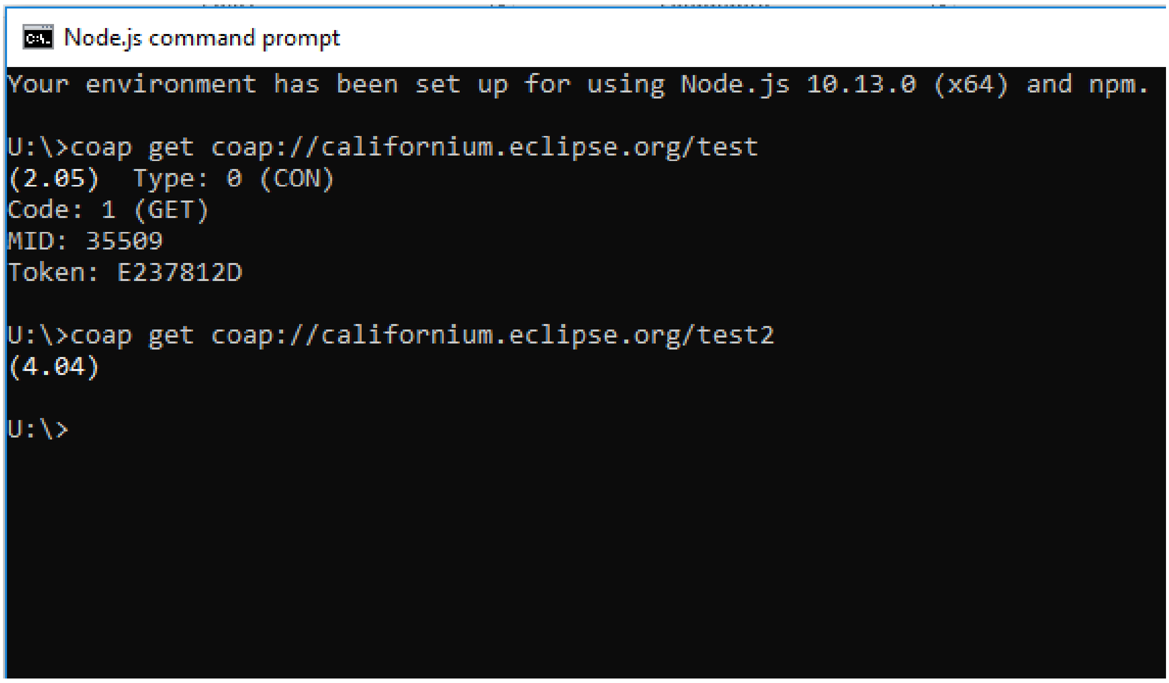
IOT Prototypying Lab2 - 16 Feb 2019

Sahar Hosseini

Quoc Trung Pham

Host machine

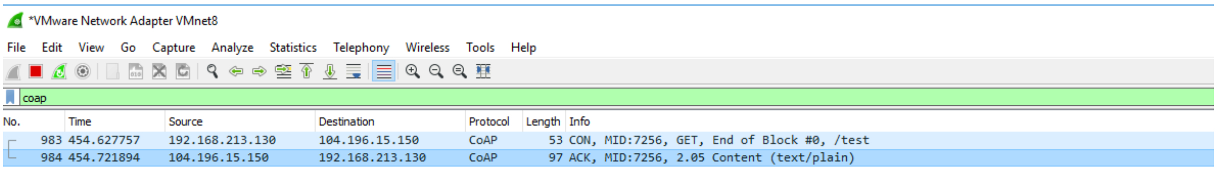
After instalation coap protocol on, We try to test different different message content from server with get method first test the exist path on the server coap://californium.eclipse.org/test so we recive the Token and MID with method GET, then we try to test path not exist in the server we get 4.04 error message means not found.



```
Your environment has been set up for using Node.js 10.13.0 (x64) and npm.

U:\>coap get coap://californium.eclipse.org/test
(2.05)  Type: 0 (CON)
Code: 1 (GET)
MID: 35509
Token: E237812D

U:\>coap get coap://californium.eclipse.org/test2
(4.04)

U:\>
```

First step with coap

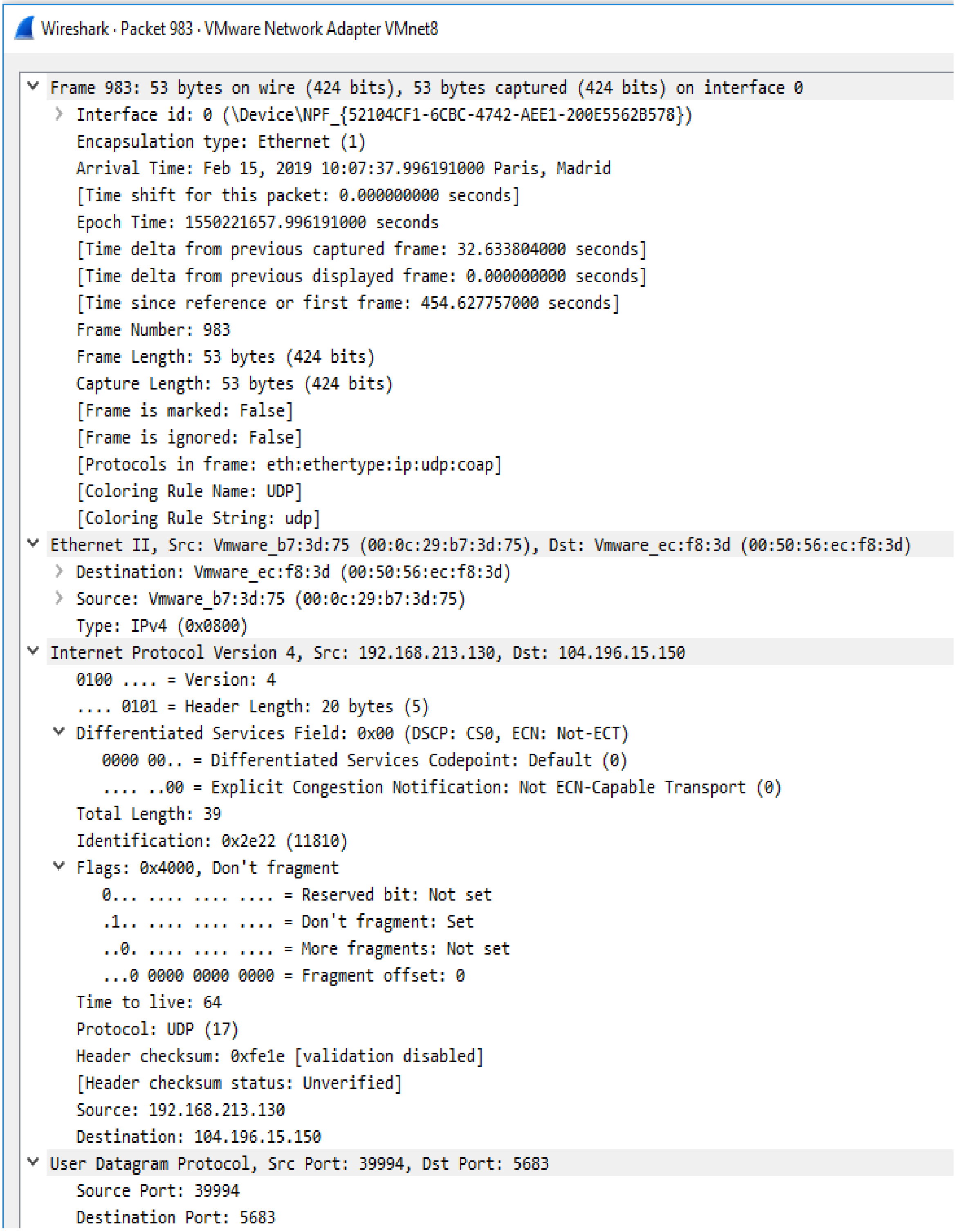
We start capturing VMware machine with wiresharke and run script file name ”lab2.js” to check the path. Below figure shows the connection request to the server and receive acknowledgment message to establish the connation with coap protocol from wiresharke.



coap

No.	Time	Source	Destination	Protocol	Length	Info
983	454.627757	192.168.213.130	104.196.15.150	CoAP	53	CON, MID:7256, GET, End of Block #0, /test
984	454.721894	104.196.15.150	192.168.213.130	CoAP	97	ACK, MID:7256, 2.05 Content (text/plain)

Details of first message to request establish the Connection

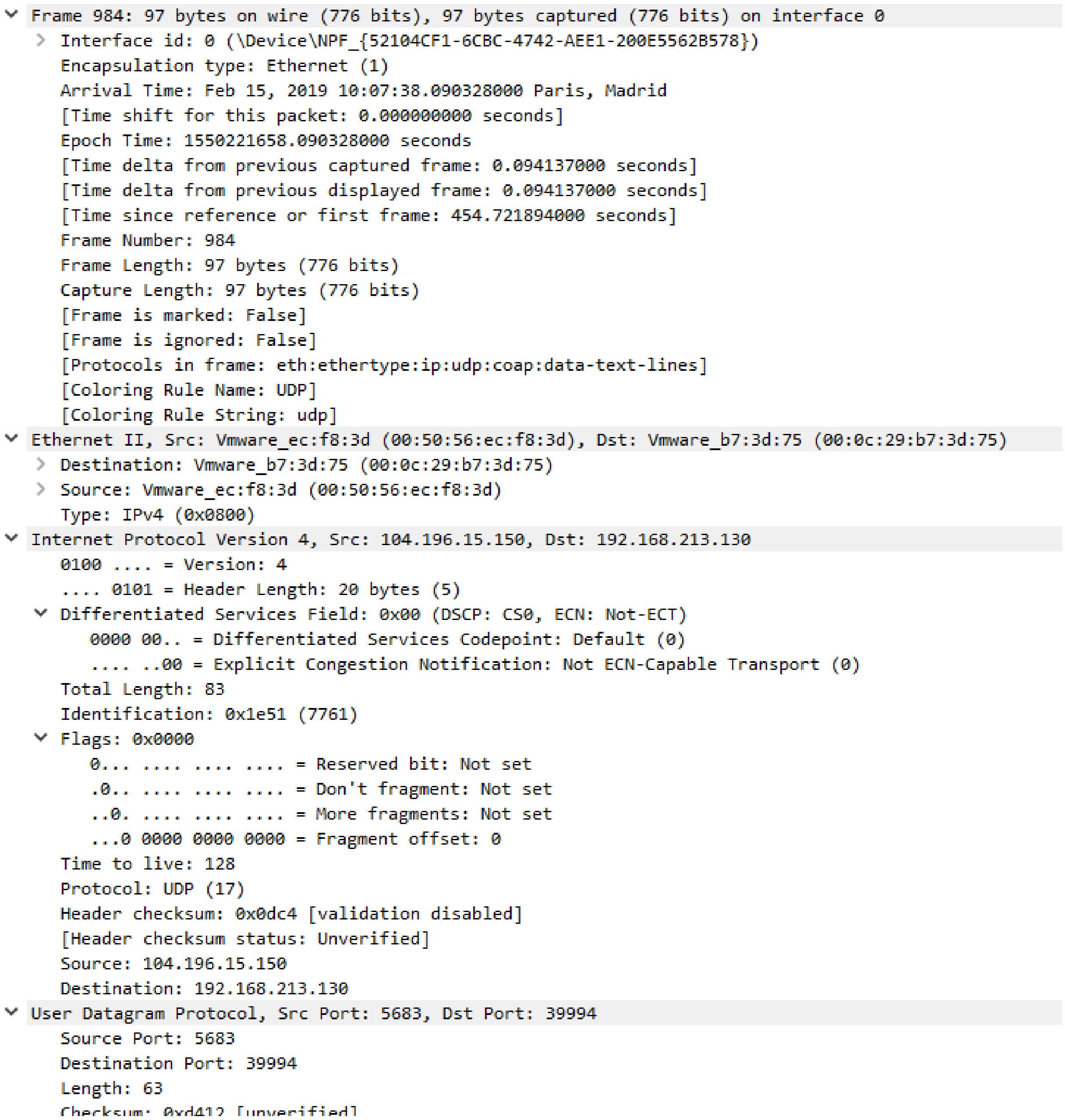


Wireshark · Packet 983 · VMware Network Adapter VMnet8

```
v Frame 983: 53 bytes on wire (424 bits), 53 bytes captured (424 bits) on interface 0
  > Interface id: 0 (\Device\NPF_{52104CF1-6CBC-4742-AEE1-200E5562B578})
    Encapsulation type: Ethernet (1)
    Arrival Time: Feb 15, 2019 10:07:37.996191000 Paris, Madrid
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1550221657.996191000 seconds
    [Time delta from previous captured frame: 32.633804000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 454.627757000 seconds]
    Frame Number: 983
    Frame Length: 53 bytes (424 bits)
    Capture Length: 53 bytes (424 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:udp:coap]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
v Ethernet II, Src: Vmware_b7:3d:75 (00:0c:29:b7:3d:75), Dst: Vmware_ec:f8:3d (00:50:56:ec:f8:3d)
  > Destination: Vmware_ec:f8:3d (00:50:56:ec:f8:3d)
  > Source: Vmware_b7:3d:75 (00:0c:29:b7:3d:75)
    Type: IPv4 (0x0800)
v Internet Protocol Version 4, Src: 192.168.213.130, Dst: 104.196.15.150
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  v Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      0000 00.. = Differentiated Services Codepoint: Default (0)
      .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    Total Length: 39
    Identification: 0x2e22 (11810)
  v Flags: 0x4000, Don't fragment
      0... .... .... .... = Reserved bit: Not set
      .1.. .... .... .... = Don't fragment: Set
      ..0. .... .... .... = More fragments: Not set
      ...0 0000 0000 0000 = Fragment offset: 0
    Time to live: 64
    Protocol: UDP (17)
    Header checksum: 0xfe1e [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.213.130
    Destination: 104.196.15.150
v User Datagram Protocol, Src Port: 39994, Dst Port: 5683
    Source Port: 39994
    Destination Port: 5683
```

Informations like frame numebr ,
Protocol name,
Source and destination ID
Frame length

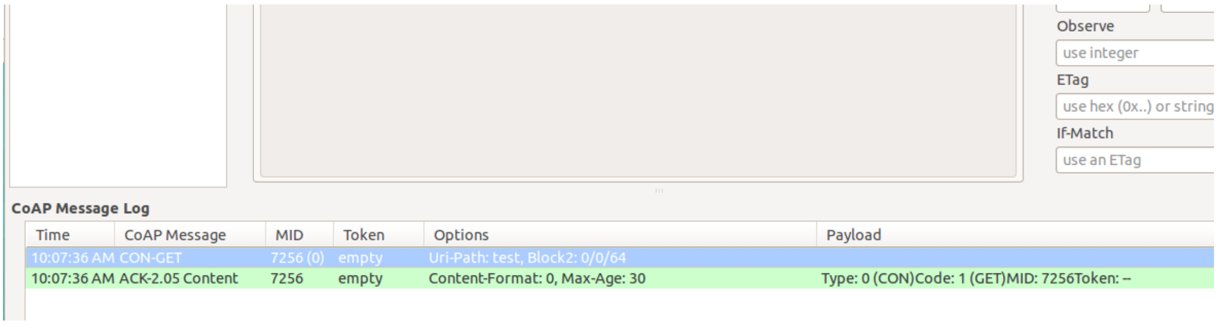
Details of ackknowledgement of the request



```
⌄ Frame 984: 97 bytes on wire (776 bits), 97 bytes captured (776 bits) on interface 0
  › Interface id: 0 (\Device\NPF_{52104CF1-6CBC-4742-AEE1-200E5562B578})
    Encapsulation type: Ethernet (1)
    Arrival Time: Feb 15, 2019 10:07:38.090328000 Paris, Madrid
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1550221658.090328000 seconds
    [Time delta from previous captured frame: 0.094137000 seconds]
    [Time delta from previous displayed frame: 0.094137000 seconds]
    [Time since reference or first frame: 454.721894000 seconds]
    Frame Number: 984
    Frame Length: 97 bytes (776 bits)
    Capture Length: 97 bytes (776 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:udp:coap:data-text-lines]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
⌄ Ethernet II, Src: Vmware_ec:f8:3d (00:50:56:ec:f8:3d), Dst: Vmware_b7:3d:75 (00:0c:29:b7:3d:75)
  › Destination: Vmware_b7:3d:75 (00:0c:29:b7:3d:75)
  › Source: Vmware_ec:f8:3d (00:50:56:ec:f8:3d)
    Type: IPv4 (0x0800)
⌄ Internet Protocol Version 4, Src: 104.196.15.150, Dst: 192.168.213.130
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ⌄ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
       0000 00.. = Differentiated Services Codepoint: Default (0)
       .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    Total Length: 83
    Identification: 0x1e51 (7761)
  ⌄ Flags: 0x0000
       0... .... .... .... = Reserved bit: Not set
       .0.. .... .... .... = Don't fragment: Not set
       ..0. .... .... .... = More fragments: Not set
       ...0 0000 0000 0000 = Fragment offset: 0
    Time to live: 128
    Protocol: UDP (17)
    Header checksum: 0x0dc4 [validation disabled]
    [Header checksum status: Unverified]
    Source: 104.196.15.150
    Destination: 192.168.213.130
⌄ User Datagram Protocol, Src Port: 5683, Dst Port: 39994
    Source Port: 5683
    Destination Port: 39994
    Length: 63
    Checksum: 0xd412 [unverified]
```

Informations like frame numebr ,
Protocol name,
Source and destination ID
Frame length

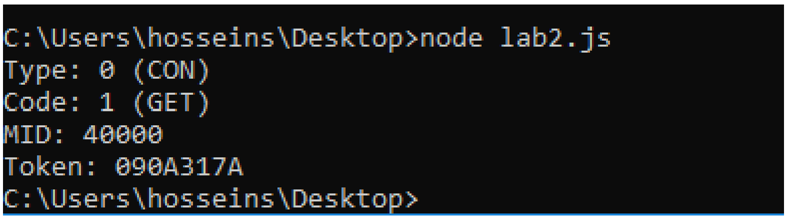
Coap message fist script on vmware



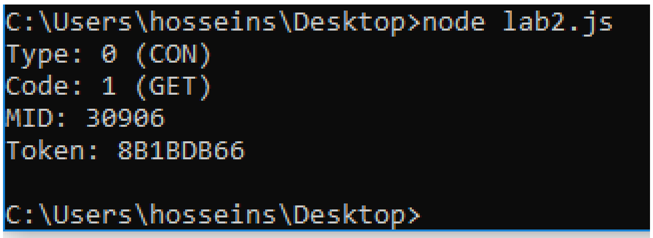
CoAP Message Log

Time	CoAP Message	MID	Token	Options	Payload
10:07:36 AM	CON-GET	7256 (0)	empty	Uri-Path: test, Block2: 0/0/64	
10:07:36 AM	ACK-2.05 Content	7256	empty	Content-Format: 0, Max-Age: 30	Type: 0 (CON)Code: 1 (GET)MID: 7256Token: --

We run the firts script of lab 2 print the response with pipe

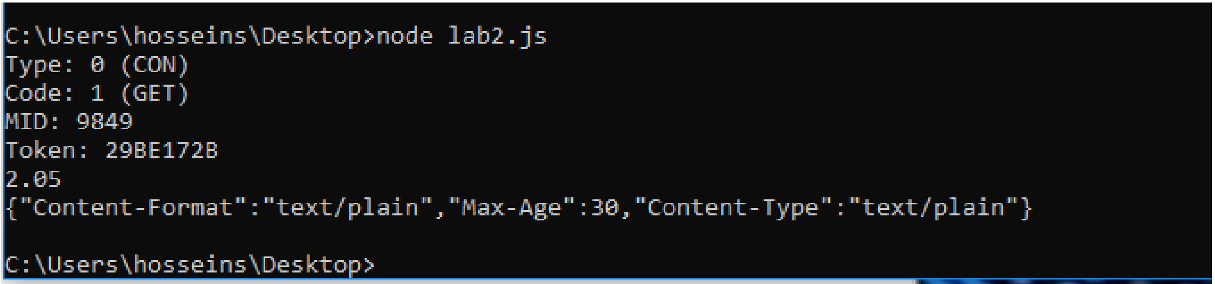

```
C:\Users\hosseins\Desktop>node lab2.js
Type: 0 (CON)
Code: 1 (GET)
MID: 40000
Token: 090A317A
C:\Users\hosseins\Desktop>
```

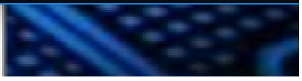
After that we replace pipe with payload


```
C:\Users\hosseins\Desktop>node lab2.js
Type: 0 (CON)
Code: 1 (GET)
MID: 30906
Token: 8B1BDB66

C:\Users\hosseins\Desktop>
```

No different between the messages MID and Token number is different we could use both of command to see the response value from the server.

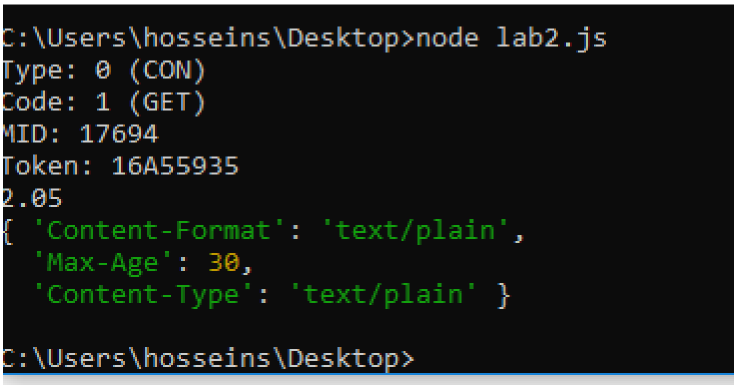
We print out the content code and header information.


```
C:\Users\hosseins\Desktop>node lab2.js
Type: 0 (CON)
Code: 1 (GET)
MID: 9849
Token: 29BE172B
2.05
{"Content-Format":"text/plain","Max-Age":30,"Content-Type":"text/plain"}

C:\Users\hosseins\Desktop>
```



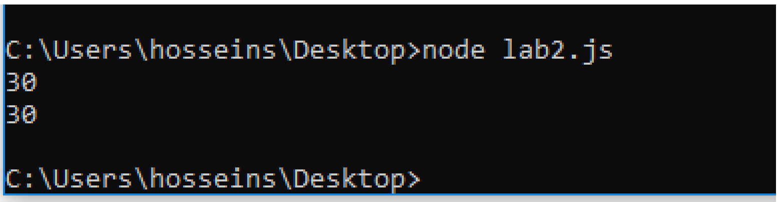
We use toString() to convert information to string sometimes based on need we must convert our information to do some work on them.
Using JSON.stringify() convert json data as a string some times we cannot send json data as a querystring or other situation so we convert data to string and pass them.

without convert we see information like below



```
C:\Users\hosseins\Desktop>node lab2.js
Type: 0 (CON)
Code: 1 (GET)
MID: 17694
Token: 16A55935
2.05
{ 'Content-Format': 'text/plain',
  'Max-Age': 30,
  'Content-Type': 'text/plain' }

C:\Users\hosseins\Desktop>
```

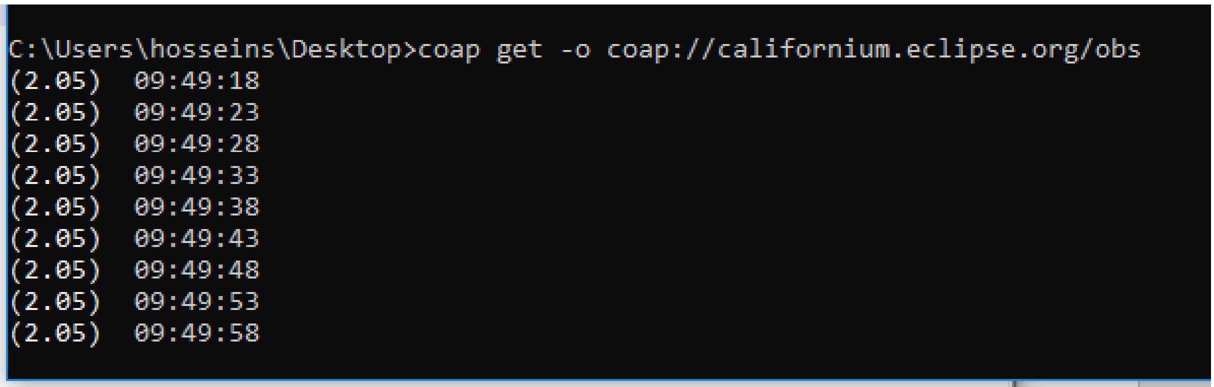
Max age : live duration for request



```
C:\Users\hosseins\Desktop>node lab2.js
30
30

C:\Users\hosseins\Desktop>
```

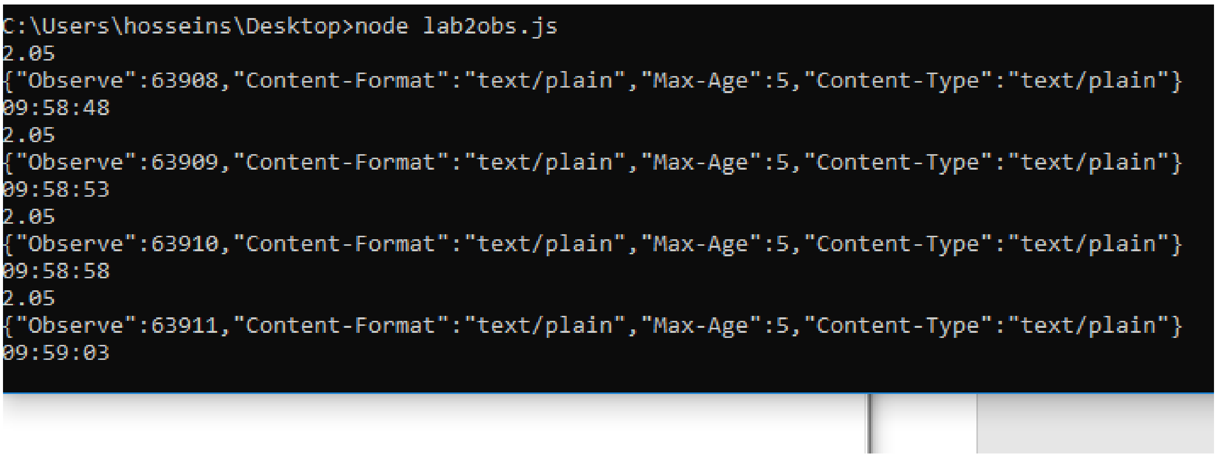
Observation

We write file ‘lab2obs.js’ to use observable option. After running the script we get this response each 5 second.



```
C:\Users\hosseins\Desktop>coap get -o coap://californium.eclipse.org/obs
(2.05)  09:49:18
(2.05)  09:49:23
(2.05)  09:49:28
(2.05)  09:49:33
(2.05)  09:49:38
(2.05)  09:49:43
(2.05)  09:49:48
(2.05)  09:49:53
(2.05)  09:49:58
```

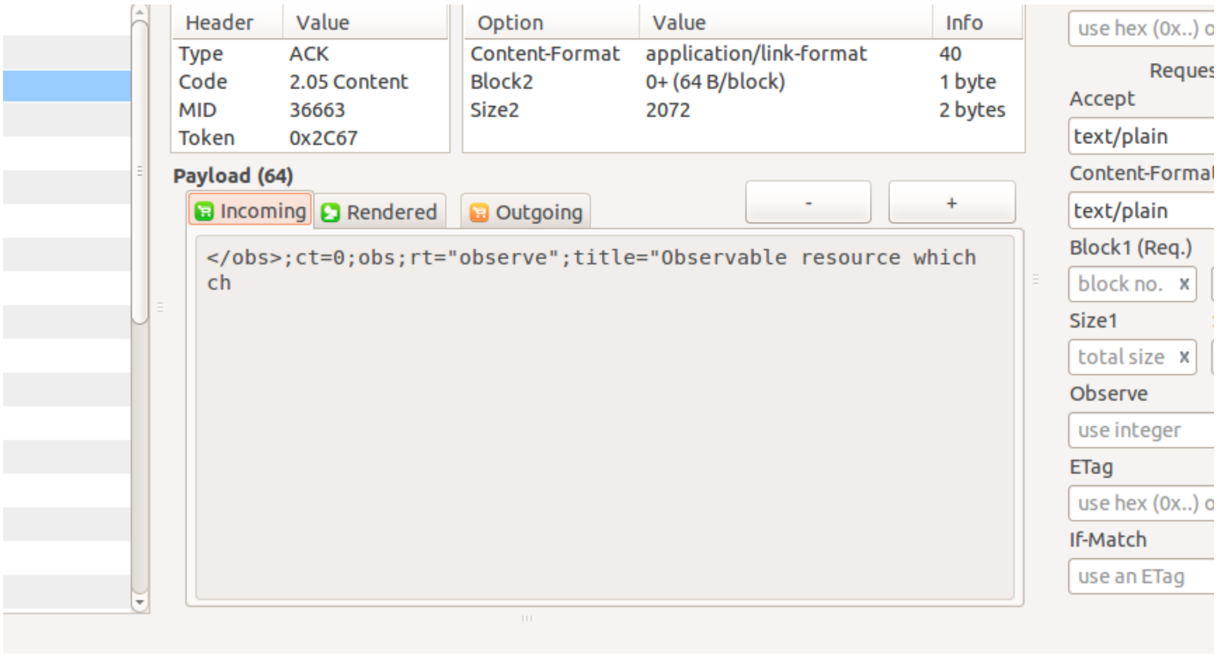
We change the script as described in the question we received the response content message type, header infromation and observe on the screen.



```
C:\Users\hosseins\Desktop>node lab2obs.js
2.05
{"Observe":63908,"Content-Format":"text/plain","Max-Age":5,"Content-Type":"text/plain"}
09:58:48
2.05
{"Observe":63909,"Content-Format":"text/plain","Max-Age":5,"Content-Type":"text/plain"}
09:58:53
2.05
{"Observe":63910,"Content-Format":"text/plain","Max-Age":5,"Content-Type":"text/plain"}
09:58:58
2.05
{"Observe":63911,"Content-Format":"text/plain","Max-Age":5,"Content-Type":"text/plain"}
09:59:03
```

Max age is 5 second so we observe the response every 5 second as seen in the screen
Data = Each new observe message from the server is a new 'data' event.
Remote = we define our request inforamtion as json object this way is easy and clear to use different option on the request.

Discovery



Header	Value
Type	ACK
Code	2.05 Content
MID	36663
Token	0x2C67

Option	Value	Info
Content-Format	application/link-format	40
Block2	0+ (64 B/block)	1 byte
Size2	2072	2 bytes

Payload (64)

```
</obs>;ct=0;obs;rt="observe";title="Observable resource which
ch
```

Accept: text/plain

Content-Format: text/plain

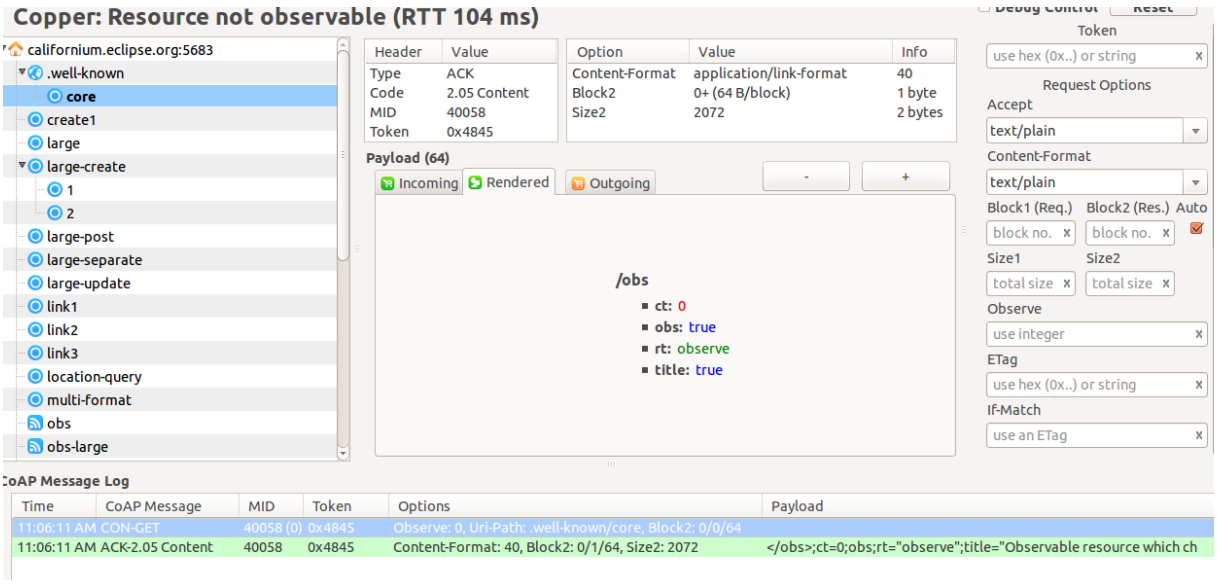
Block1 (Req.): block no.

Size1: total size

Observe: use integer

ETag: use hex (0x..)

If-Match: use an ETag



Copper: Resource not observable (RTT 104 ms)

- californium.eclipse.org:5683
  - .well-known
    - core
  - create1
  - large
  - large-create
    - 1
    - 2
  - large-post
  - large-separate
  - large-update
  - link1
  - link2
  - link3
  - location-query
  - multi-format
  - obs
  - obs-large

Header	Value
Type	ACK
Code	2.05 Content
MID	40058
Token	0x4845

Option	Value	Info
Content-Format	application/link-format	40
Block2	0+ (64 B/block)	1 byte
Size2	2072	2 bytes

Payload (64)

/obs
- ct: 0
- obs: true
- rt: observe
- title: true

Request Options

Accept: text/plain

Content-Format: text/plain

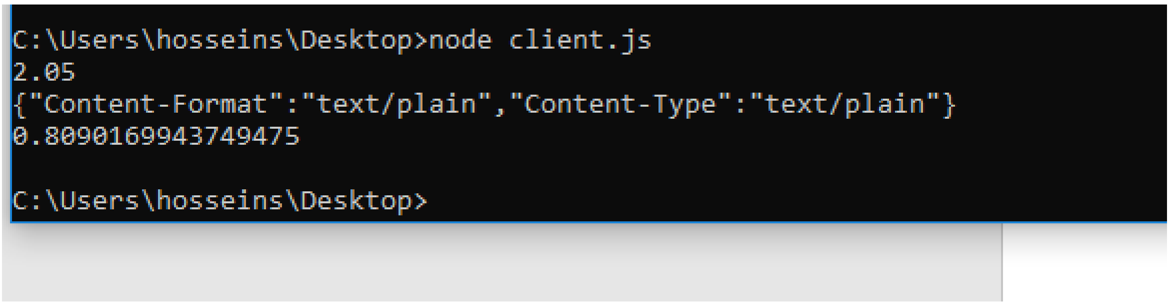
Auto ☑

CoAP Message Log

Time	CoAP Message	MID	Token	Options	Payload
11:06:11 AM	CON-GET	40058 (0)	0x4845	Observe: 0, Uri-Path: .well-known/core, Block2: 0/0/64	
11:06:11 AM	ACK-2.05 Content	40058	0x4845	Content-Format: 40, Block2: 0/1/64, Size2: 2072	</obs>;ct=0;obs;rt="observe";title="Observable resource which ch

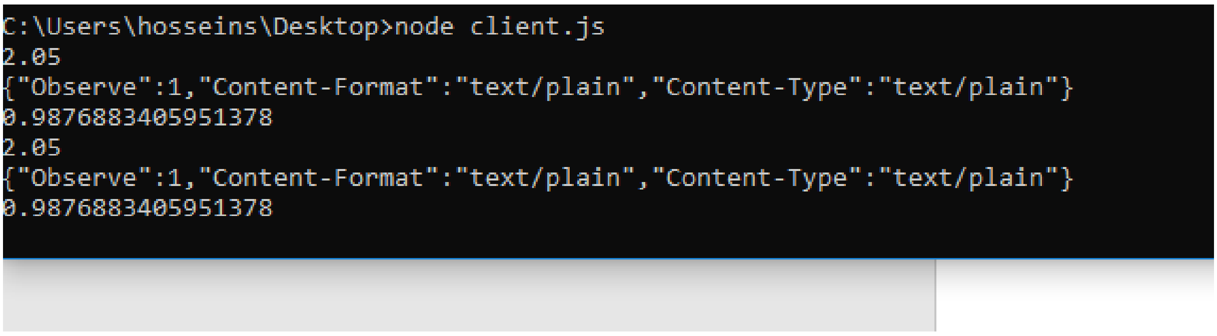
Server

Request path temp file names ‘client.js’ and ‘server.js’



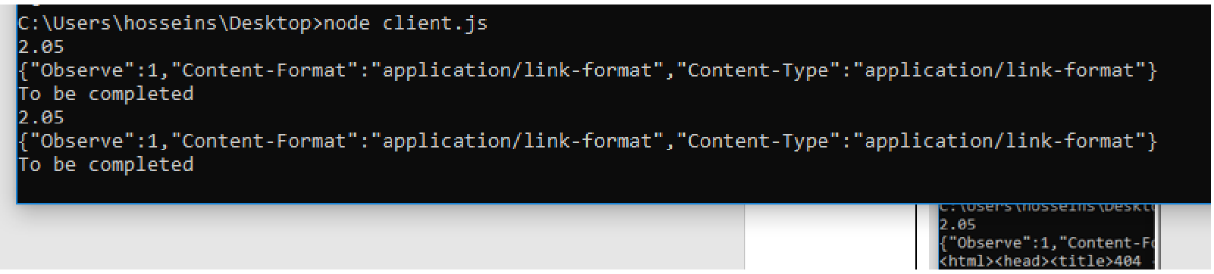
```
C:\Users\hosseins\Desktop>node client.js
2.05
{"Content-Format":"text/plain","Content-Type":"text/plain"}
0.8090169943749475

C:\Users\hosseins\Desktop>
```

Request path temp observation mode client side



```
C:\Users\hosseins\Desktop>node client.js
2.05
{"Observe":1,"Content-Format":"text/plain","Content-Type":"text/plain"}
0.9876883405951378
2.05
{"Observe":1,"Content-Format":"text/plain","Content-Type":"text/plain"}
0.9876883405951378
```

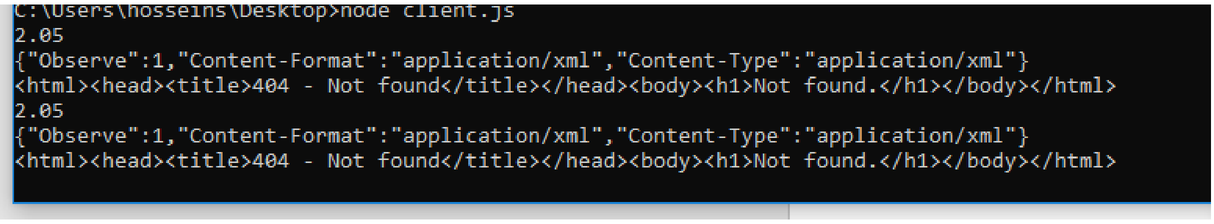
Request path well-known/core



```
C:\Users\hosseins\Desktop>node client.js
2.05
{"Observe":1,"Content-Format":"application/link-format","Content-Type":"application/link-format"}
To be completed
2.05
{"Observe":1,"Content-Format":"application/link-format","Content-Type":"application/link-format"}
To be completed
```

```
C:\Users\hosseins\Deskt
2.05
{"Observe":1,"Content-Fo
<html><head><title>404
```

Request path test2 this path is not exist



```
C:\Users\hosseins\Desktop>node client.js
2.05
{"Observe":1,"Content-Format":"application/xml","Content-Type":"application/xml"}
<html><head><title>404 - Not found</title></head><body><h1>Not found.</h1></body></html>
2.05
{"Observe":1,"Content-Format":"application/xml","Content-Type":"application/xml"}
<html><head><title>404 - Not found</title></head><body><h1>Not found.</h1></body></html>
```

To modify the client.js to send observe request we add observe:true to the request option
And in server.js check this option with if (req.headers['Observe'] !== 0)