Homework 3 - PDF response file

Internet of Things
Politecnico di Milano

Sara Acquaviva (943806), Davide Savoldelli (928676)

May 17, 2020

1 CoAP

1.1 What's the difference between the message with MID: 3978 and the one with MID: 22636?

The packet with message id 3978 is "Confirmable", while the one with message id 22636 is "Non Confirmable". This means that the former packet requires an acknowledgement for its reception, while the latter does not. In fact an ACK for the packet with mid 3978 is provided (the packet No. 6702).

1.2 Does the client receive the response of message No. 6949?

Yes, the packet number 6949 is of the type "Confirmable", hence we can know if it has been received or not. As an ACK for the same MID exists (the packet No. 6953), the 6949 got an answer.

```
frame.number == 6949
coap.mid == 28357
```

Actually, the status of the ACK is "Method not Allowed". Hence, the GET does not obatain a proper response, just an ACK.

1.3 How many replies of type Confirmable and result code "Content" are received by the server "localhost"?

There are eight packets that satisfy the filter.

```
coap.code == 69 && ip.dst == 127.0.0.1
```

2 MQTT

2.1 How many messages containing the topic "factory/department*/+" are published by a client with user name: "jane"? (where * replaces the dep. number, e.g. factory/department1/+, factory/department2/+ and so on.)

First, we filter on the message sent by the username "jane".

```
mqtt.username == "jane"
```

The are four messages of type "Connect", which connect to localhost from the TCP ports number 42821, 40989, 40005 and 50985. We use those ports to handle the connection to the username, as it appears only in Connect type messages. Filtering the "Publish" messages (type 0x11) which connect from the same ports and whose topic matches the regex "factory/department*/, we find six packets:

```
mqtt.topic matches "factory/department[0-9]/" &&
mqtt.msgtype == 3 && ip.src == 127.0.0.1 &&
(tcp.srcport == 42821 || tcp.srcport == 40989 || tcp.srcport
   == 40005 || tcp.srcport == 50985)
```

Being the packets of type "Published", obviously they cannot contain a topic with wildcards: these can be used only by subscribers. Unfortunately, none of the packet can be filtered by an eventual subscriber using the "+" wildcard after the department* level in the topic, because all of them contain an extra sublevel. To filter the packets a subscriber should specify a topic made like this: "factory/department*/+/+", being the "+" wildcard single level only.

2.2 How many clients connected to the broker “hivemq” have specified a will message?

First, we have to understand what IP address is related to the ”hivemq” name. To do so, we have a lookup on the DNS received packets.

```
dns.qry.name contains "hivemq" && dns.flags.response && dns.a
```

After that, we can filter the packets sent to that IP address that match the former predicate.

```
mqtt.conflag.willflag == 1 && (ip.dst == 3.120.68.56 ||
   ip.dst == 18.185.199.22)
```

That said, there are 16 matching packets. Among them there are two packets with the same clientId but different TCP source ports.

2.3 How many publishes with QoS 1 don’t receive the ACK?

We can filter all the publishes (129 packets) with:

```
mqtt.qos == 1 && mqtt.msgtype == 3
```

Then, we filter only the Acknowledgements (74 packets) with:

```
mqtt.msgtype == 4
```

Substracting the latter result from the former we obtain 55 packets without ACK. Among the Publishes, we want to select only the publisher-broker packets (77):

```
mqtt.qos == 1 && mqtt.msgtype == 3 && tcp.dstport == 1883
```

Finally, we can conclude that 3 (77 - 74) of the 55 packets which unreceived ACK are from publisher-broker relationship, while the other 52 are from the broker-subscriber one.

2.4 How many last will messages with QoS set to 0 are actually delivered?

First, filtering the packets which bring a Last Will Message, we notice that it starts with the keyword "error".

```
mqtt.conflag.willflag == 1
```

To know which Last Will Messages are actually delivered, we just look for the delivered messages with QoS 1 starting with "error".

```
mqtt.msg contains "error" && mqtt.qos == 1
```

There are 4 packets which satisfy the condition.

2.5 Are all the messages with QoS greater than 0 published by the client "4m3DWYzWr40pce6OaBQAfk" correctly delivered to the subscribers?

We must firstly find all the messages sent by "4m3DWYzWr40pce6OaBQAfk" with QoS greater than 0. Filtering the clientId:

```
mqtt.clientid == "4m3DWYzWr40pce6OaBQAfk"
```

And looking at his socket, we filter his messages by QoS:

```
mqtt.msgtype == 3 && ip.src == 10.0.2.15 && tcp.srcport ==
   58313 && mqtt.qos > 0
```

For everyone of them (there is only one message with QoS 2) we may check the reception of the PUBREC:

```
mqtt.msgtype == 5&& ip.dst == 10.0.2.15 && tcp.dstport==58313
```

And then we check if the PUBREL has been sent back:

```
mqtt.msgtype == 6&& ip.src == 10.0.2.15 && tcp.srcport ==
   58313
```

We find that there are no result out of the filter. Sticking to the schema QoS 2 (publisher-broker) + QoS 2 (broker-subscriber), we notice that the broker

cannot proceed until the PUBREL has been received. Hence, we can infer that the message has not been sent to the subscriber(s). In every case, the process is incomplete.

2.6 What is the average message length of a connect msg using mqttv5 protocol? Why messages have different size?

There are 63 packets of type "Connect" and MQTT version 5:

```
mqtt.msgtype == 1 && mqtt.ver == 5
```

The average size of the packets is 91 bytes. They have different size because of the optional fields that can be left empty. Moreover the MQTTv5 specifications explain that Metadata in the header of the message are available and they can be used to specify user defined properties, which are, obviously, of different dimensions.

2.7 Why aren't there any REQ/RESP pings in the pcap?

The absence of the pings in the pcap is motivated by the fact that they are not required if time specified in the Keep-Alive field (similar to HTTP's) of Connection packets is greater than the time required for the recipient(s) to answer.