Western Institute of Technology and Higher Education

Networks for Embedded Systems (Wireless)

O2023_ESI1118O

Estephania Martínez García

Practice 2 – Thread

Nelida Paulina Hernández Moya

ie727824

Mauricio Peralta Osorio

ie728593

November 22nd, 2023

Concepts

During the evolution of this procedure, we actively interacted with various essential ideas. This not only eased a better execution but also enriched our comprehension of contextualization. These principles acted as crucial foundation elements for our efforts, playing a substantial role in the comprehensive triumph of the initiative:

Thread is an open standard, IPv6-based networking protocol designed for the Internet of Things (IoT). It's specifically created to provide reliable, secure, and scalable networking solutions for smart home and building automation devices. Thread is built on existing standards, such as IPv6 and 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks), and it uses the IEEE 802.15.4 wireless protocol.

An **RTOS**, or Real-Time Operating System, is a specialized software system designed for applications that require precise and predictable timing and response. It manages hardware resources and tasks, ensuring that critical processes are executed within predefined time constraints. RTOS is commonly used in embedded systems, robotics, aerospace, and automotive applications.

The **PAN ID**, or Personal Area Network Identifier, in In Thread is a unique identifier that distinguishes one wireless network or personal area network (PAN) from another. It helps devices within a PAN to identify and connect to the correct network. PAN IDs are a fundamental part of the addressing scheme in Thread networks, ensuring that devices communicate within the intended network.

In Thread, a **channel** refers to a specific radio frequency on which devices within a wireless network communicate. The standard defines multiple channels with different frequencies, allowing for coexistence and reducing interference in wireless networks.

In Thread, a **network** refers to a collection of interconnected devices that communicate with each other using the Thread networking protocol, providing a reliable and secure framework for devices to form wireless mesh networks.

In Thread, a **router leader** is a router that assumes the responsibility of being the leader among routers. The leader is responsible for coordinating and managing the activities of other routers in the network. Some key responsibilities of a Router Leader in a Thread network include network management, path selection, device joining, routing information, network stability and synchronization.

In Thread, a **router** plays a critical role in establishing and maintaining communication within a Thread network. Here's an overview of the main functionalities of a router in the Thread IoT protocol: routing functionality, mesh networking, network infrastructure, end device communication, border router connection and routing table management.

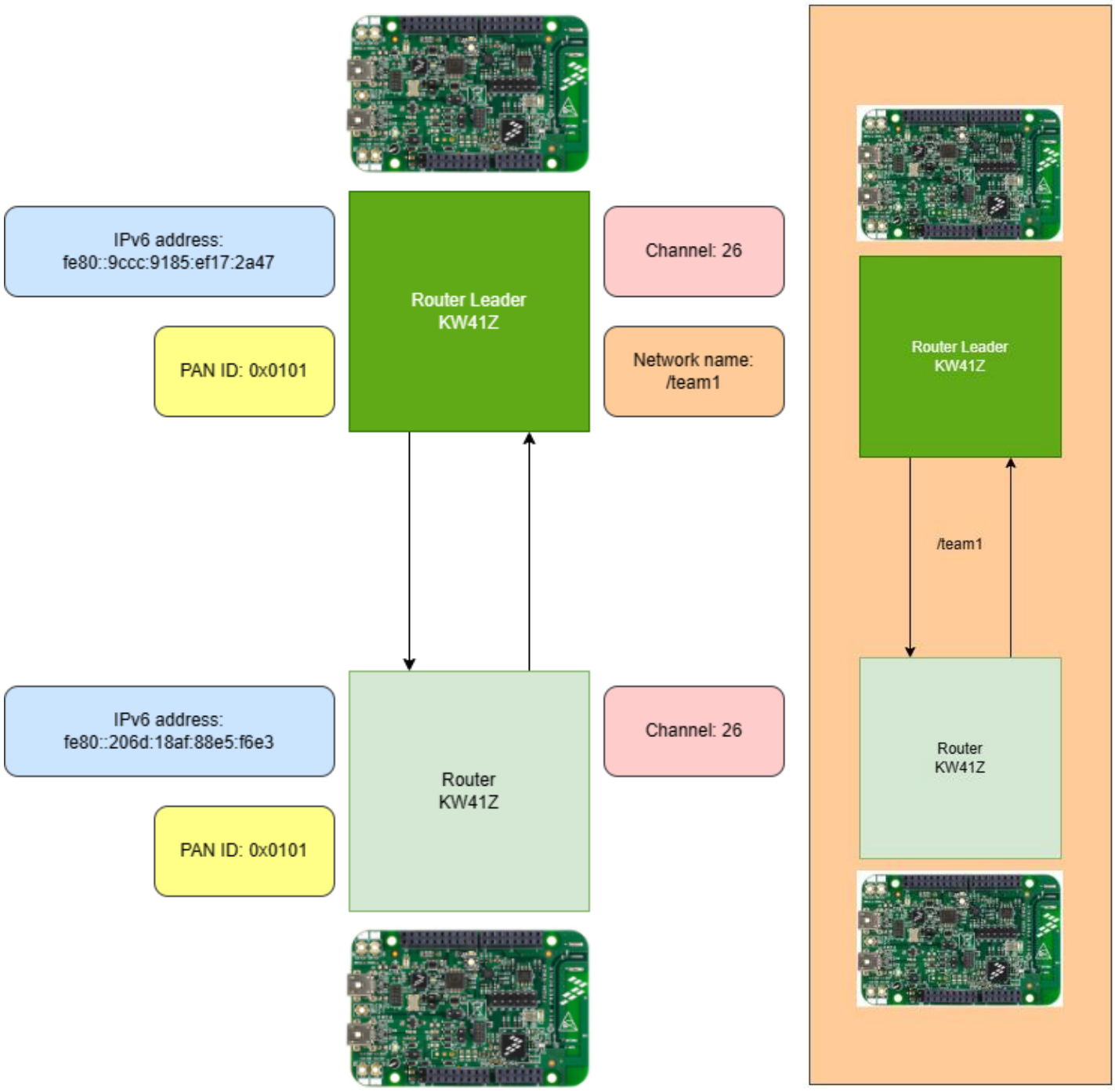
In Thread, **CoAP** (Constrained Application Protocol) is an application layer protocol designed to enable communication between devices in a constrained environment, such as those commonly found in IoT devices. Thread, being an IPv6-based networking protocol for IoT, leverages CoAP to facilitate communication between devices in a Thread network.

In Thread, **URI** stands for Uniform Resource Identifier. A URI is a string of characters that uniquely identifies a particular resource. It's a fundamental concept in networking and is widely used in the context of web communication. In the context of CoAP, URIs are used to identify resources that devices can interact with. Thread devices use CoAP to exchange information, and the URIs play a crucial role in addressing specific resources.

In Thread, **message** typically refers to the data packets that devices exchange within a Thread network. These messages are used for communication between devices, conveying information such as sensor readings, control commands, or other data relevant to the IoT application. The Thread protocol leverages the Constrained Application Protocol (CoAP) as its application layer protocol, and messages are structured according to CoAP

principles, a **Confirmable (CON) or Non-Confirmable (NON) CoAP message** is sent as a POST request to the resource identified by the correspondent URI, and the payload of the message contains a JSON object representing the temperature value.

Network Topology of the Final Setup and Network Diagram



Modifications in the Code

Both code implementations were developed based on the SDK example named router_eligible_device from the Thread protocol.

Starting with the preeminent router, distinguished by its extensive array of implementations, we initiated modifications by altering the channel, PAN ID, and network name. Subsequently, we incorporated the requisite libraries for accelerometer initialization and data reading. Additionally, we added the code by initiating definitions for URIs and associated callbacks. Supplementary definitions were integrated to facilitate timer and counter functionalities, along with their respective prototypes, URI callbacks, and paths.

To facilitate the commissioning joiner acceptance process, we allocated and initialized a timer. This timer is invoked within the commissioning handler, progressing incrementally from 1 to 200.

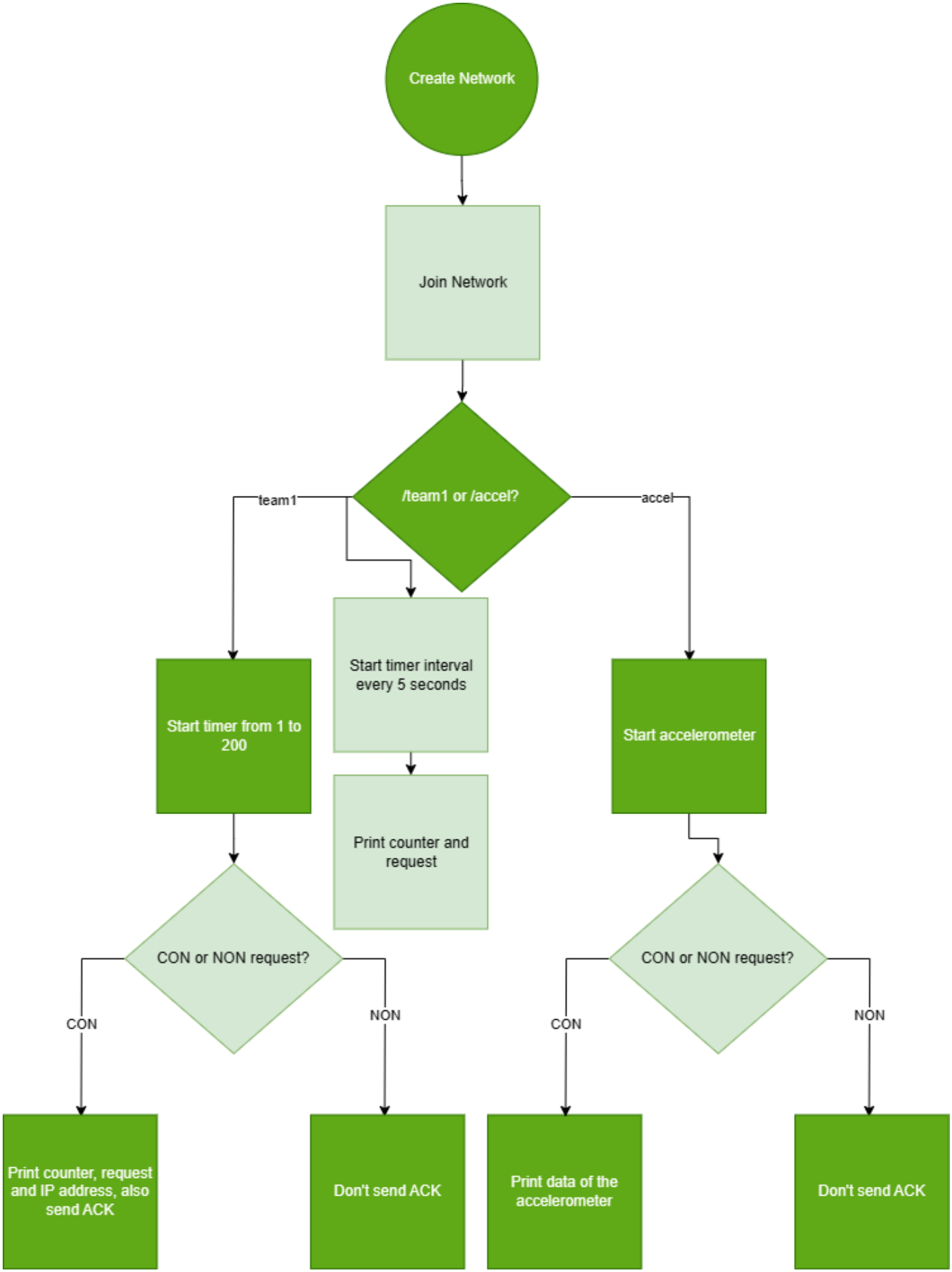
The callback functions for both /team1 and /accel closely mirrored those observed in the Thread laboratory. We declared and initialized these functions in a manner consistent with the original example. Subsequently, we refined the instructions within the CON or NON conditionals, addressing specific requirements such as sending ACK, adapting accelerometer data according to the bubble demo specifications, and other pertinent adjustments.

About the router component, the initial procedural steps closely paralleled one another, encompassing the declaration of prototypes for pivotal functions, URI, variables, and more, alongside the concomitant establishment of its corresponding callback.

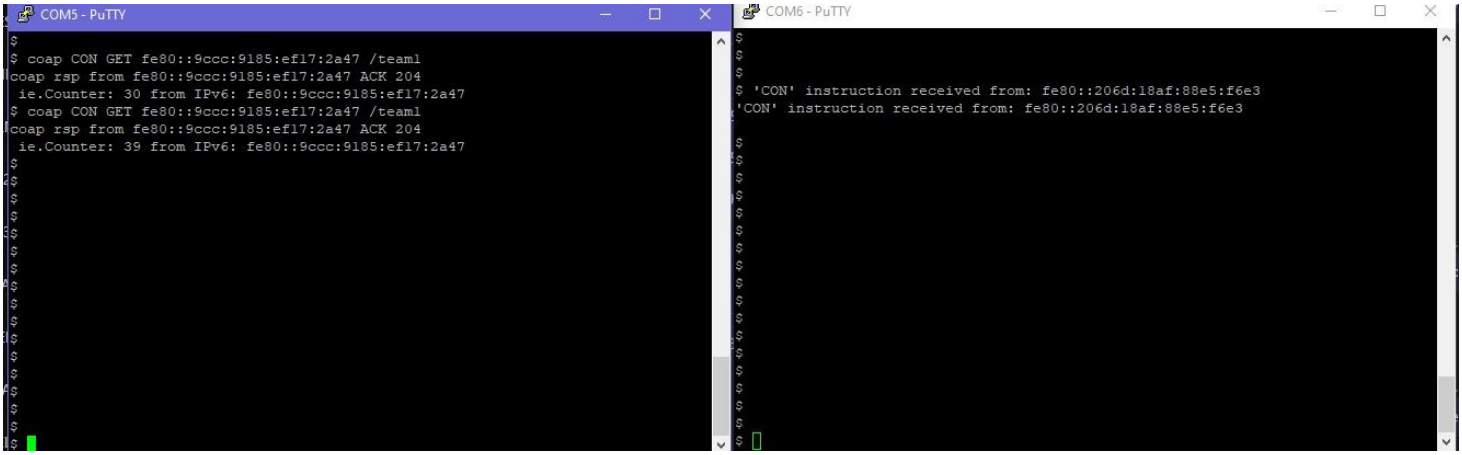
In this iteration, the timer assumed a recurrent 5-second interval, with its allocation executed in a manner consistent with prior practices. Activation transpired specifically in the event of the mesh joiner being accepted, aligning with the point at which network connection initiation takes place.

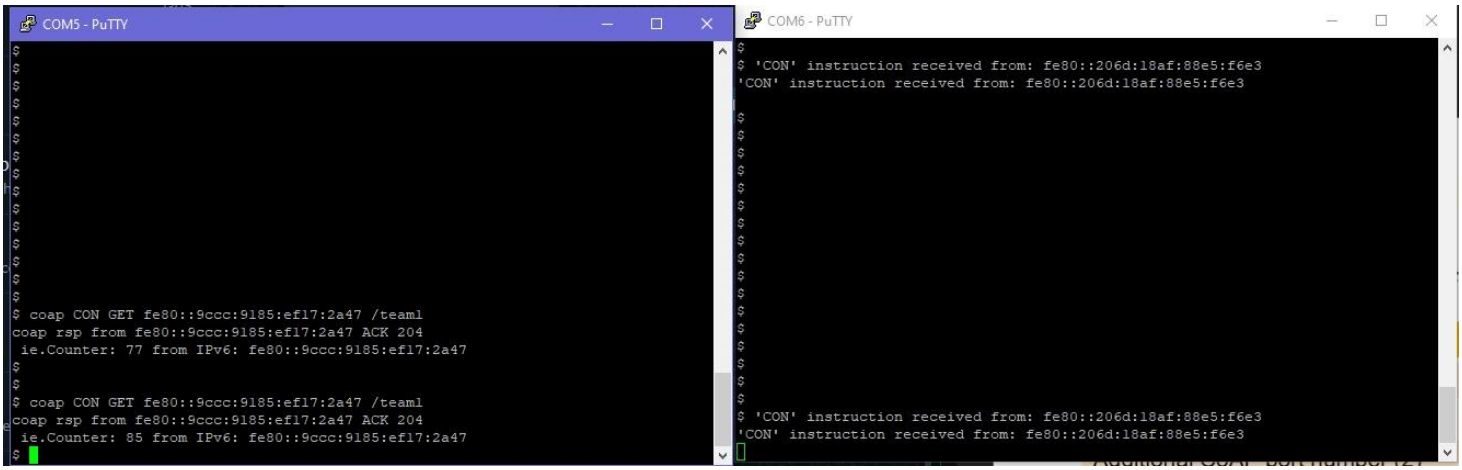
However, a noteworthy challenge emerged during this phase, as we grappled with the implementation of an automated request mechanism recurring every 5 seconds. Regrettably, the means to effectuate this periodicity in an automated fashion eluded us, and as a result, the implementation remained manual. Our attempts to replicate this automated behavior within the counter callback proved unsuccessful, highlighting a nuanced aspect of the development process where further exploration and refinement may be necessary to achieve the desired functionality.

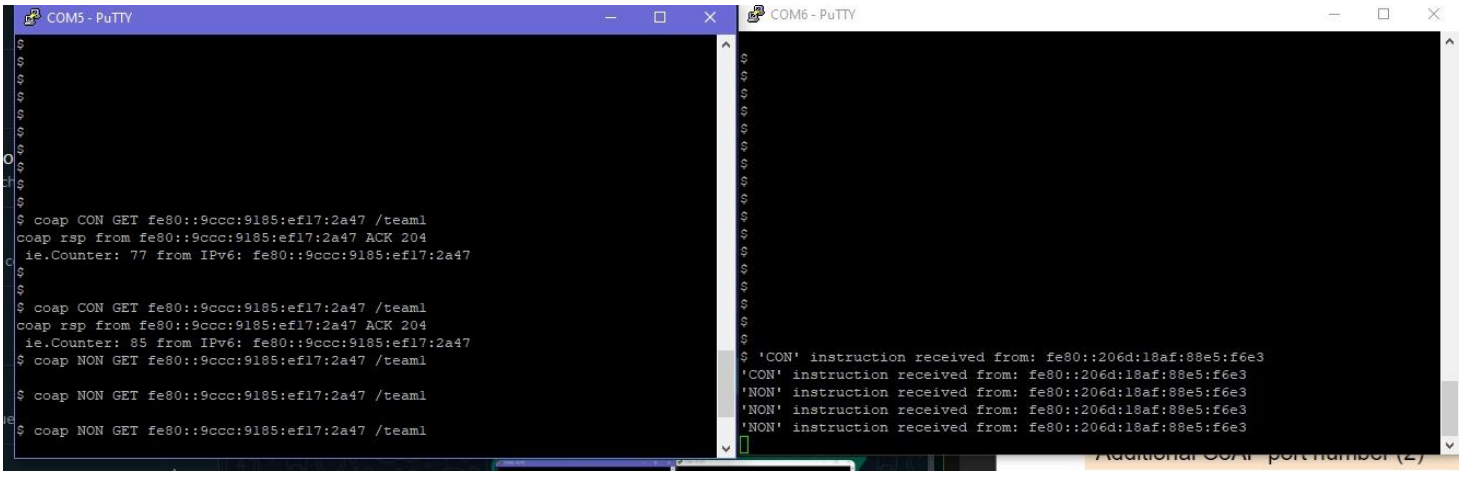
Sequence Diagram

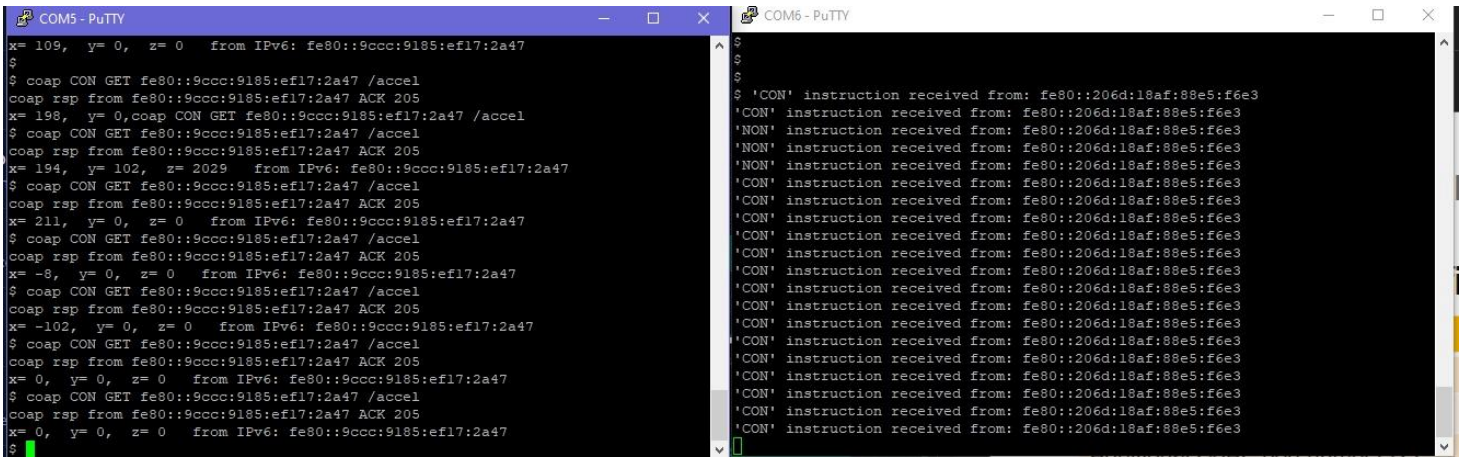


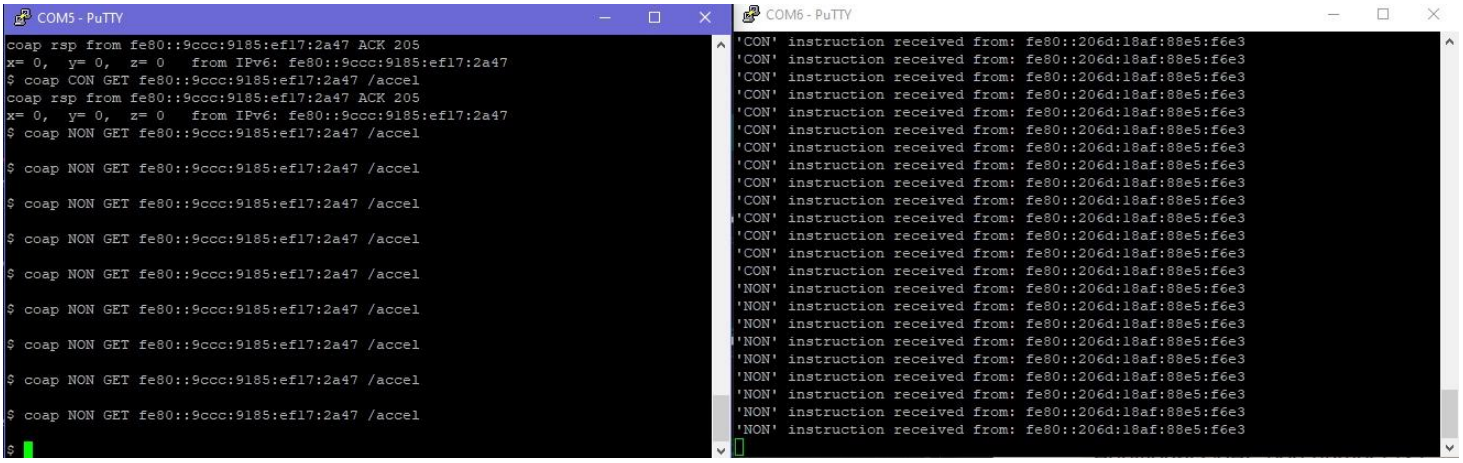
Terminal Screenshots Displaying CoAP Messages





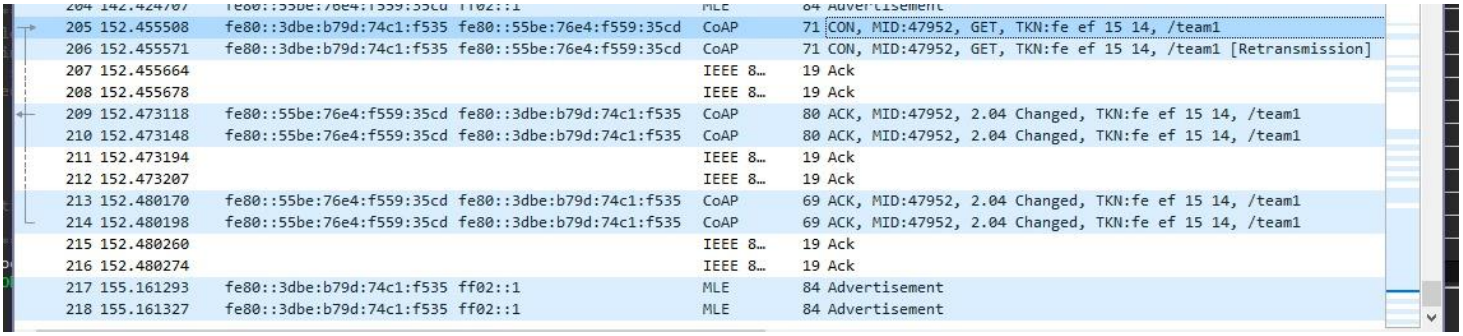




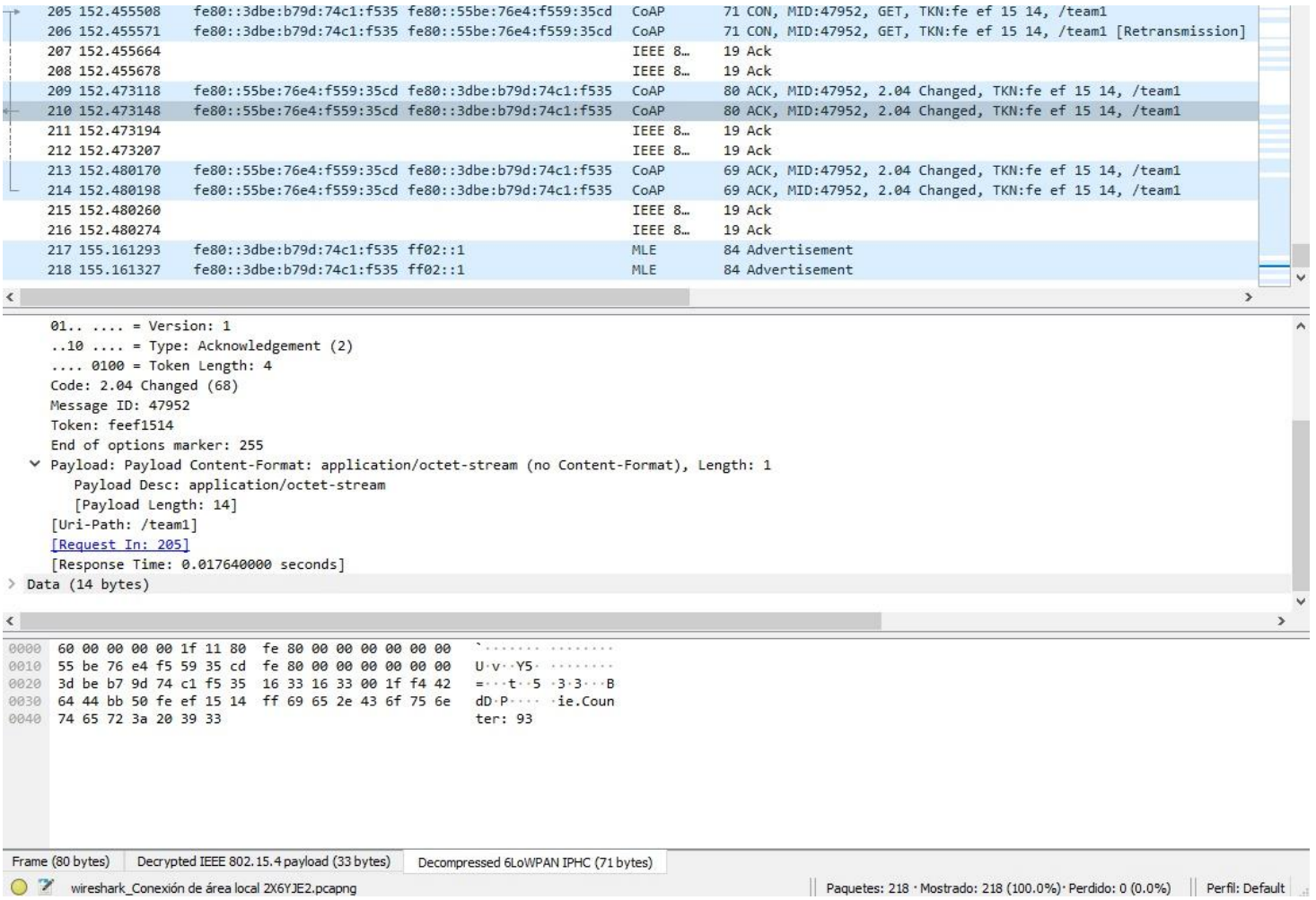


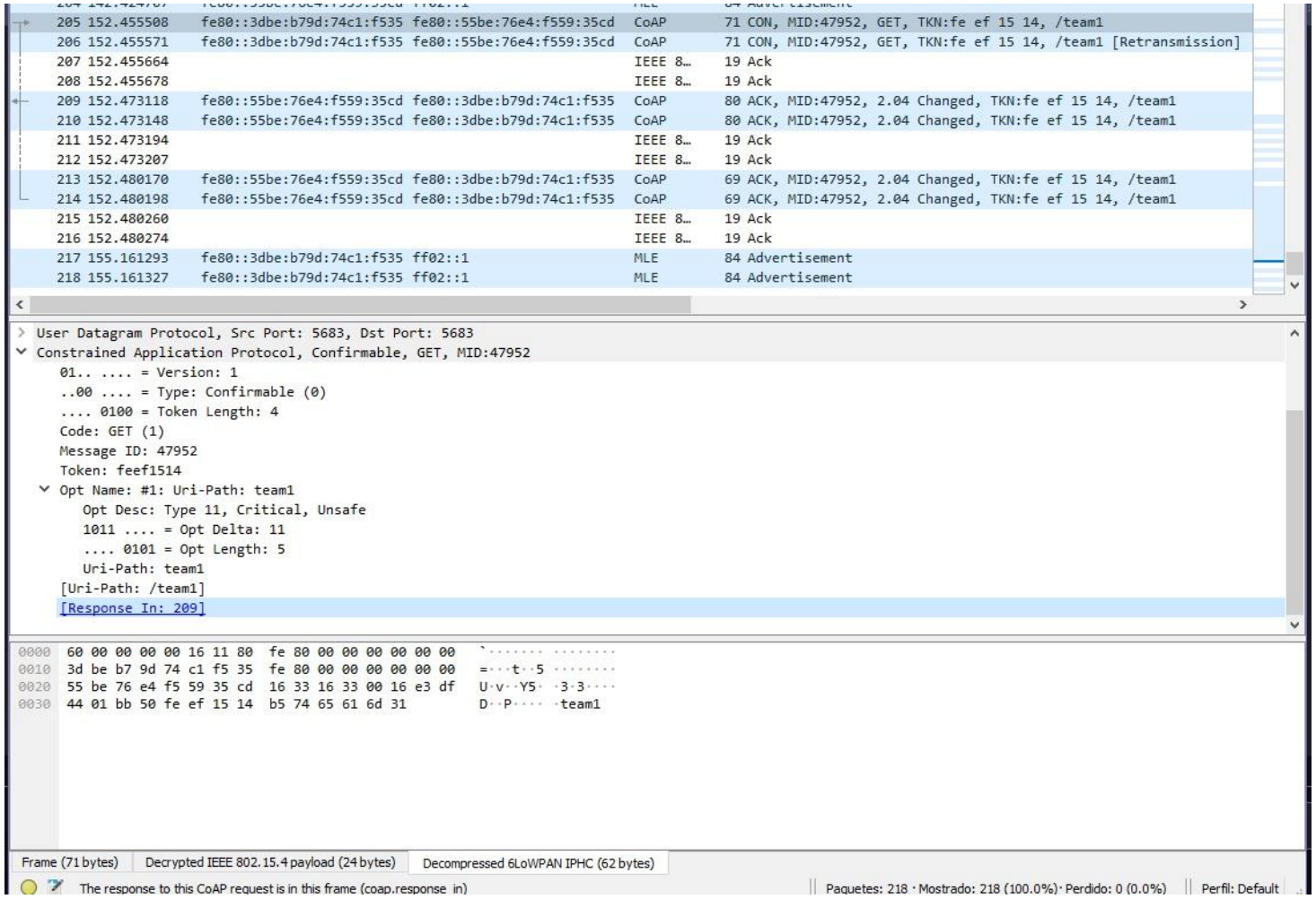
Sniffer Screenshots of the Joining Process and CoAP Messages

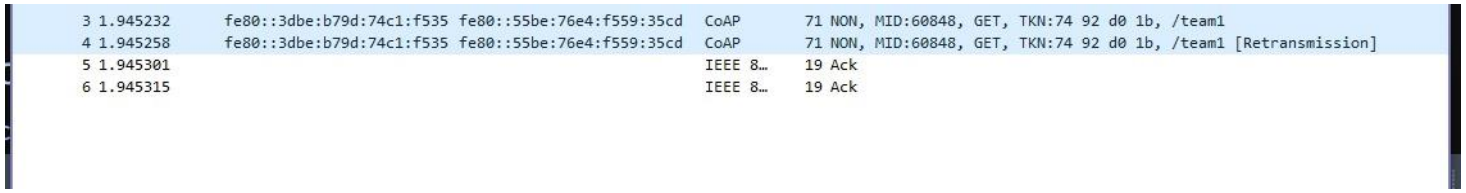
/team1



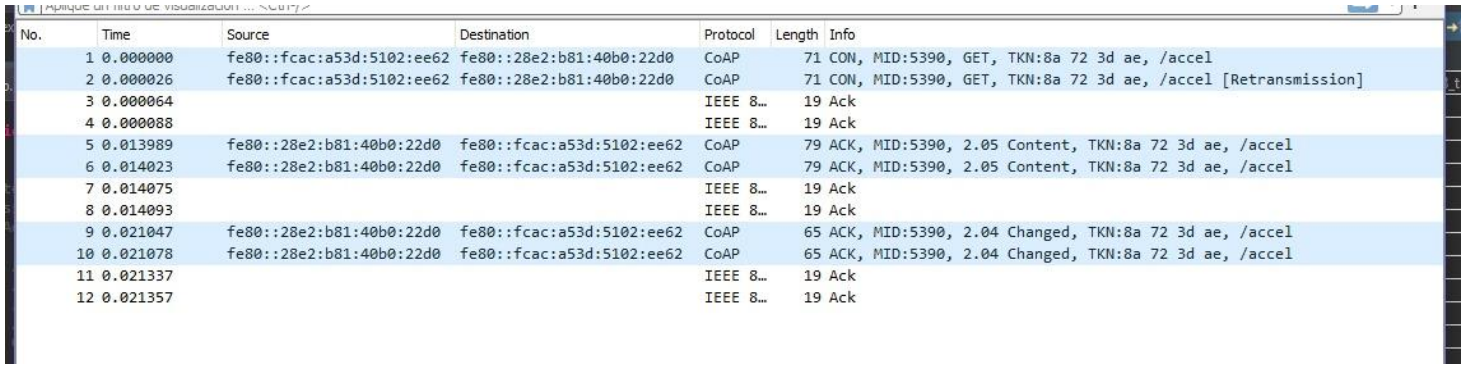
205	152.455508	fe80::3dbe:b79d:74c1:f535	fe80::55be:76e4:f559:35cd	CoAP	71 CON, MID:47952, GET, TKN:fe ef 15 14, /team1
206	152.455571	fe80::3dbe:b79d:74c1:f535	fe80::55be:76e4:f559:35cd	CoAP	71 CON, MID:47952, GET, TKN:fe ef 15 14, /team1 [Retransmission]
207	152.455664			IEEE 8…	19 Ack
208	152.455678			IEEE 8…	19 Ack
209	152.473118	fe80::55be:76e4:f559:35cd	fe80::3dbe:b79d:74c1:f535	CoAP	80 ACK, MID:47952, 2.04 Changed, TKN:fe ef 15 14, /team1
210	152.473148	fe80::55be:76e4:f559:35cd	fe80::3dbe:b79d:74c1:f535	CoAP	80 ACK, MID:47952, 2.04 Changed, TKN:fe ef 15 14, /team1
211	152.473194			IEEE 8…	19 Ack
212	152.473207			IEEE 8…	19 Ack
213	152.480170	fe80::55be:76e4:f559:35cd	fe80::3dbe:b79d:74c1:f535	CoAP	69 ACK, MID:47952, 2.04 Changed, TKN:fe ef 15 14, /team1
214	152.480198	fe80::55be:76e4:f559:35cd	fe80::3dbe:b79d:74c1:f535	CoAP	69 ACK, MID:47952, 2.04 Changed, TKN:fe ef 15 14, /team1
215	152.480260			IEEE 8…	19 Ack
216	152.480274			IEEE 8…	19 Ack
217	155.161293	fe80::3dbe:b79d:74c1:f535	ff02::1	MLE	84 Advertisement
218	155.161327	fe80::3dbe:b79d:74c1:f535	ff02::1	MLE	84 Advertisement

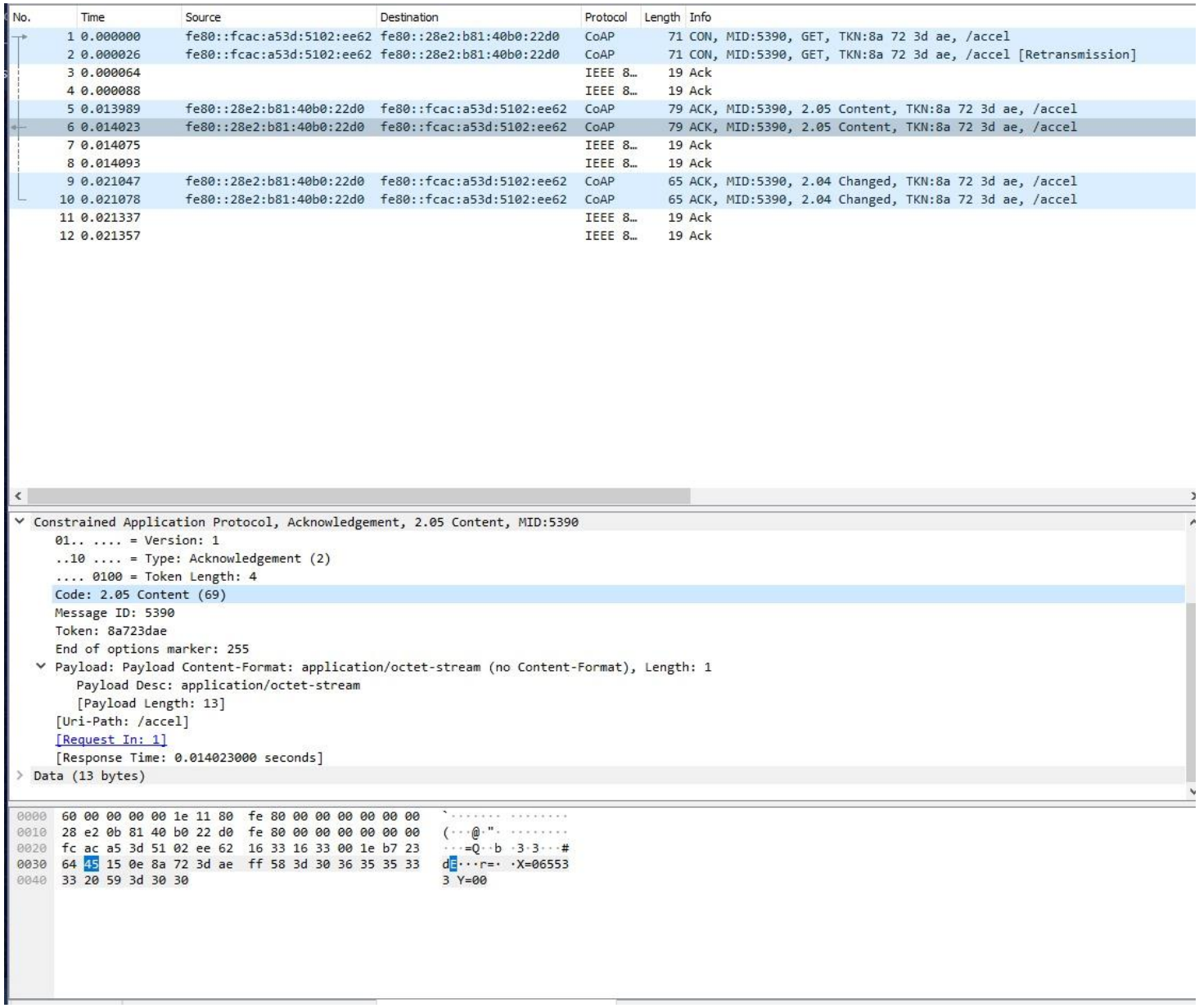

205 152.455508 fe80::3dbe:b79d:74c1:f535 fe80::55be:76e4:f559:35cd CoAP 71 CON, MID:47952, GET, TKN:fe ef 15 14, /team1
206 152.455571 fe80::3dbe:b79d:74c1:f535 fe80::55be:76e4:f559:35cd CoAP 71 CON, MID:47952, GET, TKN:fe ef 15 14, /team1 [Retransmission]
207 152.455664 IEEE 8… 19 Ack
208 152.455678 IEEE 8… 19 Ack
209 152.473118 fe80::55be:76e4:f559:35cd fe80::3dbe:b79d:74c1:f535 CoAP 80 ACK, MID:47952, 2.04 Changed, TKN:fe ef 15 14, /team1
210 152.473148 fe80::55be:76e4:f559:35cd fe80::3dbe:b79d:74c1:f535 CoAP 80 ACK, MID:47952, 2.04 Changed, TKN:fe ef 15 14, /team1
211 152.473194 IEEE 8… 19 Ack
212 152.473207 IEEE 8… 19 Ack
213 152.480170 fe80::55be:76e4:f559:35cd fe80::3dbe:b79d:74c1:f535 CoAP 69 ACK, MID:47952, 2.04 Changed, TKN:fe ef 15 14, /team1
214 152.480198 fe80::55be:76e4:f559:35cd fe80::3dbe:b79d:74c1:f535 CoAP 69 ACK, MID:47952, 2.04 Changed, TKN:fe ef 15 14, /team1
215 152.480260 IEEE 8… 19 Ack
216 152.480274 IEEE 8… 19 Ack
217 155.161293 fe80::3dbe:b79d:74c1:f535 ff02::1 MLE 84 Advertisement
218 155.161327 fe80::3dbe:b79d:74c1:f535 ff02::1 MLE 84 Advertisement
01.. .... = Version: 1
..10 .... = Type: Acknowledgement (2)
.... 0100 = Token Length: 4
Code: 2.04 Changed (68)
Message ID: 47952
Token: feef1514
End of options marker: 255
Payload: Payload Content-Format: application/octet-stream (no Content-Format), Length: 1
Payload Desc: application/octet-stream
[Payload Length: 14]
[Uri-Path: /team1]
[Request In: 205]
[Response Time: 0.017640000 seconds]
Data (14 bytes)
0000 60 00 00 00 00 1f 11 80 fe 80 00 00 00 00 00 00
0010 55 be 76 e4 f5 59 35 cd fe 80 00 00 00 00 00 00
0020 3d be b7 9d 74 c1 f5 35 16 33 16 33 00 1f f4 42
0030 64 44 bb 50 fe ef 15 14 ff 69 65 2e 43 6f 75 6e
0040 74 65 72 3a 20 39 33
Frame (80 bytes)
Decrypted IEEE 802.15.4 payload (33 bytes)
Decompressed 6LoWPAN IPHC (71 bytes)
wireshark_Conexión de área local 2X6YJE2.pcapng
Paquetes: 218 · Mostrado: 218 (100.0%) · Perdido: 0 (0.0%)
Perfil: Default


205 152.455508 fe80::3dbe:b79d:74c1:f535 fe80::55be:76e4:f559:35cd CoAP 71 CON, MID:47952, GET, TKN:fe ef 15 14, /team1
206 152.455571 fe80::3dbe:b79d:74c1:f535 fe80::55be:76e4:f559:35cd CoAP 71 CON, MID:47952, GET, TKN:fe ef 15 14, /team1 [Retransmission]
207 152.455664 IEEE 8… 19 Ack
208 152.455678 IEEE 8… 19 Ack
209 152.473118 fe80::55be:76e4:f559:35cd fe80::3dbe:b79d:74c1:f535 CoAP 80 ACK, MID:47952, 2.04 Changed, TKN:fe ef 15 14, /team1
210 152.473148 fe80::55be:76e4:f559:35cd fe80::3dbe:b79d:74c1:f535 CoAP 80 ACK, MID:47952, 2.04 Changed, TKN:fe ef 15 14, /team1
211 152.473194 IEEE 8… 19 Ack
212 152.473207 IEEE 8… 19 Ack
213 152.480170 fe80::55be:76e4:f559:35cd fe80::3dbe:b79d:74c1:f535 CoAP 69 ACK, MID:47952, 2.04 Changed, TKN:fe ef 15 14, /team1
214 152.480198 fe80::55be:76e4:f559:35cd fe80::3dbe:b79d:74c1:f535 CoAP 69 ACK, MID:47952, 2.04 Changed, TKN:fe ef 15 14, /team1
215 152.480260 IEEE 8… 19 Ack
216 152.480274 IEEE 8… 19 Ack
217 155.161293 fe80::3dbe:b79d:74c1:f535 ff02::1 MLE 84 Advertisement
218 155.161327 fe80::3dbe:b79d:74c1:f535 ff02::1 MLE 84 Advertisement
User Datagram Protocol, Src Port: 5683, Dst Port: 5683
Constrained Application Protocol, Confirmable, GET, MID:47952
01.. .... = Version: 1
..00 .... = Type: Confirmable (0)
.... 0100 = Token Length: 4
Code: GET (1)
Message ID: 47952
Token: feef1514
Opt Name: #1: Uri-Path: team1
Opt Desc: Type 11, Critical, Unsafe
1011 .... = Opt Delta: 11
.... 0101 = Opt Length: 5
Uri-Path: team1
[Uri-Path: /team1]
[Response In: 209]
0000 60 00 00 00 00 16 11 80 fe 80 00 00 00 00 00 00
0010 3d be b7 9d 74 c1 f5 35 fe 80 00 00 00 00 00 00
0020 55 be 76 e4 f5 59 35 cd 16 33 16 33 00 16 e3 df
0030 44 01 bb 50 fe ef 15 14 b5 74 65 61 6d 31
Frame (71 bytes)
Decrypted IEEE 802.15.4 payload (24 bytes)
Decompressed 6LoWPAN IPHC (62 bytes)
The response to this CoAP request is in this frame (coap.response_in)
Paquetes: 218 · Mostrado: 218 (100.0%) · Perdido: 0 (0.0%)
Perfil: Default


3 1.945232 fe80::3dbe:b79d:74c1:f535 fe80::55be:76e4:f559:35cd CoAP 71 NON, MID:60848, GET, TKN:74 92 d0 1b, /team1
4 1.945258 fe80::3dbe:b79d:74c1:f535 fe80::55be:76e4:f559:35cd CoAP 71 NON, MID:60848, GET, TKN:74 92 d0 1b, /team1 [Retransmission]
5 1.945301 IEEE 8… 19 Ack
6 1.945315 IEEE 8… 19 Ack

/accel

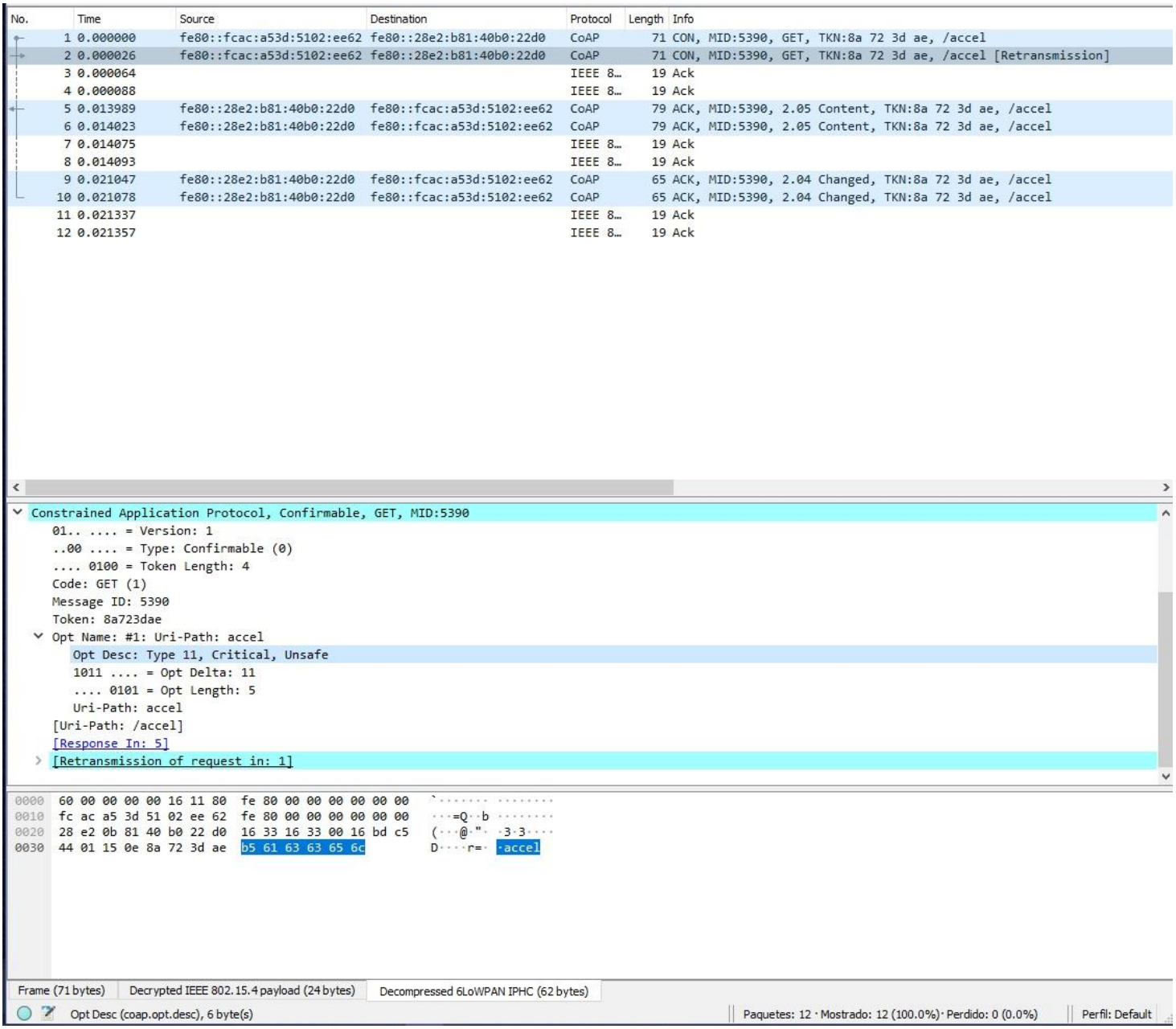

No. Time Source Destination Protocol Length Info
1 0.000000 fe80::fcac:a53d:5102:ee62 fe80::28e2:b81:40b0:22d0 CoAP 71 CON, MID:5390, GET, TKN:8a 72 3d ae, /accel
2 0.000026 fe80::fcac:a53d:5102:ee62 fe80::28e2:b81:40b0:22d0 CoAP 71 CON, MID:5390, GET, TKN:8a 72 3d ae, /accel [Retransmission]
3 0.000064 IEEE 8… 19 Ack
4 0.000088 IEEE 8… 19 Ack
5 0.013989 fe80::28e2:b81:40b0:22d0 fe80::fcac:a53d:5102:ee62 CoAP 79 ACK, MID:5390, 2.05 Content, TKN:8a 72 3d ae, /accel
6 0.014023 fe80::28e2:b81:40b0:22d0 fe80::fcac:a53d:5102:ee62 CoAP 79 ACK, MID:5390, 2.05 Content, TKN:8a 72 3d ae, /accel
7 0.014075 IEEE 8… 19 Ack
8 0.014093 IEEE 8… 19 Ack
9 0.021047 fe80::28e2:b81:40b0:22d0 fe80::fcac:a53d:5102:ee62 CoAP 65 ACK, MID:5390, 2.04 Changed, TKN:8a 72 3d ae, /accel
10 0.021078 fe80::28e2:b81:40b0:22d0 fe80::fcac:a53d:5102:ee62 CoAP 65 ACK, MID:5390, 2.04 Changed, TKN:8a 72 3d ae, /accel
11 0.021337 IEEE 8… 19 Ack
12 0.021357 IEEE 8… 19 Ack

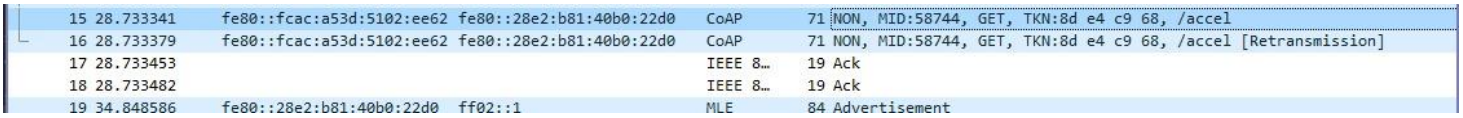


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::fcac:a53d:5102:ee62	fe80::28e2:b81:40b0:22d0	CoAP	71	CON, MID:5390, GET, TKN:8a 72 3d ae, /accel
2	0.000026	fe80::fcac:a53d:5102:ee62	fe80::28e2:b81:40b0:22d0	CoAP	71	CON, MID:5390, GET, TKN:8a 72 3d ae, /accel [Retransmission]
3	0.000064			IEEE 8…	19	Ack
4	0.000088			IEEE 8…	19	Ack
5	0.013989	fe80::28e2:b81:40b0:22d0	fe80::fcac:a53d:5102:ee62	CoAP	79	ACK, MID:5390, 2.05 Content, TKN:8a 72 3d ae, /accel
6	0.014023	fe80::28e2:b81:40b0:22d0	fe80::fcac:a53d:5102:ee62	CoAP	79	ACK, MID:5390, 2.05 Content, TKN:8a 72 3d ae, /accel
7	0.014075			IEEE 8…	19	Ack
8	0.014093			IEEE 8…	19	Ack
9	0.021047	fe80::28e2:b81:40b0:22d0	fe80::fcac:a53d:5102:ee62	CoAP	65	ACK, MID:5390, 2.04 Changed, TKN:8a 72 3d ae, /accel
10	0.021078	fe80::28e2:b81:40b0:22d0	fe80::fcac:a53d:5102:ee62	CoAP	65	ACK, MID:5390, 2.04 Changed, TKN:8a 72 3d ae, /accel
11	0.021337			IEEE 8…	19	Ack
12	0.021357			IEEE 8…	19	Ack

```
Constrained Application Protocol, Acknowledgement, 2.05 Content, MID:5390
    01.. .... = Version: 1
    ..10 .... = Type: Acknowledgement (2)
    .... 0100 = Token Length: 4
    Code: 2.05 Content (69)
    Message ID: 5390
    Token: 8a723dae
    End of options marker: 255
  Payload: Payload Content-Format: application/octet-stream (no Content-Format), Length: 1
      Payload Desc: application/octet-stream
      [Payload Length: 13]
    [Uri-Path: /accel]
    [Request In: 1]
    [Response Time: 0.014023000 seconds]
Data (13 bytes)
```

```
0000  60 00 00 00 00 1e 11 80  fe 80 00 00 00 00 00 00   `······ ········
0010  28 e2 0b 81 40 b0 22 d0  fe 80 00 00 00 00 00 00   (···@·"· ········
0020  fc ac a5 3d 51 02 ee 62  16 33 16 33 00 1e b7 23   ···=Q··b ·3·3···#
0030  64 45 15 0e 8a 72 3d ae  ff 58 3d 30 36 35 35 33   dE···r=· ·X=06553
0040  33 20 59 3d 30 30                                 3 Y=00
```





Conclusions Nelida

- **Is Thread useful? Why?**
  Yes, it is. Because is a low-power, wireless mesh IoT protocol, that excels in energy efficiency for battery-operated devices. Its mesh network ensures reliability through multiple communication paths, supporting scalability for numerous interconnected devices. Incorporating security features and compatibility with IP-based technologies, Thread prioritizes secure and interoperable communication. With a user-friendly design and industry-backed standardization taking as a base the IEEE 802.15.4, it addresses key IoT challenges, fostering adoption in diverse scenarios.
- **What went wrong in the process?**
  The challenge we encountered might stem from the router aspect and the frequency of making requests every 5 seconds. It's plausible that a lack of clarity regarding the objective of this segment complicated our implementation.

Furthermore, we faced some difficulties in normalizing the accelerometer data. The intricacy arose from our struggle to identify the precise methodology to achieve the normalization in the desired and anticipated manner.

- **Was it easy to implement?**
  Broadly speaking, I perceive the implementation as straightforward. The foundation provided by the Thread laboratory significantly facilitated our progress, particularly with URI callbacks. Adapting it to meet the specific requirements was a relatively uncomplicated task.
  Additionally, leveraging the bubble demo as a reference point for extracting raw accelerometer data played a pivotal role in simplifying the implementation of this particular aspect for our team.

Conclusions Mauricio

- **Is Thread useful? Why?**
  Thread is useful as it provides network services where you can use CoAP to manage the necessary information from sensors or their processing. Additionally, the mesh topology aspect helps ensure there's always a leader in the network, so the connection won't be lost if it fails; it can anticipate such situations. Another critical aspect is the security and encryption of data. Additionally, Thread is important because it provides a reliable and secure communication protocol specifically designed for Internet of Things (IoT) devices. It ensures robustness, low power consumption, and scalability in network connectivity, allowing for seamless interaction between devices in a secure manner.
- **What went wrong in the process?**
  We encountered issues with sending messages every 5 seconds, so we adjusted it to be manually done using CoAP GET. Moreover, we were incorrectly identifying when it was a NON or a CON, but we understood that a NON would never receive a response when making a GET request of that type.
- **Was it easy to implement?**
  The practical implementation was relatively easy once we comprehended the different message types. Everything flowed smoothly, and dealing with timers and enabling the accelerometer wasn't too complicated. It was just a matter of being cautious when initiating and closing the session while working with CoAP.

GitHub Repository

https://github.com/hanilizalo23/P2_RSE_E1_Wireless.git

Implementation Video

P2 - Thread.mp4