

#1 Following steps to subscribe to each of the varying QoS levels:

1. client initiates connection with the CONNECT message

Important points noted:

- ClientId identifies each MQTT client that connects to an MQTT broker
- Keep Alive identifies the time interval in seconds which the client specifies and communicates to the broker. This interval defines the longest period of the time that the broker and client could endure without sending a message

2. Broker response with a CONNACK message

Points noted:

- Return code is '0' means connection has been accepted
- Sessionpresent is not visible

3. the client sends a SUBSCRIBE message to the MQTT broker.

Points noted:

- Subscribe message for each qos level is the same except for the qos level being passed.
- Message/packet identifier uniquely identifies a message as it flows between the client and broker.

4. the broker sends a SUBACK acknowledgement message to the client. Which consists of the same identifier used in subscribe with return code whether or not the subscription has been accepted for the corresponding qos level(0 for qos=0,1 for qos=1...)

5. For QoS=0 the only packet received is the actual message

For QoS=1, two packets are seen the one which is the actual message and next the acknowledging packet the PUBACK has the same identifier as the original message

For QoS=2, four packets are captured. The original message, PUBREC from the receiver, PUBREL from the sender and PUBCOMP from the receiver. PUBREC indicates that the publish has been received. PUBREL when the sender receives the PUBREC it responds with the PUBREL. PUBCOMP when the receiver receives PUBEL it responds with PUBCOMP After the sender receives the PUBCOMP packet, the packet identifier of the published message becomes available for reuse.

When to choose each QoS level..

QoS 0 when the connection between sender and receiver is mostly stable perhaps over a wired connection. Few messages loss are tolerable

QoS 1 when the receiver can handle duplicates properly and process them accordingly since it can send more than once the same messages and a likely less overhead than QoS 2

QoS 2 when the messages are critical and not be received only once. In cases where duplicity can harm the running application.

784	26.986319449	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message	[counter/slow/q0]
788	28.012307385	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message	[counter/slow/q0]
792	29.036461215	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message	[counter/slow/q0]
797	30.061421589	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message	[counter/slow/q0]
813	31.084597800	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message	[counter/slow/q0]
828	32.117099302	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message	[counter/slow/q0]
4246	33.139627545	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message	[counter/slow/q0]
4357	33.426924106	192.168.112.26	52.15.211.32	MQTT	113 90:e2:ba:...	Connect Command	
4360	33.670754599	52.15.211.32	192.168.112.26	MQTT	70 d8:9e:f3:...	Connect Ack	
4362	33.672001702	192.168.112.26	52.15.211.32	MQTT	88 90:e2:ba:...	Subscribe Request (id=1)	[counter/slow/q0]
4366	33.917677104	52.15.211.32	192.168.112.26	MQTT	71 d8:9e:f3:...	Subscribe Ack (id=1)	
4384	34.205797387	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message	[counter/slow/q0]
4387	35.193988537	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message	[counter/slow/q0]
4395	36.222616415	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message	[counter/slow/q0]
4398	37.249251418	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message	[counter/slow/q0]
4403	38.317312534	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message	[counter/slow/q0]
▶ [SEQ/ACK analysis] ▶ [Timestamps] TCP payload (47 bytes) [PDU Size: 47] * MQ Telemetry Transport Protocol, Connect Command ▶ Header Flags: 0x10, Message Type: Connect Command 0001 = Message Type: Connect Command (1) 0000 = Reserved: 0 Msg Len: 45 Protocol Name Length: 4 Protocol Name: MQTT Version: MQTT v3.1.1 (4) ▶ Connect Flags: 0xc2, User Name Flag, Password Flag, QoS Level: At most once delivery (Fire and Forget), Clean Session Flag Keep Alive: 60 Client ID Length: 13 Client ID: 3310-u6439994 User Name Length: 8 User Name: students Password Length: 8 Password: 33106331							
0000	90 e2 ba 29 d1 cc d8 9e f3 73 91 34 08 00 45 00	...).... s.4...E.					
0010	00 63 63 8d 40 00 40 06 9f 15 c0 a8 70 1a 34 0f	.cc.@.@...p.4.					
0020	d3 20 a7 1f 07 5b 4c 99 c2 e7 87 18 68 c7 80 18	...[L...h...					
0030	00 1d 38 48 00 00 01 01 08 0a 74 9b 10 78 15 cc	...8H...t...x...					
0040	cc 4f 16 2d 00 04 4d 51 54 54 04 c2 00 3c 00 0d	0MQTT...<...					
0050	33 33 31 30 2d 75 36 34 33 39 39 39 34 00 08 73	3310-u64 39994..s					
0060	74 75 64 65 6e 74 73 00 08 33 33 31 30 36 33 33	tudents..3310633					
0070	31	1					

Image shows the packet connect message .

828	32.117099302	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message	[counter/slow/q0]
4246	33.139627545	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message	[counter/slow/q0]
4357	33.426924106	192.168.112.26	52.15.211.32	MQTT	113 90:e2:ba:...	Connect Command	
4360	33.670754599	52.15.211.32	192.168.112.26	MQTT	70 d8:9e:f3:...	Connect Ack	
4362	33.672001702	192.168.112.26	52.15.211.32	MQTT	88 90:e2:ba:...	Subscribe Request (id=1)	[counter/slow/q0]
4366	33.917677104	52.15.211.32	192.168.112.26	MQTT	71 d8:9e:f3:...	Subscribe Ack (id=1)	
4384	34.205797387	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message	[counter/slow/q0]
4387	35.193988537	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message	[counter/slow/q0]
4395	36.222616415	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message	[counter/slow/q0]
4398	37.249251418	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message	[counter/slow/q0]
4403	38.317312534	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message	[counter/slow/q0]
1000 = Header Length: 32 bytes (8) ▶ Flags: 0x018 (PSH, ACK) Window size value: 210 [Calculated window size: 26880] [Window size scaling factor: 128] Checksum: 0x07bd [unverified] [Checksum Status: Unverified] Urgent pointer: 0 ▶ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps ▶ [SEQ/ACK analysis] ▶ [Timestamps] TCP payload (4 bytes) [PDU Size: 4] * MQ Telemetry Transport Protocol, Connect Ack ▶ Header Flags: 0x20, Message Type: Connect Ack 0010 = Message Type: Connect Ack (2) 0000 = Reserved: 0 Msg Len: 2 ▶ Acknowledge Flags: 0x00 Return Code: Connection Accepted (0)							
0000	d8 9e f3 73 91 34 90 e2 ba 29 d1 cc 08 00 45 00	...s.4...)....E.					
0010	00 38 3e 00 40 00 1b 06 e9 cd 34 0f d3 20 c0 a8	.8>.@...4...					
0020	70 1a 07 5b a7 1f 87 18 68 c7 4c 99 c3 16 80 18	p>[... h.L...					
0030	00 d2 07 bd 00 00 01 01 08 0a 15 cc cd 43 74 9bCT.					
0040	10 78 26 02 00 00	.X...					

Images shows the packet of connectack message from broker

4357	33.426924186	192.168.112.26	52.15.211.32	MQTT	113 90:e2:ba:...	Connect Command
4360	33.670754599	52.15.211.32	192.168.112.26	MQTT	70 d8:9e:f3:...	Connect Ack
4362	33.672001702	192.168.112.26	52.15.211.32	MQTT	88 90:e2:ba:...	Subscribe Request (id=1) [counter/slow/q0]
4366	33.917677104	52.15.211.32	192.168.112.26	MQTT	71 d8:9e:f3:...	Subscribe Ack (id=1)
4384	34.205797387	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message [counter/slow/q0]
4387	35.193988537	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message [counter/slow/q0]
4395	36.222616415	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message [counter/slow/q0]
4398	37.249251418	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message [counter/slow/q0]
4403	38.317312534	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message [counter/slow/q0]

Window size value: 29
[Calculated window size: 29696]
[Window size scaling factor: 1024]
Checksum: 0x382f [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[SEQ/ACK analysis]
[Timestamps]
TCP payload (22 bytes)
[PDU Size: 22]

MQ Telemetry Transport Protocol, Subscribe Request

Header Flags: 0x02, Message Type: Subscribe Request

1000 = Message Type: Subscribe Request (8)
.... 0010 = Reserved: 2
Msg Len: 20
Message Identifier: 1
Topic Length: 15
Topic: counter/slow/q0
Requested QoS: At most once delivery (Fire and Forget) (0)

0000 90 e2 ba 29 d1 cc d8 9e f3 73 91 34 08 00 45 00s.4..E.
0010 00 4a 63 8f 40 00 40 06 9f 2c c0 a8 70 1a 34 0f .Jc.@@. ., .p.4.
0020 d3 20 a7 1f 07 5b 4c 99 c3 16 87 18 68 cb 80 18[L.h...
0030 00 1d 38 2f 00 00 01 01 08 0a 74 9b 11 6d 15 cc .8/.... .t..m..
0040 cd 43 82 14 00 01 00 0f 63 6f 75 6e 74 65 72 2f .C..... counter/
0050 73 6c 6f 77 2f 71 30 00 slow/q0

Packet for subscribe request for QoS 0

4362	33.672001702	192.168.112.26	52.15.211.32	MQTT	88 90:e2:ba:...	Subscribe Request (id=1) [counter/slow/q0]
4366	33.917677104	52.15.211.32	192.168.112.26	MQTT	71 d8:9e:f3:...	Subscribe Ack (id=1)
4384	34.205797387	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message [counter/slow/q0]
4387	35.193988537	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message [counter/slow/q0]
4395	36.222616415	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message [counter/slow/q0]
4398	37.249251418	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message [counter/slow/q0]
4403	38.317312534	52.15.211.32	192.168.112.26	MQTT	88 d8:9e:f3:...	Publish Message [counter/slow/q0]

1000 = Header Length: 32 bytes (8)
Flags: 0x018 (PSH, ACK)
Window size value: 210
[Calculated window size: 26880]
[Window size scaling factor: 128]
Checksum: 0x95b3 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[SEQ/ACK analysis]
[Timestamps]
TCP payload (5 bytes)
[PDU Size: 5]

MQ Telemetry Transport Protocol, Subscribe Ack

Header Flags: 0x00, Message Type: Subscribe Ack

1001 = Message Type: Subscribe Ack (9)
.... 0000 = Reserved: 0
Msg Len: 3
Message Identifier: 1
Granted QoS: At most once delivery (Fire and Forget) (0)

0000 d8 9e f3 73 91 34 90 e2 ba 29 d1 cc 08 00 45 00s.4..E.
0010 00 39 3e 01 40 00 1b 06 e9 cb 34 0f d3 20 c0 a8 .9>.@... .4...
0020 70 1a 07 5b a7 1f 87 18 68 cb 4c 99 c3 2c 80 18 p>[.... h.L...
0030 00 d2 95 b3 00 00 01 01 08 0a 15 cc ce 3a 74 9b:t..
0040 11 6d 90 03 00 01 00 .m.....

Packet for subscribe ack for QoS 0

1494	5.279727387	192.168.112.26	52.15.211.32	MQTT	70 90:e2:ba:...	Publish Received (id=1)
1497	5.526838275	52.15.211.32	192.168.112.26	MQTT	70 d8:9e:f3:...	Publish Release (id=1)
1498	5.528349441	192.168.112.26	52.15.211.32	MQTT	70 90:e2:ba:...	Publish Complete (id=1)
1510	6.267482146	52.15.211.32	192.168.112.26	MQTT	91 d8:9e:f3:...	Publish Message (id=2) [counter/slow/q2]
1511	6.267824487	192.168.112.26	52.15.211.32	MQTT	70 90:e2:ba:...	Publish Received (id=2)
1513	6.515011766	52.15.211.32	192.168.112.26	MQTT	70 d8:9e:f3:...	Publish Release (id=2)
1514	6.516071669	192.168.112.26	52.15.211.32	MQTT	70 90:e2:ba:...	Publish Complete (id=2)
1521	7.296147522	52.15.211.32	192.168.112.26	MQTT	91 d8:9e:f3:...	Publish Message (id=3) [counter/slow/q2]
1522	7.296447866	192.168.112.26	52.15.211.32	MQTT	70 90:e2:ba:...	Publish Received (id=3)
2076	7.543255387	52.15.211.32	192.168.112.26	MQTT	70 d8:9e:f3:...	Publish Release (id=3)
2079	7.544286679	192.168.112.26	52.15.211.32	MQTT	70 90:e2:ba:...	Publish Complete (id=3)
2122	8.364020975	52.15.211.32	192.168.112.26	MQTT	91 d8:9e:f3:...	Publish Message (id=4) [counter/slow/q2]
2123	8.364245514	192.168.112.26	52.15.211.32	MQTT	70 90:e2:ba:...	Publish Received (id=4)

Acknowledgment number: 35 (relative ack number)
 1000 = Header Length: 32 bytes (8)
 ▶ Flags: 0x018 (PSH, ACK)
 Window size value: 29
 [Calculated window size: 29696]
 [Window size scaling factor: 1024]
 Checksum: 0x381d [unverified]
 [Checksum Status: Unverified]
 Urgent pointer: 0
 ▶ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
 ▶ [SEQ/ACK analysis]
 ▶ [Timestamps]
 TCP payload (4 bytes)
 [PDU Size: 4]
 MQ Telemetry Transport Protocol, Publish Received
 ▼ Header Flags: 0x50, Message Type: Publish Received
 0101 = Message Type: Publish Received (5)
 0000 = Reserved: 0
 Msg Len: 2
 Message Identifier: 1

Image captured of message received

33522	312.463765595	192.168.112.26	52.15.211.32	MQTT	70 90:e2:ba:...	Publish Ack (id=1)
33525	313.451470840	52.15.211.32	192.168.112.26	MQTT	90 d8:9e:f3:...	Publish Message (id=2) [counter/slow/q1]
33526	313.451725530	192.168.112.26	52.15.211.32	MQTT	70 90:e2:ba:...	Publish Ack (id=2)
33534	314.483948006	52.15.211.32	192.168.112.26	MQTT	90 d8:9e:f3:...	Publish Message (id=3) [counter/slow/q1]
33535	314.484262737	192.168.112.26	52.15.211.32	MQTT	70 90:e2:ba:...	Publish Ack (id=3)
33540	315.468572905	52.15.211.32	192.168.112.26	MQTT	90 d8:9e:f3:...	Publish Message (id=4) [counter/slow/q1]
33541	315.468790099	192.168.112.26	52.15.211.32	MQTT	70 90:e2:ba:...	Publish Ack (id=4)
33547	316.576011133	52.15.211.32	192.168.112.26	MQTT	90 d8:9e:f3:...	Publish Message (id=5) [counter/slow/q1]

Acknowledgment number: 34 (relative ack number)
 1000 = Header Length: 32 bytes (8)
 ▶ Flags: 0x018 (PSH, ACK)
 Window size value: 29
 [Calculated window size: 29696]
 [Window size scaling factor: 1024]
 Checksum: 0x381d [unverified]
 [Checksum Status: Unverified]
 Urgent pointer: 0
 ▶ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
 ▶ [SEQ/ACK analysis]
 ▶ [Timestamps]
 TCP payload (4 bytes)
 [PDU Size: 4]
 MQ Telemetry Transport Protocol, Publish Ack
 ▼ Header Flags: 0x40, Message Type: Publish Ack
 0100 = Message Type: Publish Ack (4)
 0000 = Reserved: 0
 Msg Len: 2
 Message Identifier: 1

```

0000  90 e2 ba 29 d1 cc d8 9e f3 73 91 34 08 00 45 00  ... ).... s 4..E.
0010  00 30 df 4c 40 00 40 06 23 81 c0 a8 70 1a 34 0f  .8.L@.#...p.4.
0020  d3 20 db 01 07 5b 40 2c 70 24 07 ea a8 c9 80 18  ...[@,p$.....
0030  00 1d 38 1d 00 00 01 01 08 0a 74 9f 52 72 15 d1  ..8.....t.Rr..
0040  0e 54 40 02 00 01  ...T@...
  
```

Image captured of the acknowledgement send over the message received.

1498	5.528349441	192.168.112.26	52.15.211.32	MQTT	70 90:e2:ba:...	Publish Complete (id=1)
1510	6.267482146	192.168.112.26	52.15.211.32	MQTT	91 d8:9e:f3:...	Publish Message (id=2) [counter/slow/q2]
1511	6.267824487	192.168.112.26	52.15.211.32	MQTT	70 90:e2:ba:...	Publish Received (id=2)
1513	6.515011766	192.168.112.26	52.15.211.32	MQTT	70 d8:9e:f3:...	Publish Release (id=2)
1514	6.516071669	192.168.112.26	52.15.211.32	MQTT	70 90:e2:ba:...	Publish Complete (id=2)
1521	7.296147522	192.168.112.26	52.15.211.32	MQTT	91 d8:9e:f3:...	Publish Message (id=3) [counter/slow/q2]
1522	7.296447866	192.168.112.26	52.15.211.32	MQTT	70 90:e2:ba:...	Publish Received (id=3)
2076	7.543255387	192.168.112.26	52.15.211.32	MQTT	70 d8:9e:f3:...	Publish Release (id=3)
2079	7.544286679	192.168.112.26	52.15.211.32	MQTT	70 90:e2:ba:...	Publish Complete (id=3)
2122	8.364020975	192.168.112.26	52.15.211.32	MQTT	91 d8:9e:f3:...	Publish Message (id=4) [counter/slow/q2]
2123	8.364245514	192.168.112.26	52.15.211.32	MQTT	70 90:e2:ba:...	Publish Received (id=4)

Acknowledgment number: 74 (relative ack number)
1000 = Header Length: 32 bytes (8)
Flags: 0x018 (PSH, ACK)
Window size value: 210
[Calculated window size: 26880]
[Window size scaling factor: 128]
Checksum: 0xa9c3 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[SEQ/ACK analysis]
[Timestamps]
TCP payload (4 bytes)
[PDU Size: 4]
MQ Telemetry Transport Protocol, Publish Release
Header Flags: 0x62, Message Type: Publish Release
0110 = Message Type: Publish Release (6)
.... 0010 = Reserved: 2
Msg Len: 2
Message Identifier: 1

PUBREL packet for QoS 2

1498	5.528349441	192.168.112.26	52.15.211.32	MQTT	70 90:e2:ba:...	Publish Complete (id=1)
1510	6.267482146	192.168.112.26	52.15.211.32	MQTT	91 d8:9e:f3:...	Publish Message (id=2) [counter/slow/q2]
1511	6.267824487	192.168.112.26	52.15.211.32	MQTT	70 90:e2:ba:...	Publish Received (id=2)
1513	6.515011766	192.168.112.26	52.15.211.32	MQTT	70 d8:9e:f3:...	Publish Release (id=2)
1514	6.516071669	192.168.112.26	52.15.211.32	MQTT	70 90:e2:ba:...	Publish Complete (id=2)
1521	7.296147522	192.168.112.26	52.15.211.32	MQTT	91 d8:9e:f3:...	Publish Message (id=3) [counter/slow/q2]
1522	7.296447866	192.168.112.26	52.15.211.32	MQTT	70 90:e2:ba:...	Publish Received (id=3)
2076	7.543255387	192.168.112.26	52.15.211.32	MQTT	70 d8:9e:f3:...	Publish Release (id=3)
2079	7.544286679	192.168.112.26	52.15.211.32	MQTT	70 90:e2:ba:...	Publish Complete (id=3)
2122	8.364020975	192.168.112.26	52.15.211.32	MQTT	91 d8:9e:f3:...	Publish Message (id=4) [counter/slow/q2]
2123	8.364245514	192.168.112.26	52.15.211.32	MQTT	70 90:e2:ba:...	Publish Received (id=4)

Acknowledgment number: 39 (relative ack number)
1000 = Header Length: 32 bytes (8)
Flags: 0x018 (PSH, ACK)
Window size value: 29
[Calculated window size: 29696]
[Window size scaling factor: 1024]
Checksum: 0x381d [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[SEQ/ACK analysis]
[Timestamps]
TCP payload (4 bytes)
[PDU Size: 4]
MQ Telemetry Transport Protocol, Publish Complete
Header Flags: 0x70, Message Type: Publish Complete
0111 = Message Type: Publish Complete (7)
.... 0000 = Reserved: 0
Msg Len: 2
Message Identifier: 1

```

00  90 e2 ba 29 d1 cc d8 9e f3 73 91 34 08 00 45 00  ...  s 4  E
10  00 38 4c 68 40 00 40 06 b6 65 c0 a8 70 1a 34 0f  8Lh@  e  p 4
20  d3 20 e3 f5 07 5b 2f b3 26 1e 8f 95 aa c6 80 18  ...  /  & .....
30  00 1d 38 1d 00 00 01 01 08 0a 74 a2 38 a3 15 d3  8  t 8
40  f4 8b 70 02 00 01  ...  p

```

PUBCOMP packet for QoS 2

#4 a) with just as much only 4 GB of RAM And a 3 GHz intel dual core processor could handle upto 60k clients running mosquito however when it is required to maintain persistent messages for about a million client then the CPU and memory power could be held more accountable.

Often the case the maximum allowed memory could be used up entirely by the client leaving the CPU in an unsteady state and also if the server is single threaded then could only utilize half the CPU power.

When networks grow to join billions of devices centralized systems will turn into a bottleneck. Such systems will require huge investments and spending in maintaining cloud servers that can handle such large amounts of information exchange, and entire systems can go down if the server becomes unavailable.

With scalability network security can also impose a challenge but with a well-established Internet communication protocols armed with modern cryptography algorithms make it virtually impossible for hackers to decipher data in transmission.

b) In a situation where high reliability of messages are not required and a stable network can be guaranteed QoS 0 can be proved the hugely beneficial. Likewise, if the overhead in communication is acceptable at the cost of power for a guaranteed delivery then QoS 2. Its highly likely that any of the quality of services could be proved advantageous depending upon the need and availability of the resources at hand which one of them is affordable to be traded for another.

#2c) Following the observation of \$SYS topics mainly these:

\$SYS/broker/load/publish/received/+

\$SYS/broker/load/publish/sent/+

\$SYS/broker/load/publish/dropped/+

A strong sense of correlation could be built with the values that I had arrived with counting the number of messages received and lost. However, the SYS topics were subscribed soon before subscribing the counter topics it's assumed that the deviation couldn't have been very since the number of active clients over all the time were constant

It made more sense to see that the rate of messages captured fluctuating with the change in number of messages received and published by the broker.

\$SYS/broker/heap/current size

\$SYS/broker/heap/maximum size

By observing these topics , it was noted that at a particular interval the broker wasn't completely utilizing its maximum allocated heap memory it was only about 25% at a time.