

Internet stvari

2. laboratorijska vježba

Ema Popović

0036506085

CoAP

1. Korištene su datoteke MyCoapServer.java i MyCoapClient1.java.

a) Pri korištenju poruke CON za zahtjev resursa izmjene se dvije poruke, CON i ACK. Klijent šalje GET zahtjev za resursom i dobiva odgovor ACK od poslužitelja (unutar kojeg se javlja da resurs nije pronađen). MID je jednak u obje poruke, kao i TKN.

b) Pri korištenju NON izmjene se dvije poruke, zahtjev i odgovor. Tip poruke je non-confirmable i nema ACK potvrda primitka. Poruka koja se vraća od poslužitelja je također tipa NON i javlja da resurs nije pronađen. TKN je isti u obje poruke, a MID-ovi su različiti.

c) Na slici 1.1. je prikazan isječak iz alata Wireshark i 4 poruke koje se spominju u zadacima a) i b). U sadržaju ACK poruke pod rednim brojem 62 vidljivo je da resurs nije pronađen. U sadržaju NON poruke pod brojem 64 nalazi se isti „Not Found“ sadržaj.

61	6.301426	127.0.0.1	127.0.0.1	CoAP	57 CON, MID:54304, GET, TKN:50 88 37 05 9f 00 bf 80, /sensors/temp
62	6.347763	127.0.0.1	127.0.0.1	CoAP	44 ACK, MID:54304, 4.04 Not Found, TKN:50 88 37 05 9f 00 bf 80, /sensors/temp
63	6.364104	127.0.0.1	127.0.0.1	CoAP	57 NON, MID:54305, GET, TKN:9c 3c 27 2d 7d 20 10 b3, /sensors/temp
64	6.365881	127.0.0.1	127.0.0.1	CoAP	44 NON, MID:63954, 4.04 Not Found, TKN:9c 3c 27 2d 7d 20 10 b3, /sensors/temp

>	Frame 62: 44 bytes on wire (352 bits), 44 bytes captured (352 bits) on interface \Device\NPF_{...}, id 0
>	Null/Loopback
>	Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
>	User Datagram Protocol, Src Port: 5683, Dst Port: 52735
▼	Constrained Application Protocol, Acknowledgement, 4.04 Not Found, MID:54304
	01.. = Version: 1
	..10 = Type: Acknowledgement (2)
 1000 = Token Length: 8
	Code: 4.04 Not Found (132)

Slika 1.1. Dohvaćanje nepostojećeg resursa u Wiresharku

2. Korištene su datoteke SensorsResource.java, MyCoapClient2.java i MyCoapServer.java.

a) `server.add(new SensorsResource());`

b) URI = putanja + ime. Putanja je kombinacija svih predaka resursa do odvojenih s „/“ i ne uključuje ime. URI je spoj putanje i imena i označava identifikator resursa.

- Sensors
 - Ime: sensors
 - Putanja: /
 - URI: /sensors

- Temp
 - Ime: temp
 - Putanja: /sensors/
 - URI: /sensors/temp

c) Činjenica da je resurs stvoren javlja se kodom 2.01 – CREATED.

d) Nedozvoljena metoda javlja se kodom 4.05 – METHOD NOT ALLOWED.

```
@Override
public void handleGET(CoapExchange exchange) {
    exchange.respond(METHOD_NOT_ALLOWED);
}
```

e) Na slici 1.2. je isječak iz alata Wireshark. Klijent šalje post metodu s payloadom „test“. Kako se koristi tip CON klijent dobiva natrag potvrdu ACK kojom ga se obavještava o stvaranju resursa temp i koristi se kod CREATED. Klijent šalje GET metodu, ali zbog koda iz d) podzadatka u ACK povratnoj poruci dobiva kod METHOD NOT ALLOWED.

47 9.288523	127.0.0.1	127.0.0.1	CoAP	58 CON, MID:63977, POST (text/plain), TKN:ac 03 c3 e5 0c 95 98 6c, /sensors
48 9.328162	127.0.0.1	127.0.0.1	CoAP	50 ACK, MID:63977, 2.01 Created (text/plain), TKN:ac 03 c3 e5 0c 95 98 6c, /sensors
49 9.341980	127.0.0.1	127.0.0.1	CoAP	52 CON, MID:63978, GET, TKN:48 77 0f 58 c8 34 b6 63, /sensors
50 9.342837	127.0.0.1	127.0.0.1	CoAP	44 ACK, MID:63978, 4.05 Method Not Allowed, TKN:48 77 0f 58 c8 34 b6 63, /sensors

Slika 1.2. Slanje POST i nedopuštene GET metode u Wiresharku

3. Korištene su datoteke MyCoapServer.java, TempResource.java, MyCoapClient2.java i MyCoapClient3.java.

a) Za pretplaćivanje se koristi opcija observe (Observe:0). Resurs postavlja

```
setObservable(true);
setObserveType(Type.CON);
getAttributes().setObservable();
```

što omogućava pretplate koje će se aktualizirati putem CON poruka. Pretplatnicima se pomoću `changed()` javlja promjena resursa. Klijent koristi metodu `client.observe()` unutar koje definira kako postupa kada primi notifikaciju na koju je pretplaćen i kako postupa u slučaju greške.

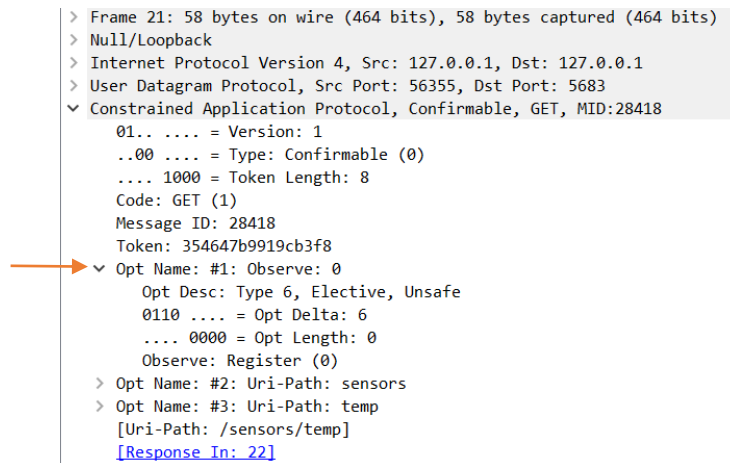
b) MID je različit za svaki par poruka CON – ACK. Klijent prima CON poruku s novom vrijednosti temperature i šalje potvrdu ACK s istim MID-om. TKN je jednak u svim CON i ACK porukama koje prenose vrijednost resursa na kojeg je klijent pretplaćen. Zahtjev za stvaranjem resursa na početku i pripadajući ACK imaju svoj TKN. Nedozvoljeni GET zahtjev iz 1.d) i pripadajući ACK također imaju jednak TKN.

c) Za prekid pretplate šalje se GET poruka Observe:1.

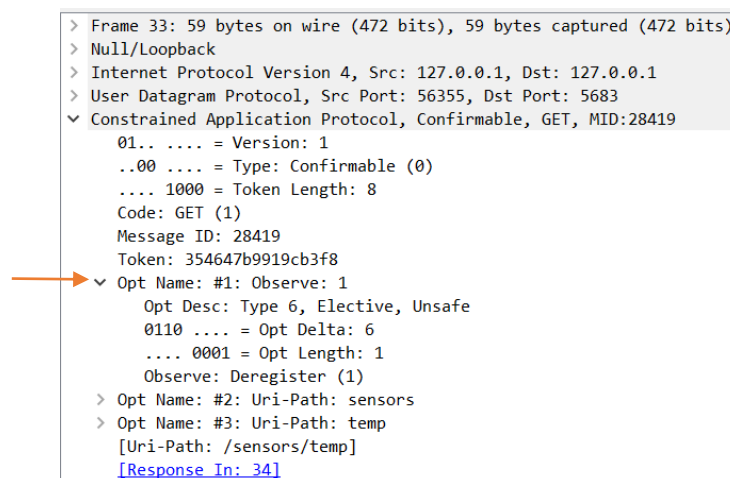
d) Slika 1.3. prikazuje isječak iz alata Wireshark gdje je cijela komunikacija. Slika 1.4. prikazuje sadržaj paketa 21, u kojem se klijent pretplati. Označena je opcija Observe:0. Slika 1.5. prikazuje sadržaj paketa 33, u kojem klijent otkaže pretplatu. Označena je opcija Observe:1. Ispod ovih opcija naveden je i URI resursa na koji se pretplata odnosi.

17	9.041552	127.0.0.1	127.0.0.1	CoAP	58 CON, MID:9489, POST (text/plain), TKN:f0 cb ee 27 10 be 0a b9, /sensors
18	9.107897	127.0.0.1	127.0.0.1	CoAP	50 ACK, MID:9489, 2.01 Created (text/plain), TKN:f0 cb ee 27 10 be 0a b9, /sensors
19	9.117849	127.0.0.1	127.0.0.1	CoAP	52 CON, MID:9490, GET, TKN:58 2e 79 18 cf 5e 2a fb, /sensors
20	9.118731	127.0.0.1	127.0.0.1	CoAP	44 ACK, MID:9490, 4.05 Method Not Allowed, TKN:58 2e 79 18 cf 5e 2a fb, /sensors
21	19.328054	127.0.0.1	127.0.0.1	CoAP	58 CON, MID:28418, GET, TKN:35 46 47 b9 91 9c b3 f8, /sensors/temp
22	19.336707	127.0.0.1	127.0.0.1	CoAP	56 ACK, MID:28418, 2.05 Content (text/plain), TKN:35 46 47 b9 91 9c b3 f8, /sensors/temp
23	20.193283	127.0.0.1	127.0.0.1	CoAP	55 CON, MID:11484, 2.05 Content (text/plain), TKN:35 46 47 b9 91 9c b3 f8, /sensors/temp
24	20.199181	127.0.0.1	127.0.0.1	CoAP	36 ACK, MID:11484, Empty Message
25	21.191974	127.0.0.1	127.0.0.1	CoAP	55 CON, MID:11485, 2.05 Content (text/plain), TKN:35 46 47 b9 91 9c b3 f8, /sensors/temp
26	21.193840	127.0.0.1	127.0.0.1	CoAP	36 ACK, MID:11485, Empty Message
27	22.194306	127.0.0.1	127.0.0.1	CoAP	55 CON, MID:11486, 2.05 Content (text/plain), TKN:35 46 47 b9 91 9c b3 f8, /sensors/temp
28	22.195891	127.0.0.1	127.0.0.1	CoAP	36 ACK, MID:11486, Empty Message
29	23.203623	127.0.0.1	127.0.0.1	CoAP	55 CON, MID:11487, 2.05 Content (text/plain), TKN:35 46 47 b9 91 9c b3 f8, /sensors/temp
30	23.205925	127.0.0.1	127.0.0.1	CoAP	36 ACK, MID:11487, Empty Message
31	24.216399	127.0.0.1	127.0.0.1	CoAP	55 CON, MID:11488, 2.05 Content (text/plain), TKN:35 46 47 b9 91 9c b3 f8, /sensors/temp
32	24.219380	127.0.0.1	127.0.0.1	CoAP	36 ACK, MID:11488, Empty Message
33	24.778007	127.0.0.1	127.0.0.1	CoAP	59 CON, MID:28419, GET, TKN:35 46 47 b9 91 9c b3 f8, /sensors/temp
34	24.781322	127.0.0.1	127.0.0.1	CoAP	53 ACK, MID:28419, 2.05 Content (text/plain), TKN:35 46 47 b9 91 9c b3 f8, /sensors/temp

Slika 1.3. Pretplata na resurs u alatu Wireshark



Slika 1.4. Sadržaj CoAP paketa za početak pretplate



Slika 1.5. Sadržaj CoAP paketa za prestanak pretplate

MQTT

1. Korištena je datoteka MyMqttClientZad1.java.

a) MQTT broker se definira kroz URI, u primjeru kroz liniju `String broker = "tcp://127.0.0.1:1883";`. Navodi se korištenje protokola TCP/IP, adresa i port. Pri uključivanju Mosquitto brokera navedeno je da se koriste default postavke i port 1883.

b) Parametri poruke koji se mogu postaviti su tema (na koju se korisnici mogu pretplatiti), sadržaj poruke, kvaliteta usluge (QoS) i identifikacijska oznaka klijenta koji objavljuje poruku.

c) Kvaliteta usluge se prije objavljivanja definira linijom `message.setQos(qos);`. Varijabla `qos` predstavlja broj u intervalu [0, 2]. Za svaku objavu pojedinačno se odabire kvaliteta usluge.

d) Prvo je isprobano slanje s kvalitetom usluge 0, vidljivo na slici 2.1. Slike su isječki poruka pročitanih na Mosquitto brokeru. U ovom slučaju koristi se samo PUBLISH i to najviše jednom.

```
1618734659: Received PUBLISH from 0036506085 (d0, q0, r0, m0, 'mqtt-spy/temp', ... (5 bytes))
1618734659: Received PUBLISH from 0036506085 (d0, q0, r0, m0, 'mqtt-spy/hum', ... (2 bytes))
```

Slika 2.1. Slanje informacija, QoS = 0

Iduća kvaliteta usluge bila je 1, komunikacija je prikazana slikom 2.2. Koriste se PUBLISH i PUBACK (potvrda). U ovoj razini kvalitete usluge objava se može ponovno poslati ako klijent ne dobije PUBACK, ali u primjeru to nije bilo potrebno.

```
1618734666: Received PUBLISH from 0036506085 (d0, q1, r0, m1, 'mqtt-spy/temp', ... (5 bytes))
1618734666: Sending PUBACK to 0036506085 (m1, rc0)
1618734666: Received PUBLISH from 0036506085 (d0, q1, r0, m2, 'mqtt-spy/hum', ... (2 bytes))
1618734666: Sending PUBACK to 0036506085 (m2, rc0)
```

Slika 2.2. Slanje informacija, QoS = 1

U zadnjem primjeru promatra se kvaliteta usluge razine 2, prikazana na slici 2.3. Ova razina uključuje isporuku objave točno jedanput i zahtijeva potvrdu u više poruka. Za svaku informaciju potrebni su PUBLISH, PUBREC, PUBREL i PUBCOMP.

```
1618734673: Received PUBLISH from 0036506085 (d0, q2, r0, m1, 'mqtt-spy/temp', ... (5 bytes))
1618734673: Sending PUBREC to 0036506085 (m1, rc0)
1618734673: Received PUBREL from 0036506085 (Mid: 1)
1618734673: Sending PUBCOMP to 0036506085 (m1)
1618734673: Received PUBLISH from 0036506085 (d0, q2, r0, m2, 'mqtt-spy/hum', ... (2 bytes))
1618734673: Sending PUBREC to 0036506085 (m2, rc0)
1618734673: Received PUBREL from 0036506085 (Mid: 2)
1618734673: Sending PUBCOMP to 0036506085 (m2)
```

Slika 2.3. Slanje informacija, QoS = 2

2. U alatu mqtt-spy stvorene su 3 pretplate: mqtt-spy/#, mqtt-spy/+ i mqtt-spy/temp. Poslane su 4 objave s različitim rezultatima.

Objava kojoj je tema „mqtt-spy“ prikazala se kod preplate mqtt-spy/# i mqtt-spy/+. mqtt-spy/temp nije ju primio jer u temi nije naznačeno temp.

Objavu s temom „mqtt-spy/temp“ primile su sve 3 pretplate.

Objavu s temom „mqtt-spy/temp/today“ primila je samo pretplata mqtt-spy/#. mqtt-spy/+ prima objave koje su maksimalno jedan stupanj ispod mqtt-spy, a ova objave je 2 stupnja ispod. S druge strane # označava prihvatanje svih stupnjeva hijerarhije. Kod pretplate mqtt-spy/temp je označeno da prihvaća objave samo iz tog stupnja u hijerarhiji pa također ne prima poruku.

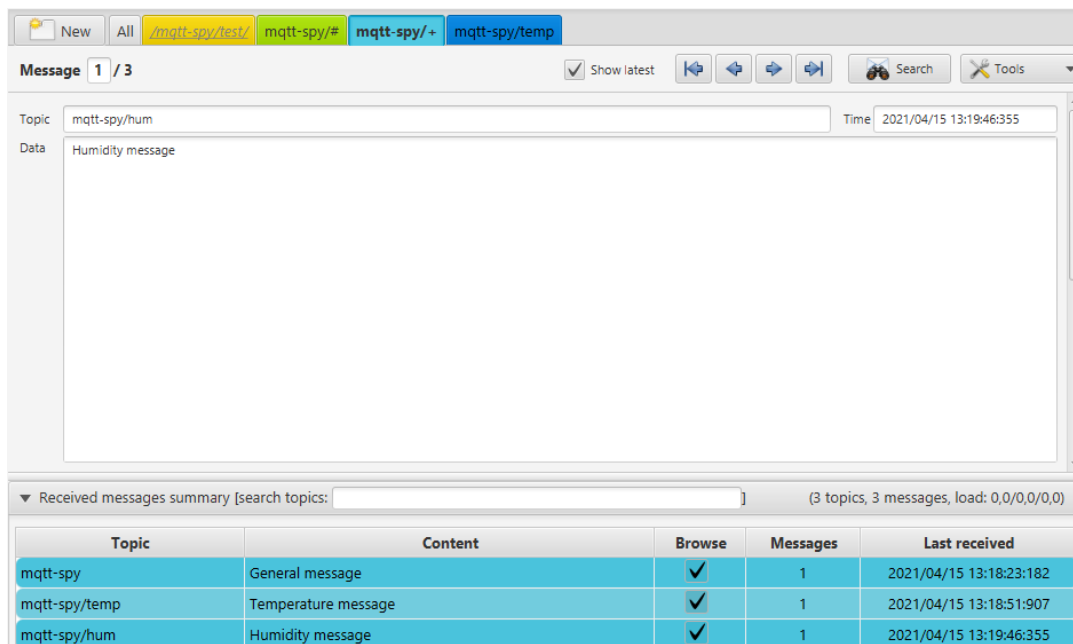
Objave s temom „mqtt-spy/hum“ primaju pretplate mqtt-spy/# i mqtt-spy/+ kako je tema jedan stupanj niže u hijerarhiji od mqtt-spy. mqtt-spy/temp ne prima ovu objavu jer je označena s hum, a ne temp.

U slikama 2.4. – 2.6. prikazane su objave dobivene za svaku pretplatu.

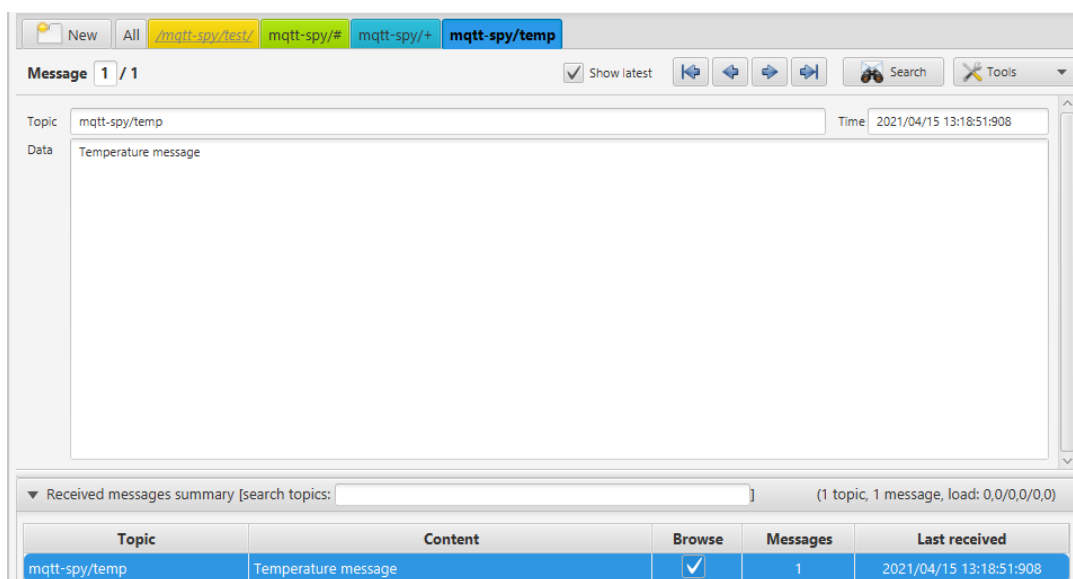
The screenshot shows the MQTT Spy application interface. At the top, there are tabs for different topics: `/mqtt-spy/test`, `mqtt-spy/#`, `mqtt-spy/+`, and `mqtt-spy/temp`. The `mqtt-spy/#` tab is selected. Below the tabs, the 'Message' pane shows the details of the selected message. The topic is `mqtt-spy/hum` and the data is 'Humidity message'. The time is 2021/04/15 13:19:46:355. At the bottom, there is a 'Received messages summary' table with the following data:

Topic	Content	Browse	Messages	Last received
mqtt-spy	General message	✓	1	2021/04/15 13:18:23:179
mqtt-spy/temp	Temperature message	✓	1	2021/04/15 13:18:51:906
mqtt-spy/temp/today	Temperature today message	✓	1	2021/04/15 13:19:23:852
mqtt-spy/hum	Humidity message	✓	1	2021/04/15 13:19:46:355

Slika 2.4. Primljene objave – pretplata mqtt-spy/#



Slika 2.5. Primljene objave – pretplata mqtt-spy/+



Slika 2.6. Primljene objave – pretplata mqtt-spy/temp

3. Korištene su datoteke MyMqttClientZad1.java i MyMqttClientZad3.java.

a) Za različite kvalitete usluge dolazi različit broj poruka. U svakom isprobanom slučaju poruke su došle do odredišta. U zadatku 1 prikazano je kako izgledaju poruke za različite kvalitete usluge objavljiivača. Ovdje je opisan utjecaj kvalitete i pretplatnika i objavljiivača na slanje jedne objave.

- QoS pretplatnika = 2
 - QoS objavljiivača = 0: 2 poruke – PUBLISH od objavljiivača do brokera, PUBLISH od brokera do pretplatnika
 - QoS objavljiivača = 1: 3 poruke - PUBLISH od objavljiivača do brokera, PUBLISH od brokera do pretplatnika, PUBACK potvrda od brokera do objavljiivača
 - QoS objavljiivača = 2: 7 poruka - PUBLISH od objavljiivača do brokera, PUBREC potvrda o primitku od brokera do objavljiivača, PUBREL potvrda za otpuštanje poruke od objavljiivača do brokera, PUBLISH od brokera do pretplatnika, PUBCOMP potvrda o uspješnom slanju od brokera do objavljiivača, PUBREC potvrda o primitku od pretplatnika do brokera, PUBREL potvrda za otpuštanje poruke od brokera do pretplatnika
- QoS pretplatnika = 1
 - QoS objavljiivača 0 i 1: isti broj poruka
 - Qos Objavljiivača 2: 5 poruka - PUBLISH od objavljiivača do brokera, PUBREC potvrda o primitku od brokera do objavljiivača, PUBREL potvrda za otpuštanje poruke od objavljiivača do brokera, PUBLISH od brokera do pretplatnika, PUBCOMP potvrda o uspješnom slanju od brokera do objavljiivača
- QoS pretplatnika = 0
 - Rezultati su jednaki kao i u slučaju kad je QoS pretplatnika = 1