International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.

http://ijpsat.ijsht-journals.org
Vol. 24 No. 2 January 2021, pp. 119-139

Research On IoT Communication Protocols Bluetooth Mesh

Andrei Vulpe[1], Ciprian Lupu[2]

[1]Phd. Student, Faculty Of Automatic Control And Computer Science, University Politehnica Of Bucharest, Romania,

[2]Prof., Faculty Of Automatic Control And Computer Science, University Politehnica Of Bucharest, Romania,

Abstract – The paper focuses on the aspect of communication inside an IoT infrastructure. Choosing the most suitable protocol for your environment is without any doubt one of the most important things to consider. We reviewed and compared 5 of the most common protocols in terms of range, data rate, security, power consumption and scalability. The main focus is on Bluetooth Low Energy alongside Bluetooth Mesh. We managed to implement the new BLE mesh functionality on 3 real devices, being able to test the functionality and observe the protocol at work. A first possible IoT application is presented.

Keywords – IoT, communication protocols, BLE, mesh, wireless networks.

I. Introduction

The so called "Internet of Things" (IoT) is no longer about the future. IoT is here and more and more companies are investing in solutions for already well-known concepts like connected cars, smart homes or smart cities.

Technology advancements like the 5G networking and cheaper sensors of any kind will bring the IoT technologies to the front, with an estimated global economic value between 3.9 and 11.1 trillion dollars by 2025 (McKinsey Global Institute). This multi-trillion-dollar opportunity requires innovation and research like never before.

But what exactly is the internet of things?

In a definition given by [1], IoT is a system of connected computing devices, mechanical and digital machines, objects, animals or people that have unique identifiers and the ability to transfer data over a network without requiring human intervention.

An IoT system (Fig. 1) is a collection of devices based on embedded systems, such as processors, sensors and communication hardware, which acquire, send or act according to the data they collect from the environment. Even if data can also be processed locally, usually an IoT system is connected to a gateway and all the data is carried to the cloud, where it is analyzed.

Apart from offering all the attractive smart devices that contribute to the home automation and increase the comfort of living, IoT definitely has a more important impact on businesses, enabling accurate monitoring of business processes, enhancing productivity, improving customer experience and saving time and money by achieving new levels of efficiency. Finance, retail and manufacturing are currently benefiting from the usage of some sort of IoT, but businesses such as healthcare and agriculture are not far behind.

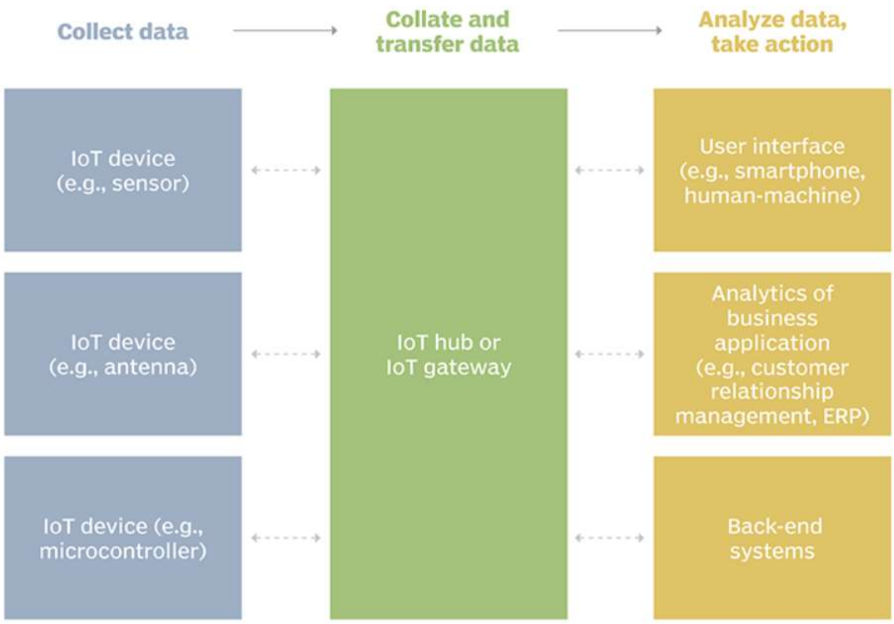

Fig. 1. Example of an IoT system[1]

Some of the challenges that the IoT world is facing are related to security and compatibility between devices from different manufactures, since there is yet no international compatibility standard for IoT.

Choosing the right connectivity protocol is an important challenge and also not an easy task. There are quite a few available protocols that could prove suitable for some kind of IoT application, but each of these has its own strengths and weaknesses.

In this paper we will revise some of the most suitable protocols for IoT connectivity, with the main focus on the Bluetooth low energy, for which we will present our own implementation of the Bluetooth mesh on a few physical devices.

Next chapter presents an analysis on a few connectivity protocols in terms of range, bandwidth, security, data rate, interoperability, scalability and power consumption.

II. IoT CONNECTIVITY PROTOCOLS

As stated before, comparing connectivity protocols is difficult [2], but one must consider the following things to perform a correct analysis:

- Range the distances at which communication happens within the IoT network.
- Even if everyone would be tempted to say that the bigger the range, the better, shorter range protocols could prove better in terms of security concerns, limiting the physical range of communication.
- Interoperability the ability of connected devices, applications or sensors to communicate with each other, especially if they are produced by different manufactures.
- With the daily advancements in the IoT world, backwards compatibility must be one of the main concerns for the future, otherwise further developments might face serious limitations.
- Security measures that are taken so that data is protected during transmission, reception and storage.
- Security is the biggest concern of the IoT world. Any application that operates with sensitive data such as, for example, medical data, must ensure protective measures are taken. A few of the methods commonly used to decrease vulnerabilities include encryption, authentication and port protection.
- Data rate the rate at which data can be transmitted through a network, usually measured in Kbps or Mbps.

This parameter depends on the range and bandwidth of the communication protocol. Its importance is closely linked to the type of application. For example, in a network of sensors that monitor occupancy in an indoor space, it might not be very critical to have as little latency as possible.

- Power consumption → the power consumed by the devices when using the selected protocol to communicate over the network.
- Apart for the need that IoT networks are eco-friendly, it is an important advantage if devices can be battery powered.
- Scalability the ability of a protocol to operate normally as the number of connected devices inside the network is increasing.

When choosing the right protocol, any future developments should be considered, to ensure responsiveness and quality of operation is kept at good levels as the network grows.

2.1. Wi-Fi

Wi-Fi is a trademark of the Wi-Fi Alliance (over 800 companies from around the world in 2017). The first version of the 802.11 protocol was released in 1997. Today, Wi-Fi connectivity is practically everywhere, with an estimate of over 8 billion devices around the world. It is the most common way for any device to connect wirelessly to local networks and to the internet. When it appeared, IoT was not taken into consideration, but now every laptop or smartphone has Wi-Fi connectivity.

Wi-Fi devices use radio waves to send signals, just like cell phones, radios or television. Communication in a wireless network works like this:

- A wireless adapter transforms any data into a radio signal and transmits it using an antenna.
- A wireless router receives the signal, decodes it and sends it to the Internet by using a physical Ethernet connection.

The mainstream Wi-Fi (802.11n or 802.11ac) uses two main frequency bands: 2.4GHz and 5GHz, allowing the signal to carry more data. However, the existence of other protocols that use the same bands make reception less clean. There are 2 main challenges: interference and contention.

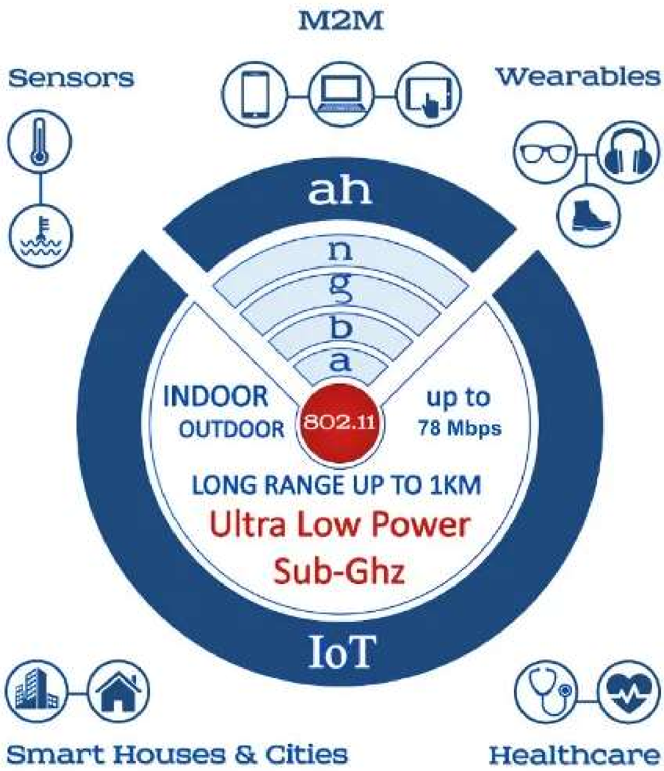

Fig. 2. Different ranges capabilities according to Wi-Fi versions[3]

With the introduction of orthogonal frequency-division multiplexing (OFDM), the interference is greatly reduced, this efficient coding technique performing a split on the signal into sub-signals before reaching the receiver.

The latest standard, IEEE 802.11ax, marketed as Wi-Fi 6 by the Wi-Fi Alliance proposes a new band of 6GHz and speeds up to 11Gbit/s.

2.2. ZigBee

The ZigBee trademark is owned by the ZigBee Alliance. It is an IEEE 802.15.4 – based specification, which defines the physical and MAC layers. The main applications include control and monitoring where the levels of data transmitted do not require a very good throughput. The range is somewhere between 10 and 100 meters[4], but the very low power consumption is definitely a key strength.

According to the protocol specification, three are 3 available bands of operation:

- 868MHz, with maximum data rate of 20kbps
- 915MHz, with maximum data rate of 40kbps
- 2.4GHz, with maximum data rate of up to 250kbps

The network topologies supported by ZigBee include:

- Star:

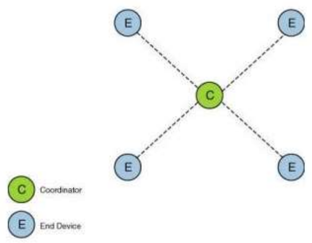

Fig. 3. Star topology[5]

All the devices connect to the coordinator node and communication happens via this coordinator. The drawback is that there is only one path from source to destination.

- Cluster tree or hybrid networks :

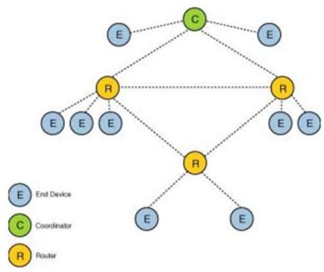

Fig. 4. Tree topology[5]

Tree topology consists of a coordinator, a few routers and many end devices. Network coverage is extended by the routers, which, as well as coordinators are considered “parent” nodes for the end devices. There is a big weakness of this topology, the fact that if a parent enters fault state, all its children will be isolated from the rest of the network.

- Mesh:

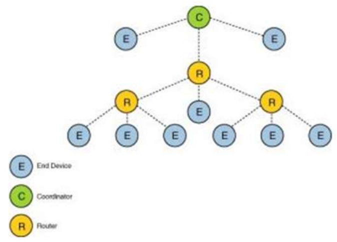

Fig. 5. Mesh topology[5]

The peer-to-peer or mesh topology also consists of a coordinator, some routers and the end devices. If one path fails during transmission, an alternative path to destination will be found, therefore eliminating the dead zones. It is easy to add or remove nodes, since they are able to communicate with any of the destination devices on the network.

2.3. Z-Wave

The Z-Wave protocol was first developed in 1999[6] by a Danish company called Zensys. In 2005, the Z-Wave Alliance emerged, which led to a great success of the protocol, with an estimate of over 2600 Z-Wave certified interoperable products in 2019.

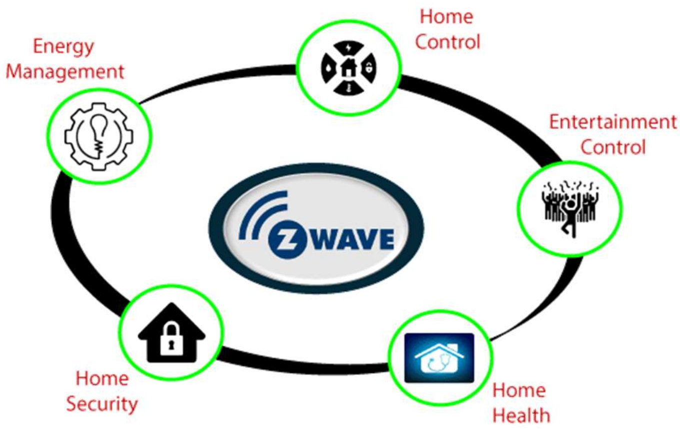

Fig. 6. Z-Wave applications

Z-Wave uses an unlicensed frequency band in the 900 MHz range, being designed to offer reliable, ultra-low power, low latency transmission of small data packets at low rates, as opposed to many other wireless protocols based on IEEE 802.11, which have high data rates capabilities. The speed ranges from 40 to 100 kbps.

The protocol is based on mesh network topology, so each installed device can become a signal repeater, provided that it does not rely on battery power. This encourages network expansion, taking into account that the maximum theoretical open air range is around 100 meters, but for indoor applications, it is recommended to have a Z-Wave device every 10 meters.

As opposed to a ZigBee network, which can support up to 65000 devices, Z-Wave has a limitation of 4 hops inside the network, with maximum 232 devices, but it is still very powerful for smart home applications.

2.4. Thread

The Thread protocol was released by the Thread Group in 2015. It uses a low-power IPv6 based protocol, 6LoWPAN, which is based on the well-known IEEE 802.15.4 wireless protocol. However, Thread is IP-addressable, AES encryption based and has cloud access capability.

Thread supports mesh networking, granting no single point of failure, provided that the network is not set up with only one edge router (or border router, as called in the protocol specification).

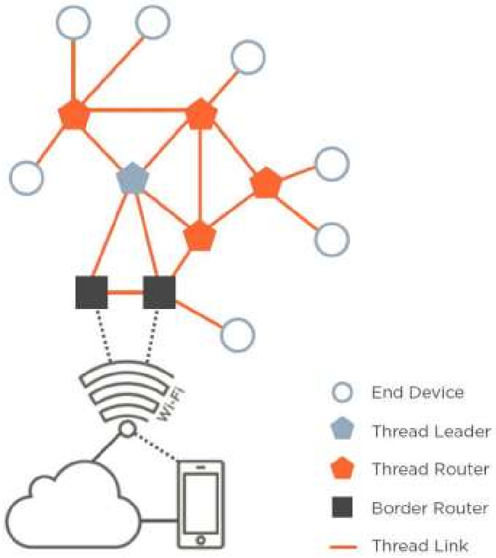

Fig. 7. Basic Thread Network Topology and Devices[7]

All the devices in a Thread network that have an IPv6 address can be accessed directly by local devices in a home area network or off-network if Thread-capable IP routers (border routers) are used.

The low-power operation is achieved using “sleepy nodes” (devices that maintain a low-power sleep state most of the time), messages to such nodes being buffered by their parents. When a “sleepy node” wakes up, it can poll its parent for any inbound messages.

Thread network can support up to 250 nodes, with 32 maximum active routers that know about all routes. It is robust network with self-healing capacity. If any router fails, the network is capable of rerouting all the traffic around it. Also, if a Leader fails, another router takes his place as the new Leader.

In terms of security, Thread is considered to be a strong protocol[8], using AES encryption for protecting networking transactions at MAC level and ECC+J-PAKE algorithms for secure addition of devices to a network.

2.5. Bluetooth Low Energy (BLE)

The classic Bluetooth wireless technology standard emerged in 1999 and is managed by the Bluetooth Special Interest Group (SIG). It uses UHF radio waves in frequency range from 2402 GHz to 2480 GHz to transmit data over relatively short distances.

Bluetooth uses a radio frequency technology called frequency-hoping spread spectrum, dividing data into packets that are transmitted on one of the 79 channels available in the frequency band(1 MHz intervals) or 40 channels in case of BLE, which uses 2 MHz spacing. With adaptive frequency-hopping(AFH) enabled, around 1600 hops are performed each second.

In 2010, the Bluetooth SIG launched Bluetooth 4.0, which first included the specifications for BLE protocol. As the name suggests, BLE is intended for extremely reduced power consumption, but it also achieves lower costs without significantly affecting the communication ranges. It is especially designed for IoT use-cases.

A BLE connected device can have one of the following 4 different functions:

- Broadcaster → used like a server, its purpose being just to transmit data to other devices.
- Observer → used like a client, monitoring data that is sent by the broadcasters.
- Central → usually a capable device such as a smartphone. It can provide two types of connections: in advertising mode or in connected mode.
- Peripheral → allows transfers of data and connections with “Central” devices.

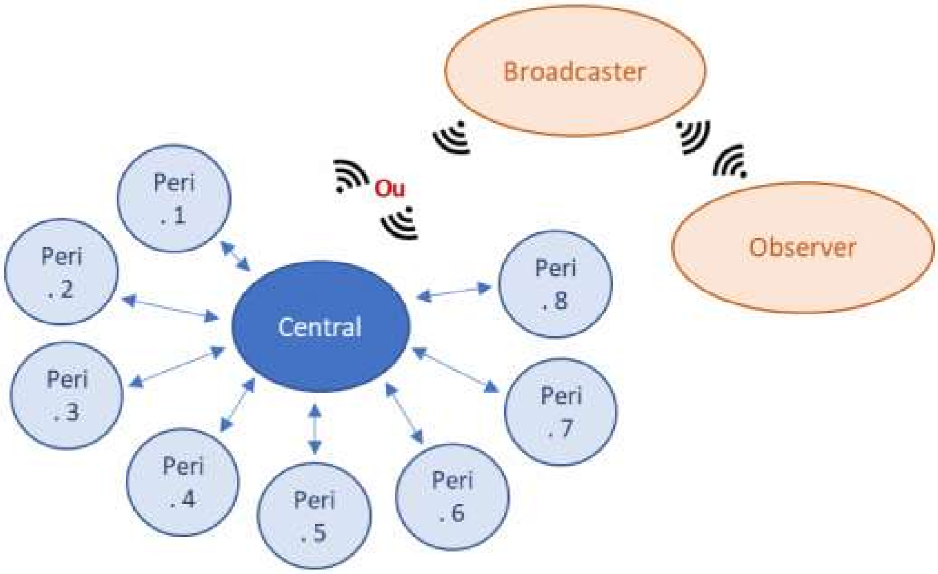

Fig. 8. Bluetooth Low Energy operation[9]

BLE enables the connection of devices based on the master-slave principle, with the GATT (Generic ATTribute) protocol being used for transmitting data.

Although Bluetooth uses a similar frequency range to WiFi, there is no other similarity. There are a few dedicated channels for advertising.

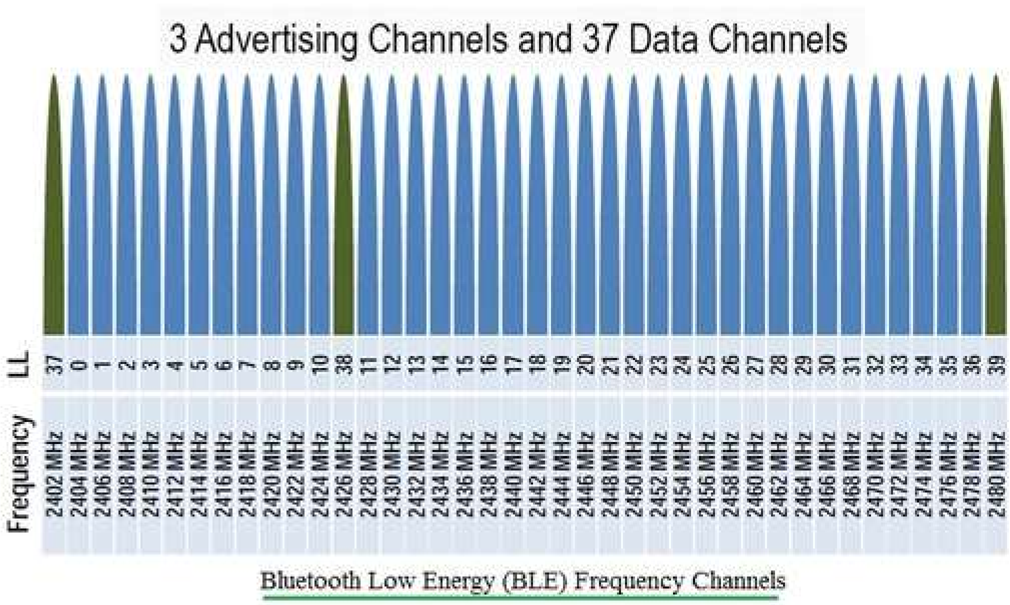

Fig. 9. BLE advertising and data channels[10]

With the appearance of Bluetooth 5.0, the range is pushed to up to 240 meters and data transfers can reach speeds of 2Mbps. Of course, these depends on the configuration of BLE parameters and there is a trade-off between range and speed. High speeds can be achieved on shorter ranges.

Table 1 presents a comparison of the 5 revised protocols with respect to frequency, data rate, mesh capabilities, power consumption and smartphone support.

Table 1. Basic protocols comparison

	ZigBee	Z-Wave	Wi-Fi	Thread	BLE with Bluetooth 5
Frequency	Both 915 MHz and 2.4 GHz	Multiple sub-GHz frequencies depending on the region	5 GHz and 2.4 GHz	2.4 GHz	2.4 GHz
Data rate	250 kbps	100 kbps	Over 1 Gbps	250 kbps	Up to 2 Mbps
Mesh capabilities	Mesh topology	Mesh topology	Limited mesh topology support	Mesh topology	Mesh topology support using Bluetooth Mesh
Power consumption	Low power consumption	Low power consumption	High power consumption	Low power consumption	Very low power consumption
Smartphone support	No support	No support	Full support	No support	Full support

Network performance of Bluetooth Mesh, Thread and ZigBee was evaluated in a benchmarking conducted by the Silicon Labs[11]. The environment used for testing was an office building which also had Wi-Fi connectivity active.

In a first experiment, the latency over 4 hops versus payload size was evaluated, with a few conclusions:

- Thread protocol achieves the best latency performance when payload size increases.
- ZigBee shows good efficiency, but with fragmentation at the application layer.
- Bluetooth mesh has bad latency when payload size increases.

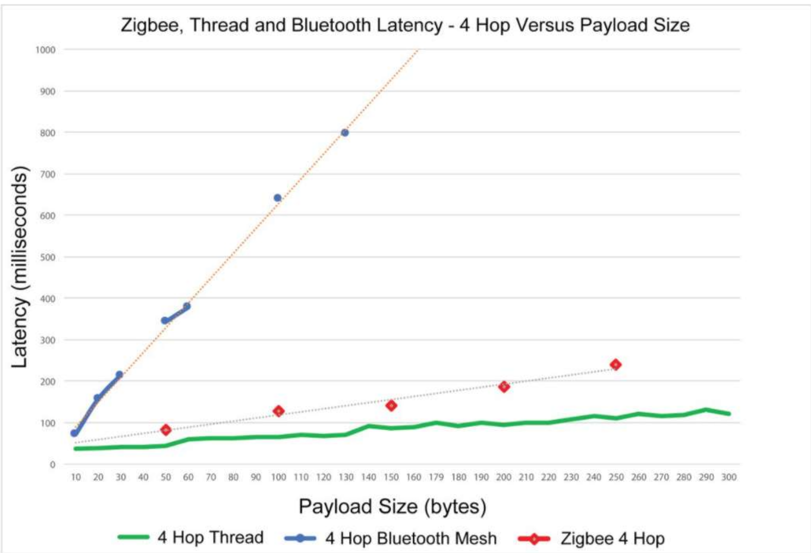

Fig. 10. Latency over 4 hops[11]

All the protocols show quite similar latency when working with small payloads (Fig. 10).

Next experiments evaluate latency while varying network and payload sizes.

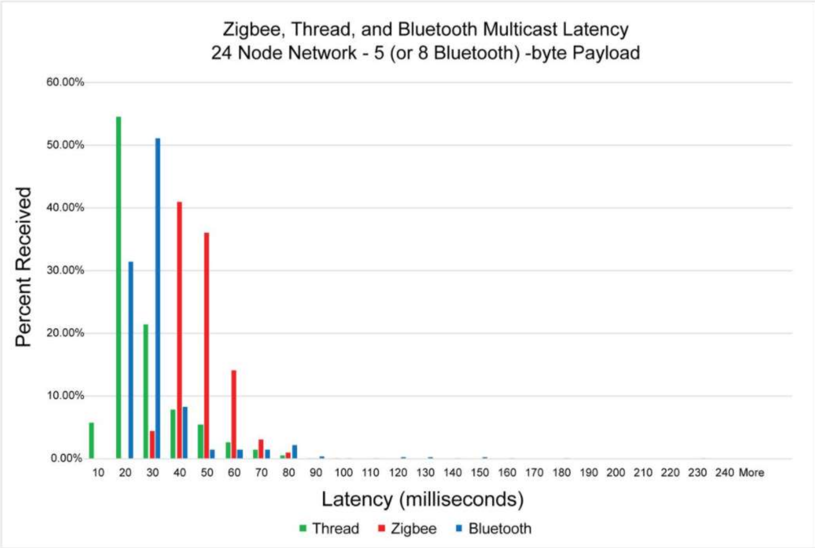

Fig. 11. Small network, small payload[11]

When transmitting small payloads (5-8 bytes) over quite small networks (24 nodes), all three protocols peak below 50ms and show high reliability for multicast (Fig. 11).

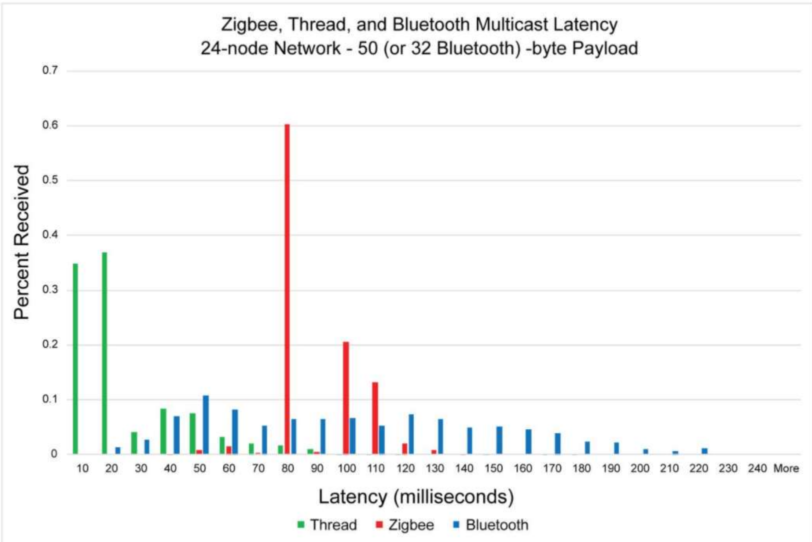

Fig. 12. Small network, bigger payload[11]

When the payload size is increased (Fig. 12), the gap between protocols begins to show up:

- Thread has the best latency (lower than 100ms).
- Zigbee receives most packets at around 80ms latency.
- Bluetooth mesh latency varies a lot (between 20 and 200ms).

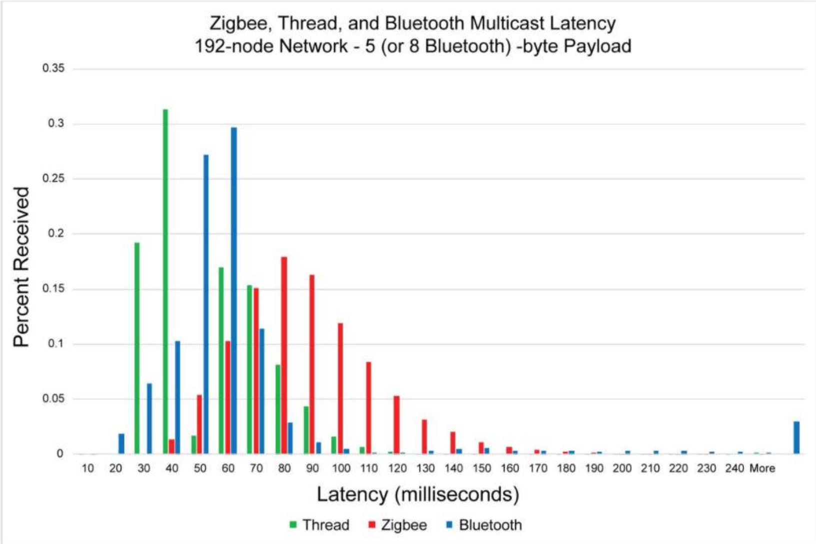

Fig. 13. Big network, small payload[11]

For a large network (192 nodes) with small payloads (5-8 bytes), Thread and ZigBee performs quite similarly as for the small network with bigger payload (Fig. 13), while Bluetooth mesh receives most packets at around 60ms latency.

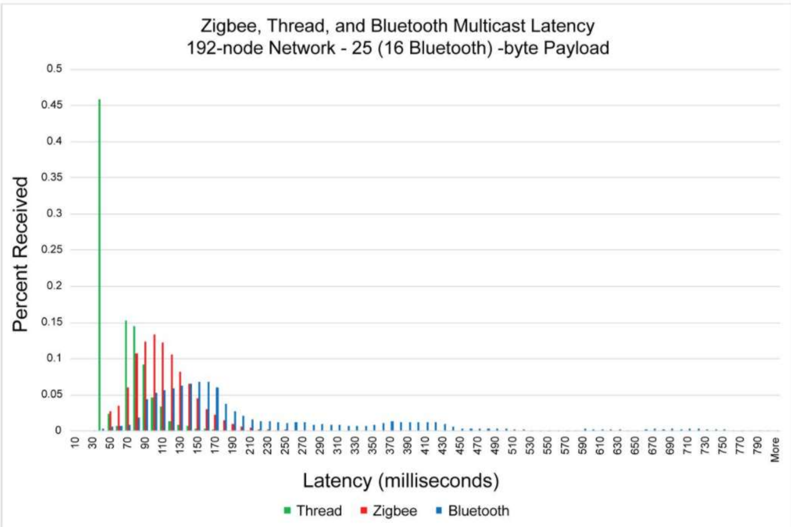

Fig. 14. Big network, big payload[11]

For big networks and big payload, Thread protocol performs best, receiving most packets at around 40ms latency.

After careful analysis of the 5 protocols, we can say that there is no best protocol for IoT. Each of the analyzed protocols could prove most suitable for different types of applications.

Apart from the frequency band, range, data rate or security of the wireless protocol, if we emphasize on the low cost of implementation, eco-friendliness (lowest power consumption), technology spread and future improvements potential, BLE (with Bluetooth Mesh) is a very strong candidate for implementing an IoT network of reasonable size.

Next section presents the Bluetooth Mesh in detail.

III. Bluetooth Mesh

The Bluetooth Mesh specification was released in 2017. The primary goal was to increase the range of Bluetooth networks and add support for more industrial applications utilizing BLE technology.

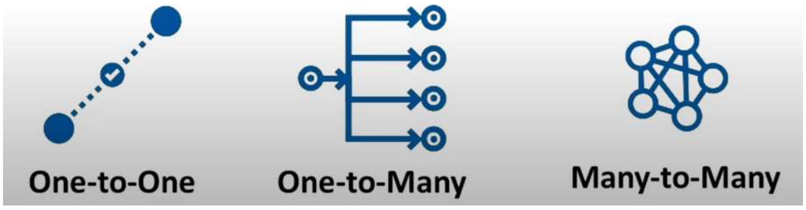

Fig. 15. BLE network topologies

Earlier version of Bluetooth supported only one-to-one and one-to-many network topologies. With Bluetooth Mesh, devices can now operate in a many-to-many topology (Fig. 15).

Bluetooth Mesh is built on top of BLE and it specifically utilizes the advertising states of BLE. Devices in a Bluetooth Mesh network do not connect to each other like traditional BLE devices do. They rather use the advertising and scanning states in order to relay messages to each other.

The Bluetooth Mesh standard defines how different devices operate in a network based on their roles. It supports all BLE versions (back to 4.0).

Devices that are part of a mesh network are called nodes, while the others are called unprovisioned devices. Once an unprovisioned device gets provisioned, it becomes a node.

Nodes within a mesh network send messages to each other. These messages are used to control the nodes, relay information between nodes or to report status to each other.

Messages are split in two categories:

- Acknowledged: response is required
- Unacknowledged: no response is required

The addresses of nodes are used to identify the source and destination of a message. There are three types of addresses:

- Unicast: uniquely identifies a node, assigned during provisioning process.
- Group address: identifies a group of nodes, SIG fixed or dynamic (defined by the user)
- Virtual addresses: assigned to one or more elements

The benefit of using groups or virtual addresses is that adding or removing nodes does not require reconfiguration of the network.

Functionality of a given element in the network is defined through models, which can be split in three categories:

- Server model → a collection of states, state transitions, state bindings and messages which an element containing the model may send or receive.
- Client model → it focuses on the messages such as the get, set and status messages sent to a server model.
- Control model → contains both the server and the client model, allowing communication with other server and client models.

The Bluetooth Mesh network introduces the concept of scenes, which is a stored collection of states identified by a 16-bit number that is unique inside the network. A scene allows setting multiple states of different nodes in one action.

Nodes can be assigned optional features that give them special capabilities inside the network:

- Relay nodes → supports the relay feature, which means that this node can retransmit messages that are broadcasted by other nodes, allowing them to traverse the network and extend the reach.

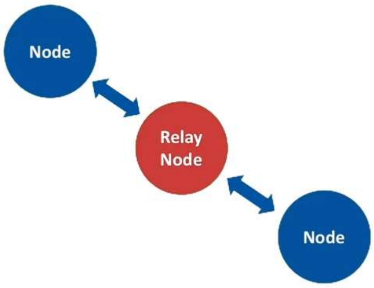

Fig. 16. Relay nodes

The TTL (Time To Live) is tightly related to the relay nodes, controlling whether a message is relayed or not. Value of TTL gives the maximum number of hops. However, a value of 1 means that the message will not be relayed further, while 0 means that the message was not and will not be relayed.

- Low power nodes → they are power-constraint nodes that need to stay asleep most of the time and turn off the radio. They are mostly concerned with sending messages (like sensors that constantly read data from the environment and send it to the network).

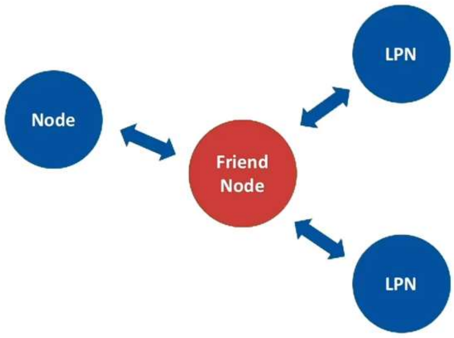

Fig. 17. LPN and Friend nodes

- Friend nodes → they are not power-constraint and need to keep the radio on all the time to be able to listen to broadcast messages, so they can relay them to the low power nodes (Fig. 17). Also, friend nodes can cash messages until the low power nodes wake up and can receive them.
- Proxy nodes → implements both the Mesh and the BLE stack. It uses GATT to expose what is called a proxy service, which allows a non-mesh device to interact with the mesh network indirectly.

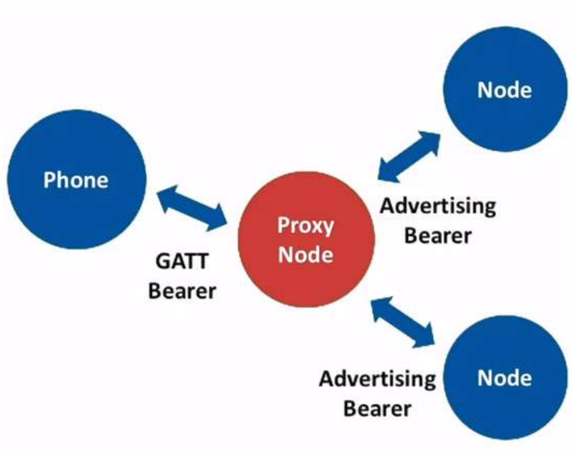

Fig. 18. Proxy nodes

In Fig. 18, the phone is not part of the mesh network, but it is able to interact with it via the proxy service exposed by the proxy node, which transforms the GATT read and write messages into mesh messages that can be sent over the advertising bearer.

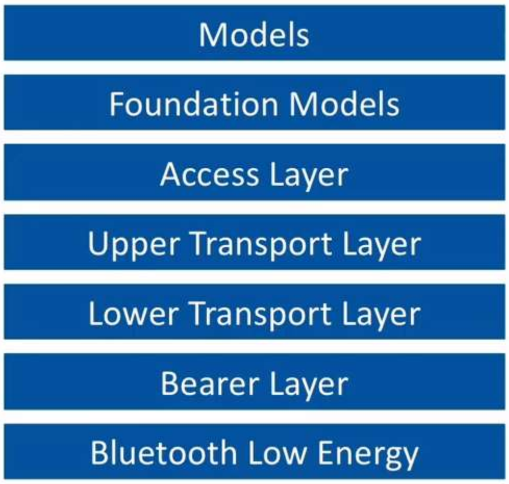

Fig. 19. Bluetooth Mesh architecture

The architecture of a Bluetooth Mesh network contains 7 layers (Fig. 19):

- BLE layer → as Bluetooth Mesh is build on top of BLE, this means it needs full implementation of the BLE stack. It uses the advertising and scanning feature of BLE (advertising bearer) and supports the GATT for proxy nodes (GATT bearer).
- Bearer layer → defines how mesh packets (PDUs) are handled inside the mesh. There are two bearers: the advertising bearer and the GATT bearer.
- Lower Transport layer → it handles the segmentation of packets from the upper transport layer and the reassembly of packets from the bearer layer.
- Upper Transport layer → it is responsible for encryption, decryption, authentication and transport control messages (heartbeat, friendship, etc.).
- Access layer → it defines how the application uses the upper transport layer, including the format of application data, defining and controlling the encryption and decryption performed in the upper transport layer and verification of data received.
- Foundation Models layer → concerned with network configuration and management models.
- Models layer → addresses the implementation of models, including behaviors, messages, states or state bindings.

In a Bluetooth Mesh network, messages are broadcasted, not routed to specific nodes, which means that all nodes receive all messages from nodes in range. However, the protocol defines a “managed flooding” technique through a few features:

- Heartbeats: transmitted by a node periodically to indicate that it is being active
- TTL: controls how far a message can traverse the network.
- Message caching: messages that already exist in the cache are immediately discarded.
- Friendship: allows nodes such as the LPNs to keep radio off for longer times and poll Friend nodes for previous mesh messages.

The most important process in a mesh network is the provisioning, which allows external devices to connect to the mesh and become nodes. A provisioner is usually a smartphone or a tablet. There are five steps involved:

1. Beaconing → the unprovisioned device announces its availability to be provisioned by sending out mesh beacon advertisements.

Fig. 20. Beaconing

2. Invitation → when the provisioner discovers the unprovisioned device via the beacons being sent, it sends an invitation to this device using a new type of PDU (Provisioning Invite Packet). The unprovisioned device responds with information about itself through a Provisioning Capabilities PDU (Fig. 21). Information sent include the number of elements supported by the device, the set of security algorithms supported, the ability for this device to output a value to the user, as well as the ability to allow a value to be input by the user.
3.

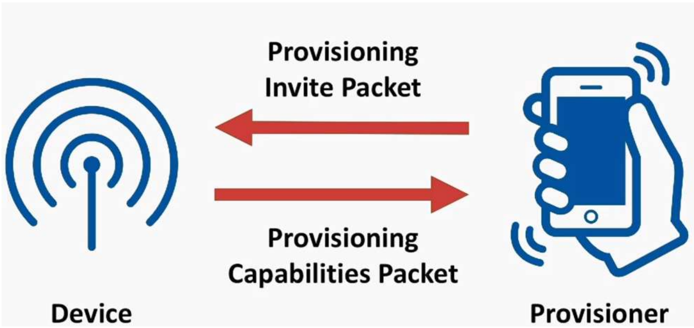

Fig. 21. Invitation

4. Public Key Exchange → security in Bluetooth mesh involves the use of a combination of symmetric and asymmetric keys such as the ECC. In this step, public keys are exchanged between the provisioner and the device to be provisioned. This is done either directly or via an out of band channel.
5.

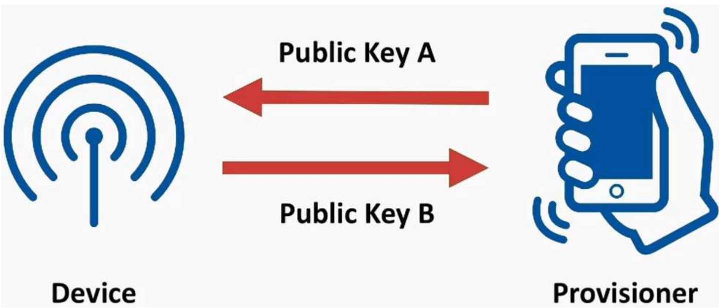

Fig. 22. Public Key Exchange

6. Authentication → the next step is to authenticate the unprovisioned device. It usually requires action by the user by interacting with both the provisioner and the unprovisioned device. The authentication method depends heavily on the capabilities of the both devices used.
7. Provision Data Distribution after authentication is complete, each device derives a session key using their private key and the public key that was received from the other device. The session key is then used to secure the connection or exchange

additional provisioning data (network key, a security parameter also known as the IV index, and the unicast address allocated)

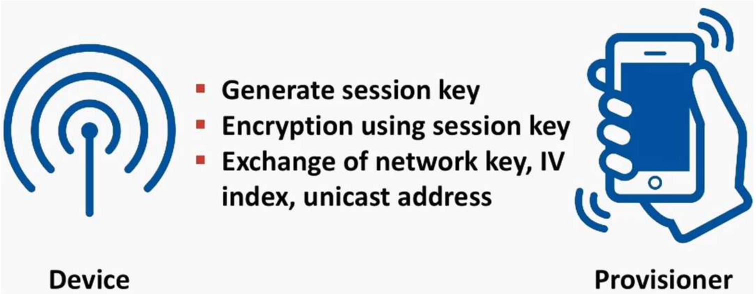

Fig. 23. Provision Data Distribution

After successful completion of all the 5 steps, the unprovisioned device becomes a node of the mesh network.

In terms of security, all messages are encrypted and authenticated in a Bluetooth Mesh network. However, network security, application security and device security are handled separately. Also, the keys can change thorough the lifetime of the mesh network.

Because of the fact that security is handled separately, there are three types of keys, each addressing a different concern:

- The network key → possession of this shared key makes the device a part of the network. There are two keys derived from the network key: the network encryption key and the privacy key. A possession of the network key allows a node to decrypt and authenticate up to the network layer, allowing relaying of messages, but not application data decryption.
- The application key → a key shared between a subset of nodes within a mesh network. Normally these nodes participate in a common application (for example, a lighting app). The key is used to decrypt and authenticate messages and is not valid across other mesh networks.
- The device key → a device specific key that is used during provisioning for securing communication between the device and the provisioner
-

Fig. 24. Security keys for Bluetooth Mesh

For security reasons, removing a node form the network includes adding it to a so called black list and refreshing of keys across the network. This prevents “trashcan attacks” → gaining access to the mesh by obtaining the keys from a previously removed node.

Privacy is achieved by using a privacy key (obtained from the network key) to obfuscate the message header. Source addresses are obfuscated also and this prevents tracking of devices.

The replay attacks, another security concern is addressed by using a sequence number (SEQ) and an IV index for messages[12].

Next section presents an actual implementation of the Bluetooth Mesh.

IV. Implementing the Bluetooth Mesh

For demonstrating the Bluetooth Mesh capabilities, we implemented a network of 3 devices which could be extended further more depending on the needs. First step for implementing the Bluetooth Mesh was choosing the appropriate hardware and software producer.

4.1. Choosing the appropriate hardware

After analyzing the list of product manufacturers agreed by the Bluetooth SIG, we chose to use nRF52840, the most advanced member of the nRF52 family from Nordic Semiconductor.

Fig. 25. nRF52840 chip[13]

Some of the key features of nRF52840 (Fig. 25) include generous Flash (1 MB) and RAM (256 KB) memories, concurrent support for protocols such as Bluetooth 5, Bluetooth Mesh, Thread, ANT, ZigBee and 802.15.4, dedicated ARM CryptoCell 310 cryptographic accelerator and numerous peripherals or interfaces such as UART, SPI, PDM, I2S, QSPI, TWI. With a 2.4 GHz transceiver, the device has over 8 dBm TX Power. Also, with an advanced adaptive power management system, very low energy consumption can be achieved, which is ideal for an eco-friendly IoT system.

4.2. Choosing the appropriate software

Our devices use FreeRTOS as operating system. Apart from the dedicated nRF52840 software development kit and the nRF5 SDK for mesh, we chose the fully compatible so-called "SoftDevice" S140, which is a Bluetooth 5 protocol stack. The SoftDevices from Nordic Semiconductor come as pre-compiled binaries which have no dependencies at runtime and reside in a dedicated memory location.

S140 Bluetooth 5 protocol stack can support 20 Bluetooth LE links concurrently for any of the four possible roles (observer, broadcaster, central or peripheral). Also, it is fully compatible with latest features, like the long range and high throughput.

We successfully integrated all the needed software stacks and developed our customized application on top.

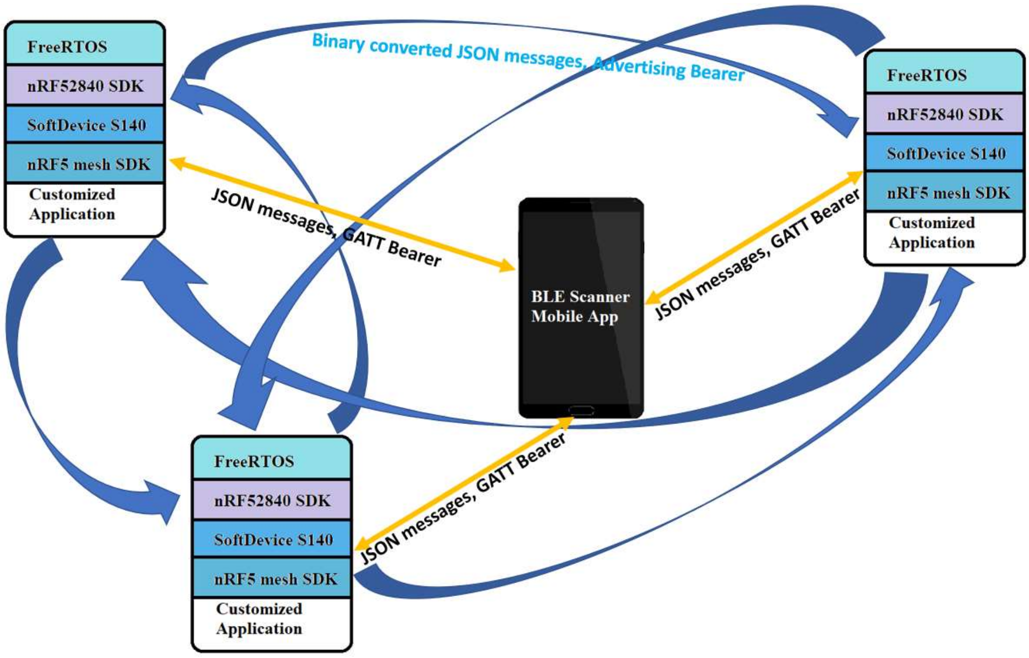

Fig. 26. Our Bluetooth Mesh implementation on 3 nRF52840 devices

For easy interaction with the mesh, we chose to use a BLE Scanner mobile application (Fig. 26). The user can scan for BLE devices and connect to the desired one. In order to easily identify the devices, we modified the advertising data with a unique name identifier.

We chose the versatile JSON format for sending/receiving messages between the mobile device and the BLE device over the GATT bearer. For increased speed and reduced payload, we transform the JSON messages to binary format when crossing the mesh network, carried by the Advertising bearer.

When connecting to a BLE device through the BLE scanner, a pairing request is first being sent. If successful, communication can be realized through Custom Service 00E0, using the Client Characteristic Configuration Descriptor. Our application waits for the user to input the authentication key. If the key matches the one that is known by the embedded device, connection is established and JSON messages can now be sent.

Our application is capable of reacting to one of the following actions:

- Setting a network (creating the mesh).
- Adding a node to an existing network (provisioning of a device).
- Setting/Updating Mesh parameters (NodeId, MeshId, NetworkId, BLEAddress, TTL, TXPower).
- Reading device specific mesh information.
- Setting data and time (using the RTC).

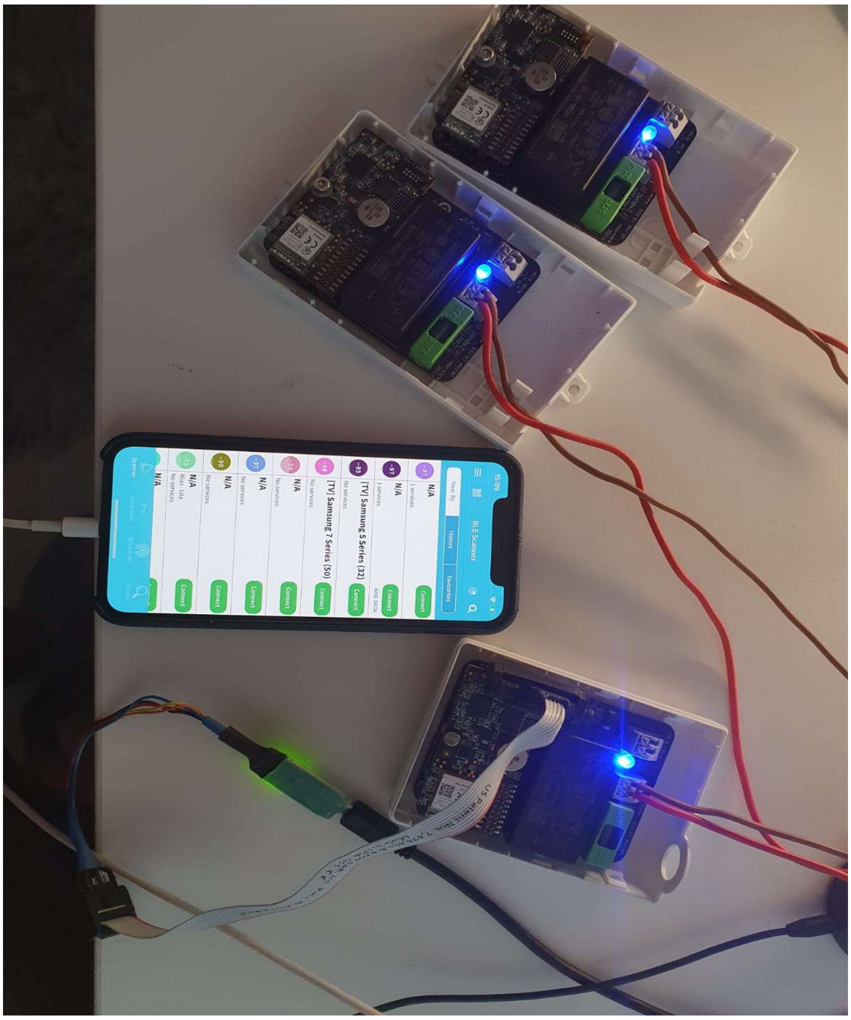

Fig. 27. The Bluetooth Mesh at work

An example of a JSON command for setting a network is shown below:

{"c": "NET","nID": "1596", "mID": "a2c455ef", "neID": "23f5", "bleA": "2563a1b47882", "ttl": "1", "txP": "08" }

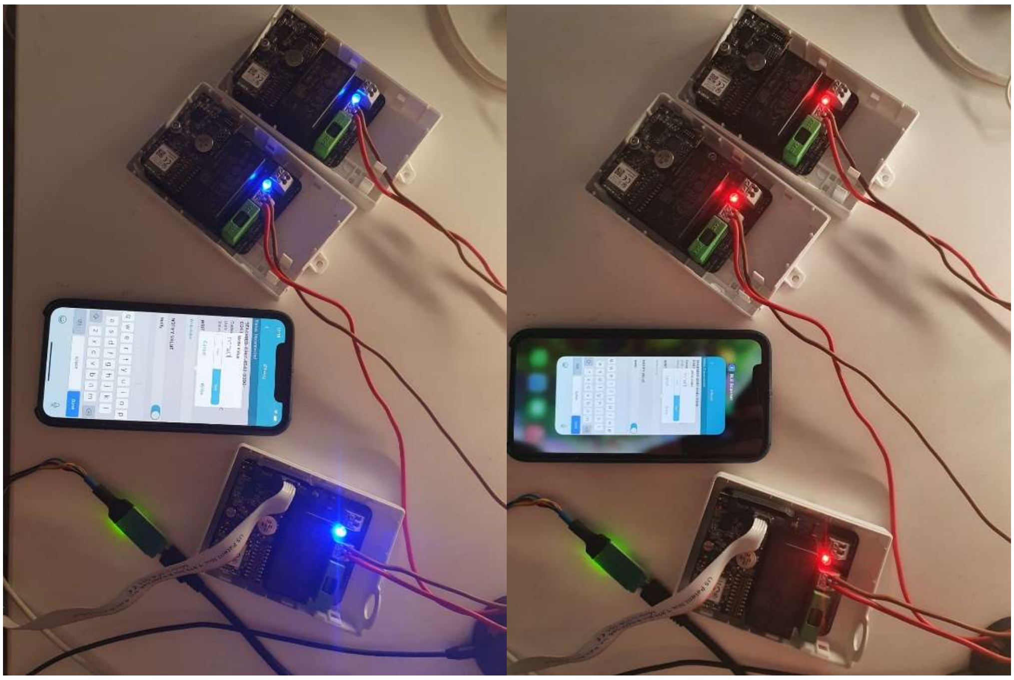

Fig. 28. Sending basic command in mesh

Our mesh network is set to use the 1 Mbps PHY, which provides a balance[12] between range and data rate. In order to test mesh basic functionality, we implemented a very simple test in which a command for setting RGB led value is being sent across the network. The response is almost instant on all three devices in the mesh (Fig. 28), no difference in timing being observable by the human eye. Also, by filming the led color transition and analyzing the video while being played at very slow speed, we can say that the difference in timing is lower than 100ms, which is ok for most IoT applications. Changing the PHY to 2 Mbps didn't show a big difference. However, the devices are in close proximity (Fig. 27).

In order to correctly evaluate the mesh performance (data rate, range, latency) using different settings, the placement of devices should also be modified. Another important aspect to consider when installing the mesh network is the RSSI. For this purpose, the best solution is to design an antenna that would increase the signal strength and offer similar coverage in all directions, so the routes across the network are predictable.

V. FUTURE DEVELOPMENTS

Having successfully implemented the mesh functionality, many IoT applications can now be developed. Normally, any application (for example: temperature, humidity, air quality, CO_2 monitoring) would require adding different types of sensor beacons to the mesh network[14]. However, we propose a simple solution for occupancy detection in a closed space like and office floor[15]. The solution relies on the fact that BLE devices are more and more popular and it is assumed that the majority of workers carry on at least one such device with Bluetooth turned ON.

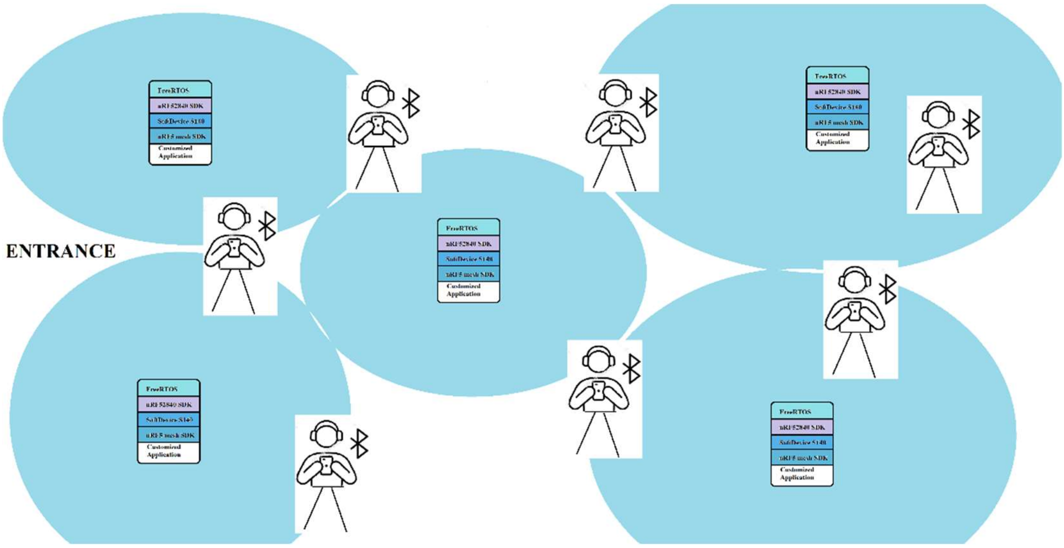

Fig. 29. Occupancy detection through Bluetooth advertising

Using the BLE advertisements and the RSSI received on each node in the mesh network (Fig. 29), the number of people inside the facility could be approximated. If we assume that the nodes would have same coverage in all directions through the use of a customized antenna, they could be installed in such a way that there would be almost no “blind spots”.

Considering that one person could carry more BLE devices at the same time, it is important to analyze the advertising packets and distinguish between different types of devices. Also, different assumptions could be made (one person does not carry 2 mobile phones, 2 headsets or 2 fitness trackers).

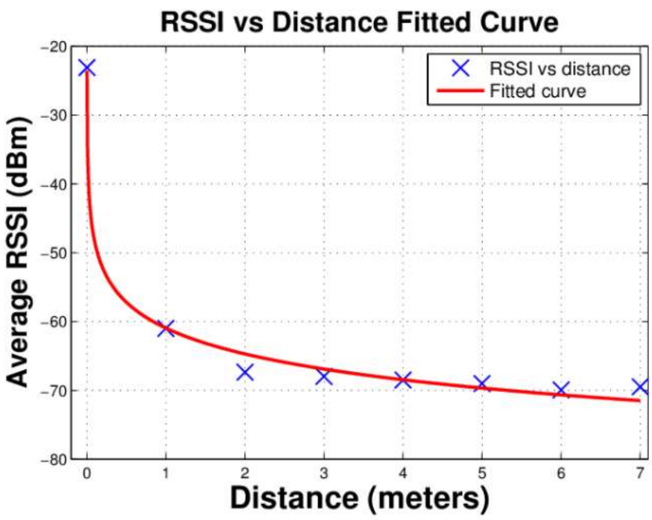

Fig. 30. Occupancy detection through Bluetooth advertising[16]

Distance between scanner and advertising device could be estimated by using only the BLE Received Signal Strength Indication (RSSI), which could be averaged over fixed periods of time for increased accuracy (Fig 30).

For best results, it would be necessary to send as much data from mesh as possible to cloud and do an extensive analysis through dedicated algorithms. However, considering GDPR regulations, storing things such as BLE addresses of devices that are being detected by the mesh network might be considered a violation of privacy.

A more simple use case would be just monitoring the number of BLE devices that are in the range of a single node and executing visual warnings in case a defined threshold is reached (led blinking red). This could be used for COVID-19 virus spreading prevention.

VI. Conclusions

This paper presented a review of the most common communication protocols to be used for implementing and IoT network. Comparing the 5 protocols we came to the conclusion that these 3 are most suitable for IoT applications: ZigBee, Thread and BLE with Bluetooth Mesh.

Considering the low cost of implementation, eco-friendliness (lowest power consumption), technology spread and future improvements potential, we chose to investigate further more the Bluetooth Mesh.

After integrating the necessary software stacks, we developed our own custom application on top. The application can execute a few basic commands like setting the mesh network, provisioning nodes, interrogating the mesh for specific details, updating mesh parameters or setting the date and time.

We used 3 nRF52840 devices to set up a mesh and tested the responsiveness of nodes when commands are being sent across the network. The presented mesh can be extended further more with different types of sensor beacons and many IoT applications can be developed on top of it. We also propose a very cost-effective solution for analyzing the occupancy of a space without the need for installing any additional sensors.

Acknowledgment

The work has been funded by the Operational Programme Human Capital of the Ministry of European Funds through the Financial Agreement 51675/09.07.2019, SMIS code 125125.

References

[1] Available [Online] at : https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT

[2] S. Tsantilas, Christos Spandonidis, "A Comparative Study of Wireless Communication Protocols in a Computer Vision System for improving the Autonomy of the Visually Impaired", Journal of Engineering Science and Technology Review, 2020.

[3] Available [Online] at : https://semiengineering.com/wifi-evolves-for-the-iot/

[4] Ievgeniia Kuzminykh, Arkadii Snihurov, Anders Carlsson, "Testing of communication range in ZigBee technology", 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics, 2017.

[5] Available [Online] at : https://www.e-spincorp.com/zigbee-network-topology/

[6] Available [Online] at : https://en.wikipedia.org/wiki/Z-Wave

[7] Available [Online] at : https://www.threadgroup.org/Portals/0/documents/support/Thread%20Fundamentals_v3.pdf

[8] Ishaq Unwala, Zafar Taqvi, Jiang Lu, "IoT Security: ZWave and Thread", IEEE Green Technologies Conference, 2018.

[9] Available [Online] at : https://elainnovation.com/what-is-ble.html

[10] Available [Online] at : https://www.rfwireless-world.com/Terminology/BLE-Advertising-channels-and-Data-channels-list.html

[11] Available [Online] at : https://www.silabs.com/products/wireless/learning-center/mesh-performance

[12] Ángela Hernández-Solana, David Pérez-Díaz-De-Cerio, Mario García-Lozano, "Bluetooth Mesh Analysis, Issues, and Challenges", IEEE Access, 2020.

[13] Available [Online] at : https://www.nordicsemi.com/Products/Low-power-short-range-wireless/nRF52840

[14] Nikita Adkar, Akshata Talele, Chinmayee Mundhe, Anita Gunjal, “Bluetooth Beacon Applications in Retail Market”, International Conference On Advances in Communication and Computing Technology, 2018.

[15] Martin Jürgens, Dennis Meis, Dominik Möllers, Felix Nolte, Etienne Stork, Gottfried Vossen, Christian Werner, “Bluetooth Mesh Networks for Indoor Localization”, 20th IEEE International Conference on Mobile Data Management, 2019.

[16] Available [Online] at : https://smartsensordevices.com/distance-measuring-solution-for-covid-19-using-bluetooth-low-energy/

[17] nRF52840 datasheet.

[18] S140 SoftDevice Specification.

[19] Mesh Profile Specification 1.0.1, Nordic Semiconductor, 2019.