node and Relay Node roles
  - Configurable scanning interval and duty cycle (from 3ms-10240ms)
  - Configurable advertisement interval (from 20ms-10240ms)
- Broadcast flooding mesh
  - Theoretically up to 32,000 nodes
  - No routing feature or routing tables
  - No single point of failure
  - Node to Node and Node to group communication
  - Configurable time to live (TTL)
- Example applications and proprietary models
- Two-layer 128-bit AES-CCM network and transport security
- Provisioning support
  - Provisioning over advertisement bearer
  - Proprietary remote provisioning over relaying nodes
- Support for concurrent standard GATT/GAP and Mesh connections
  - For connectivity to PCs and tablets
  - For beaconing
- Serializer to support network processor configurations
- Python shell based test and demo framework for PC
- Support for over-the-air secure background DFU
- Cross-platform toolchain
  - ARM GCC support
  - Segger Embedded Studio project files

APPLICATIONS

You can see that you can configure the mesh for relay nodes as well as end nodes, okay. Then the stack actually talks about if you look at the software stack you can be talking about advertising bearer, which are these hard lines that you see here. You see these hard lines these are all advertising bearers. And then there is a gap also which is from the old traditional generic access protocol part which is also out there.

So it is a combination of adding additional things into the protocol stack which will support the mesh systems. So you can support securely several hundreds of devices and this is a nice picture, okay.

(Refer Slide Time: 06:46)

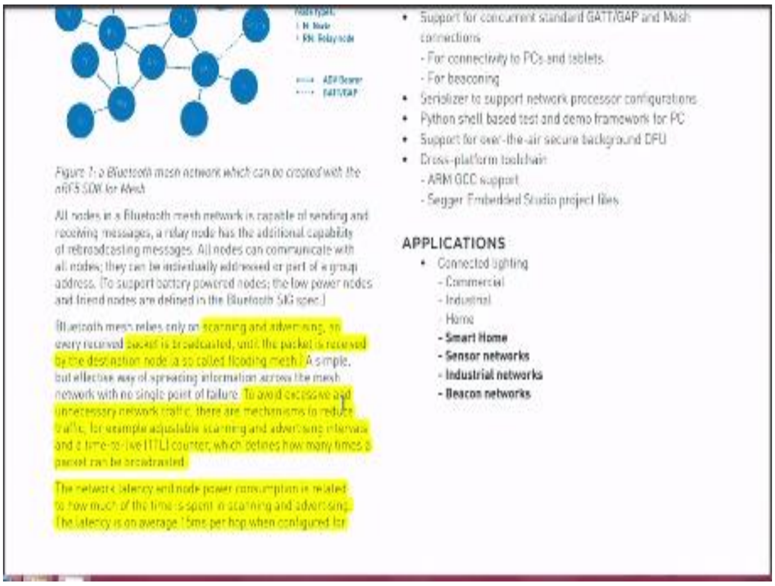


Figure 1: a Bluetooth mesh network which can be created with the nRF5 SDK for Mesh

All nodes in a Bluetooth mesh network is capable of sending and receiving messages, a relay node has the additional capability of rebroadcasting messages. All nodes can communicate with all nodes; they can be individually addressed or part of a group address. (To support battery powered nodes; the low power nodes and friend nodes are defined in the Bluetooth SIG spec.)

Bluetooth mesh relies only on scanning and advertising, so every received packet is broadcasted, until the packet is received by the destination node (a so called flooding mesh). A simple, but effective way of spreading information across the mesh network with no single point of failure. To avoid excessive and unnecessary network traffic, there are mechanisms to reduce traffic, for example adjustable scanning and advertising intervals and a time-to-live (TTL) counter, which defines how many times a packet can be broadcasted.

The network latency and node power consumption is related to how much of the time is spent in scanning and advertising. The latency is on average 15ms per hop when configured for

- Support for concurrent standard GATT/GAP and Mesh connections
  - For connectivity to PCs and tablets
  - For beaconing
- Serializer to support network processor configurations
- Python shell based test and demo framework for PC
- Support for over-the-air secure background DFU
- Cross-platform toolchain
  - ARM GCC support
  - Segger Embedded Studio project files

APPLICATIONS

- Connected lighting
  - Commercial
  - Industrial
  - Home
  - **Smart Home**
  - **Sensor networks**
  - **Industrial networks**
  - **Beacon networks**

So how does this mesh work? It purely works on scanning and advertising. So every packet is broadcasted until the packet is received by the destination. So you use what is known as flooding. So it is a flooding mesh. May not be all that energy efficient, but surely it is a reliable mesh because of the fact that it floods and then packets are received by multiple nodes. The relay node has the job of re-relaying the packet.

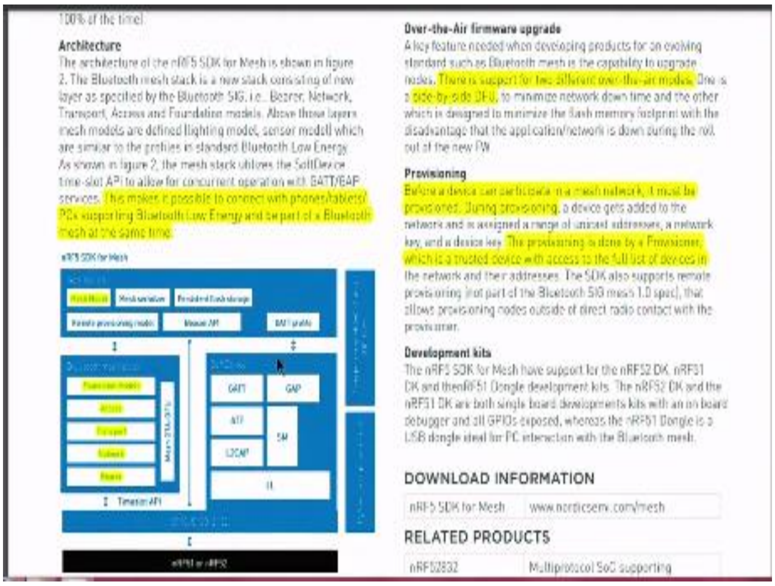
That is the only difference between the RN and the end node, okay. Now you may also want to say that is it not going to add to a lot of excessive traffic? Yes, indeed because each time you are flooding and each node that picks up has to re-broadcast it, it is a lot of flooding, right? So to avoid the excessive and unnecessary network traffic, there are mechanisms okay, to reduce the traffic.

For example, adjustable scanning and advertising intervals you can sort of tune and time-to-live is also an important parameter here and how many times the packet can be broadcasted is also there. Each time you receive it, you broadcast it maybe account is decremented by 1. And then once it becomes 0 you stop broadcasting it. So you can control it. So latency and power consumption are related, right?

So how much time is spent in scanning and advertising means if you put it very large number there is obviously going to be a latency hit for you. So an average latency is on an average is 15 milliseconds per hop when configured for minimum latency and power consumption, okay. And it is determined by a large extent by the receive current of the device.

Of course, when the state machine at the receiver is of the transceiver chip is in the receive mode, the current consumed and the receive mode is the one that actually determines. So these details are given here.

(Refer Slide Time: 08:38)



100% of the time!

Architecture

The architecture of the nRF5 SDK for Mesh is shown in figure 2. The Bluetooth mesh stack is a new stack consisting of new layer as specified by the Bluetooth SIG, i.e., Bearer, Network, Transport, Access and Foundation models. Above those layers mesh models are defined (lighting model, sensor model) which are similar to the profiles in standard Bluetooth Low Energy. As shown in figure 2, the mesh stack utilizes the SoftDevice time-slot API to allow for concurrent operation with GATT/GAP services. This makes it possible to connect with phones/tablets/PCs supporting Bluetooth Low Energy and be part of a Bluetooth mesh at the same time.

nRF5 SDK for Mesh

Over-the-Air firmware upgrade

A key feature needed when developing products for an evolving standard such as Bluetooth mesh is the capability to upgrade nodes. There is support for two different over-the-air modes. One is a side-by-side DFU, to minimize network down time and the other which is designed to minimize the flash memory footprint with the disadvantage that the application/network is down during the roll out of the new FW.

Provisioning

Before a device can participate in a mesh network, it must be provisioned. During provisioning, a device gets added to the network and is assigned a range of unicast addresses, a network key, and a device key. The provisioning is done by a Provisioner, which is a trusted device with access to the full list of devices in the network and their addresses. The SDK also supports remote provisioning (not part of the Bluetooth SIG mesh 1.0 spec), that allows provisioning nodes outside of direct radio contact with the provisioner.

Development kits

The nRF5 SDK for Mesh have support for the nRF52 DK, nRF51 DK and thenRF51 Dongle development kits. The nRF52 DK and the nRF51 DK are both single board developments kits with an on board debugger and all GPIOs exposed, whereas the nRF51 Dongle is a USB dongle ideal for PC interaction with the Bluetooth mesh.

DOWNLOAD INFORMATION

nRF5 SDK for Mesh	www.nordicsemi.com/mesh

RELATED PRODUCTS

nRF52832	Multiprotocol SoC supporting

Now I mentioned to you about the architecture, which is lot more exhaustive compared to the old Bluetooth that we know very well. You can see the mesh figure is shown here and you have bearer network which is added. You have transport, access and foundation models. Now all these above layers, above all these above this layers, above these layers, the mesh models are defined.

You have lighting model, you have sensor model and so on. So you have different types of models which are up out there, okay. These are called the mesh models. You can define several of them independently. There is also, so the idea here is it is possible to connect with phones, tablets, and all that and become part of the Bluetooth mesh at the same time.

This is the exciting part that I mentioned that you can go with your phone and start participating into, you can get into the mesh quite easily.

(Refer Slide Time: 09:40)

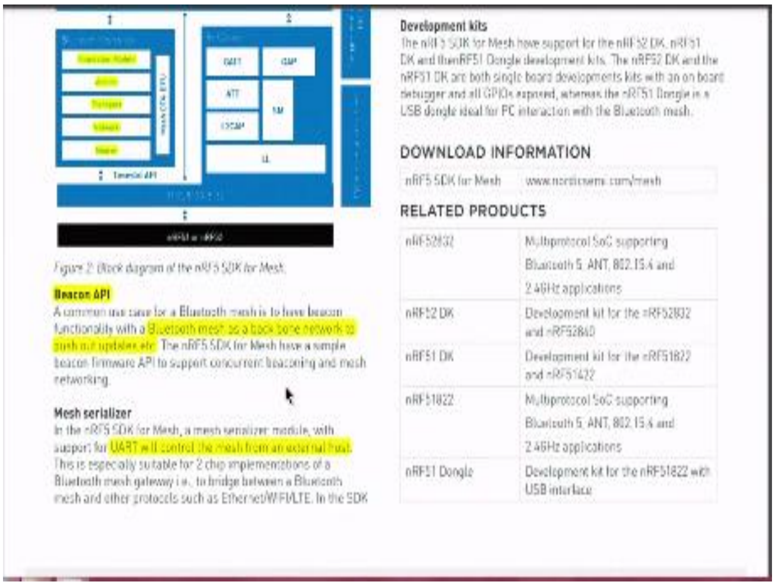


Figure 2: Block diagram of the nRF5 SDK for Mesh.

Beacon API

A common use case for a Bluetooth mesh is to have beacon functionality with a Bluetooth mesh as a back bone network to push out updates etc. The nRF5 SDK for Mesh have a simple beacon firmware API to support concurrent beaconing and mesh networking.

Mesh serializer

In the nRF5 SDK for Mesh, a mesh serializer module, with support for UART will control the mesh from an external host. This is especially suitable for 2 chip implementations of a Bluetooth mesh gateway i.e., to bridge between a Bluetooth mesh and other protocols such as Ethernet/WiFi/LTE. In the SDK

Development kits

The nRF5 SDK for Mesh have support for the nRF52 DK, nRF51 DK and thenRF51 Dongle development kits. The nRF52 DK and the nRF51 DK are both single board developments kits with an on board debugger and all GPIOs exposed, whereas the nRF51 Dongle in a USB dongle ideal for PC interaction with the Bluetooth mesh.

DOWNLOAD INFORMATION

nRF5 SDK for Mesh	www.nordicsemi.com/mesh

RELATED PRODUCTS

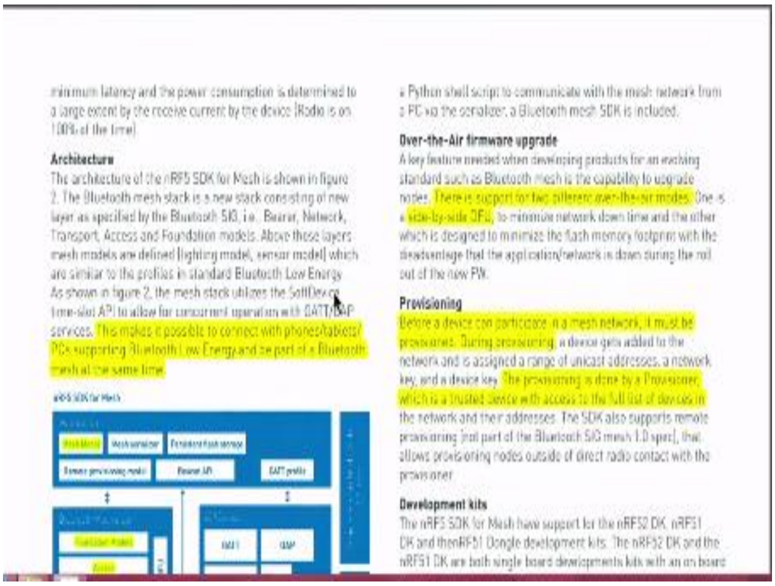
nRF52832	Multiprotocol SoC supporting Bluetooth 5, ANT, 802.15.4 and 2.4GHz applications
nRF52 DK	Development kit for the nRF52832 and nRF52840
nRF51 DK	Development kit for the nRF51822 and nRF51422
nRF51822	Multiprotocol SoC supporting Bluetooth 5, ANT, 802.15.4 and 2.4GHz applications
nRF51 Dongle	Development kit for the nRF51822 with USB interface

The beacon API, which I show you here is also here. As you can see, this is the beacon API. And what this is, is essentially it is the mesh Bluetooth mesh as the backbone network to push out the updates, okay. It has, the common use case of Bluetooth mesh is to have a beacon functionality. This SDK supports beacon firmware API to support concurrent beaconing and mesh networking.

And there is a mesh serializer as you can see here, there is a mesh serializer here. And its job is you can control it with a UART. You want to build let us say a mesh gateway, you want to bridge between Bluetooth mesh and other protocols, you can use this mesh serializer to pull out the data and feed it into Ethernet and so on. So it will support you a UART okay, with the support for UART.

So it is nothing UART support for the Bluetooth mesh by which you can take the data and recast it as Ethernet or any one of them, okay.

(Refer Slide Time: 10:45)



minimum latency and the power consumption is determined to a large extent by the receive current by the device (Radio is on 100% of the time).

Architecture

The architecture of the nRF5 SDK for Mesh is shown in figure 2. The Bluetooth mesh stack is a new stack consisting of new layer as specified by the Bluetooth SIG, i.e., Bearer, Network, Transport, Access and Foundation models. Above these layers mesh models are defined (lighting model, sensor model) which are similar to the profiles in standard Bluetooth Low Energy. As shown in figure 2, the mesh stack utilizes the SoftDevice time-slot API to allow for concurrent operation with GATT/GAP services. This makes it possible to connect with phones/tablets/PCs supporting Bluetooth Low Energy and be part of a Bluetooth mesh at the same time.

nRF5 SDK for Mesh

a Python shell script to communicate with the mesh network from a PC via the serializer, a Bluetooth mesh SDK is included.

Over-the-Air firmware upgrade

A key feature needed when developing products for an evolving standard such as Bluetooth mesh is the capability to upgrade nodes. There is support for two different over-the-air modes. One is a side-by-side DFU, to minimize network down time and the other which is designed to minimize the flash memory footprint with the disadvantage that the application/network is down during the roll out of the new FW.

Provisioning

Before a device can participate in a mesh network, it must be provisioned. During provisioning, a device gets added to the network and is assigned a range of unicast addresses, a network key, and a device key. The provisioning is done by a Provisioner, which is a trusted device with access to the full list of devices in the network and their addresses. The SDK also supports remote provisioning (not part of the Bluetooth SIG mesh 1.0 spec), that allows provisioning nodes outside of direct radio contact with the provisioner.

Development kits

The nRF5 SDK for Mesh have support for the nRF52 DK, nRF51 DK and thenRF51 Dongle development kits. The nRF52 DK and the nRF51 DK are both single board developments kits with an on board

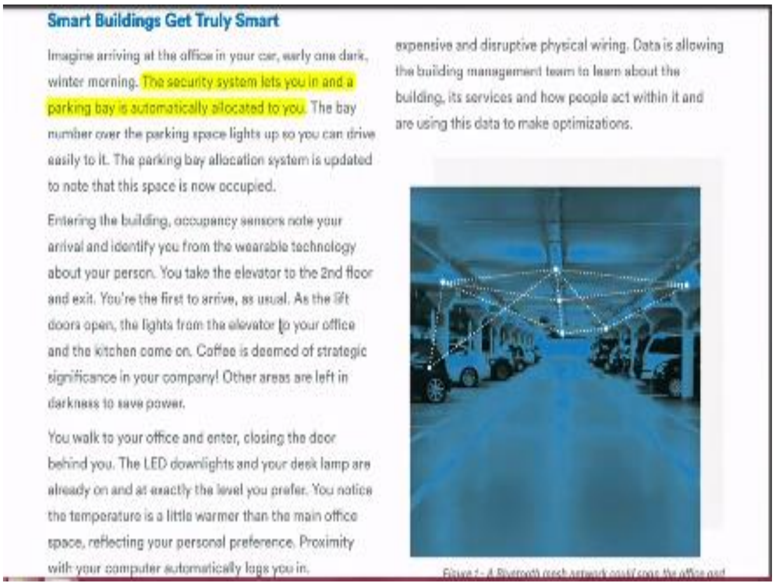
Then there is over-the-air firmware upgrade. So there are different over-the-air modes. And you can use them as well. So what is very critical in this whole discussion is about provisioning. Now before any device, including the example I gave you about my mobile phone entering that network, before you can start participating in the mesh, it must be provisioned, okay.

What exactly this means is during this provisioning process, the device gets added to the network and is assigned a range of unicast addresses. It is assigned some address, a network key as well as a device key. These three things will be there every time you do a provisioning, minimum these three. Unicast addresses, a network key and a device key, all three are there.

Now once the provisioning is done by the provisioner, the provisioner itself is expected to be a trusted one. So it will go and if you use the provisioner and provision the node, then you get these three which I mentioned to you, unicast address, network key, and device key. You are now part of the network and you can be, you can start participating receiving data, rebroadcasting it and so on and so forth, okay. So the SDK here actually supports all of that.

We will now go and see a little bit of what standard Bluetooth mesh networking standard supports for us. So let us read up a little bit of it into a little more detail on the mesh itself.

(Refer Slide Time: 12:16)



Smart Buildings Get Truly Smart

Imagine arriving at the office in your car, early one dark, winter morning. The security system lets you in and a parking bay is automatically allocated to you. The bay number over the parking space lights up so you can drive easily to it. The parking bay allocation system is updated to note that this space is now occupied.

Entering the building, occupancy sensors note your arrival and identify you from the wearable technology about your person. You take the elevator to the 2nd floor and exit. You're the first to arrive, as usual. As the lift doors open, the lights from the elevator to your office and the kitchen come on. Coffee is deemed of strategic significance in your company! Other areas are left in darkness to save power.

You walk to your office and enter, closing the door behind you. The LED downlights and your desk lamp are already on and at exactly the level you prefer. You notice the temperature is a little warmer than the main office space, reflecting your personal preference. Proximity with your computer automatically logs you in.

expensive and disruptive physical wiring. Data is allowing the building management team to learn about the building, its services and how people act within it and are using this data to make optimizations.

Figure 1 - A Bluetooth mesh network could span the office and

Yeah, so if you talk about mesh systems, you can take a nice example of let us say, you imagine arriving at an office, come out, alight from the car or from your vehicle, the security system will allow you to, will allow you inside and put your and show you your parking slot automatically. So once you are in the parking slot, you go there, drive it and leave it there.

Then what you do? You start entering the park and then enter the main building. Now once you enter the building, there are other sensors which will note that you have come in and they identify you from the wearable technology about your person. And you go to the elevator automatically. The elevator will open, close, and then actuate it and then take you directly to the second floor.

Once you arrive as usual, the lift door opens, lights from the elevator to your office and kitchen come on. Coffee is deemed of strategic importance. So other areas are left in the darkness to save power. So you see how the mesh is getting working. Only a few nodes are active and the other nodes in the mesh, keep quiet. And then you walk to your office and enter, close the door behind you.

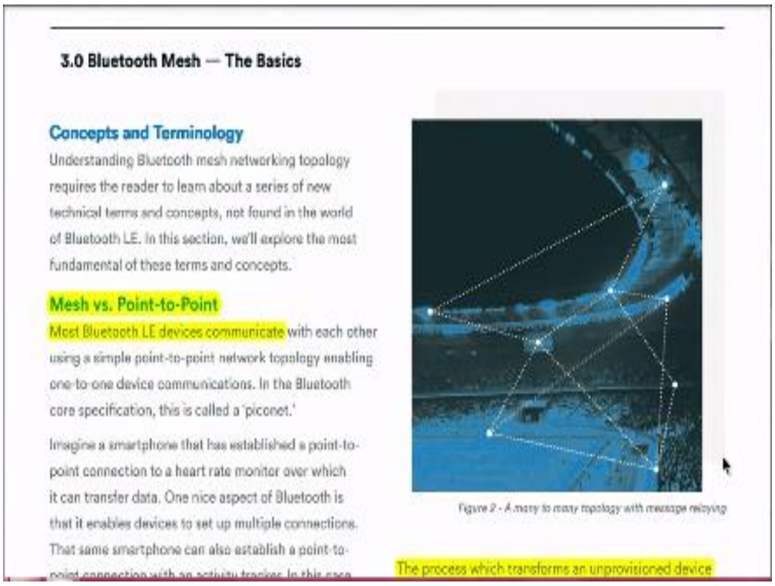
The LED downlights and your desk lamp are ready and at exactly the same level that you prefer. Go back to the original example I described about the lighting, indoor lighting example which we discussed right in the beginning. This is quite similar to that, right? And everything is suited for your convenience and for your ambience. The ambience is set up for your view, okay. So why that is all working.

It is because of the fact that you have a Bluetooth mesh okay, and you have a mesh lighting system. If you look at the stack, the mesh model, one application which Nordic has already given us, is the mesh model related to lighting, okay. This example is so simple that you can just download and install it, it should start working. And it will nicely connect to the blinds example that we were talking about.

What is the light indoor, what is the light outdoor? If you add sensors, you can directly connect the sensors to them and you will have an application running, right? Instead of you know the blind control essentially here it is the same lighting control. It is just about not closing or opening blinds. But indeed about lighting inside the house. In fact, we had also introduced the lighting as an additional node in that example.

So everything is connecting folks and how easy it is for you to look at the mesh model of that system here.

(Refer Slide Time: 15:04)



3.0 Bluetooth Mesh — The Basics

Concepts and Terminology

Understanding Bluetooth mesh networking topology requires the reader to learn about a series of new technical terms and concepts, not found in the world of Bluetooth LE. In this section, we'll explore the most fundamental of these terms and concepts.

Mesh vs. Point-to-Point

Most Bluetooth LE devices communicate with each other using a simple point-to-point network topology enabling one-to-one device communications. In the Bluetooth core specification, this is called a 'piconet.'

Imagine a smartphone that has established a point-to-point connection to a heart rate monitor over which it can transfer data. One nice aspect of Bluetooth is that it enables devices to set up multiple connections. That same smartphone can also establish a point-to-point connection with an activity tracker. In this case

Figure 2 - A many to many topology with message relaying

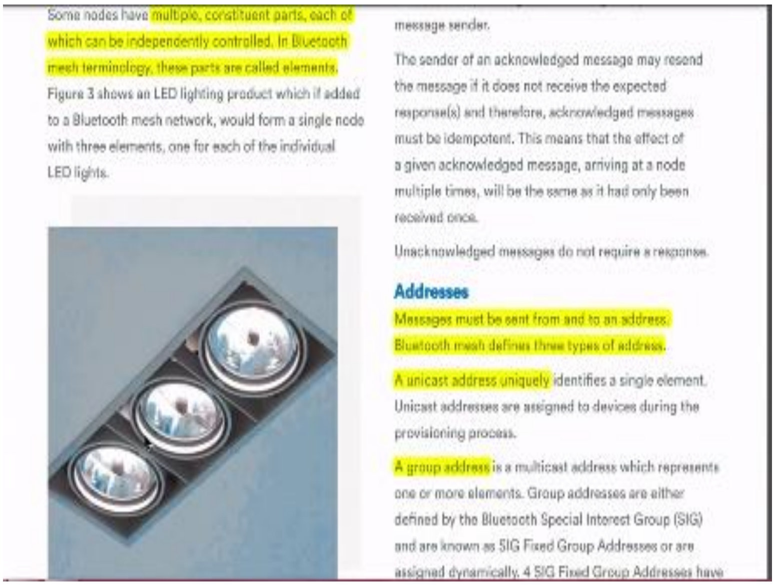
The process which transforms an unprovisioned device

So essentially what we have here in the demo here is, you can, if you read this article it will tell you about advantages of mesh versus point-to-point communication. Basically you are talking about many-to-many topology where each device is able to communicate with every other device.

And you have devices and nodes which are part of the mesh and are called nodes and those which are not are called unprovisioned nodes. Unless you provision it into the mesh, you cannot actually start using the services of the mesh. So the process by which it transforms an unprovisioned device into a node is called provisioning.

And I said about that already that provisioning is a secure procedure and it transfers information from the provisioner onto the end device. All right. So all nodes in a mesh possess at least one network net key and it is position of this key which makes a device a member of the corresponding network and as such a node. I mentioned about this already that during provisioning all these things are actually happening.

(Refer Slide Time: 16:12)



Some nodes have multiple, constituent parts, each of which can be independently controlled. In Bluetooth mesh terminology, these parts are called elements. Figure 3 shows an LED lighting product which if added to a Bluetooth mesh network, would form a single node with three elements, one for each of the individual LED lights.

message sender.

The sender of an acknowledged message may resend the message if it does not receive the expected response(s) and therefore, acknowledged messages must be idempotent. This means that the effect of a given acknowledged message, arriving at a node multiple times, will be the same as it had only been received once.

Unacknowledged messages do not require a response.

Addresses

Messages must be sent from and to an address. Bluetooth mesh defines three types of address.

A unicast address uniquely identifies a single element. Unicast addresses are assigned to devices during the provisioning process.

A group address is a multicast address which represents one or more elements. Group addresses are either defined by the Bluetooth Special Interest Group (SIG) and are known as SIG Fixed Group Addresses or are assigned dynamically. 4 SIG Fixed Group Addresses have

Now if you look at elements, some nodes have multiple constituent parts each of which can be independently controlled. Now if you look at the mesh terminology, these are called elements. Now look at this picture here. There are three LED lights which are stacked one below the other. These are three different independent LED lights, okay. This is a lighting product which if added to the Bluetooth mesh network would form a single node with three elements.

Can abstract them as a single node okay, with three elements, node with three elements; element one, element two, and element three, right. Now when a node needs to query the status of other nodes you essentially or it needs control from other nodes in some way it sends a message of suitable type and if the node needs to report its status to other nodes, it also sends out a message, okay.

All communication network is message oriented. This is absolutely critical part and we will see a little bit of that as well here. Now messages are coming under acknowledged and unacknowledged. Acknowledged messages require a response essentially. That also we can see out there. Now addresses. Messages must be sent from and to an address. Bluetooth mesh defines three types of addresses.

(Refer Slide Time: 17:35)

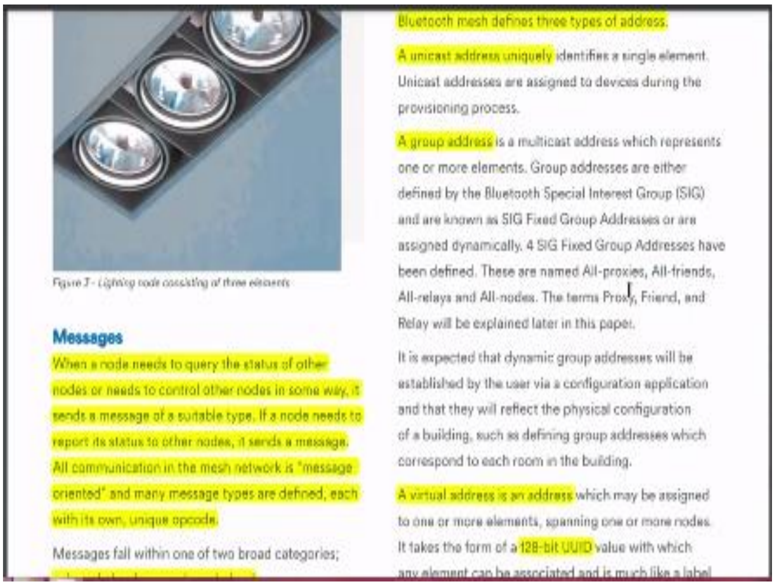


Figure 3 - Lighting node consisting of three elements

Messages

When a node needs to query the status of other nodes or needs to control other nodes in some way, it sends a message of a suitable type. If a node needs to report its status to other nodes, it sends a message. All communication in the mesh network is "message oriented" and many message types are defined, each with its own, unique opcode.

Messages fall within one of two broad categories;

Bluetooth mesh defines three types of address.

A unicast address uniquely identifies a single element. Unicast addresses are assigned to devices during the provisioning process.

A group address is a multicast address which represents one or more elements. Group addresses are either defined by the Bluetooth Special Interest Group (SIG) and are known as SIG Fixed Group Addresses or are assigned dynamically. 4 SIG Fixed Group Addresses have been defined. These are named All-proxies, All-friends, All-relays and All-nodes. The terms Proxy, Friend, and Relay will be explained later in this paper.

It is expected that dynamic group addresses will be established by the user via a configuration application and that they will reflect the physical configuration of a building, such as defining group addresses which correspond to each room in the building.

A virtual address is an address which may be assigned to one or more elements, spanning one or more nodes. It takes the form of a 128-bit UUID value with which any element can be associated and is much like a label

There is a unicast address, identifies a single element. Then you can have group address. It is a multicast address which represent one or more elements. And you can also have what are known as virtual address. This is also an address which may be assigned to one or more elements spanning one or more nodes, right. So it takes the form of 128 bit UUID value with which any element can be associated and is much like a label.

(Refer Slide Time: 18:04)

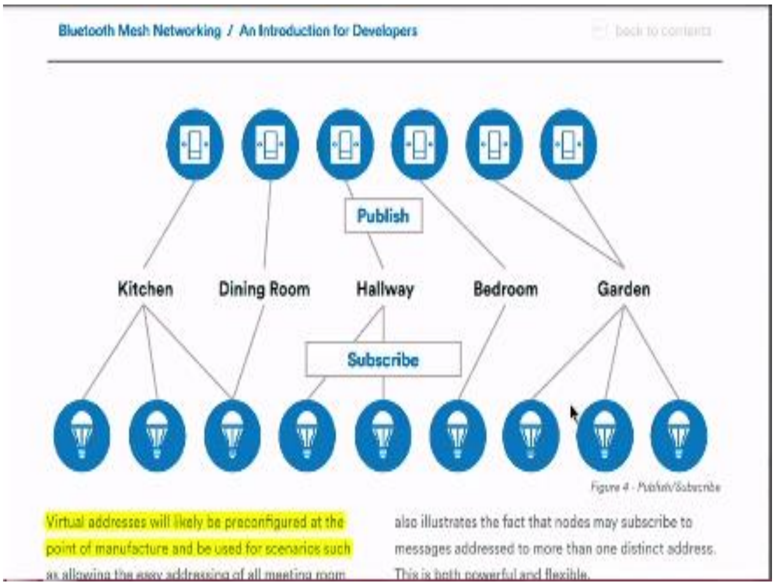


Figure 4 - Publish/Subscribe

Virtual addresses will likely be preconfigured at the point of manufacture and be used for scenarios such as allowing the easy addressing of all meeting room

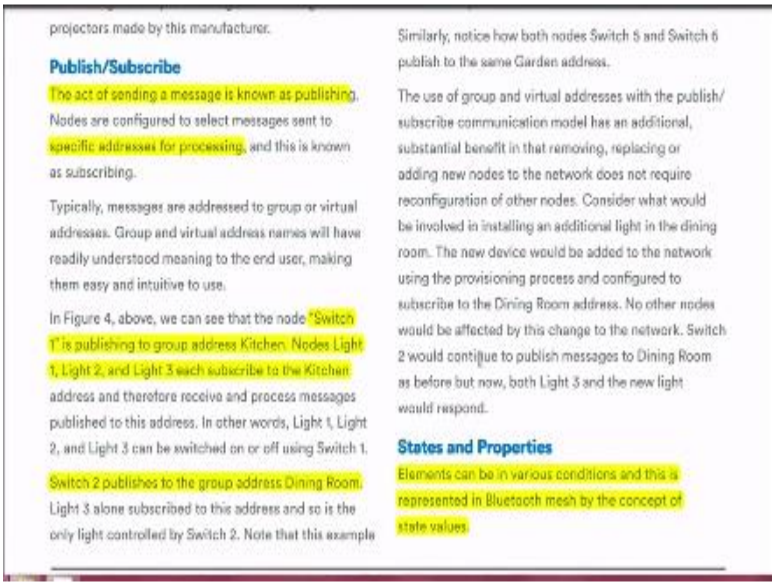
also illustrates the fact that nodes may subscribe to messages addressed to more than one distinct address. This is both powerful and flexible.

This picture is interesting, right? You have a node which is out here. It publishes data to kitchen and then there are three receivers or subscribers to it. These three subscribers get the data and depending on what the command is, that particular subscriber may actuate. It may either be on, off and so on. For example, this may have a command coming from this publisher to kitchen which says that you have to switch yourself off.

Then this simply goes off. So all of this is possible in the mesh world. So you can see these all publish and these are all subscribe, okay. It is also possible that you see this one picture like this, it is also possible to see a picture like this, where two nodes here are actually controlling this garden and then there are three subscribers to this, okay. You can think of your MQTT.

Topic can be garden and here and then there are the publishers, two of them here. And then these are subscribers. Of course you have to specify the correct wildcard routing key so that only that particular device gets actuated, whether it is on, off and so on. Alright, so that is essentially the same paradigm here, publish and subscribe. If you look at figure 4, which is this figure I suppose, yeah, this is publish subscribe.

(Refer Slide Time: 19:32)



projectors made by this manufacturer.

Publish/Subscribe

The act of sending a message is known as publishing. Nodes are configured to select messages sent to specific addresses for processing, and this is known as subscribing.

Typically, messages are addressed to group or virtual addresses. Group and virtual address names will have readily understood meaning to the end user, making them easy and intuitive to use.

In Figure 4, above, we can see that the node "Switch 1" is publishing to group address Kitchen. Nodes Light 1, Light 2, and Light 3 each subscribe to the Kitchen address and therefore receive and process messages published to this address. In other words, Light 1, Light 2, and Light 3 can be switched on or off using Switch 1.

Switch 2 publishes to the group address Dining Room. Light 3 alone subscribed to this address and so is the only light controlled by Switch 2. Note that this example

Similarly, notice how both nodes Switch 5 and Switch 6 publish to the same Garden address.

The use of group and virtual addresses with the publish/subscribe communication model has an additional, substantial benefit in that removing, replacing or adding new nodes to the network does not require reconfiguration of other nodes. Consider what would be involved in installing an additional light in the dining room. The new device would be added to the network using the provisioning process and configured to subscribe to the Dining Room address. No other nodes would be affected by this change to the network. Switch 2 would continue to publish messages to Dining Room as before but now, both Light 3 and the new light would respond.

States and Properties

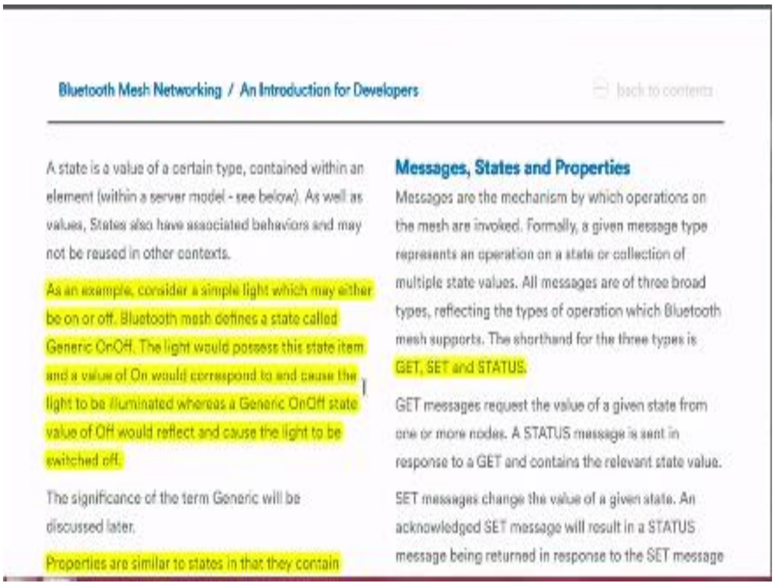
Elements can be in various conditions and this is represented in Bluetooth mesh by the concept of state values.

Switch 1 is publishing to a group address called kitchen. So this is you can think about this as the group address. These are all group addresses. And it is publishing to a group at this kitchen. Nodes light 1, light 2 and light 3 each subscribe to this group address called kitchen. And the address and therefore receive the process of message, process messages.

Light 1, light 2, light 3 each subscribe to the kitchen address and therefore receive and process messages published to this address, okay. Now switch 2, it publishes to a group called dining room. You can see this here. This is switch 2 and then it is to a dining room and the light 3 alone subscribe to it. So you can have different combinations of all of them.

Then the elements, there are what are known as states and properties. Elements can be in various conditions and this is represented in Bluetooth mesh by the concept of state values, okay.

(Refer Slide Time: 20:36)



Bluetooth Mesh Networking / An Introduction for Developers

back to contents

A state is a value of a certain type, contained within an element (within a server model - see below). As well as values, States also have associated behaviors and may not be reused in other contexts.

As an example, consider a simple light which may either be on or off. Bluetooth mesh defines a state called Generic OnOff. The light would possess this state item and a value of On would correspond to and cause the light to be illuminated whereas a Generic OnOff state value of Off would reflect and cause the light to be switched off.

The significance of the term Generic will be discussed later.

Properties are similar to states in that they contain

Messages, States and Properties

Messages are the mechanism by which operations on the mesh are invoked. Formally, a given message type represents an operation on a state or collection of multiple state values. All messages are of three broad types, reflecting the types of operation which Bluetooth mesh supports. The shorthand for the three types is GET, SET and STATUS.

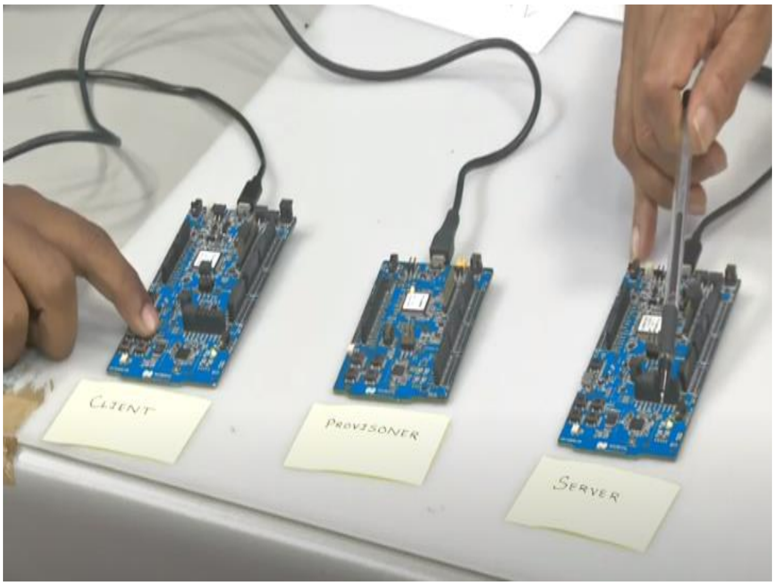
GET messages request the value of a given state from one or more nodes. A STATUS message is sent in response to a GET and contains the relevant state value.

SET messages change the value of a given state. An acknowledged SET message will result in a STATUS message being returned in response to the SET message

A state value, a state is a value of a certain type contained within an element. And an example could be consider a simple light which may either be ON or OFF, right. So Bluetooth mesh defines a state called generic ON-OFF. This is a state, name of the state.

The light would possess this state item and a value of ON would correspond to and cause the light to be illuminated whereas a generic ON-OFF state value of OFF would reflect the cause and cause the light to be switched OFF, right. So as simple as this. So let us see the demonstration and then you will perhaps be able to connect to all the discussions that we have had now.

(Refer Slide Time: 21:27)

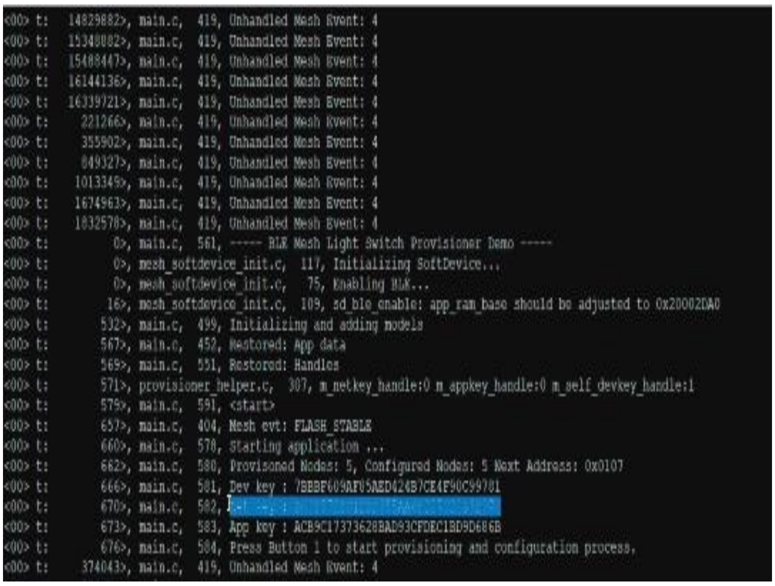


So here is the demonstration. There are three devices here. Let us start with the most important device which is the nRF52832 devices. This is the development board, you can see. And this one is the Provisioner. Here there is a client and here there is a server. Now the client will actuate the server to do something. This is a simple lightning on off example. You can see right now there is no, there are no lights on or off here.

Something after provisioning, we can actually be able to, you will be able to see that actually we have made it ready for you for the demonstration. So if a key is pressed on the client, the server will get actuated, all right. Now what should be the provisioner, what all it should do? Before you start participating in the mesh, this provisioner is a secure device. This is absolutely important, right.

It is a secure device itself and the act of provisioning is done by this provisioner and it is indeed a trusted device. And it has access to full list of devices the network and their addresses. This is absolutely critical. And every time it provisions a node either a client or a server the following will actually come through. What all will it get? It will get the unicast address and network key and a device key. All three of them will be available. We can actually have a look at that.

(Refer Slide Time: 22:57)



```
<00> t:   14829882>, main.c,  419, Unhandled Mesh Event: 4
<00> t:   15348882>, main.c,  419, Unhandled Mesh Event: 4
<00> t:   15488447>, main.c,  419, Unhandled Mesh Event: 4
<00> t:   16144136>, main.c,  419, Unhandled Mesh Event: 4
<00> t:   16339721>, main.c,  419, Unhandled Mesh Event: 4
<00> t:     221266>, main.c,  419, Unhandled Mesh Event: 4
<00> t:     355902>, main.c,  419, Unhandled Mesh Event: 4
<00> t:     849327>, main.c,  419, Unhandled Mesh Event: 4
<00> t:    1013349>, main.c,  419, Unhandled Mesh Event: 4
<00> t:    1674963>, main.c,  419, Unhandled Mesh Event: 4
<00> t:    1832578>, main.c,  419, Unhandled Mesh Event: 4
<00> t:          0>, main.c,  561, ----- BLE Mesh Light Switch Provisioner Demo -----
<00> t:          0>, mesh_softdevice_init.c,  117, Initializing SoftDevice...
<00> t:          0>, mesh_softdevice_init.c,   75, Enabling BLE...
<00> t:         16>, mesh_softdevice_init.c,  109, sd_ble_enable: app_ram_base should be adjusted to 0x20002DA0
<00> t:        532>, main.c,  499, Initializing and adding models
<00> t:        567>, main.c,  452, Restored: App data
<00> t:        569>, main.c,  551, Restored: Handles
<00> t:        571>, provisioner_helper.c,  307, m_netkey_handle:0 m_appkey_handle:0 m_self_devkey_handle:1
<00> t:        579>, main.c,  591, <start>
<00> t:        657>, main.c,  404, Mesh evt: FLASH_STABLE
<00> t:        660>, main.c,  578, Starting application ...
<00> t:        662>, main.c,  580, Provisoned Nodes: 5, Configured Nodes: 5 Next Address: 0x0107
<00> t:        666>, main.c,  581, Dev key : 7BBBF609AF05AED424B7CE4F90C99701
<00> t:        670>, main.c,  582, [illegible]
<00> t:        673>, main.c,  583, App key : ACB9C1737362BBAD93CFDEC1BD9D686B
<00> t:        676>, main.c,  584, Press Button 1 to start provisioning and configuration process.
<00> t:     374043>, main.c,  419, Unhandled Mesh Event: 4
```

So now let us see what are the keys which are part of what the provisioner does. This key is essentially the device key. This is the device key as I mentioned to you, whatever is highlighted there. And this one is the, this is the network key, whatever is highlighted now is indeed the network key. All right. So then there is another key which is called the application key.

This is perhaps for the application end-to-end you need to have a key so that you will be able to decrypt the data and you should be able to use, in a secure manner you should be able to make actuation operation, you should be able to actuate. So that is about what the provisioner does. Let us also look at what other things are possible. You also have the unicast address of the system.

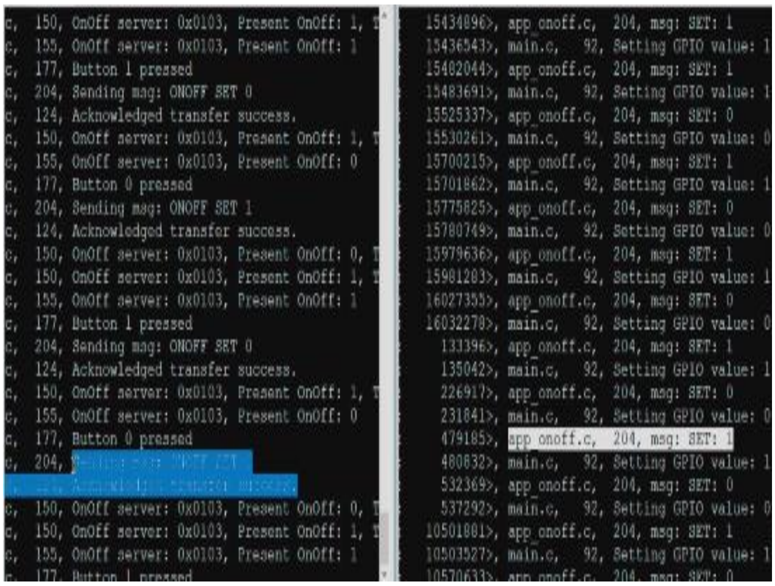
And you can see what is highlighted there indeed is the unicast address. So provisioning folks is a very involved process of several keys as well as the address being addresses being assigned to so that it now joins the network. Alright, so these are the important things. Now what we will do is having seen the provisioning part, we will now do a key press on the client side so that an actuation happens on the server side.

You do not need the provisioner anymore actually. Even without the provisioner anyway we have kept it on here, you can also keep it off. It should work without any problem. So let us see what Vasanth does. He presses a key, this is a lighting example. Ha there you are, you see he has switched on this LED. Every time he presses the LED comes up here. This is the LED of interest.

If he releases the LED is switch off and now it is on. Right, so this is essentially the lighting example I told you. Think about a light, they sent a signal so that it can switch on the lights. Now that is easy to trigger on the client side if you have a PIR sensor or any human occupancy sensor. A trigger comes and then once that trigger comes, this guy will broadcast and this server will realize that it should switch ON and then it will switch ON the light.

Whatever example I told you in the class or when what we discussed is easily realizable by interfacing several types of sensors to this board. It could be occupancy sensors or it could be HVAC sensors and so on and so forth.

(Refer Slide Time: 25:31)



```
c,  150, OnOff server: 0x0103, Present OnOff: 1, T
c,  155, OnOff server: 0x0103, Present OnOff: 1
c,  177, Button 1 pressed
c,  204, Sending msg: ONOFF SET 0
c,  124, Acknowledged transfer success.
c,  150, OnOff server: 0x0103, Present OnOff: 1, T
c,  155, OnOff server: 0x0103, Present OnOff: 0
c,  177, Button 0 pressed
c,  204, Sending msg: ONOFF SET 1
c,  124, Acknowledged transfer success.
c,  150, OnOff server: 0x0103, Present OnOff: 0, T
c,  150, OnOff server: 0x0103, Present OnOff: 1, T
c,  155, OnOff server: 0x0103, Present OnOff: 1
c,  177, Button 1 pressed
c,  204, Sending msg: ONOFF SET 0
c,  124, Acknowledged transfer success.
c,  150, OnOff server: 0x0103, Present OnOff: 1, T
c,  155, OnOff server: 0x0103, Present OnOff: 0
c,  177, Button 0 pressed
c,  204, [illegible]
[illegible]
c,  150, OnOff server: 0x0103, Present OnOff: 0, T
c,  150, OnOff server: 0x0103, Present OnOff: 1, T
c,  155, OnOff server: 0x0103, Present OnOff: 1
c,  177, Button 1 pressed
```

```
15434896>, app_onoff.c,  204, msg: SET: 1
15436543>, main.c,    92, Setting GPIO value: 1
15482044>, app_onoff.c,  204, msg: SET: 1
15483691>, main.c,    92, Setting GPIO value: 1
15525337>, app_onoff.c,  204, msg: SET: 0
15530261>, main.c,    92, Setting GPIO value: 0
15700215>, app_onoff.c,  204, msg: SET: 1
15701862>, main.c,    92, Setting GPIO value: 1
15775825>, app_onoff.c,  204, msg: SET: 0
15780749>, main.c,    92, Setting GPIO value: 0
15979636>, app_onoff.c,  204, msg: SET: 1
15981283>, main.c,    92, Setting GPIO value: 1
16027355>, app_onoff.c,  204, msg: SET: 0
16032278>, main.c,    92, Setting GPIO value: 0
  133396>, app_onoff.c,  204, msg: SET: 1
  135042>, main.c,    92, Setting GPIO value: 1
  226917>, app_onoff.c,  204, msg: SET: 0
  231841>, main.c,    92, Setting GPIO value: 0
  479185>, app_onoff.c,  204, msg: SET: 1
  480832>, main.c,    92, Setting GPIO value: 1
  532369>, app_onoff.c,  204, msg: SET: 0
  537292>, main.c,    92, Setting GPIO value: 0
10501881>, app_onoff.c,  204, msg: SET: 1
10503527>, main.c,    92, Setting GPIO value: 1
10570633>, app_onoff.c,  204, msg: SET: 0
```

Now let us see the messages. You know when a node needs to query the status of other nodes and or it needs to control other nodes in some way, it sends a message of a suitable type. And if the node needs to report its status to other nodes, it also sends a message. All communication in the mesh network is message oriented. And there are basically two types of messages.

One is acknowledged and the other is unacknowledged. So let us look at the acknowledged message, okay. So what is being highlighted now is the acknowledged message. You can see that you have an acknowledgment which says the there is a success of the sending message. Now you see, the sending message ON-OFF is set to 1.

And then you got an acknowledgment in the very next row, you see an acknowledge transfer success. So that is it folks, I would strongly encourage you to download this toolkit from Nordic and if you have the hardware, you could try practicing it. I am sure there are other vendors also who have given you several samples of this system.

You can download and install and put it up for future, building large mesh network systems. Thank you very much.