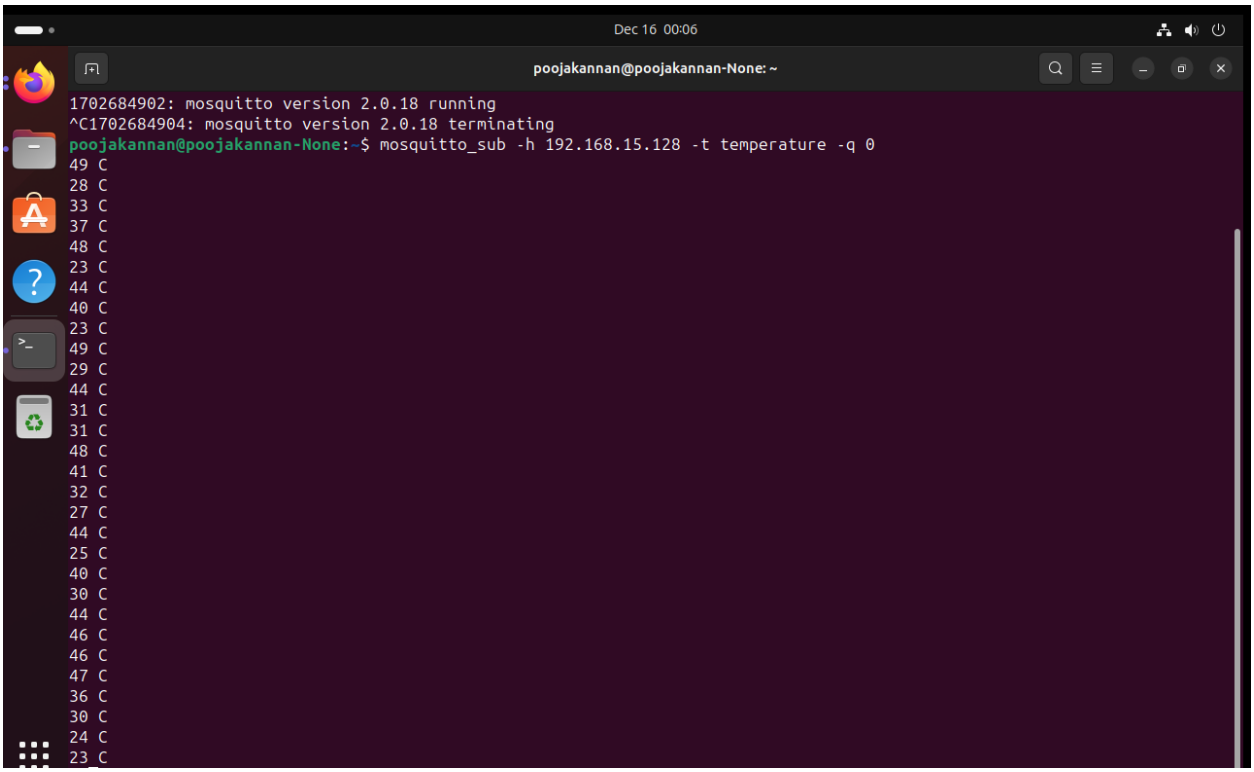
FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657

MQTT (Publish and Subscriber)

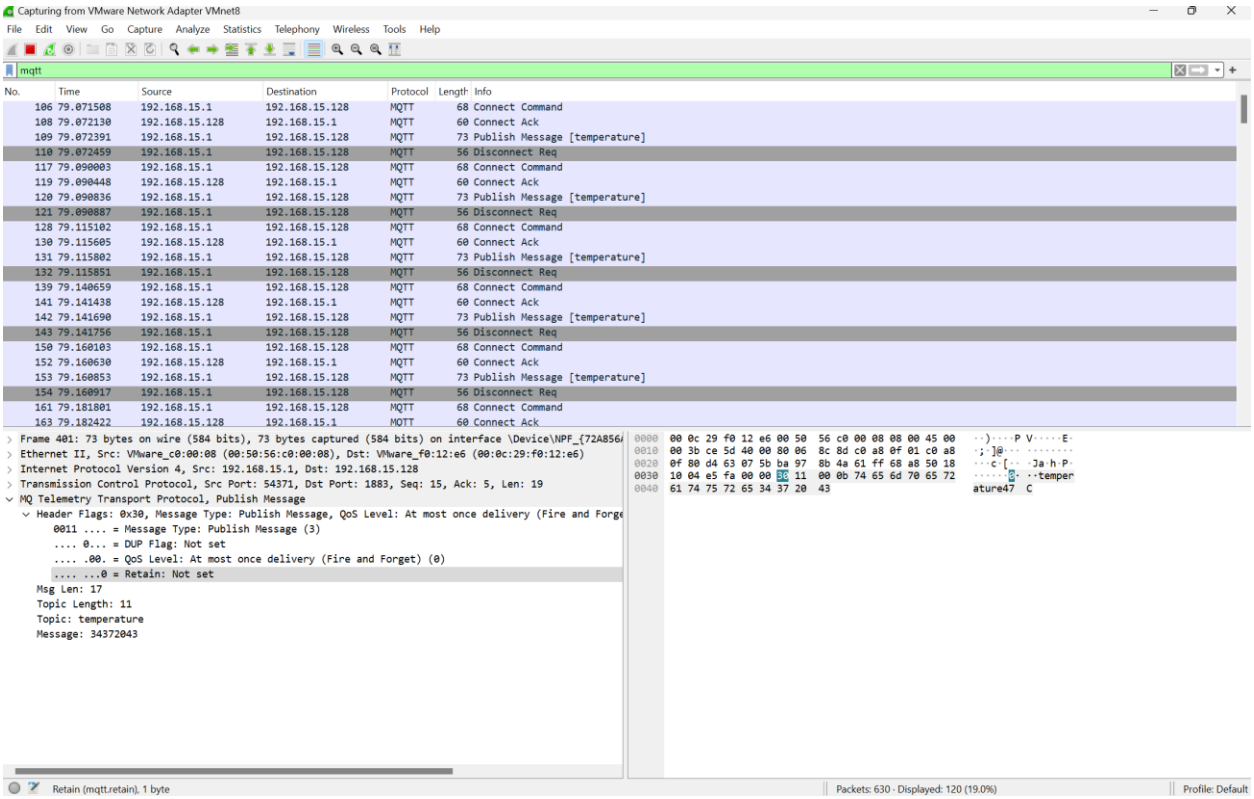
30 Readout for 0% loss



QOS 0

FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657

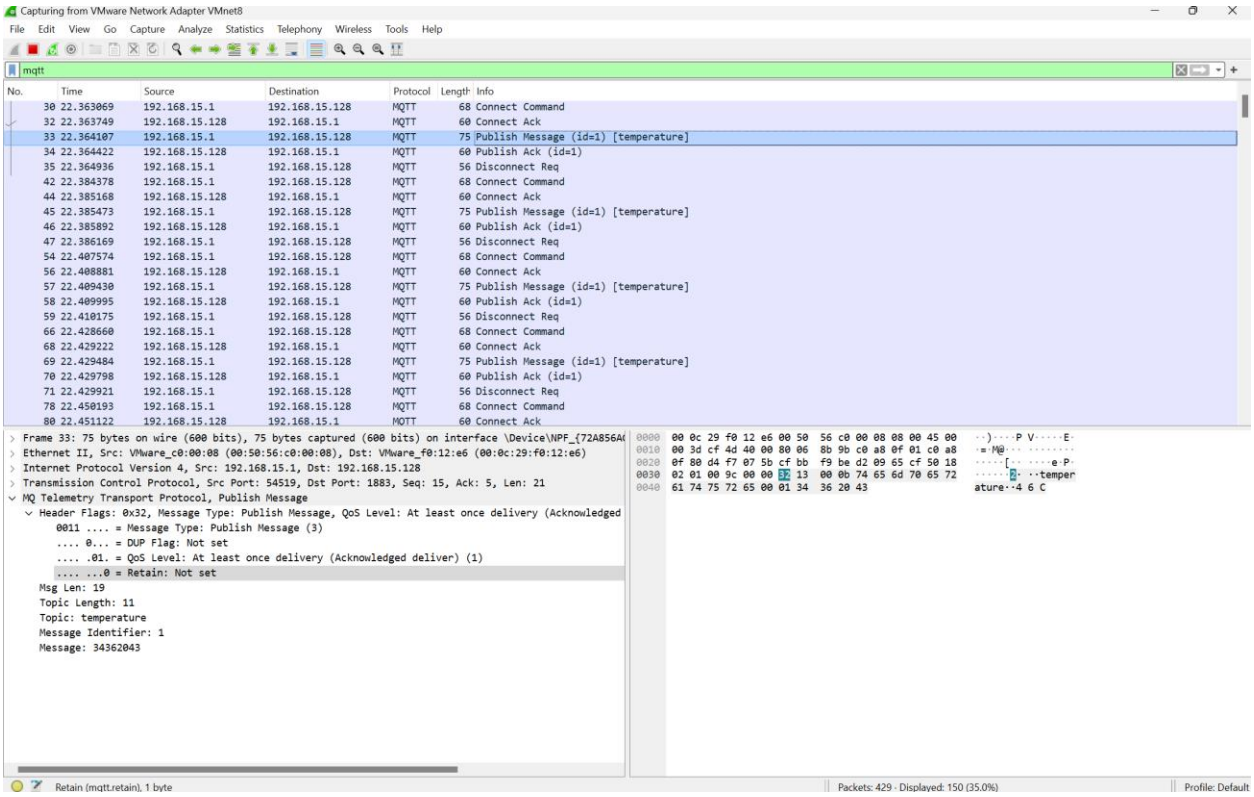


QOS 1

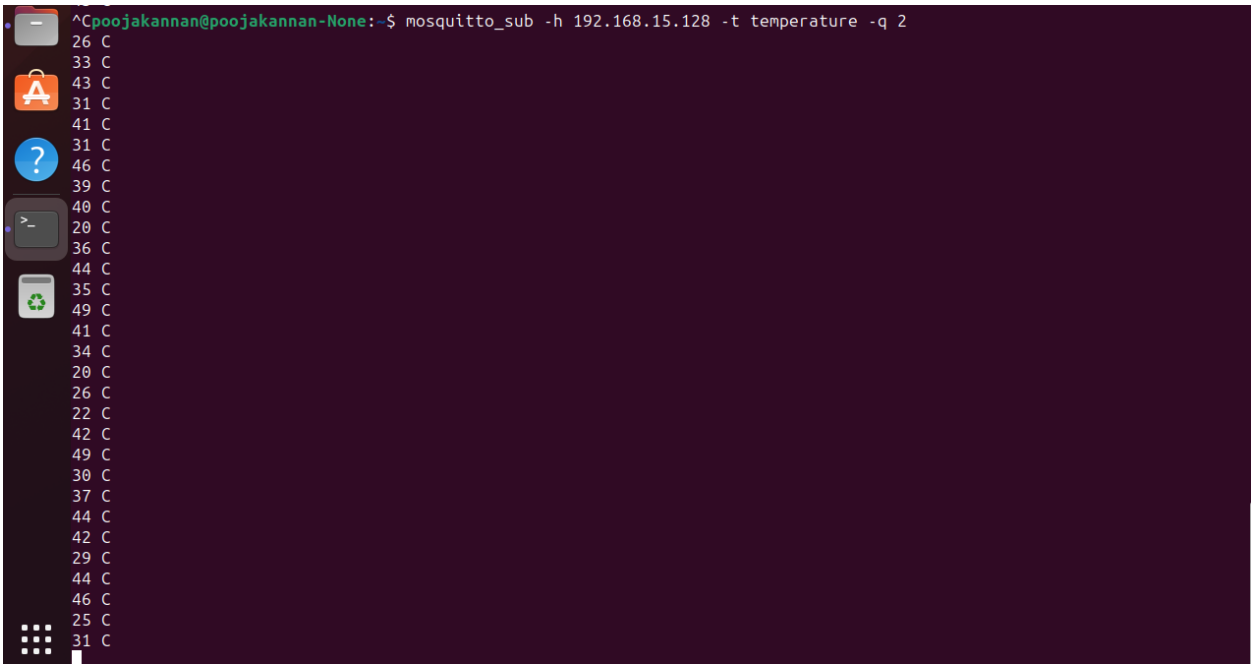
```
^Cpoojakannan@poojakannan-None:~$ mosquitto_sub -h 192.168.15.128 -t temperature -q 1
46 C
42 C
33 C
33 C
36 C
45 C
40 C
29 C
42 C
42 C
26 C
31 C
48 C
47 C
39 C
22 C
36 C
29 C
46 C
30 C
32 C
24 C
21 C
39 C
28 C
47 C
40 C
37 C
26 C
43 C
```

FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657

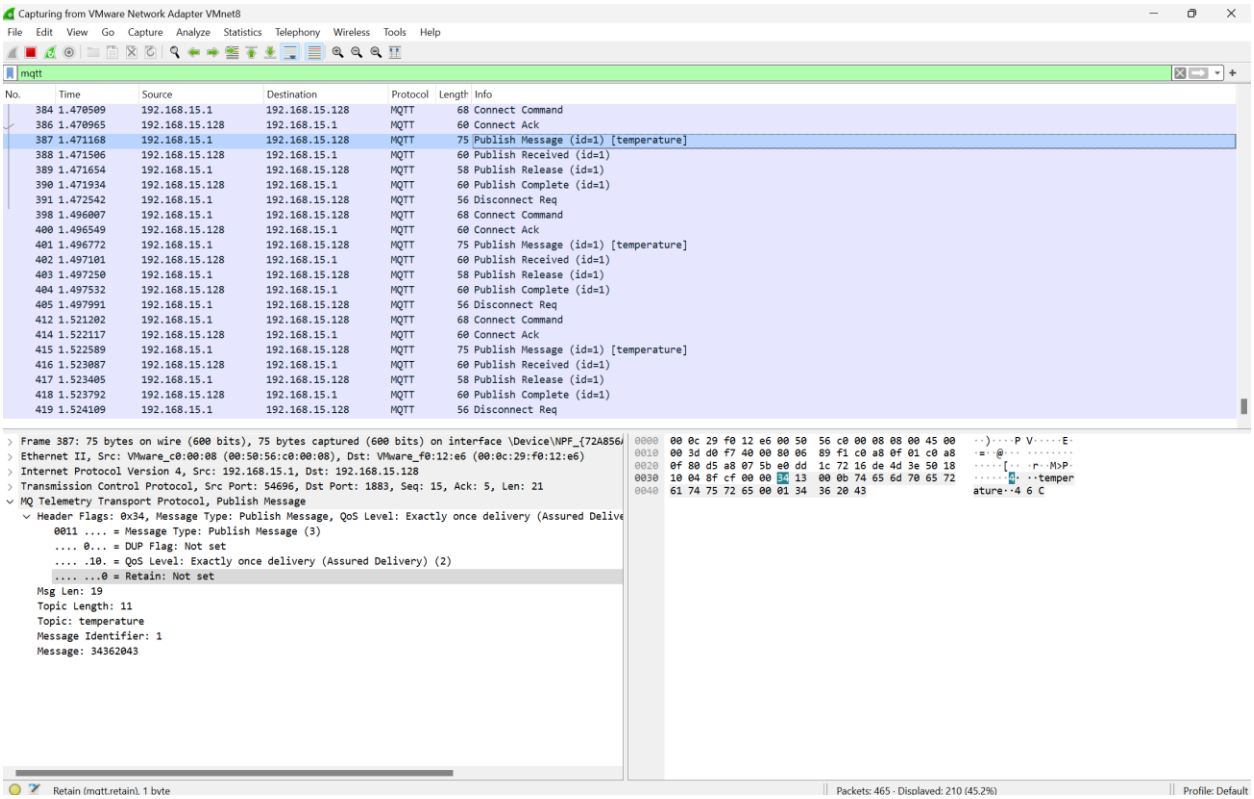


QOS 2

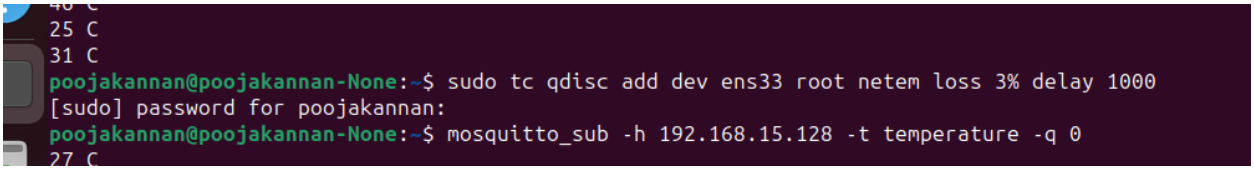


FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657



3% with 1000ms delay



FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657

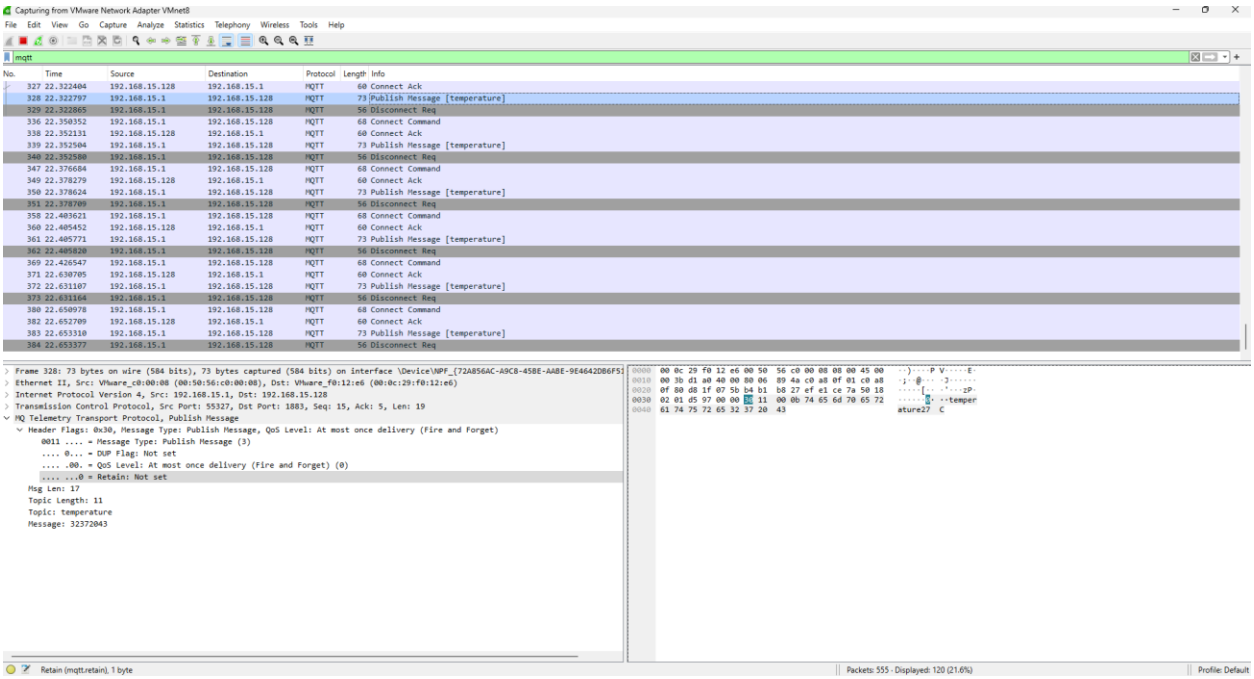
```
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\pooja>ping 192.168.15.128

Pinging 192.168.15.128 with 32 bytes of data:
Reply from 192.168.15.128: bytes=32 time=2ms TTL=64
Reply from 192.168.15.128: bytes=32 time=1ms TTL=64
Reply from 192.168.15.128: bytes=32 time=2ms TTL=64
Reply from 192.168.15.128: bytes=32 time=3ms TTL=64

Ping statistics for 192.168.15.128:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 3ms, Average = 2ms

C:\Users\pooja>
```

QOS0

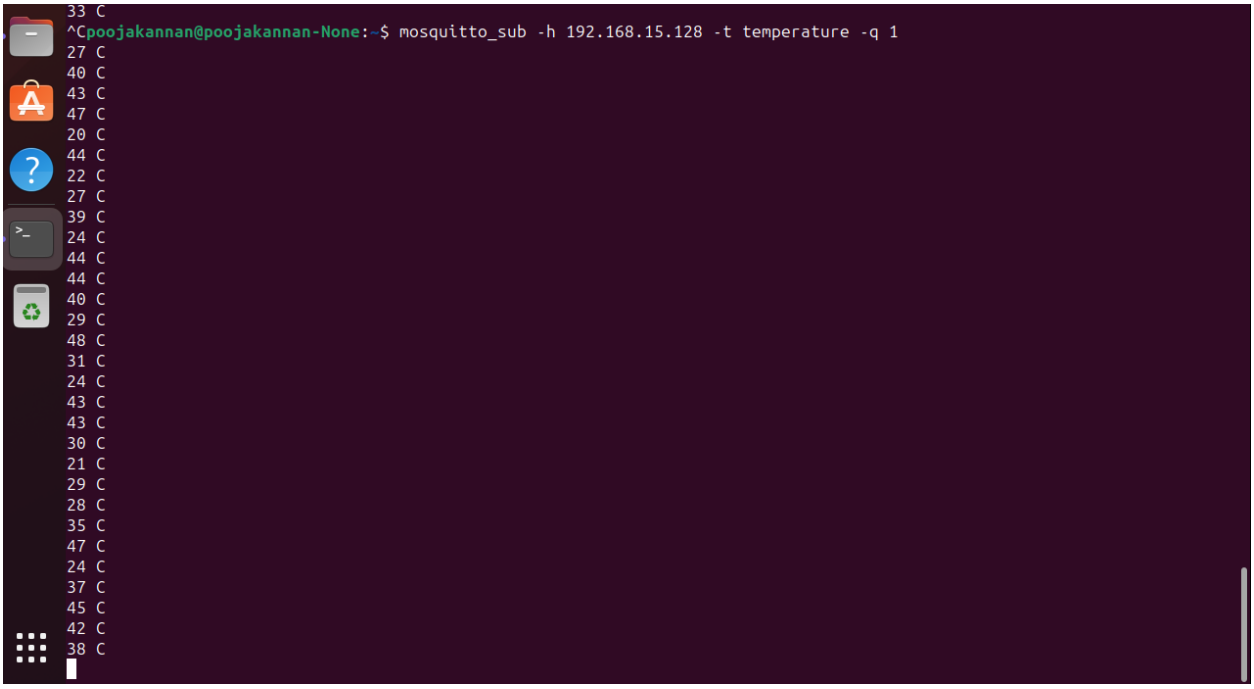
```
[sudo] password for poojakannan:
poojakannan@poojakannan-None:~$ mosquitto_sub -h 192.168.15.128 -t temperature -q 0
27 C
38 C
21 C
32 C
33 C
26 C
20 C
47 C
36 C
24 C
34 C
27 C
41 C
46 C
23 C
35 C
32 C
35 C
24 C
34 C
38 C
31 C
34 C
23 C
27 C
33 C
34 C
21 C
47 C
33 C
```

FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657

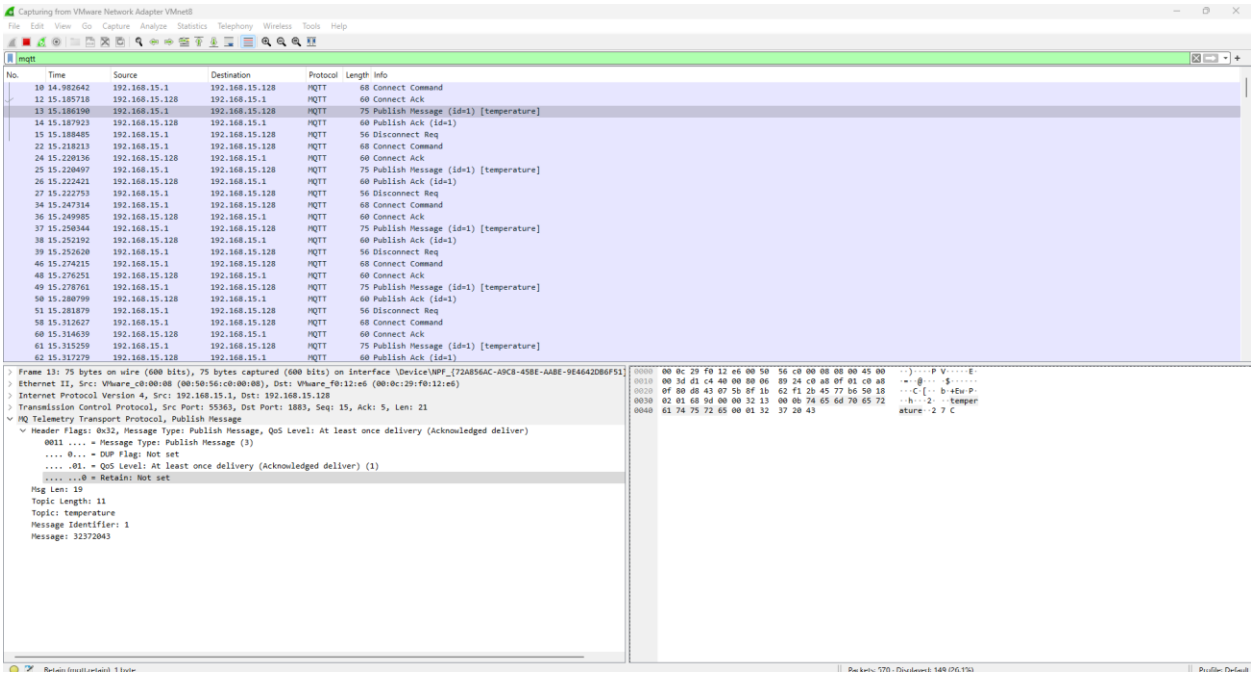


QOS1

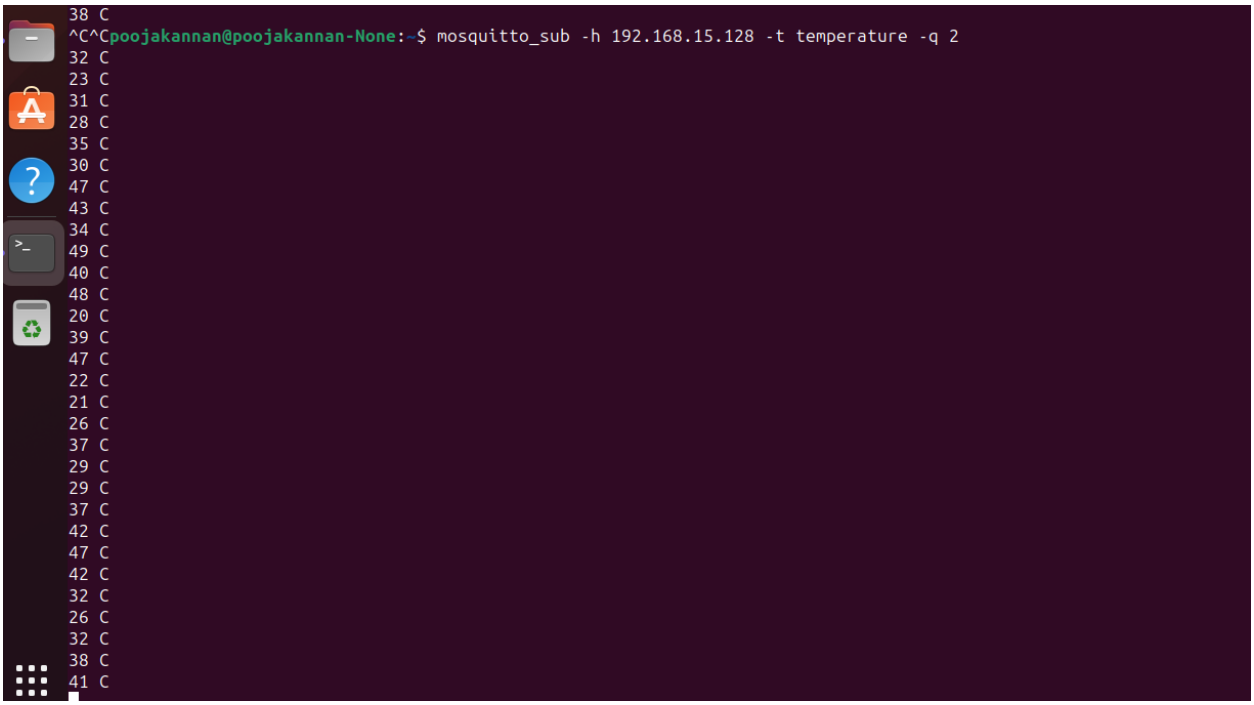


FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657

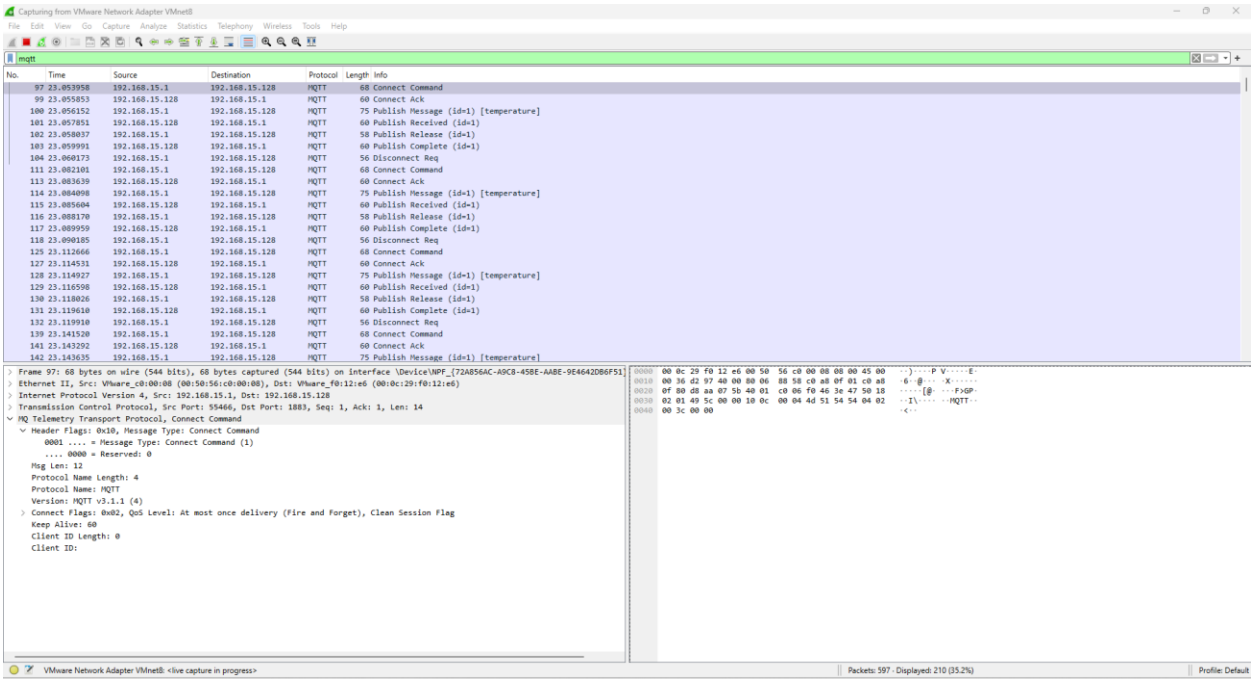


QOS2



FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657



5% with 2000ms delay

```
poojakannan@poojakannan-None:~$ sudo tc qdisc add  dev ens33 root netem loss 5% delay 2000
```

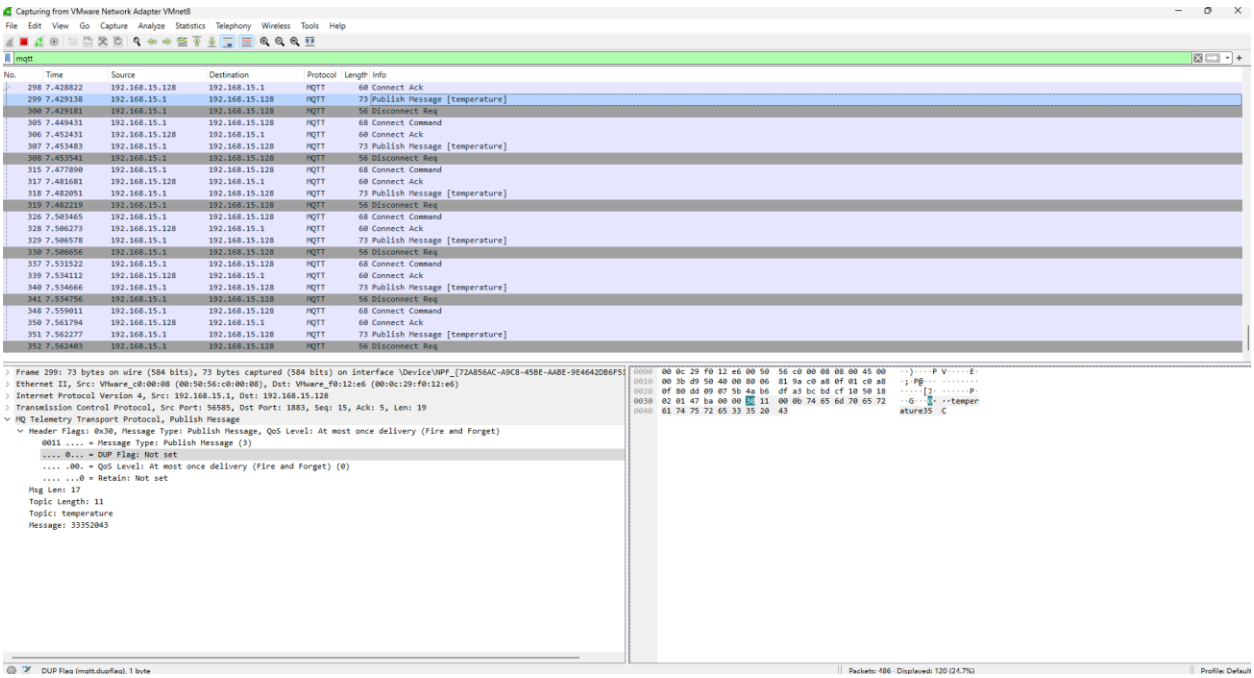
```
C:\Users\pooja>ping 192.168.15.128

Pinging 192.168.15.128 with 32 bytes of data:
Reply from 192.168.15.128: bytes=32 time=2ms TTL=64
Reply from 192.168.15.128: bytes=32 time=2ms TTL=64
Reply from 192.168.15.128: bytes=32 time=3ms TTL=64
Reply from 192.168.15.128: bytes=32 time=3ms TTL=64

Ping statistics for 192.168.15.128:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 3ms, Average = 2ms
```

QOS0

FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657

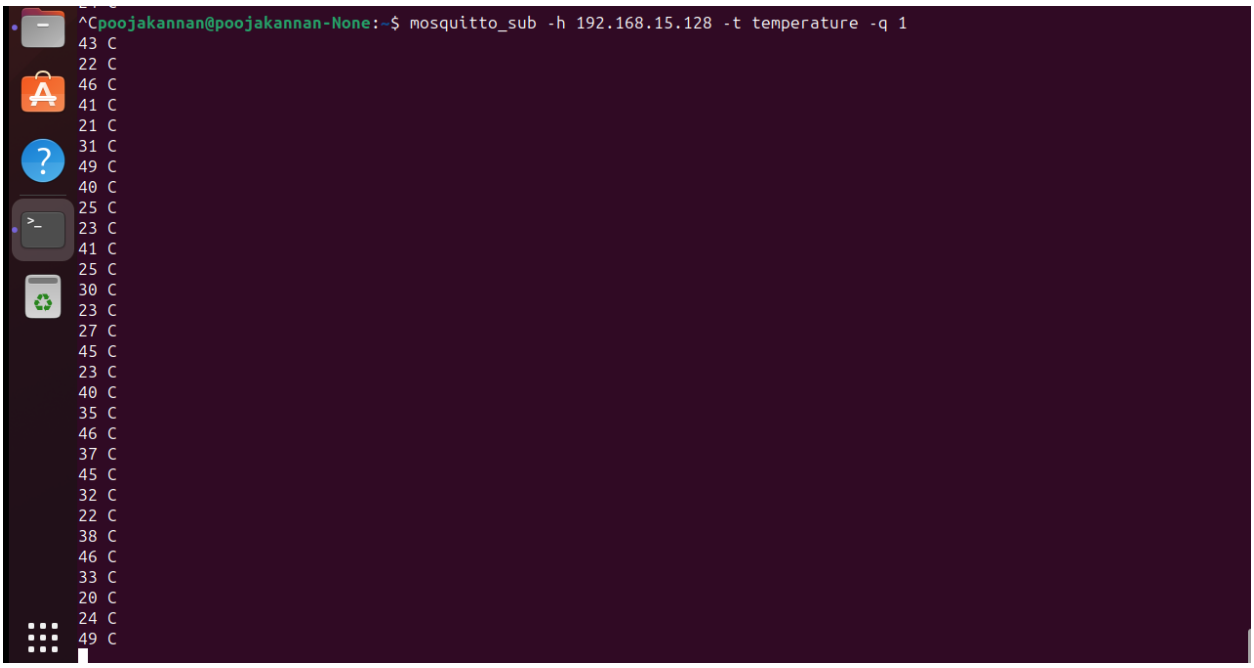
```
poojakannan@poojakannan-None:~$ mosquitto_sub -h 192.168.15.128 -t temperature -q 0
20 C
29 C
26 C
34 C
36 C
42 C
34 C
37 C
24 C
32 C
26 C
41 C
48 C
22 C
48 C
30 C
24 C
35 C
27 C
32 C
29 C
45 C
34 C
30 C
35 C
45 C
47 C
42 C
48 C
```

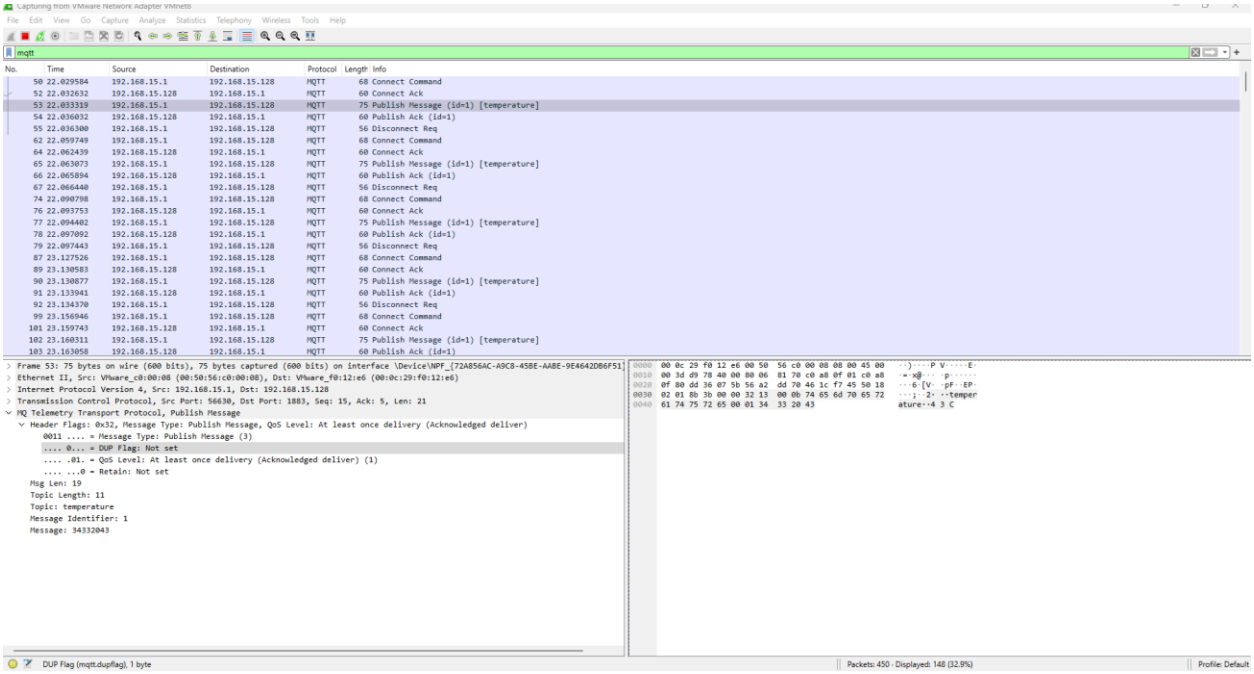


QOS1

FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657

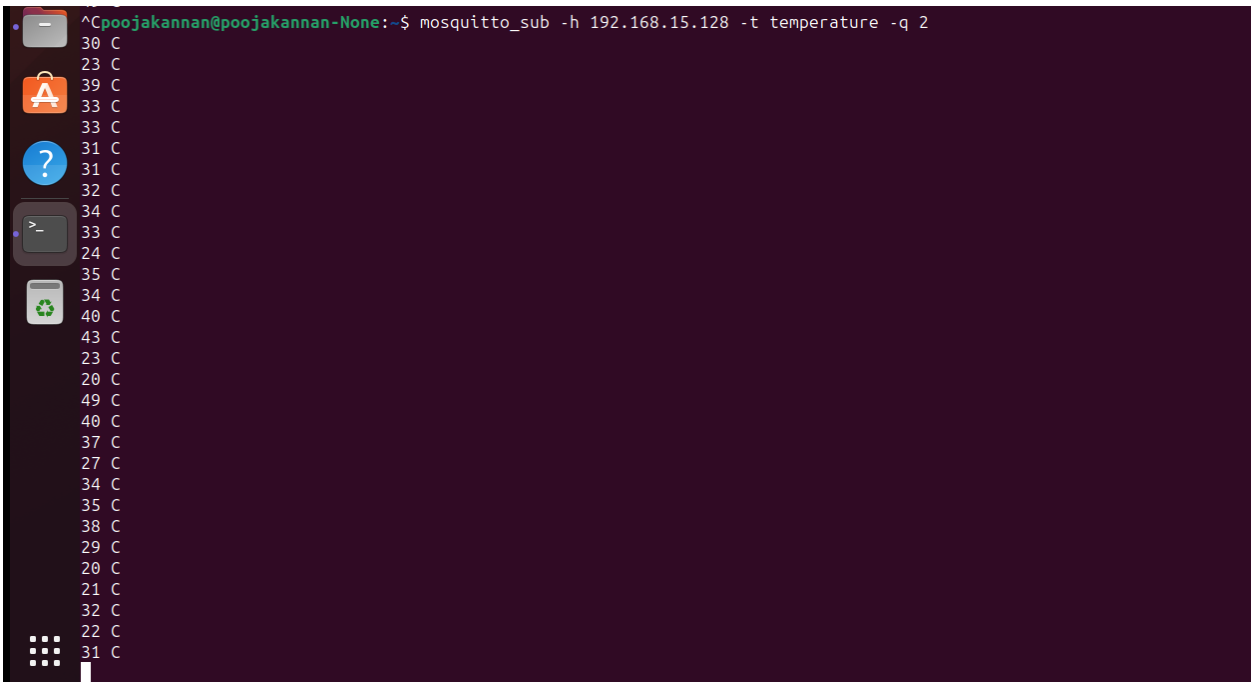


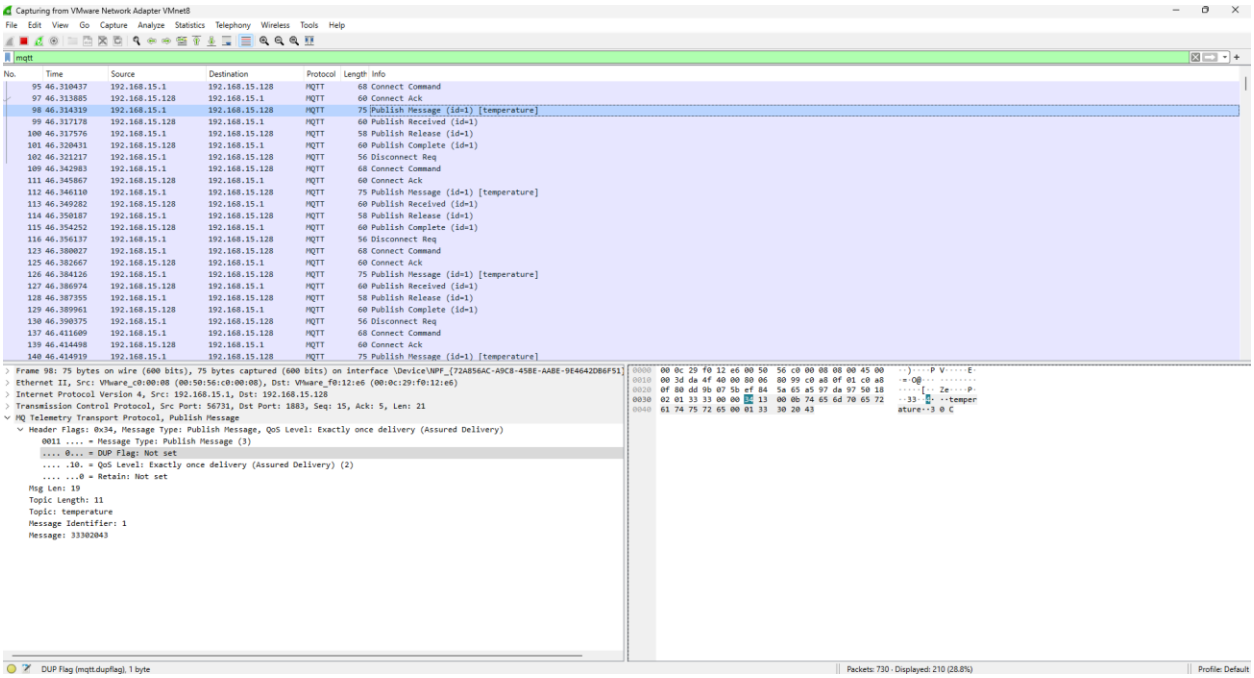


QOS2

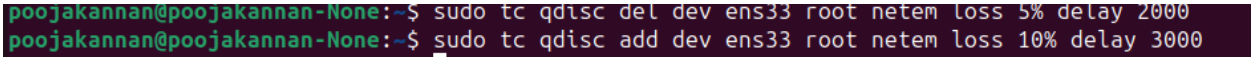
FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657





10% with 3000ms delay

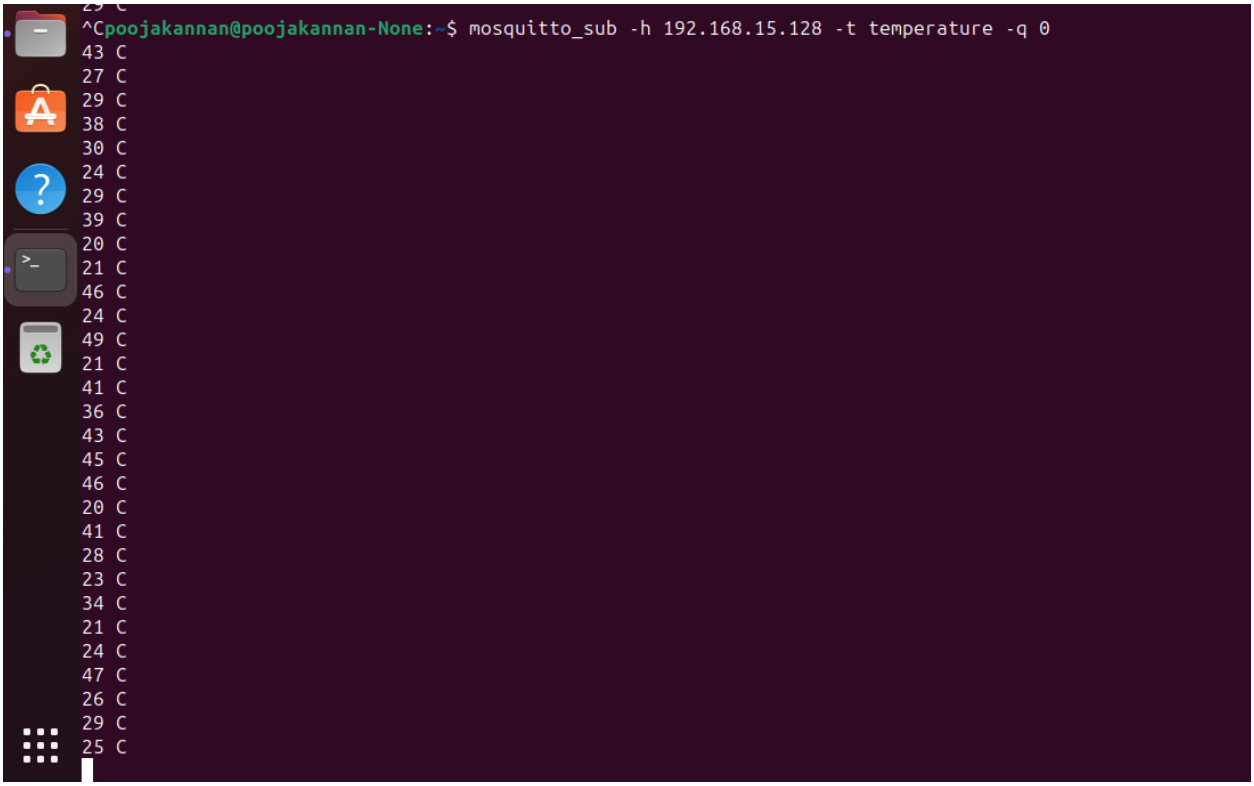


FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657

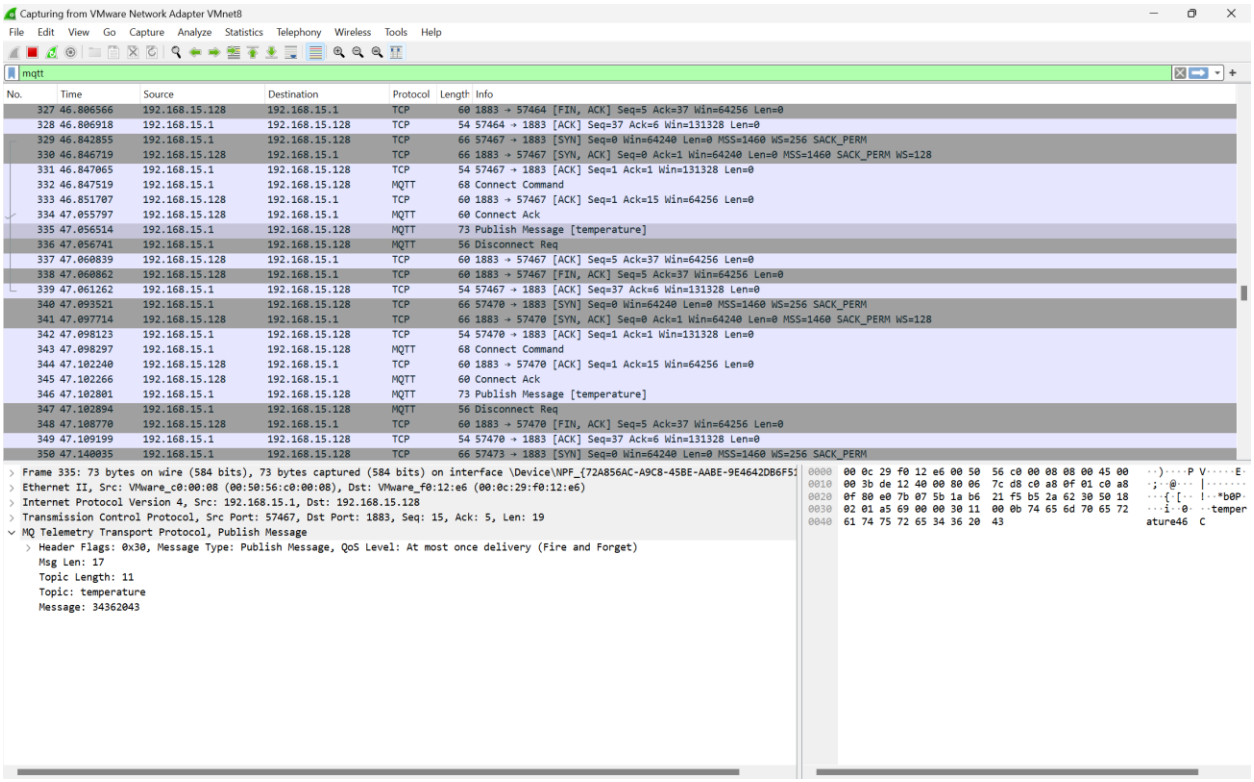
```
C:\Users\pooja>ping 192.168.15.128

Pinging 192.168.15.128 with 32 bytes of data:
Reply from 192.168.15.128: bytes=32 time=8ms TTL=64
Reply from 192.168.15.128: bytes=32 time=4ms TTL=64
Reply from 192.168.15.128: bytes=32 time=4ms TTL=64
Reply from 192.168.15.128: bytes=32 time=4ms TTL=64

Ping statistics for 192.168.15.128:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 8ms, Average = 5ms
```

QOS0

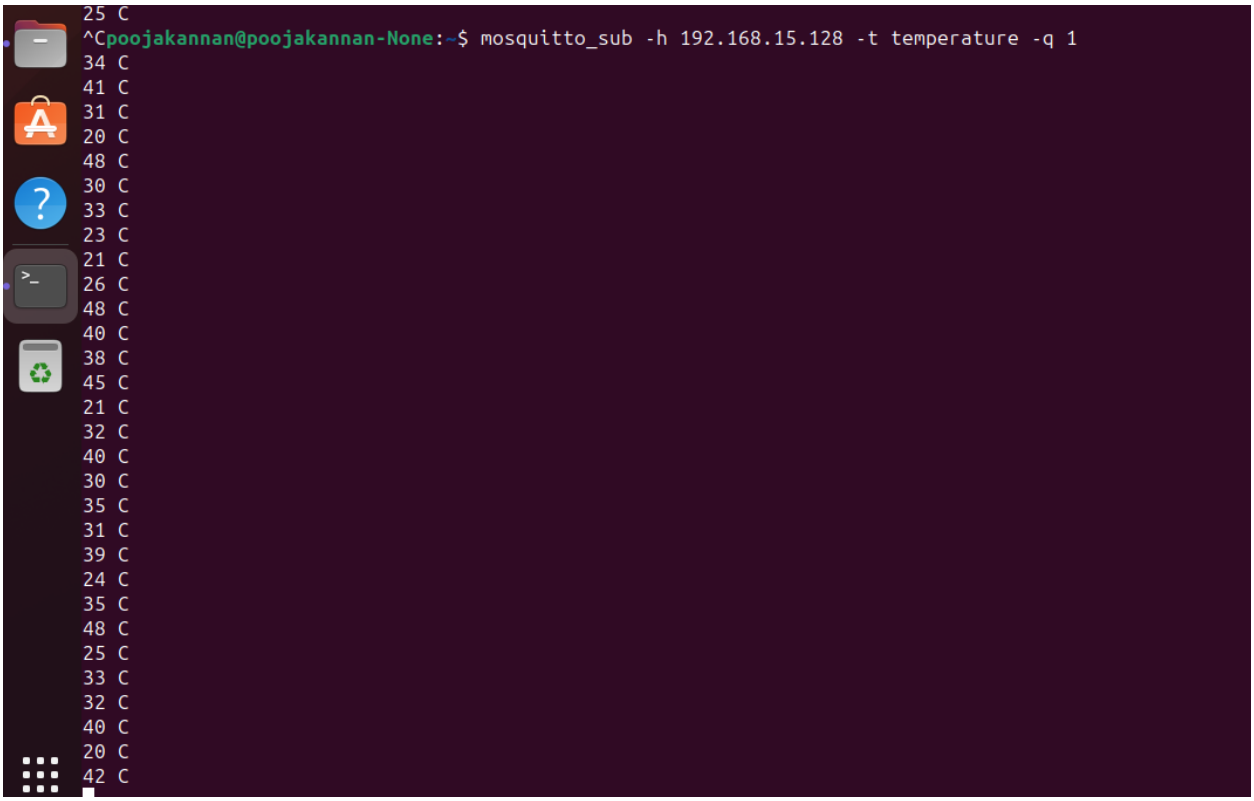


FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657

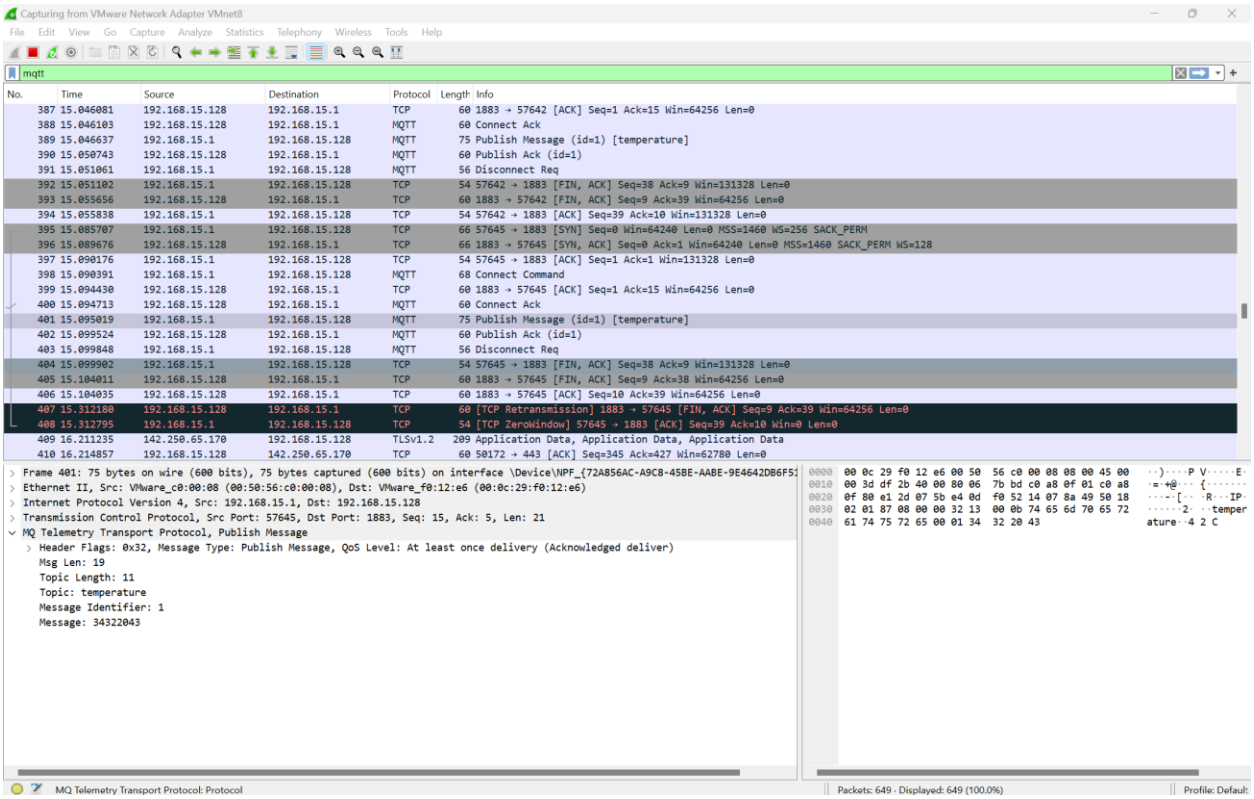


QOS1



FIOT Final Project – MQTT & COAP

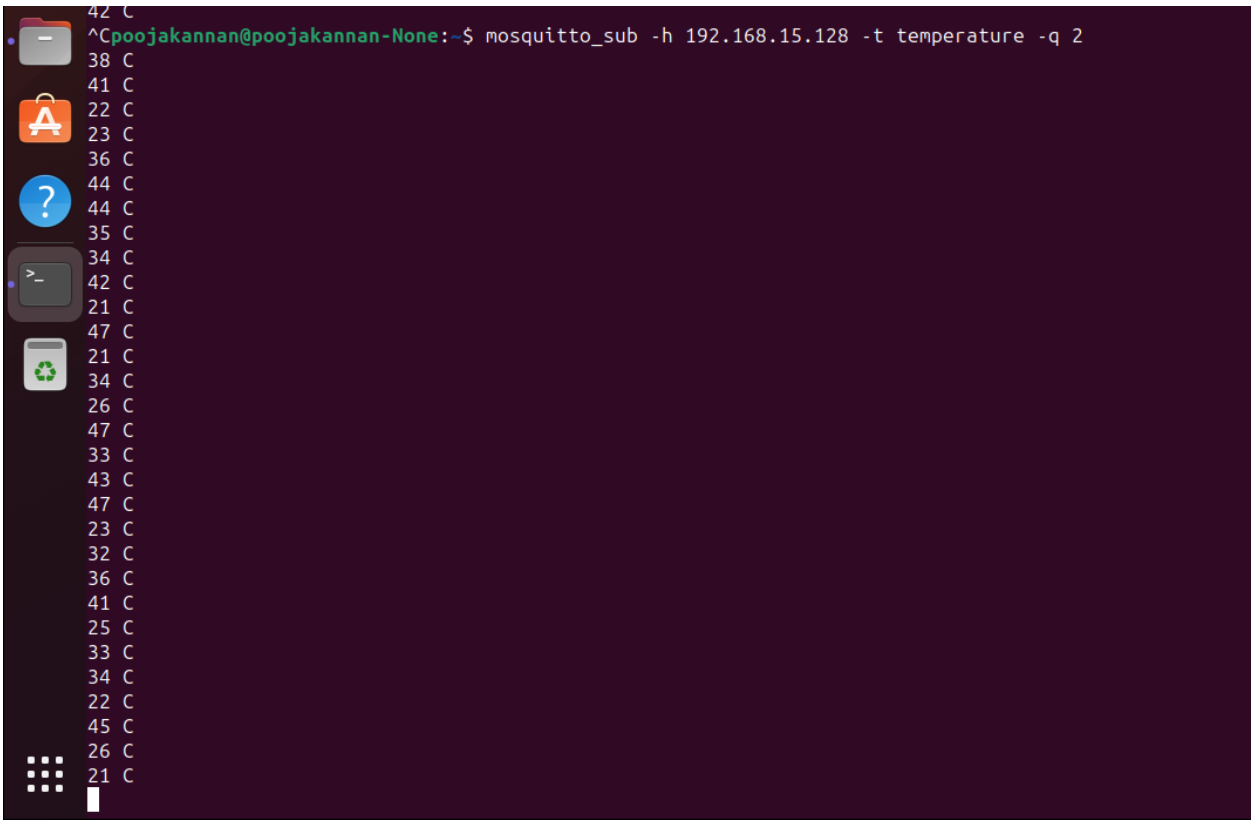
Pooja Kannan
002845657

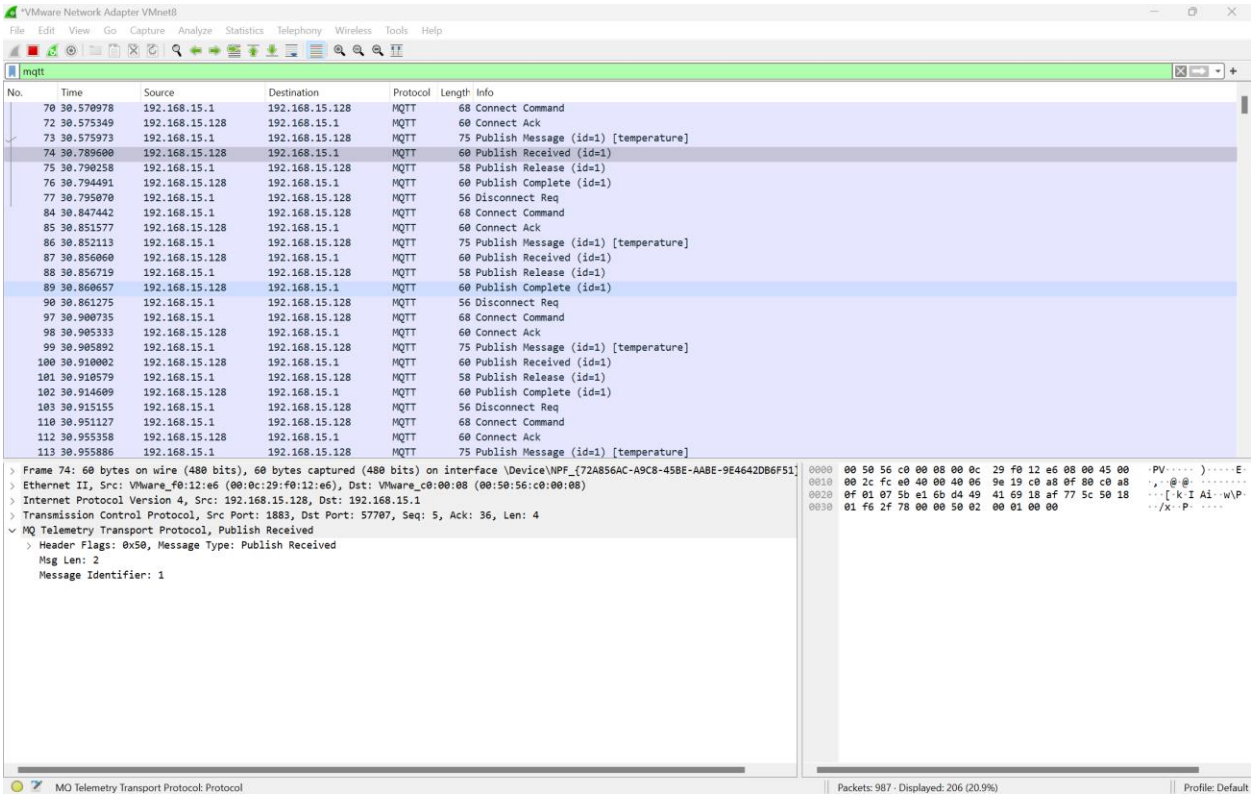


QOS2

FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657





FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657

QOS/Loss	0%	3%	5%	10%
0	0.0010079	0.00906243	0.0101478	0.01870473
1	0.00140913	0.0177371	0.02018363	0.0585904
2	0.0018835	0.02683626	0.0510157	0.091316

Describe the messages that are exchanged between devices and broker.

In an MQTT setup that includes a sensor (as the publisher), an app (as the subscriber), and a broker, the communication between these devices involves several key steps:

Establishing a Connection

1. The sensor sends a CONNECT message to the broker to initiate a connection.

2. In response, the broker sends back a CONNACK message, confirming whether the connection has been established.

Setting up a Subscription

1. The app sends a SUBSCRIBE message to the broker, indicating the specific topic it wants to subscribe to and the desired Quality of Service (QoS) level.

2. The broker acknowledges this request by sending a SUBACK message to the app, confirming the QoS level that has been granted for the subscription.

Publishing Data

1. The sensor sends a PUBLISH message to the broker, which includes the topic and the payload, in this case, temperature data.

2. Upon receiving this, the broker then forwards the message to all subscribers who have shown interest in that topic.

Delivering Data to the Subscriber

1. The broker sends a PUBLISH message to the app, which includes both the topic and the payload.

2. Finally, the app receives this message and processes the temperature data accordingly.

How does quality affect latency? Why?

The Quality of Service (QoS) level in MQTT communication significantly influences latency, as explained here:

1. QoS 0 - At Most Once: This is the fastest option since it doesn't require acknowledgments or delivery guarantees. When the sensor transmits a message, the broker forwards it to subscribers without delay. However, this method risks message loss in case of packet loss, as there is no mechanism to ensure delivery.

Pooja Kannan
002845657

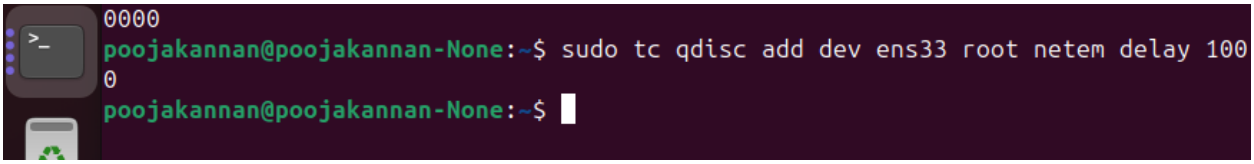
2. QoS 1 - At Least Once: Here, latency is higher due to the inclusion of acknowledgment and retry processes. Upon message transmission by the sensor, the broker acknowledges receipt and manages the message's delivery to the subscribers. If the sensor doesn't receive this acknowledgment, it retransmits the message. The need for acknowledgment and potential retransmission contributes to greater latency compared to QoS 0

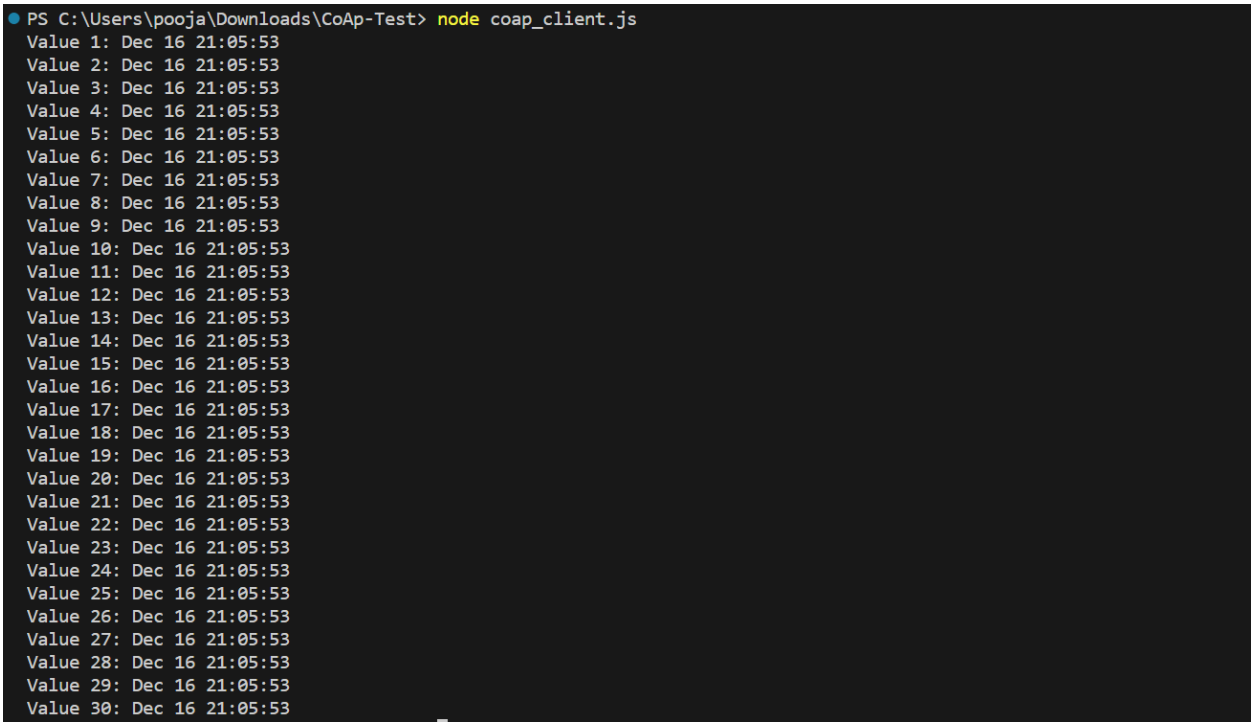
3. QoS 2 - Exactly Once: This level offers the highest reliability but also the highest latency. A four-step handshake process is used, involving multiple message exchanges between the sensor (sender) and the broker/app (receiver) to assure a single instance of delivery. This extra layer of communication and coordination adds to the latency, making it more significant than both QoS 0 and QoS 1.

To sum up, the progression from QoS 0 to QoS 2 entails increasing latency, attributable to the added complexity of acknowledgments, retries, and coordination necessary for assured message delivery.

CO-AP Server

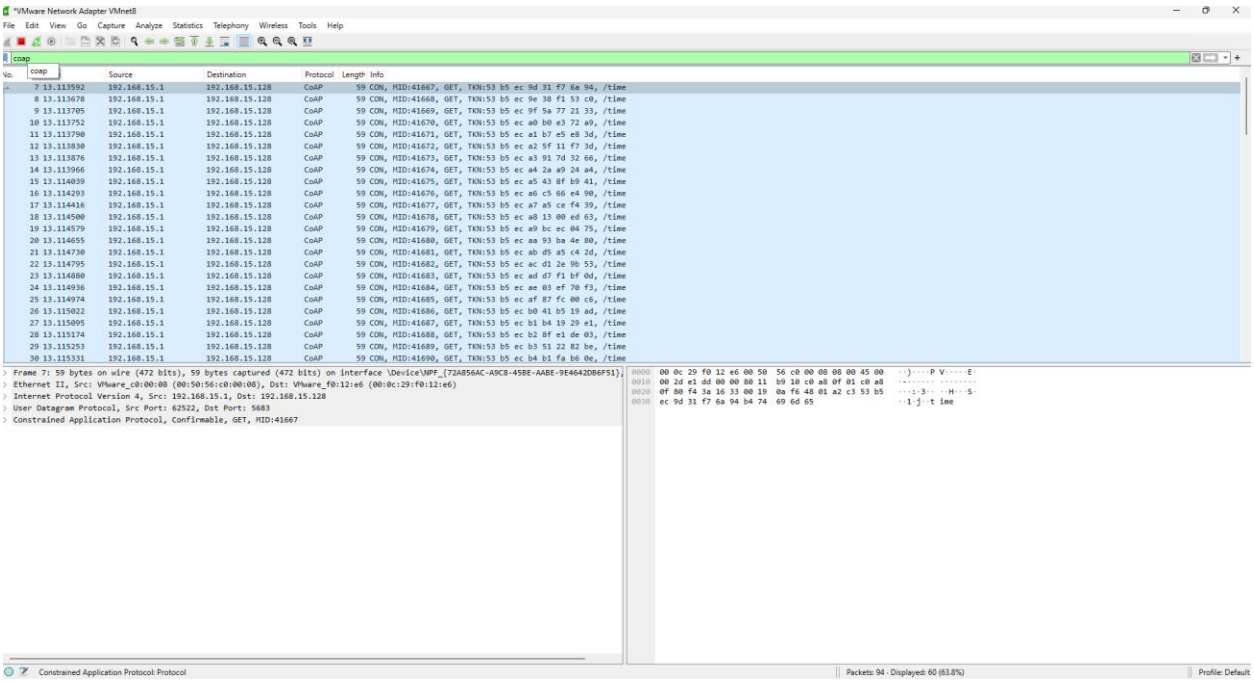
0% with 1000ms delay





FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657

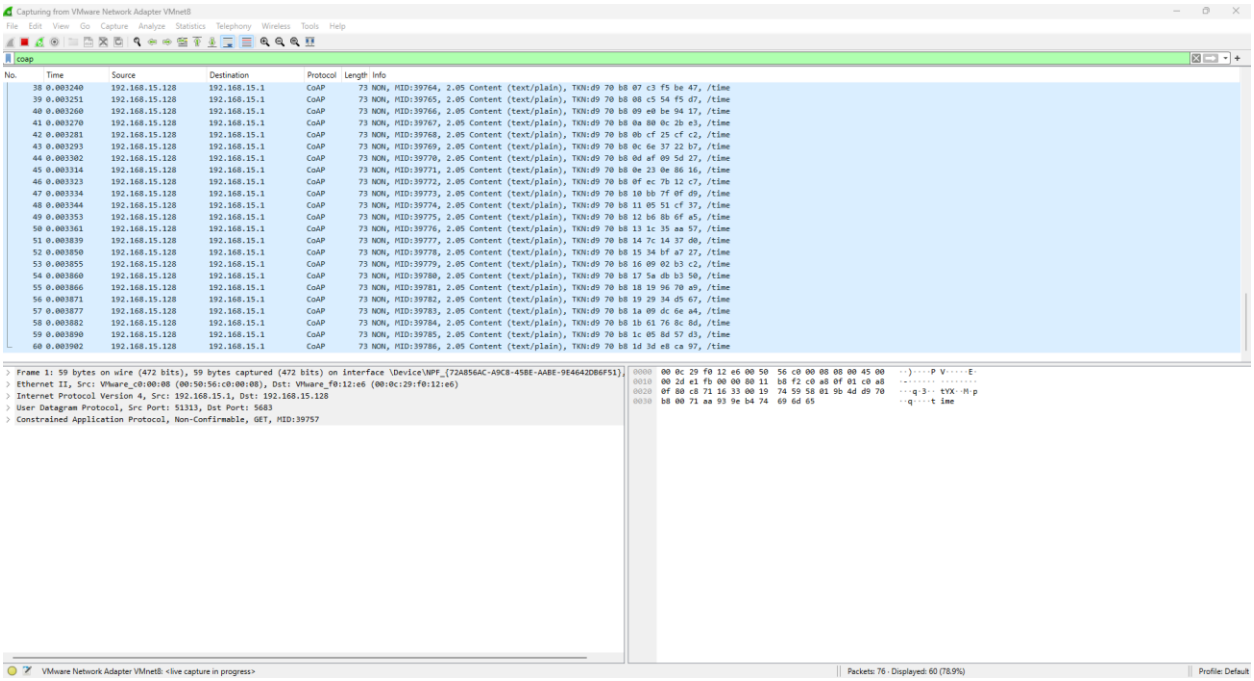


Non confirmable

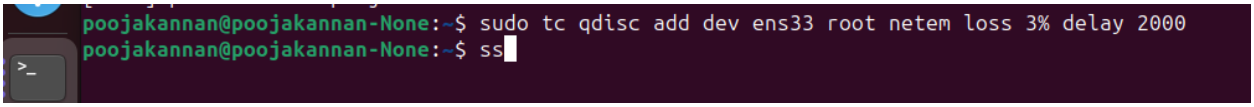
```
PS C:\Users\pooja\Downloads\CoAp-Test> node coap_client.js
  Value 1: Dec 16 21:10:01
  Value 2: Dec 16 21:10:01
  Value 3: Dec 16 21:10:01
  Value 4: Dec 16 21:10:01
  Value 5: Dec 16 21:10:01
  Value 6: Dec 16 21:10:01
  Value 7: Dec 16 21:10:01
  Value 8: Dec 16 21:10:01
  Value 9: Dec 16 21:10:01
  Value 10: Dec 16 21:10:01
  Value 11: Dec 16 21:10:01
  Value 12: Dec 16 21:10:01
  Value 13: Dec 16 21:10:01
  Value 14: Dec 16 21:10:01
  Value 15: Dec 16 21:10:01
  Value 16: Dec 16 21:10:01
  Value 17: Dec 16 21:10:01
  Value 18: Dec 16 21:10:01
  Value 19: Dec 16 21:10:01
  Value 20: Dec 16 21:10:01
  Value 21: Dec 16 21:10:01
  Value 22: Dec 16 21:10:01
  Value 23: Dec 16 21:10:01
  Value 24: Dec 16 21:10:01
  Value 25: Dec 16 21:10:01
  Value 26: Dec 16 21:10:01
  Value 27: Dec 16 21:10:01
  Value 28: Dec 16 21:10:01
  Value 29: Dec 16 21:10:01
  Value 30: Dec 16 21:10:01
  PS C:\Users\pooja\Downloads\CoAp-Test>
```

FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657



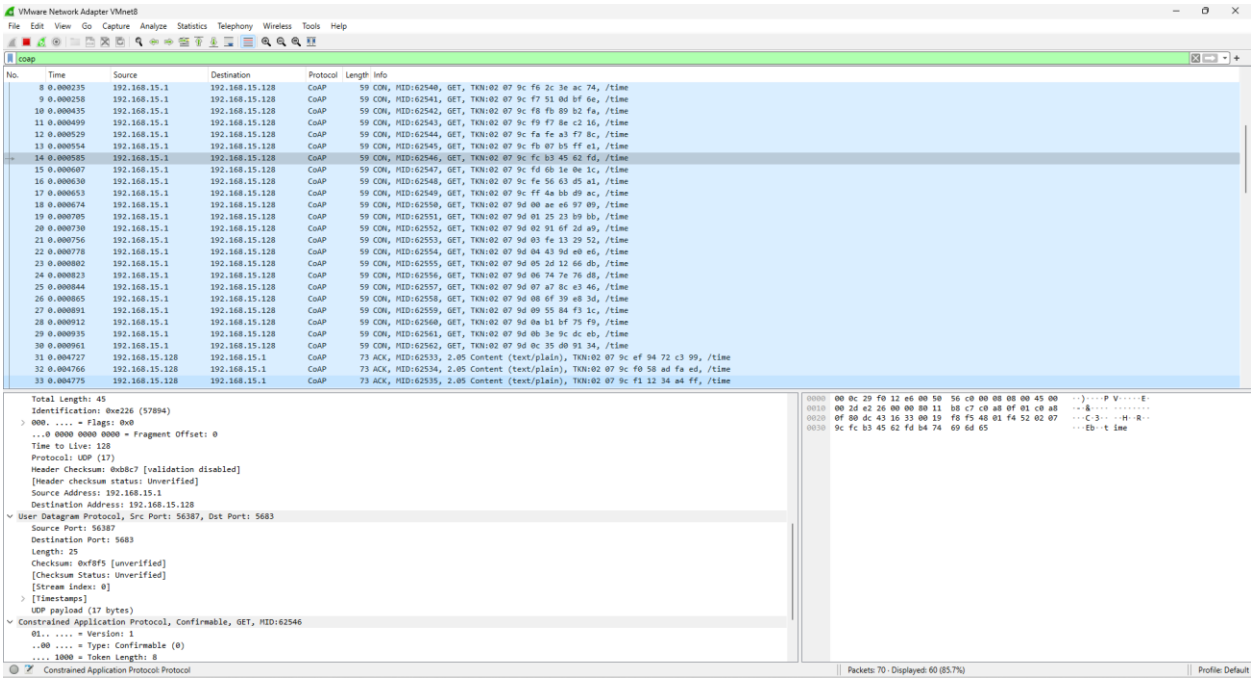
3% loss with 2000ms delay



Confirmable

FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657

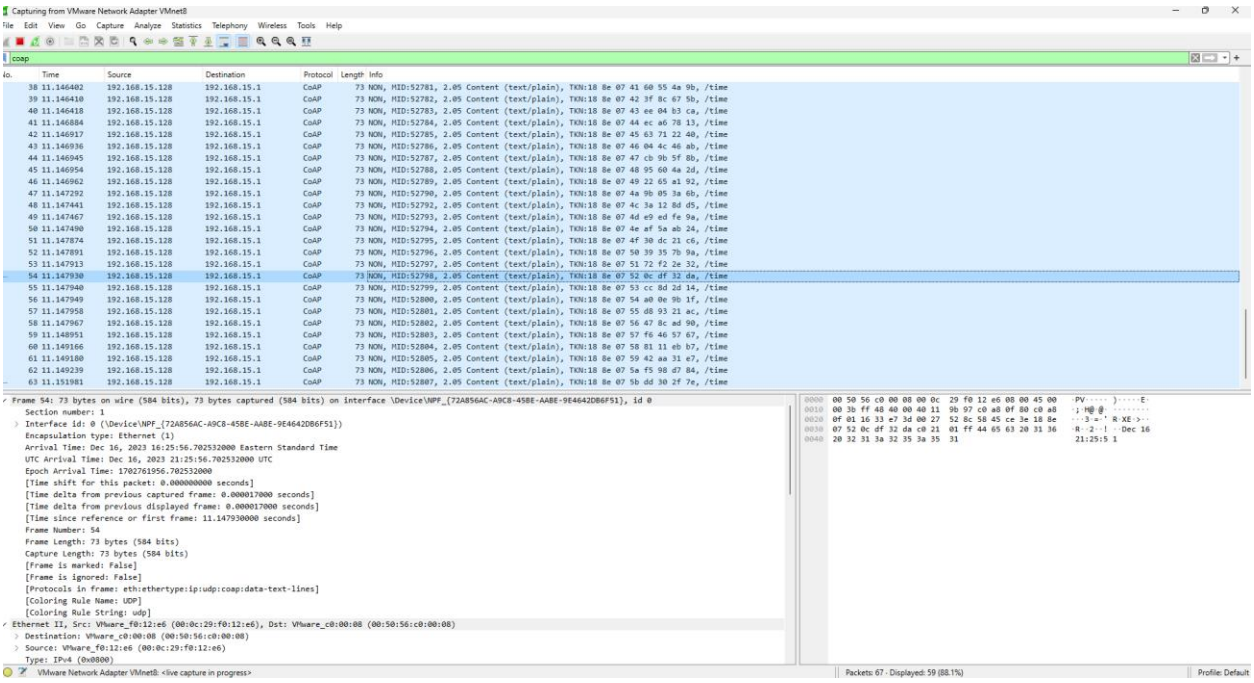


Non-confirmable

```
PS C:\Users\pooja\Downloads\CoAp-Test> node coap_client.js
  Value 1: Dec 16 21:25:51
  Value 2: Dec 16 21:25:51
  Value 3: Dec 16 21:25:51
  Value 4: Dec 16 21:25:51
  Value 5: Dec 16 21:25:51
  Value 6: Dec 16 21:25:51
  Value 7: Dec 16 21:25:51
  Value 8: Dec 16 21:25:51
  Value 9: Dec 16 21:25:51
  Value 10: Dec 16 21:25:51
  Value 11: Dec 16 21:25:51
  Value 12: Dec 16 21:25:51
  Value 13: Dec 16 21:25:51
  Value 15: Dec 16 21:25:51
  Value 16: Dec 16 21:25:51
  Value 17: Dec 16 21:25:51
  Value 18: Dec 16 21:25:51
  Value 19: Dec 16 21:25:51
  Value 20: Dec 16 21:25:51
  Value 21: Dec 16 21:25:51
  Value 22: Dec 16 21:25:51
  Value 23: Dec 16 21:25:51
  Value 24: Dec 16 21:25:51
  Value 25: Dec 16 21:25:51
  Value 26: Dec 16 21:25:51
  Value 27: Dec 16 21:25:51
  Value 28: Dec 16 21:25:51
  Value 29: Dec 16 21:25:51
  Value 30: Dec 16 21:25:51
```

FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657



5%loss with 3000ms delay

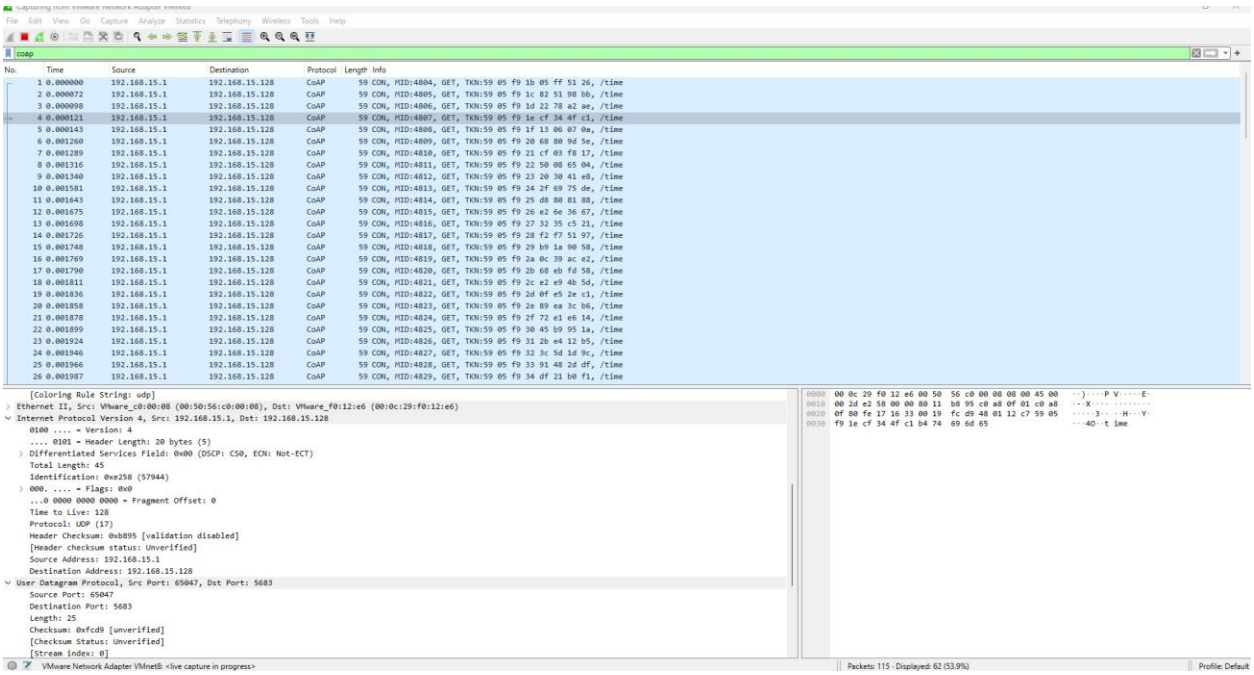
```
poojakannan@poojakannan-None:~$ sudo tc qdisc add dev ens33 root netem loss 5% delay 3000
poojakannan@poojakannan-None:~$
```

Confirmable

```
PS C:\Users\pooja\Downloads\CoAp-Test> node coap_client.js
  Value 1: Dec 16 21:29:58
  Value 2: Dec 16 21:29:58
  Value 4: Dec 16 21:29:58
  Value 5: Dec 16 21:29:58
  Value 6: Dec 16 21:29:58
  Value 7: Dec 16 21:29:58
  Value 8: Dec 16 21:29:58
  Value 9: Dec 16 21:29:58
  Value 11: Dec 16 21:29:58
  Value 12: Dec 16 21:29:58
  Value 13: Dec 16 21:29:58
  Value 14: Dec 16 21:29:58
  Value 15: Dec 16 21:29:58
  Value 16: Dec 16 21:29:58
  Value 17: Dec 16 21:29:58
  Value 18: Dec 16 21:29:58
  Value 19: Dec 16 21:29:58
  Value 20: Dec 16 21:29:58
  Value 21: Dec 16 21:29:58
  Value 22: Dec 16 21:29:58
  Value 23: Dec 16 21:29:58
  Value 24: Dec 16 21:29:58
  Value 25: Dec 16 21:29:58
  Value 26: Dec 16 21:29:58
  Value 27: Dec 16 21:29:58
  Value 28: Dec 16 21:29:58
  Value 29: Dec 16 21:29:58
  Value 30: Dec 16 21:29:58
  Value 10: Dec 16 21:30:00
  Value 3: Dec 16 21:30:00
PS C:\Users\pooja\Downloads\CoAp-Test>
```

FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657

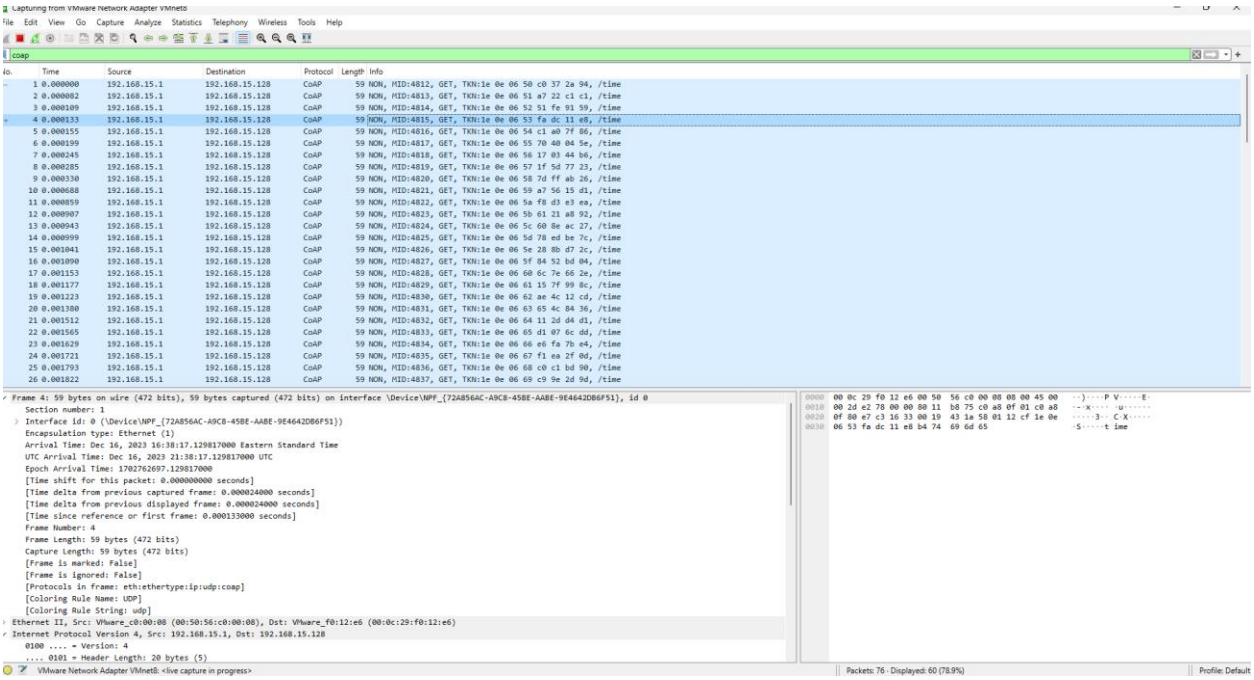


Non-confirmable

```
PS C:\Users\pooja\Downloads\CoAp-Test> node coap_client.js
  Value 1: Dec 16 21:38:11
  Value 2: Dec 16 21:38:11
  Value 3: Dec 16 21:38:11
  Value 4: Dec 16 21:38:11
  Value 5: Dec 16 21:38:11
  Value 6: Dec 16 21:38:11
  Value 7: Dec 16 21:38:11
  Value 8: Dec 16 21:38:11
  Value 9: Dec 16 21:38:11
  Value 10: Dec 16 21:38:11
  Value 11: Dec 16 21:38:11
  Value 12: Dec 16 21:38:11
  Value 13: Dec 16 21:38:11
  Value 14: Dec 16 21:38:11
  Value 15: Dec 16 21:38:11
  Value 16: Dec 16 21:38:11
  Value 17: Dec 16 21:38:11
  Value 18: Dec 16 21:38:11
  Value 19: Dec 16 21:38:11
  Value 20: Dec 16 21:38:11
  Value 21: Dec 16 21:38:11
  Value 22: Dec 16 21:38:11
  Value 23: Dec 16 21:38:11
  Value 24: Dec 16 21:38:11
  Value 25: Dec 16 21:38:11
  Value 26: Dec 16 21:38:11
  Value 27: Dec 16 21:38:11
  Value 28: Dec 16 21:38:11
  Value 29: Dec 16 21:38:11
  Value 30: Dec 16 21:38:11
PS C:\Users\pooja\Downloads\CoAp-Test>
```

FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657



10% loss with 4000ms delay

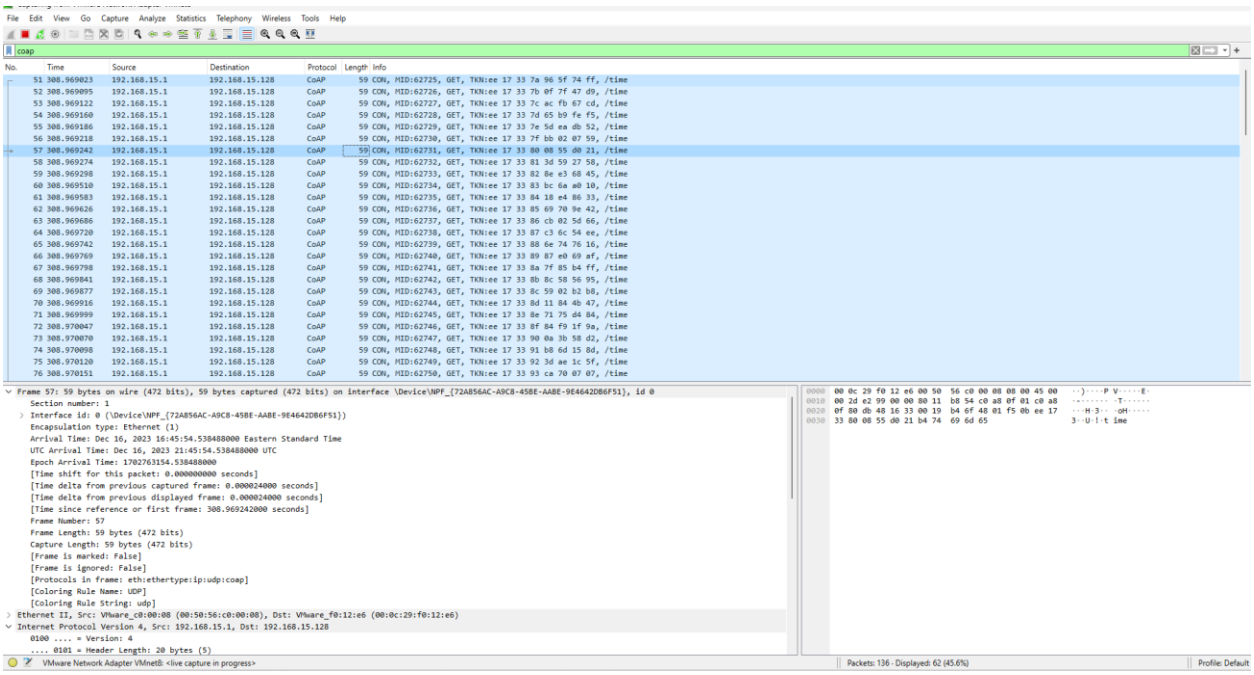
```
poojakannan@poojakannan-None:~$ sudo tc qdisc add dev ens33 root netem loss 10% delay 4000
poojakannan@poojakannan-None:~$
```

Confirmable

```
PS C:\Users\pooja\Downloads\CoAp-Test> node coap_client.js
  Value 1: Dec 16 21:45:48
  Value 2: Dec 16 21:45:48
  Value 3: Dec 16 21:45:48
  Value 4: Dec 16 21:45:48
  Value 5: Dec 16 21:45:48
  Value 6: Dec 16 21:45:48
  Value 7: Dec 16 21:45:48
  Value 8: Dec 16 21:45:48
  Value 9: Dec 16 21:45:48
  Value 10: Dec 16 21:45:48
  Value 11: Dec 16 21:45:48
  Value 12: Dec 16 21:45:48
  Value 13: Dec 16 21:45:48
  Value 14: Dec 16 21:45:48
  Value 15: Dec 16 21:45:48
  Value 16: Dec 16 21:45:48
  Value 17: Dec 16 21:45:48
  Value 18: Dec 16 21:45:48
  Value 19: Dec 16 21:45:48
  Value 20: Dec 16 21:45:48
  Value 22: Dec 16 21:45:48
  Value 23: Dec 16 21:45:48
  Value 24: Dec 16 21:45:48
  Value 25: Dec 16 21:45:48
  Value 26: Dec 16 21:45:48
  Value 28: Dec 16 21:45:48
  Value 29: Dec 16 21:45:48
  Value 30: Dec 16 21:45:48
  Value 21: Dec 16 21:45:51
  Value 27: Dec 16 21:45:51
  PS C:\Users\pooja\Downloads\CoAp-Test>
```

FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657

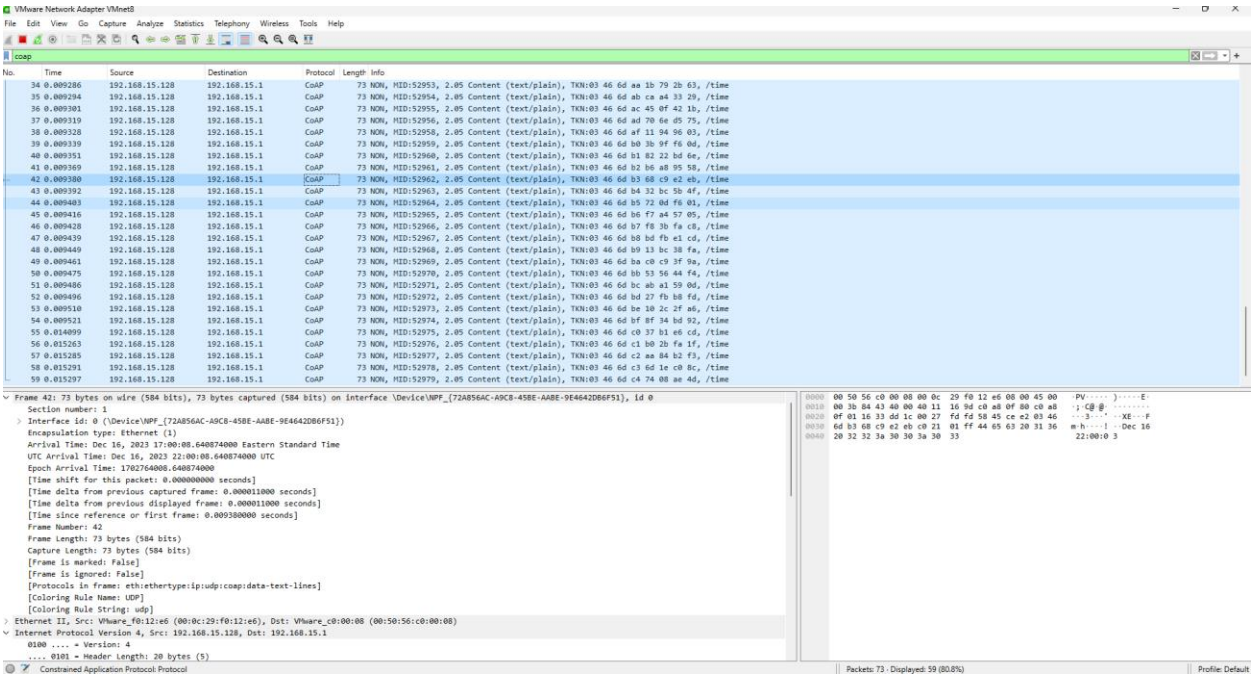


Non-confirmable

```
PS C:\Users\pooja\Downloads\CoAp-Test> node coap_client.js
  Value 1: Dec 16 22:00:03
  Value 2: Dec 16 22:00:03
  Value 3: Dec 16 22:00:03
  Value 4: Dec 16 22:00:03
  Value 5: Dec 16 22:00:03
  Value 6: Dec 16 22:00:03
  Value 7: Dec 16 22:00:03
  Value 9: Dec 16 22:00:03
  Value 10: Dec 16 22:00:03
  Value 11: Dec 16 22:00:03
  Value 12: Dec 16 22:00:03
  Value 13: Dec 16 22:00:03
  Value 14: Dec 16 22:00:03
  Value 15: Dec 16 22:00:03
  Value 16: Dec 16 22:00:03
  Value 17: Dec 16 22:00:03
  Value 18: Dec 16 22:00:03
  Value 19: Dec 16 22:00:03
  Value 20: Dec 16 22:00:03
  Value 21: Dec 16 22:00:03
  Value 22: Dec 16 22:00:03
  Value 23: Dec 16 22:00:03
  Value 24: Dec 16 22:00:03
  Value 25: Dec 16 22:00:03
  Value 26: Dec 16 22:00:03
  Value 27: Dec 16 22:00:03
  Value 28: Dec 16 22:00:03
  Value 29: Dec 16 22:00:03
  Value 30: Dec 16 22:00:03
```

FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657



Loss	Confirmable	Non-Confirmable
0%	0.0032168	0.0028064
3%	0.00540223	0.0053908
5%	0.15767796	0.0058193
10%	0.1761446	0.0097769

Describe the messages that are exchanged between client and server for each case (what are the flows?)

In CoAP communication between a client and a server, the flow of messages varies based on whether the traffic is confirmable or non-confirmable:

Confirmable (CON) Traffic:

Client Request: The client sends a confirmable CoAP request, which typically includes a method (like GET, POST), a URI, and may include optional payload or query parameters. Each request is tagged with a unique Message ID.

Server Acknowledgement (ACK): The server, upon receipt of the request, sends back an acknowledgement (ACK) message, confirming the request's reception.

Pooja Kannan
002845657

Server Response: Following the ACK, the server processes the request and sends back a response. This response, also a confirmable CoAP message, contains a status code (2.xx for successful operations, 4.xx or 5.xx for errors), and may include an optional payload and other pertinent details.

Client Response Acknowledgement (ACK): The client acknowledges the server's response by sending back an ACK, indicating successful receipt of the response.

Non-confirmable (NON) Traffic:

Client Request: The client dispatches a non-confirmable (NON) CoAP request to the server. This also encompasses a method, URI, and potentially optional payload or query parameters, along with a Message ID.

Server Response: The server, after receiving the non-confirmable request, processes it and prepares a response. This response, sent as a non-confirmable CoAP message, includes a response code and may contain an optional payload and additional information.

In both scenarios, the client and server engage in an exchange of CoAP messages to initiate requests and deliver responses. The key distinction lies in the acknowledgment process: Confirmable (CON) messages necessitate explicit acknowledgments from the recipient, whereas non-confirmable (NON) messages do not require such acknowledgments.

How does CoAP compare against MQTT for same network conditions?

CoAP (Constrained Application Protocol) and MQTT (Message Queuing Telemetry Transport) are popular protocols in Internet of Things (IoT) and machine-to-machine (M2M) communication. Here's a comparative analysis under similar network conditions:

1. Messaging Model:
   CoAP: Operates on a request-response model. Clients (CoAP nodes) make requests to servers (CoAP endpoints) and await responses.
   MQTT: Utilizes a publish-subscribe model. Publishers (MQTT clients) send messages to a broker, and subscribers then receive these messages.

2. Designed for Resource Constraints:
   CoAP: Specifically built for devices and networks with limited resources, such as low-power devices in lossy networks, using UDP for minimal overhead.
   MQTT: Although lightweight, it's not specifically designed for constrained environments. It's suitable but doesn’t offer the same level of optimization as CoAP.

3. Underlying Transport Protocol:
   CoAP: Uses UDP, which is connectionless and has low overhead, ideal for situations needing low latency and efficiency but does not ensure reliable delivery.
   MQTT: Typically employs TCP, ensuring reliable and ordered message delivery at the cost of extra overhead.

FIOT Final Project – MQTT & COAP

Pooja Kannan
002845657

4. Message Size and Network Overhead:
CoAP: Generally features smaller message sizes compared to MQTT, leading to lower network overhead. It employs a compact binary header format.
MQTT: Messages have greater overhead due to additional header information and a text-based format, impacting network utilization in constrained environments.

5. Quality of Service (QoS) Support:
CoAP: Offers various QoS levels including Confirmable, Non-confirmable, Acknowledged, and Unacknowledged, allowing trade-offs between reliability and latency.
MQTT: Provides three QoS levels (at most once, at least once, and exactly once), catering to different needs for reliability and latency.

In conclusion, CoAP and MQTT cater to different scenarios based on their design principles. CoAP is ideal for constrained devices and networks, offering efficient communication with lower overhead but less reliability. MQTT, in contrast, provides more robustness and reliability with its publish-subscribe model and TCP transport, but at a higher overhead. The selection between CoAP and MQTT should be based on the specific demands and limitations of the IoT application and the network environment.