Project 2 Report
CSCI 43300

Riya Shrestha
Zoe Yang

1. Introduction

In this project, we will be developing a simple server on our smart device for any internet client to query the sensor readings obtained on the smart device using CoAP protocol. We will be using Raspberry Pi as the server and Google chrome on our own laptop as the client. We will be learning how to use the Copper Application for the Google chrome browser to send and receive the request to and from the server. We will also be learning how to use the Temperature and Humidity sensor, DHT11 as a resource for the CoAP server. Finally, we will be learning how to capture the packets and analyze them in Wireshark software.

Hardware platform used:

- Raspberry Pi 4
- Laptop

Software platforms used:

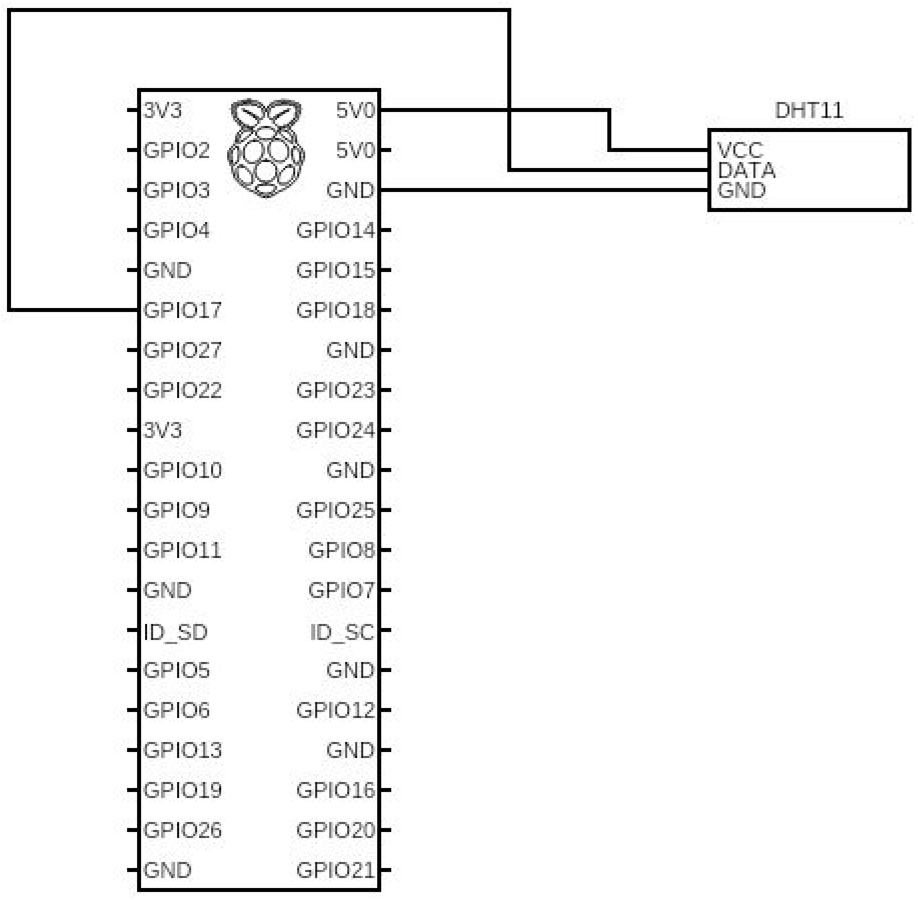
- Raspberry Pi OS
- macOS Catalina
- Python programming language
- CoAPthon library
- Copper Extension and Application
- Google chrome browser
- Wireshark

We used the Temperature and Humidity sensor DHT11, breadboard, female to male wires to build the circuit in this project.

2. Project design

Step 1:
We followed the step by step tutorial from the github link to install the CoAPthon library in python for the CoAP protocol and configured the development environment. Then we followed the tutorial to install the Copper extension and application for the Google chrome browser to run it as a client. Finally, we ran the “Hello World” application on the server which could be accessed on the client.

Step 2:
Since we chose to use the Temperature and Humidity sensor DHT11 for this project, we started by installing the Adafruit package. Then we connected the sensor DHT11 to the raspberry pi to build a circuit. As our DHT11 came in with a built in resistor, we were not required to connect an extra resistor in the circuit.
This is the circuit diagram for connecting the DHT11 to the raspberry pi.



We ran some example programs to check if our sensor was working well. After knowing that the sensor works well and is able to produce proper readings, we made a my_resource.py file and a my_coap_server.py file. In the resource file, we imported the Adafruit_DHT library for the DHT11 sensor. Then, we created a class for our sensor as a resource, for getting the readings from the sensor and passing the results to the payload. The class defines four main methods: render_GET(), render_PUT(), render_POST() and render_DELETE(), which will be triggered when the client sends any request respectively to the server.

In the server file, we imported the CoAP library and the DHT11 resource from the resource file. We then added our resource which is the DHT11 sensor to the server class and entered the IP address of our raspberry pi device to the CoAP server. In our case the IP address of the raspberry pi is 192.168.0.107.

Finally, in the raspberry pi, which is our server, we ran the my_coap_server.py program to start the server. On the client machine which is the laptop, we started our Copper application on the Google chrome browser by entering the ip address of the raspberry pi to connect to the server. After the connection is established between the client and the server, we pressed the button ‘Discover’ to look for our DHT11 resource which gets displayed on the left side of the Copper application. Then we selected our DHT11 resource and clicked the ‘“GET” button which sends the request to the server. The server then processed the request by displaying the sensor reading on the payload and sending the acknowledgement to the client.

Step 3:
We used Wireshark software to analyze the CoAP packets captured. We first filtered all the packets to display only the CoAP packets by typing “coap” on the packet display filter field. We observed two types of packets: CON and ACK. The CON packet is the GET request from the client to the server and the ACK packet is the Response from the server to the client.

Analysis of CON packet:
- Packet Number: 65
- Time since beginning of the capture: 27.645154
- Length: 56
- Source MAC Address: 3c:22:fb:79:00:d0
- Destination MAC Address: dc:a6:32:97:8d:46
- Source IP Address: 192.168.0.101
- Destination IP Address: 192.168.0.107
- Source Port: 64322
- Destination Port: 5683
- Code: GET (1)
- Message ID: 53852
- Uri Path: MyDHT11
- Uri Path (Hexadecimal):

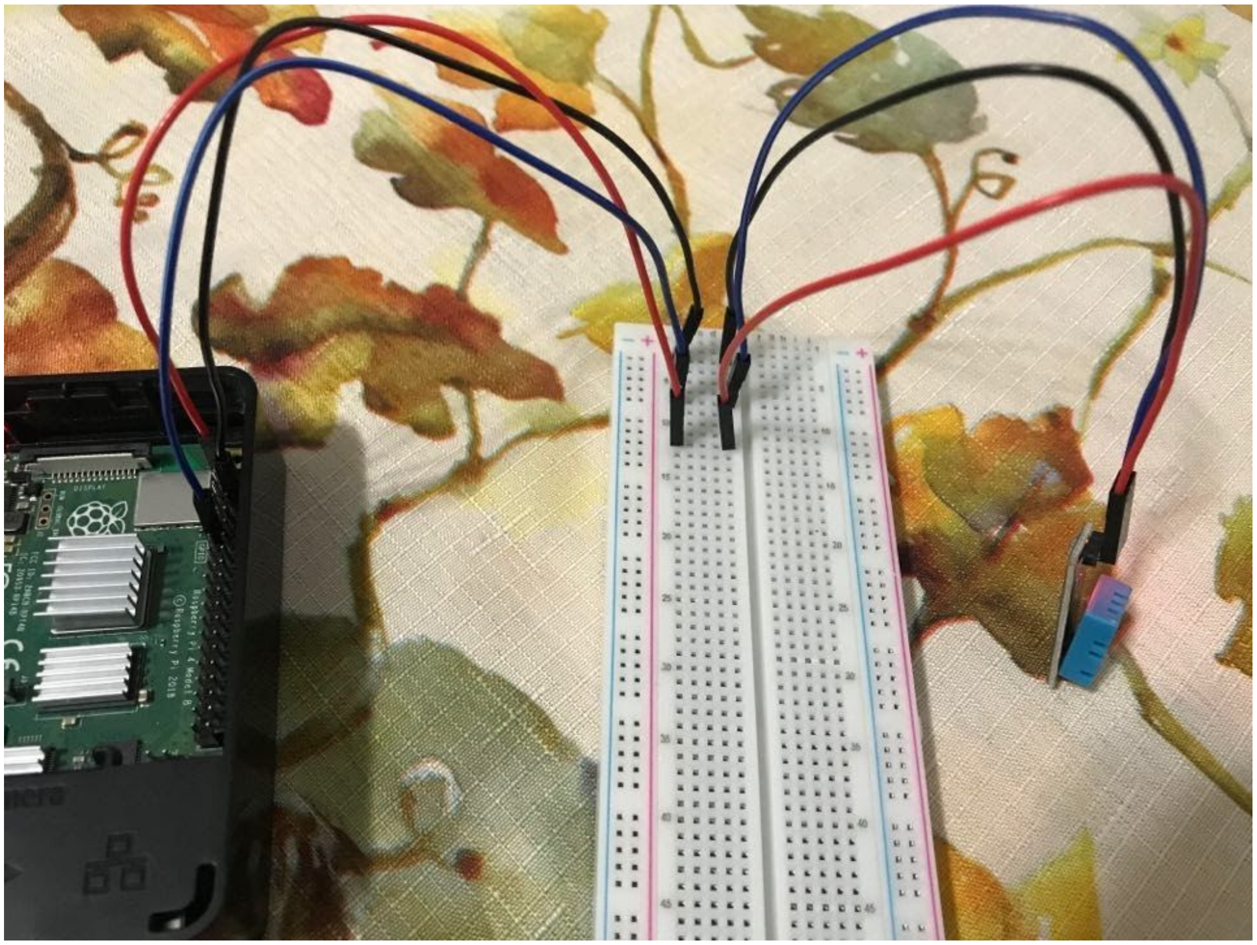
Analysis of ACK packet:
- Packet Number: 66
- Time since beginning of the capture: 28.225313
- Length: 75
- Source MAC Address: dc:a6:32:97:8d:46
- Destination MAC Address: 3c:22:fb:79:00:d0
- Source IP Address: 192.168.0.107
- Destination IP Address: 192.168.0.101
- Source Port: 5683
- Destination Port: 64322
- Code: 2.05 Content (69)
- Message ID: 53852
- Temperature Reading (Data): 25.0C
- Temperature Reading (Hexadecimal): 54 65 6d 70 3d 32 35 2e 30 43 20
- Humidity Reading (Data): 48.0%
- Humidity Reading (Hexadecimal): 48 75 6d 69 64 69 74 79 3d 34 38 2e 30 25

3. Implementation

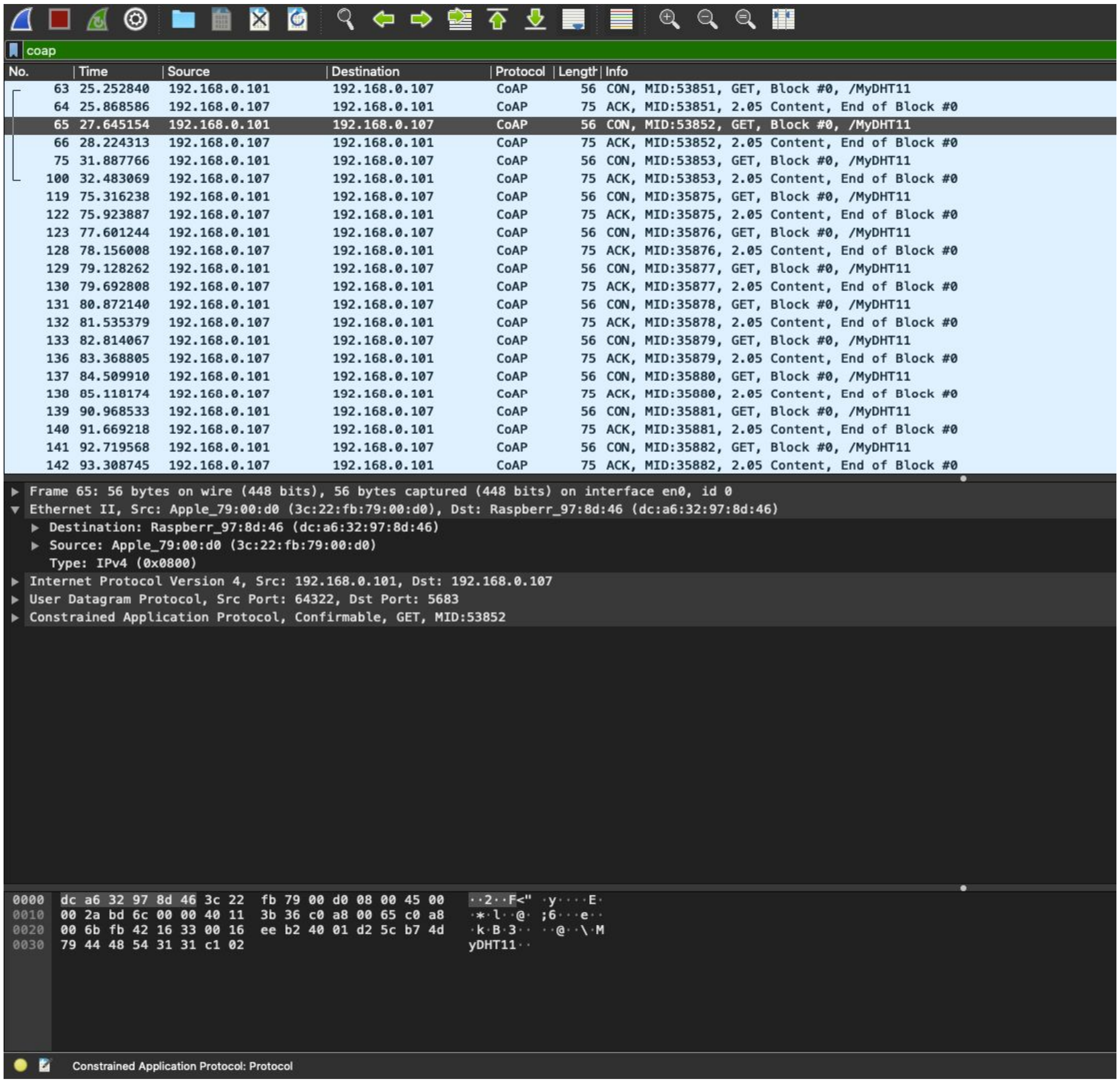
We used the programming language Python for our entire project. We imported the CoAP, Adafruit_DHT, and time libraries for this project. Our project works correctly for each step as defined in the project assignment. The first step makes the "Hello World" application available to the client on the request from the server. The second step displays the DHT11 sensor readings on the client after requesting it from the server. The third step uses Wireshark software to analyze the captured CoAP packets.

Some snapshots of the project:

Step 1 and Step 2:



Step 3:

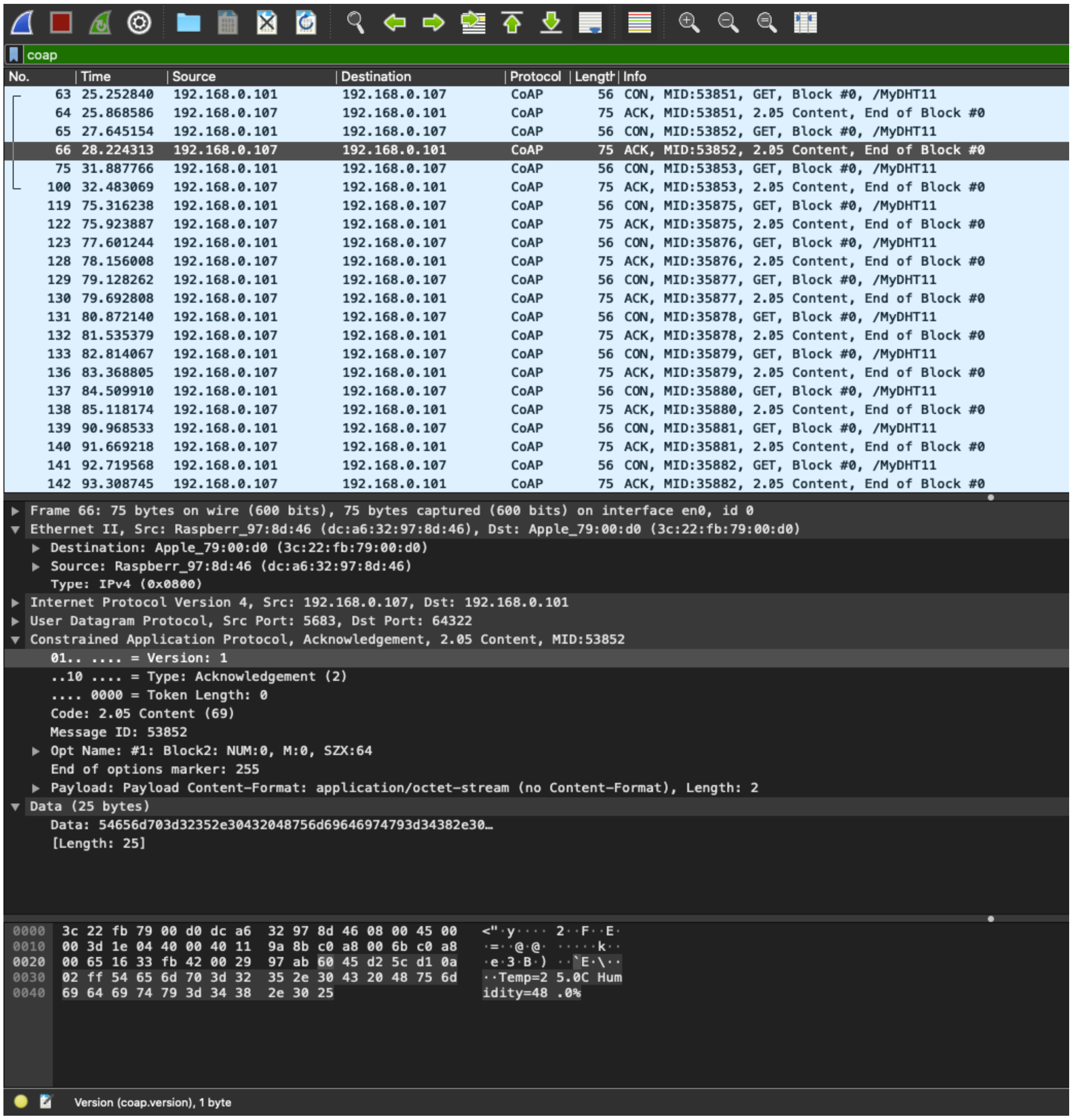

coap

```
No.    Time       Source         Destination    Protocol Length Info
   63 25.252840  192.168.0.101  192.168.0.107  CoAP       56 CON, MID:53851, GET, Block #0, /MyDHT11
   64 25.868586  192.168.0.107  192.168.0.101  CoAP       75 ACK, MID:53851, 2.05 Content, End of Block #0
   65 27.645154  192.168.0.101  192.168.0.107  CoAP       56 CON, MID:53852, GET, Block #0, /MyDHT11
   66 28.224313  192.168.0.107  192.168.0.101  CoAP       75 ACK, MID:53852, 2.05 Content, End of Block #0
   75 31.887766  192.168.0.101  192.168.0.107  CoAP       56 CON, MID:53853, GET, Block #0, /MyDHT11
  100 32.483069  192.168.0.107  192.168.0.101  CoAP       75 ACK, MID:53853, 2.05 Content, End of Block #0
  119 75.316238  192.168.0.101  192.168.0.107  CoAP       56 CON, MID:35875, GET, Block #0, /MyDHT11
  122 75.923887  192.168.0.107  192.168.0.101  CoAP       75 ACK, MID:35875, 2.05 Content, End of Block #0
  123 77.601244  192.168.0.101  192.168.0.107  CoAP       56 CON, MID:35876, GET, Block #0, /MyDHT11
  128 78.156008  192.168.0.107  192.168.0.101  CoAP       75 ACK, MID:35876, 2.05 Content, End of Block #0
  129 79.128262  192.168.0.101  192.168.0.107  CoAP       56 CON, MID:35877, GET, Block #0, /MyDHT11
  130 79.692808  192.168.0.107  192.168.0.101  CoAP       75 ACK, MID:35877, 2.05 Content, End of Block #0
  131 80.872140  192.168.0.101  192.168.0.107  CoAP       56 CON, MID:35878, GET, Block #0, /MyDHT11
  132 81.535379  192.168.0.107  192.168.0.101  CoAP       75 ACK, MID:35878, 2.05 Content, End of Block #0
  133 82.814067  192.168.0.101  192.168.0.107  CoAP       56 CON, MID:35879, GET, Block #0, /MyDHT11
  136 83.368805  192.168.0.107  192.168.0.101  CoAP       75 ACK, MID:35879, 2.05 Content, End of Block #0
  137 84.509910  192.168.0.101  192.168.0.107  CoAP       56 CON, MID:35880, GET, Block #0, /MyDHT11
  138 85.110174  192.168.0.107  192.168.0.101  CoAP       75 ACK, MID:35880, 2.05 Content, End of Block #0
  139 90.968533  192.168.0.101  192.168.0.107  CoAP       56 CON, MID:35881, GET, Block #0, /MyDHT11
  140 91.669218  192.168.0.107  192.168.0.101  CoAP       75 ACK, MID:35881, 2.05 Content, End of Block #0
  141 92.719568  192.168.0.101  192.168.0.107  CoAP       56 CON, MID:35882, GET, Block #0, /MyDHT11
  142 93.308745  192.168.0.107  192.168.0.101  CoAP       75 ACK, MID:35882, 2.05 Content, End of Block #0
```

```
▶ Frame 65: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface en0, id 0
▼ Ethernet II, Src: Apple_79:00:d0 (3c:22:fb:79:00:d0), Dst: Raspberr_97:8d:46 (dc:a6:32:97:8d:46)
  ▶ Destination: Raspberr_97:8d:46 (dc:a6:32:97:8d:46)
  ▶ Source: Apple_79:00:d0 (3c:22:fb:79:00:d0)
    Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 192.168.0.101, Dst: 192.168.0.107
▶ User Datagram Protocol, Src Port: 64322, Dst Port: 5683
▶ Constrained Application Protocol, Confirmable, GET, MID:53852
```

```
0000  dc a6 32 97 8d 46 3c 22  fb 79 00 d0 08 00 45 00   ··2··F<" ·y····E·
0010  00 2a bd 6c 00 00 40 11  3b 36 c0 a8 00 65 c0 a8   ·*·l··@· ;6···e··
0020  00 6b fb 42 16 33 00 16  ee b2 40 01 d2 5c b7 4d   ·k·B·3·· ··@··\·M
0030  79 44 48 54 31 31 c1 02                            yDHT11··
```

Constrained Application Protocol: Protocol



```
coap
No.   Time        Source          Destination     Protocol Length Info
   63 25.252840   192.168.0.101   192.168.0.107   CoAP      56 CON, MID:53851, GET, Block #0, /MyDHT11
   64 25.868586   192.168.0.107   192.168.0.101   CoAP      75 ACK, MID:53851, 2.05 Content, End of Block #0
   65 27.645154   192.168.0.101   192.168.0.107   CoAP      56 CON, MID:53852, GET, Block #0, /MyDHT11
   66 28.224313   192.168.0.107   192.168.0.101   CoAP      75 ACK, MID:53852, 2.05 Content, End of Block #0
   75 31.887766   192.168.0.101   192.168.0.107   CoAP      56 CON, MID:53853, GET, Block #0, /MyDHT11
  100 32.483069   192.168.0.107   192.168.0.101   CoAP      75 ACK, MID:53853, 2.05 Content, End of Block #0
  119 75.316238   192.168.0.101   192.168.0.107   CoAP      56 CON, MID:35875, GET, Block #0, /MyDHT11
  122 75.923887   192.168.0.107   192.168.0.101   CoAP      75 ACK, MID:35875, 2.05 Content, End of Block #0
  123 77.601244   192.168.0.101   192.168.0.107   CoAP      56 CON, MID:35876, GET, Block #0, /MyDHT11
  128 78.156008   192.168.0.107   192.168.0.101   CoAP      75 ACK, MID:35876, 2.05 Content, End of Block #0
  129 79.128262   192.168.0.101   192.168.0.107   CoAP      56 CON, MID:35877, GET, Block #0, /MyDHT11
  130 79.692808   192.168.0.107   192.168.0.101   CoAP      75 ACK, MID:35877, 2.05 Content, End of Block #0
  131 80.872140   192.168.0.101   192.168.0.107   CoAP      56 CON, MID:35878, GET, Block #0, /MyDHT11
  132 81.535379   192.168.0.107   192.168.0.101   CoAP      75 ACK, MID:35878, 2.05 Content, End of Block #0
  133 82.814067   192.168.0.101   192.168.0.107   CoAP      56 CON, MID:35879, GET, Block #0, /MyDHT11
  136 83.368805   192.168.0.107   192.168.0.101   CoAP      75 ACK, MID:35879, 2.05 Content, End of Block #0
  137 84.509910   192.168.0.101   192.168.0.107   CoAP      56 CON, MID:35880, GET, Block #0, /MyDHT11
  138 85.118174   192.168.0.107   192.168.0.101   CoAP      75 ACK, MID:35880, 2.05 Content, End of Block #0
  139 90.968533   192.168.0.101   192.168.0.107   CoAP      56 CON, MID:35881, GET, Block #0, /MyDHT11
  140 91.669218   192.168.0.107   192.168.0.101   CoAP      75 ACK, MID:35881, 2.05 Content, End of Block #0
  141 92.719568   192.168.0.101   192.168.0.107   CoAP      56 CON, MID:35882, GET, Block #0, /MyDHT11
  142 93.308745   192.168.0.107   192.168.0.101   CoAP      75 ACK, MID:35882, 2.05 Content, End of Block #0

▶ Frame 66: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface en0, id 0
▼ Ethernet II, Src: Raspberr_97:8d:46 (dc:a6:32:97:8d:46), Dst: Apple_79:00:d0 (3c:22:fb:79:00:d0)
  ▶ Destination: Apple_79:00:d0 (3c:22:fb:79:00:d0)
  ▶ Source: Raspberr_97:8d:46 (dc:a6:32:97:8d:46)
    Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 192.168.0.107, Dst: 192.168.0.101
▶ User Datagram Protocol, Src Port: 5683, Dst Port: 64322
▼ Constrained Application Protocol, Acknowledgement, 2.05 Content, MID:53852
    01.. .... = Version: 1
    ..10 .... = Type: Acknowledgement (2)
    .... 0000 = Token Length: 0
    Code: 2.05 Content (69)
    Message ID: 53852
  ▶ Opt Name: #1: Block2: NUM:0, M:0, SZX:64
    End of options marker: 255
  ▶ Payload: Payload Content-Format: application/octet-stream (no Content-Format), Length: 2
▼ Data (25 bytes)
    Data: 54656d703d32352e30432048756d69646974793d34382e30…
    [Length: 25]

0000  3c 22 fb 79 00 d0 dc a6  32 97 8d 46 08 00 45 00   <"·y····  2··F··E·
0010  00 3d 1e 04 40 00 40 11  9a 8b c0 a8 00 6b c0 a8   ·=··@·@·  ·····k··
0020  00 65 16 33 fb 42 00 29  97 ab 60 45 d2 5c d1 0a   ·e·3·B·)  ··`E·\··
0030  02 ff 54 65 6d 70 3d 32  35 2e 30 43 20 48 75 6d   ··Temp=2  5.0C Hum
0040  69 64 69 74 79 3d 34 38  2e 30 25                  idity=48  .0%

Version (coap.version), 1 byte
```

4. Experiences

We started our project early this time, so we had enough time to tackle most of our problems. We struggled mostly with trying to install the CoAP library and copper application. Another difficulty was in trying to understand the whole development environment and connections between all these libraries and applications. From this problem, we learned that it is important to have the basic knowledge between all these concepts and CoAP protocol.

5. Video demo

Demo for Step2:
https://drive.google.com/file/d/1JjtiATB6onwEu2TcYQ3Cibo508bIcGG0/view?usp=sharing

Demo for Step3:
https://drive.google.com/file/d/1db592LvEy75wmx-BpEWdw223Upx2W3Nc/view?usp=sharin
g

6. References

- CoAP Library Link: https://github.com/Tanganelli/CoAPthon
- Copper for Google Chrome Link: https://github.com/mkovatsc/Copper4Cr
- Adafruit Library Link: https://github.com/adafruit/Adafruit_Python_DHT