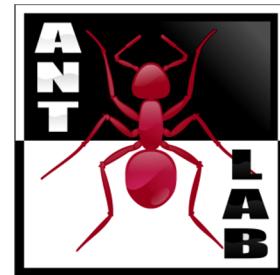




Politecnico di Milano

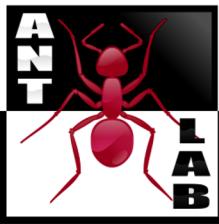
Advanced Network Technologies Laboratory



# Application layer

---

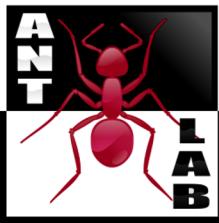
HTTP and CoAP



# Agenda

---

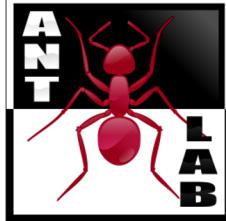
- HTTP basics
- CoAP basics
- CoAP in action
- Packets inspection



# What we'll use

---

- Virtual Machine
- RESTful API
- Command line
- Copper
- Wireshark
- Your brainz ☺



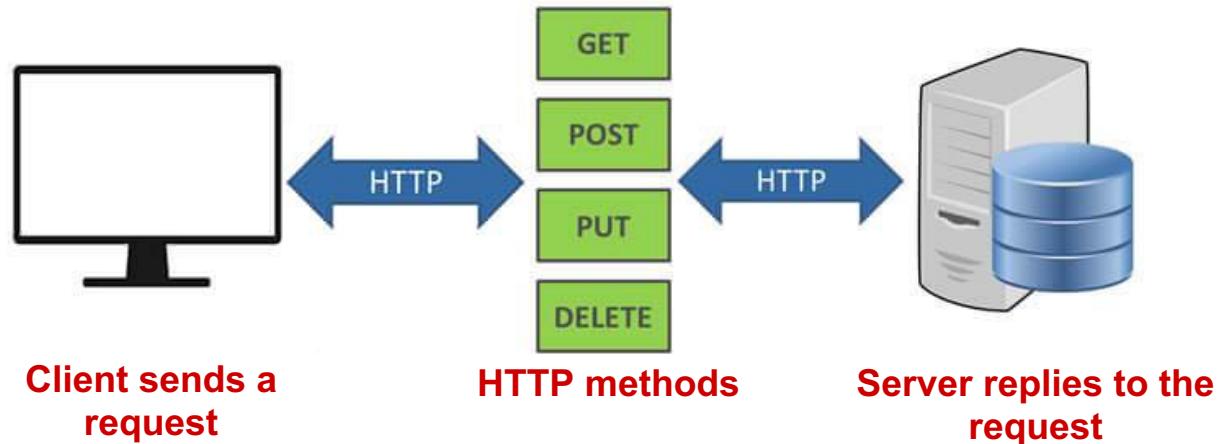
# Different needs, different protocols

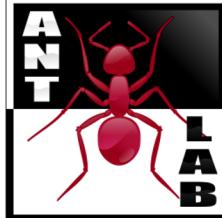
---

- End-user cloud applications
  - Display sensor data on mobile app, website dashboards
    - HTTP, Websockets...
  
- Local sensor networks
  - Transmit sensor data
    - (Usually) over **802.15.4** (zigbee), Bluetooth low energy **BLE**

# HTTP methods in a nutshell

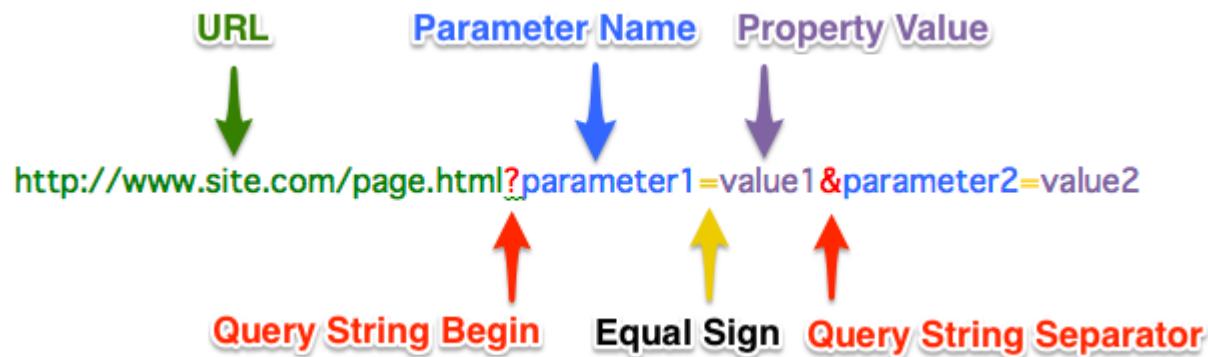
- Client-server architecture
- Restful API methods
  - GET: read resource
  - POST: create new resource
  - PUT: update resource
  - DELETE: delete resource

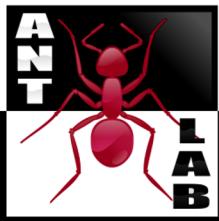




# Uniform Resource Identifier

- <https://www.example.com/>
- <http://www.example.com:1234/>
- <https://www.example.com:1234/forum/question>
- <https://www.example.com:1234/forum/question?tag=networking&order=newest>





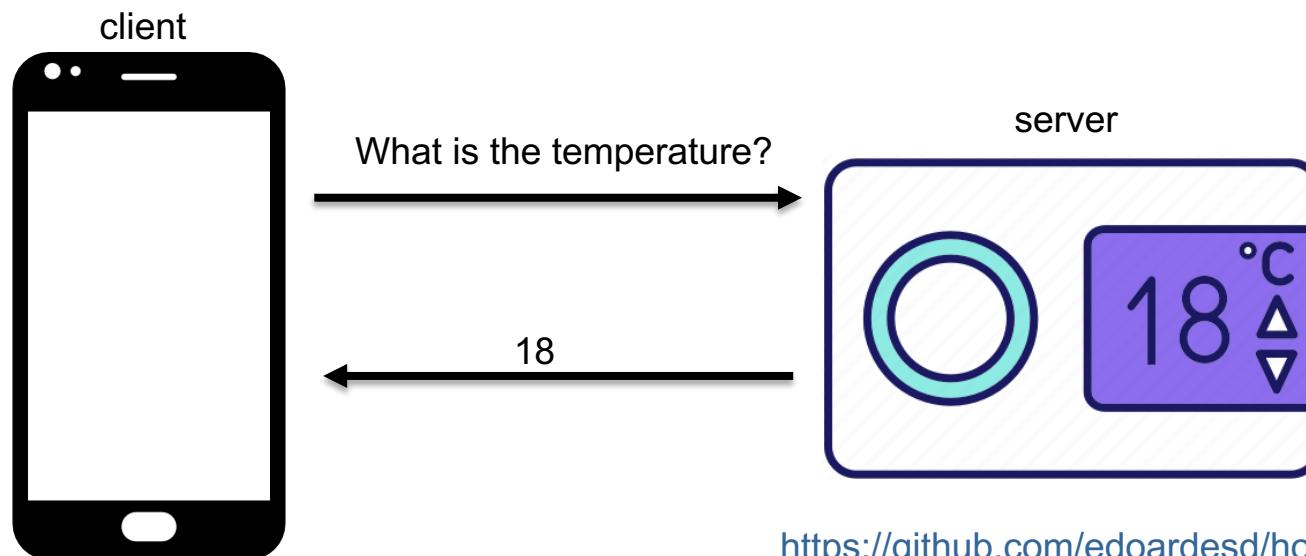
# How to send requests?

---

- Several ways to send an http request to a server
- Via browser (<https://api.thecatapi.com/v1/images/search>)
- GUI clients: Postman (<https://docs.api.getpostman.com>)
- Code library (C, python, C++, Java, JavaScript...)
- Command line

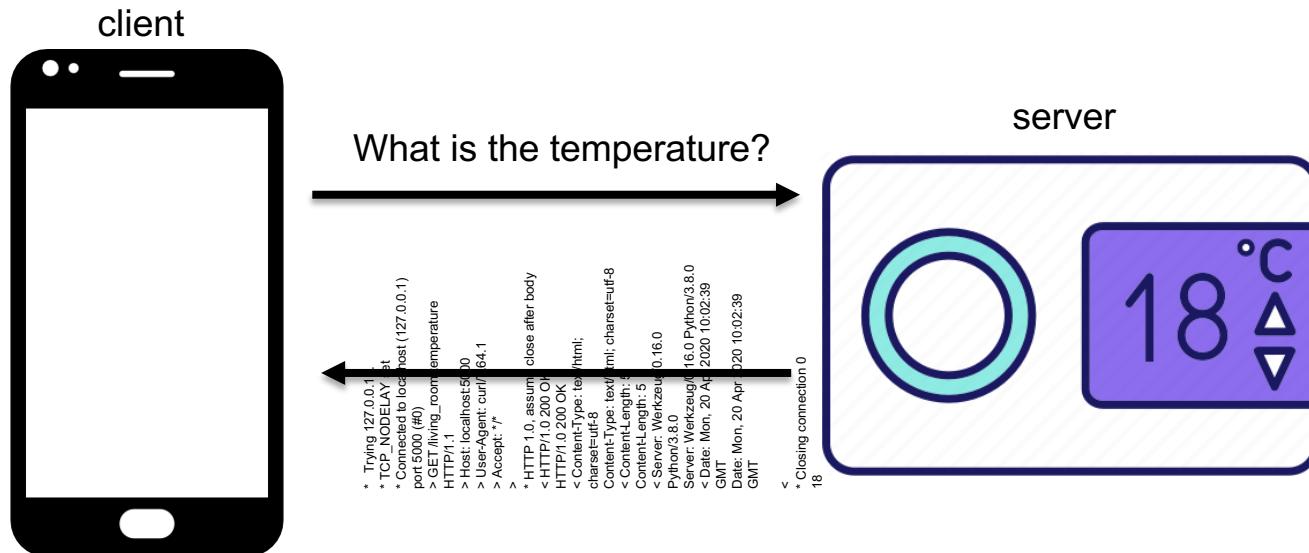
# Curl

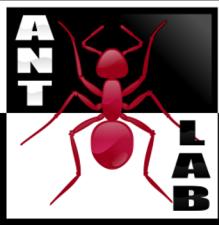
- Runs in the terminal
- Simple GET request: curl <http://hostname:port/path>
- -X specify custom method
- curl -X GET http://0.0.0.0:80/living\_room/temperature



# What's really happening?

- request header + response header
- v include the headers
- roughly 370 bytes for the request/response
  
- It's too verbose!





# Problems

---

- Long messages
- Run the server on a constrained devices

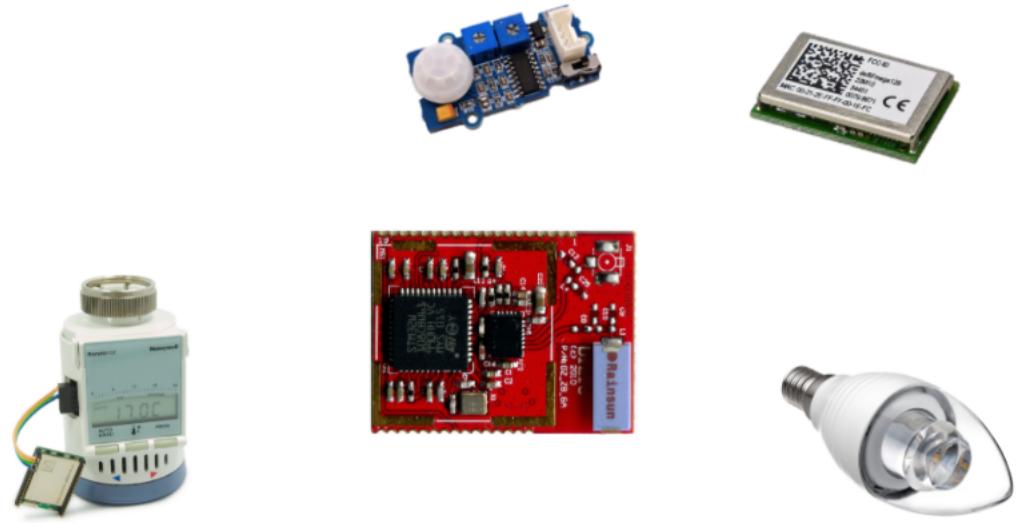
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3509	user	20	0	41708	19048	8100	S	0.7	2.0	0:05.04	python3.8
3678	user	20	0	53512	7604	4156	S	0.3	0.8	0:02.05	python2.7

A red circle highlights the %CPU column for both processes. Two red arrows point from the right side of the slide to the COMMAND column: one arrow points to the entry for PID 3509 labeled "HTTP Server" and another arrow points to the entry for PID 3678 labeled "CoAP Server".

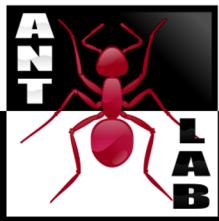
# Drawback in IoT

Target devices:

- battery-powered
- ~ \$1
- ~ 100KB Flash
- ~ 10KB RAM



HTTP is not a good fit!



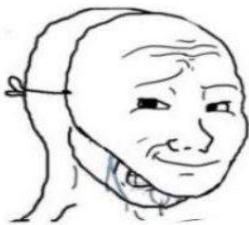
# Constrained Application Protocol

- Is adapted to the small resources of IOT devices
- Can transport RESTful calls over 'thin' networks
- Only needs small memories and slow processor
- Use the UDP protocol
- Additional features for M2M scenarios



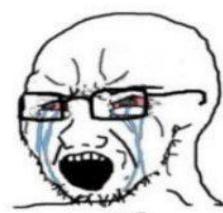


\*UDP

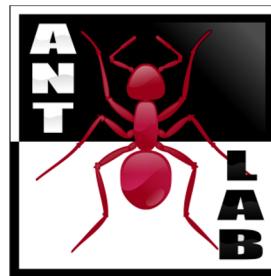


Here is your data:

01100100 01010001 01110111 00110100 01110111 00111001 01010111 01100111 01011000 01100011 01010001



No, pls wait... let me..just..



\*TCP



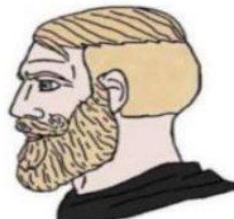
would you like to see  
a TCP/IP meme?



Yes i would like to see  
a TCP/IP meme



Ok, here is the meme  
did you recieve it?



Yes, i recievied the meme



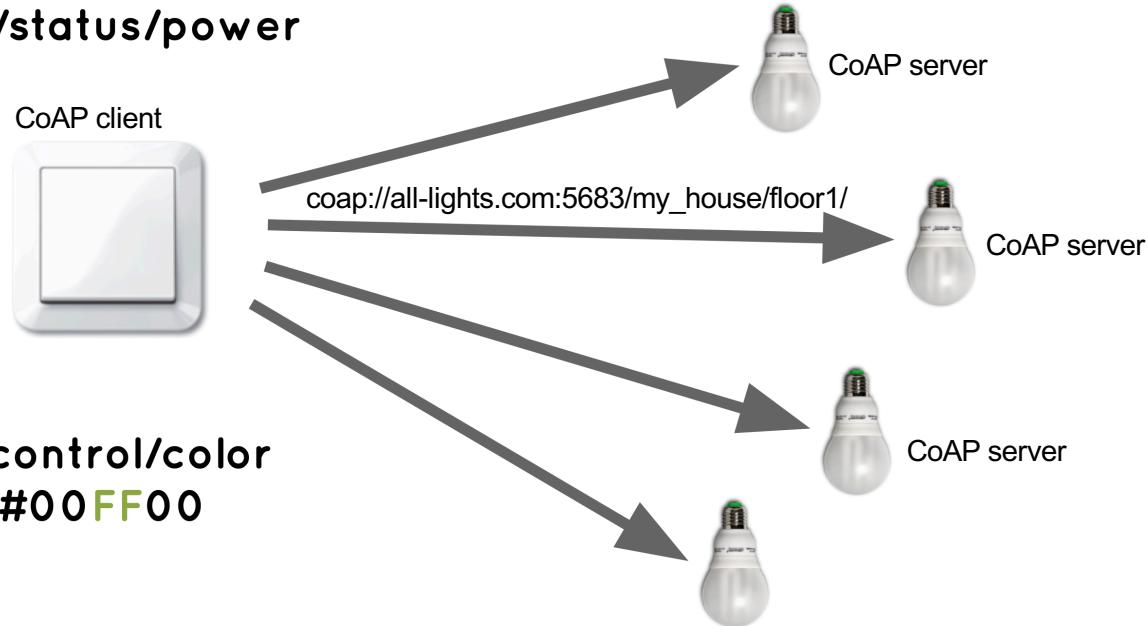
Excellent, TCP/IP meme is over.

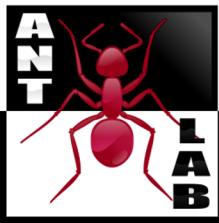


nice

# CoAP scenario

**GET /status/power**





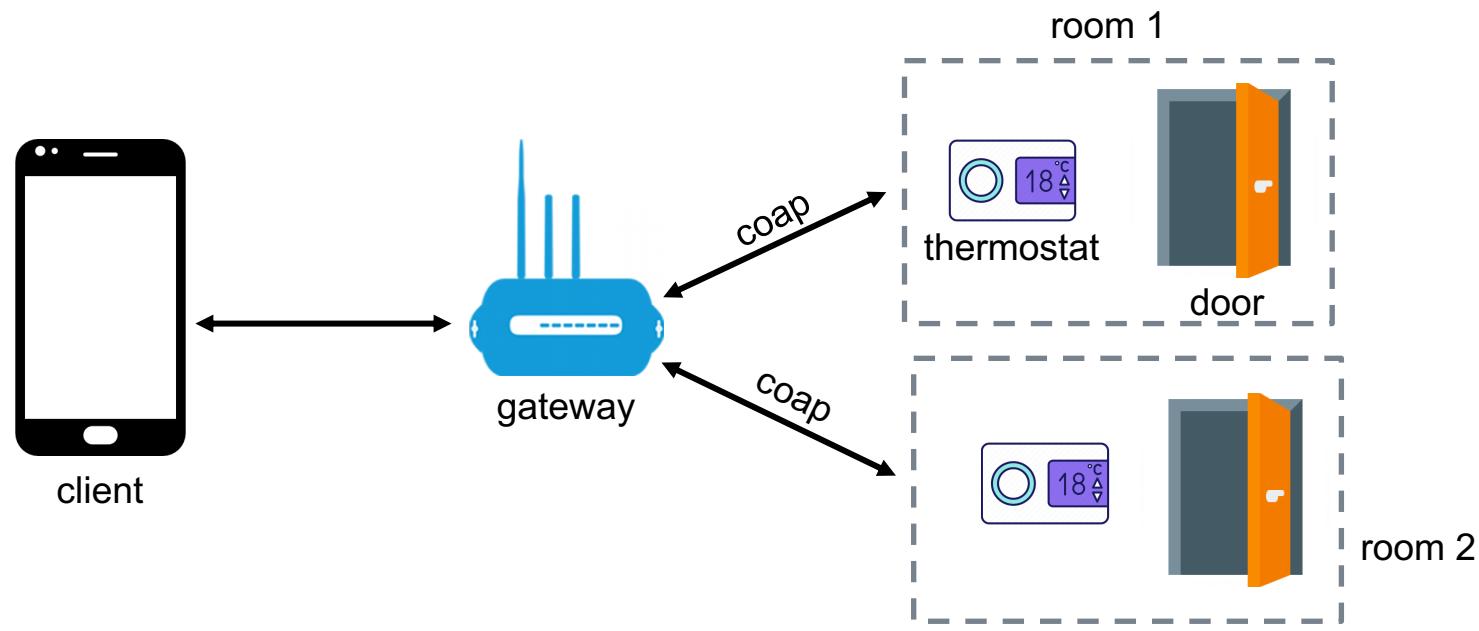
# Features

---

- RESTful API
- UDP datagram based
- Resource Discovery mode
- Observe Resources
- Confirmable/NON-Confirmable messages
- Block transfer mode

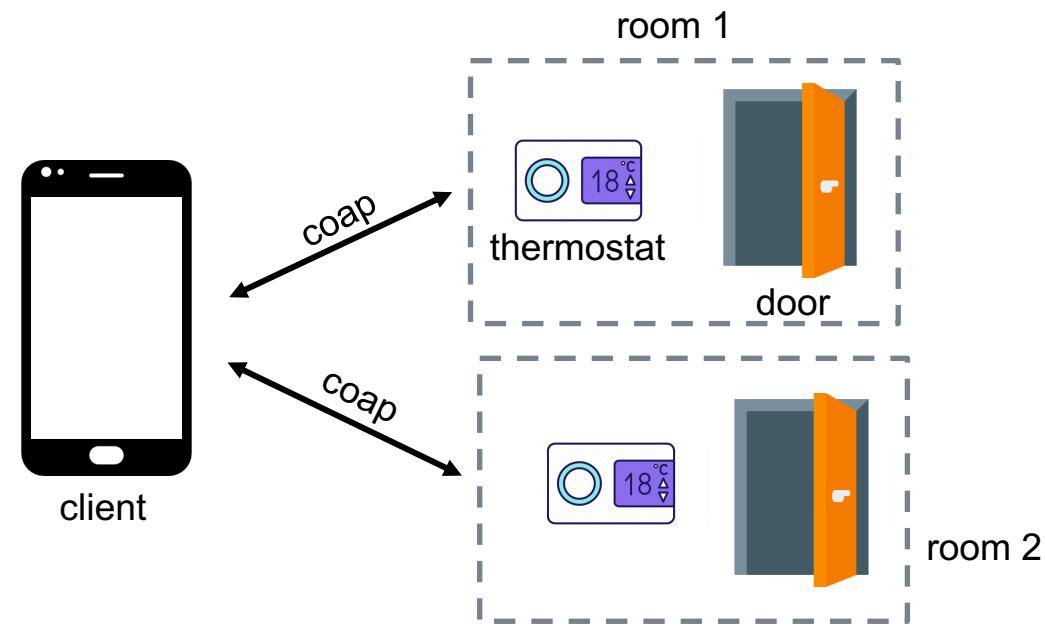
# CoAP in action

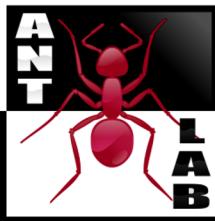
- Smart home scenario ( based on <https://github.com/edoardesd/home-CoAPServer> )
- Discovery
- Read the temperature once - > GET
- Read every change of temperature -> Observe
- Close the living room door -> PUT
- Add a smart door -> POST



# CoAP in action

- Smart home scenario (based on <https://github.com/edoardesd/home-CoAPServer>)
- Discovery
- Read the temperature once - > GET
- Read every change of temperature -> Observe
- Close the living room door -> PUT
- Add a smart door -> POST

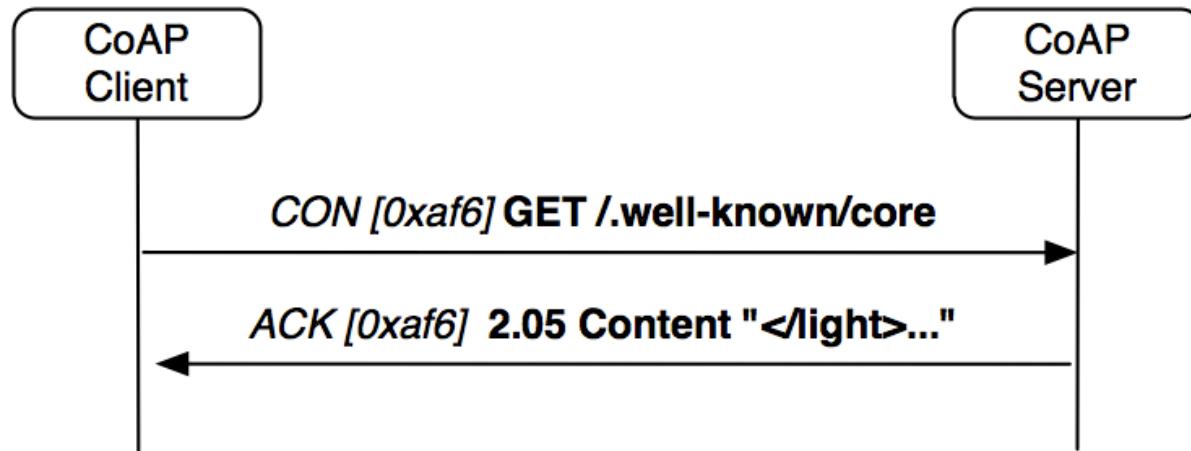




# Accessing CoAP resources

- Use as client Copper (firefox plugin)
- Open firefox
- Insert the url
  - coap://localhost:5863/

# CoRE Resource Discovery

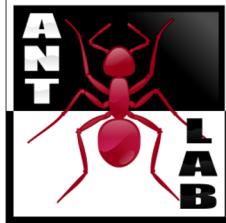


```
</dev/bat>;obs;if="";rt="ipso:dev-bat";ct="0",
</dev.mdl>;obs;if="";rt="ipso:dev-mdl";ct="0",
</dev/mfg>;obs;if="";rt="ipso:dev-mfg";ct="0",
</pwr/0/rel>;obs;if="";rt="ipso:pwr-rel";ct="0",
</pwr/0/w>;obs;if="";rt="ipso:pwr-w";ct="0",
</sen/temp>;obs;if="";rt="ucum:Cel";ct="0"
```



# GET

---



0.0.0.0:5683 - Mozilla Firefox

0.0.0.0:5683

coap://0.0.0.0:5683/living\_room/temperature

Ping Discover GET **FETCH** POST PATCH PUT iPATCH DELETE Observe Payload Behavior

### 2.05 Content (Blockwise) (Download finished)

0.0.0.0:5683

- .well-known
- core
- basic
- dinning\_room
  - door
  - temperature
- hello\_post
- hello\_world
- living\_room
  - door
  - temperature**
  - main\_door
  - test

Header	Value
Type	ACK
Code	2.05 Content
MID	14748
Token	empty

Option	Value
Block2	0 (64 B/block)

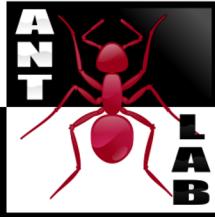
**Payload (4)**

Incoming Rendered Outgoing

22.9



# OBSERVE



0.0.0.0:5683 - Mozilla Firefox

coap://0.0.0.0:5683/living\_room/temperature

Ping Discover GET FETCH POST PATCH PUT iPATCH DELETE Cancel Payload Behavior

## 2.05 Content (Observing)

Header Value  
Type CON  
Code 2.05 Content  
MID 631  
Token 0xABOE

Option Value  
Observe 13

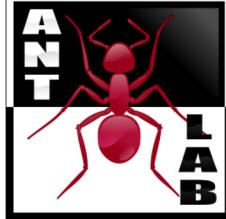
**Payload (5)**  
Incoming Rendered Outgoing  
15.52

**CoAP Message Log**

Ti...	CoAP Message	MID	Token	Options	Payload
10:16:52	CON-2.05 Content	620 (0)	0xABOE	Observe: 2	15.47
10:16:52	ACK-EMPTY	620		empty	
10:16:56	CON-2.05 Content	621 (0)	0xABOE	Observe: 3	25.18
10:16:56	ACK-EMPTY	621		empty	
10:17:02	CON-2.05 Content	622 (0)	0xABOE	Observe: 4	12.66
10:17:02	ACK-EMPTY	622		empty	
10:17:08	CON-2.05 Content	623 (0)	0xABOE	Observe: 5	19.74
10:17:08	ACK-EMPTY	623		empty	
10:17:13	CON-2.05 Content	624 (0)	0xABOE	Observe: 6	15.01
10:17:13	ACK-EMPTY	624		empty	
10:17:17	CON-2.05 Content	625 (0)	0xABOE	Observe: 7	14.6
10:17:17	ACK-EMPTY	625		empty	
10:17:21	CON-2.05 Content	626 (0)	0xABOE	Observe: 8	14.04
10:17:21	ACK-EMPTY	626		empty	
10:17:25	CON-2.05 Content	627 (0)	0xABOE	Observe: 9	21.31
10:17:25	ACK-EMPTY	627		empty	
10:17:28	CON-2.05 Content	628 (0)	0xABOE	Observe: 10	14.55
10:17:28	ACK-EMPTY	628		empty	
10:17:31	CON-2.05 Content	629 (0)	0xABOE	Observe: 11	15.42
10:17:31	ACK-EMPTY	629		empty	
10:17:37	CON-2.05 Content	630 (0)	0xABOE	Observe: 12	30.46
10:17:37	ACK-EMPTY	630		empty	
10:17:41	CON-2.05 Content	631 (0)	0xABOE	Observe: 13	15.52
10:17:41	ACK-EMPTY	631		empty	



# PUT



0.0.0.0:5683 - Mozilla Firefox

0.0.0.0:5683    coap://0.0.0.0:5683/living\_room/door?status=CLOSED

Ping Discover GET FETCH POST PATCH PUT iPATCH DELETE Observe Payload Behavior

**2.04 Changed (RTT 62 ms)**

Header Value

Type	ACK
Code	2.04 Changed
MID	5705
Token	empty

Option Value

**Payload (28)**

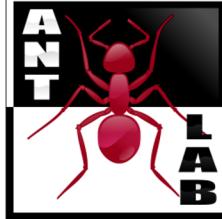
Incoming Rendered Outgoing

Response changed through PUT|

0.0.0.0:5683 .well-known core basic dining\_room door temperature hello\_post hello\_world living\_room door door2 new\_door temperature main\_door test



# POST



0.0.0.0:5683

coap://0.0.0.0:5683/living\_room/door?create

Ping Discover GET FETCH POST PATCH PUT iPATCH DELETE

### Discovering

0.0.0.0:5683

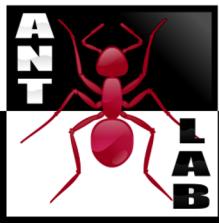
- .well-known
  - core
- basic
- dinning\_room
  - door
  - temperature
- hello\_post
- hello\_world
- living\_room
  - door
    - door2
    - new\_door
  - temperature
  - main\_door
  - test

Header	Value	Option	Value
Type			
Code			
MID			
Token			

**Payload**

Incoming Rendered Outgoing

```
new_door|
```



# Other CoAP servers

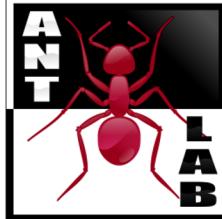
□ coap://coap.me:5683

The screenshot shows the CoAP tool interface with the following details:

- URL:** coap://coap.me:5683/path
- Content:** 2.05 Content (Blockwise) (Download finished)
- Tree View:** Shows a tree structure of resources under "coap.me:5683". The "path" node is selected, showing sub-nodes "sub1", "sub2", "query", "secret", "seg1", "separate", and "cink".
- Table View:** Displays message headers and options for a selected resource. For example, for a 2.05 Content message:

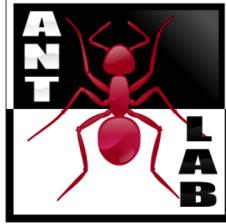
Header	Value	Option	Value	Info
Type	ACK	ETag	0x54480CA2FA8591F3	8 bytes
Code	2.05 Content	Content-Format	application/link-format	40
MID	2010	Block2	0 (64 B/block)	1 byte
Token	empty			
- Payload:** Shows the payload structure with two sub-nodes: "/path/sub1" and "/path/sub2", each with "rt: sub1" and "ct: 0".
- Log:** CoAP Message Log table showing the exchange:

T...	CoAP Message	MID	Token	Options	Payload
09:30:10	CON-GET	2010 (0)	empty	Uri-Path: path, Block2: 0/0/64	
09:30:10	ACK-2.05 Content	2010	empty	ETag: 0x54480CA2FA8591F3, Content-Format: 40, Block2: 0/0/64	</path/sub1>;rt="sub1";ct=0,</path/sub2>;rt="sub2";ct=0
- Right Panel:** Debug Control panel with various configuration options like Request Options, Content-Format, and Uri-Host.



# Response codes

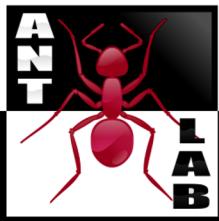
Code	Description	Reference
2.01	Created	[RFC7252]
2.02	Deleted	[RFC7252]
2.03	Valid	[RFC7252]
2.04	Changed	[RFC7252]
2.05	Content	[RFC7252]
4.00	Bad Request	[RFC7252]
4.01	Unauthorized	[RFC7252]
4.02	Bad Option	[RFC7252]
4.03	Forbidden	[RFC7252]
4.04	Not Found	[RFC7252]
4.05	Method Not Allowed	[RFC7252]
4.06	Not Acceptable	[RFC7252]
4.12	Precondition Failed	[RFC7252]
4.13	Request Entity Too Large	[RFC7252]
4.15	Unsupported Content-Format	[RFC7252]
5.00	Internal Server Error	[RFC7252]
5.01	Not Implemented	[RFC7252]
5.02	Bad Gateway	[RFC7252]
5.03	Service Unavailable	[RFC7252]
5.04	Gateway Timeout	[RFC7252]
5.05	Proxying Not Supported	[RFC7252]



# CoAP vs HTTP

---

?



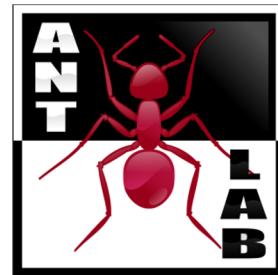
# CoAP vs HTTP

COAP	HTTP
UDP	TCP
RESTful API	RESTful API
Constrained devices	Servers/PC
Small packets	Big packets
Small overhead	Huge overhead
Resource discovery	x
Observe pattern	x
CON/NON-CON Acks	Handled by TCP
5683/5684	80/443
Still in progress/young protocol	Mature protocol



Politecnico di Milano

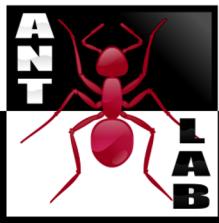
Advanced Network Technologies Laboratory



# Wireshark

---

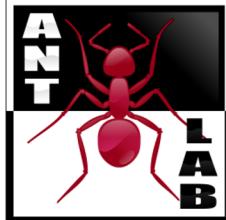
CoAP packet inspection



# Wireshark

---

- Open source packet analyzer
  - Traffic analysis
  - Packet analysis
  - Protocol analysis
  - Sniffing
  
- Process of capturing, decoding and analyzing network traffic
  - What is the network traffic pattern
  - How traffic flows among nodes



# Wireshark



Welcome to Wireshark

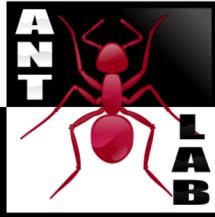
**Capture**

...using this filter:  Enter a capture filter ...

Interface	Action
enp0s3	Selected
any	—
Loopback: lo	—
hilog	—
nfqueue	—
usbmon1	—
(Cisco remote capture: ciscodump)	—
(Random packet generator: randpkt)	—
(SSH remote capture: sshdump)	—
(UDP Listener remote capture: udpdump)	—



# Packet dissector



Capturing from Loopback: lo

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression... +

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	CoAP	73	CON, MID:56593, GET, End of Block #0, /dinning_room/temperature
2	0.001072318	127.0.0.1	127.0.0.1	CoAP	83	ACK, MID:56593, 2.05 Content, Block #0 (application/json)
3	0.039829190	127.0.0.1	127.0.0.1	CoAP	73	CON, MID:56594, GET, End of Block #1, /dinning_room/temperature
4	0.041065609	127.0.0.1	127.0.0.1	CoAP	80	ACK, MID:56594, 2.05 Content, End of Block #1 (application/json)
5	409.672202041	127.0.0.1	127.0.0.1	CoAP	63	CON, MID:56595, GET, /.well-known/core
6	409.673219349	127.0.0.1	127.0.0.1	CoAP	301	ACK, MID:56595, 2.05 Content

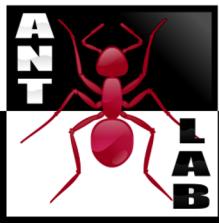
01.. .... = Version: 1  
..10 .... = Type: Acknowledgement (2)  
... 0000 = Token Length: 0  
Code: 2.05 Content (69)  
Message ID: 56595  
Opt Name: #1: Content-Format: application/link-format  
Opt Desc: Type 12, Elective, Safe  
1100 .... = Opt Delta: 12  
.... 0001 = Opt Length: 1  
Content-type: application/link-format  
End of options marker: 255  
Payload: Payload Content-Format: application/link-format, Length: 252

No.	Time	Source	Destination	Protocol	Length	Info
0020	00:01:16:33:9b:6a	01:0b:ff:1e:60:45	dd:13:c1:28	..3.j.. ..^E.. : (		
0030	ff:3c:2f:62:61:73	69:63:3e:3b:72:74	3d:22:72:74	.</basic >;rt="rt		
0040	31:22:3b:69:66:3d	22:69:66:31:22:2c	3c:2f:6d:61	1";if="i f1",</ma		
0050	69:66:5f:64:6f:6f	72:3e:3b:3c:2f:6c	69:76:69:66	in_door> ;</livin		
0060	67:5f:72:6f:6f:6d	2f:64:6f:6f:72:3e	3b:3c:2f:6c:69:76	g_room/d oor>;<d		
0070	69:6e:69:6e:67:5f	72:6f:6d:3e:3b:72	74:3d	inning_r oom>;rt=		
0080	22:72:74:31:22:3b	69:66:3d:22:69:66	31:22:2c:3c	"rt1";if ="if1",<		
0090	2f:6c:69:76:69:6e	67:5f:72:6f:6d:2f	74:65:6d	/living_ room/tem		
00a0	70:65:72:61:74:75	72:65:3e:3b:6f:62	73:2c:3c:2f	perature >;obs,<		
00b0	6c:69:76:69:6e:67	5f:72:6f:6d:3e:3b	72:74:3d	living_r oom>;rt=		
00c0	22:72:74:31:22:3b	69:66:3d:22:69:66	31:22:2c:3c	"rt1";if ="if1",<		
00d0	2f:68:65:6c:6c:6f	5f:77:6f:72:6c:64	3e:3b:3c:2f	/hello_w orld>;<		

Option Length (coap.opt.length), 1 byte

Packets: 6 · Displayed: 6 (100.0%)

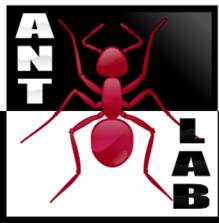
Profile: Default



# Some filters

---

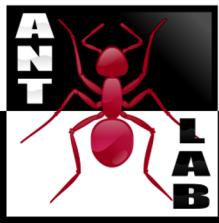
- ip.src -> source IP address
- ip.dst -> destination IP address
- ip.addr -> IP address (source or destination)
- eth.dst -> Destination MAC address
- tcp.port -> source or destination port
- udp, HTTP, coap, mqtt, ...
- ...



# Operators

---

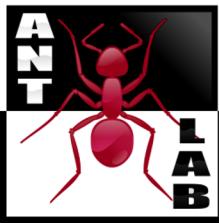
- Equal: == or eq
- Not Equal: != or ne
- Greater than: > or gt
- Less than: < or lt
- Greater than or equal: >= ge
- Less than or equal: <= le
- ...



# Examples

---

- ip.addr == 192.168.3.90
- ip.src == 10.79.0.123
- http || coap
- tcp.port == 80 && http.request



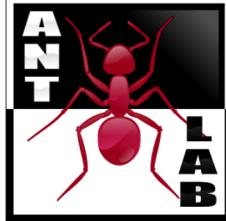
# CoAP GET

---

USB 10/100/1000 LAN: en7

coap.code == GET || coap.code == 69

No.	Time	Source	Destination	Protocol	Length	Info
167	3.092976	10.79.1.51	10.79.1.35	CoAP	60	CON, MID:15095, GET, End of Block #0, /hello_world
168	3.093895	10.79.1.35	10.79.1.51	CoAP	61	ACK, MID:15095, 2.05 Content, End of Block #0



# CoAP POST

USB 10/100/1000 LAN: en7

coap.code == POST || coap.code == 68

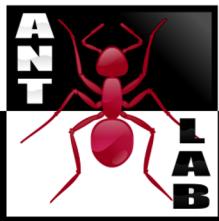
No.	Time	Source	Destination	Protocol	Length	Info
15365	128.013147	10.79.1.51	10.79.1.35	CoAP	68	CON, MID:4728, POST, /hello_post
15366	128.014216	10.79.1.35	10.79.1.51	CoAP	57	ACK, MID:4728, 2.04 Changed



# Message Format



- Version: CoAP version number
- Type:
  - 0) confirmable (CON)
  - 1) Non-confirmable (NON-CON)
  - 2) Acknowledgement (ACK)
- TKL: token length
- Code (next slide)
- Message ID (MID): sequential number



# CoAP methods wireshark codes

---

- Request: code class 0
  - GET = 0.01
  - POST = 0.02
  - PUT = 0.03
  - DELETE = 0.04
- Success response: starts with 2
- Client error response: starts with 4
- Server error response: starts with 5

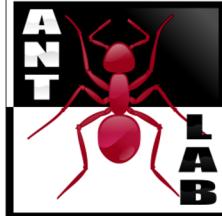


# Response codes

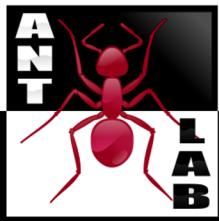
Code	Description	Reference
2.01	Created	[RFC7252]
2.02	Deleted	[RFC7252]
2.03	Valid	[RFC7252]
2.04	Changed	[RFC7252]
2.05	Content	[RFC7252]
4.00	Bad Request	[RFC7252]
4.01	Unauthorized	[RFC7252]
4.02	Bad Option	[RFC7252]
4.03	Forbidden	[RFC7252]
4.04	Not Found	[RFC7252]
4.05	Method Not Allowed	[RFC7252]
4.06	Not Acceptable	[RFC7252]
4.12	Precondition Failed	[RFC7252]
4.13	Request Entity Too Large	[RFC7252]
4.15	Unsupported Content-Format	[RFC7252]
5.00	Internal Server Error	[RFC7252]
5.01	Not Implemented	[RFC7252]
5.02	Bad Gateway	[RFC7252]
5.03	Service Unavailable	[RFC7252]
5.04	Gateway Timeout	[RFC7252]
5.05	Proxying Not Supported	[RFC7252]



# OBSERVE



USB 10/100/1000 LAN: en7						
No.	Time	Source	Destination	Protocol	Length	Info
21446	185.491607	10.79.1.35	10.79.1.51	CoAP	59	ACK, MID:2767, 2.05 Content, TKN:f6 ea, End of Block #0, /living
21493	185.773108	10.79.1.35	10.79.1.51	CoAP	56	CON, MID:291, 2.05 Content, TKN:f6 ea, /living_room/temperature
21549	185.805100	10.79.1.51	10.79.1.35	CoAP	60	ACK, MID:291, Empty Message
22378	191.530022	10.79.1.35	10.79.1.51	CoAP	56	CON, MID:292, 2.05 Content, TKN:f6 ea, /living_room/temperature
22389	191.721054	10.79.1.51	10.79.1.35	CoAP	60	ACK, MID:292, Empty Message
22978	196.397494	10.79.1.35	10.79.1.51	CoAP	56	CON, MID:293, 2.05 Content, TKN:f6 ea, /living_room/temperature
22980	196.467054	10.79.1.51	10.79.1.35	CoAP	60	ACK, MID:293, Empty Message
23258	202.217167	10.79.1.35	10.79.1.51	CoAP	56	CON, MID:294, 2.05 Content, TKN:f6 ea, /living_room/temperature
23259	202.289594	10.79.1.51	10.79.1.35	CoAP	60	ACK, MID:294, Empty Message
23698	207.081772	10.79.1.35	10.79.1.51	CoAP	56	CON, MID:295, 2.05 Content, TKN:f6 ea, /living_room/temperature
23700	207.104646	10.79.1.51	10.79.1.35	CoAP	60	ACK, MID:295, Empty Message
24346	212.830136	10.79.1.35	10.79.1.51	CoAP	56	CON, MID:296, 2.05 Content, TKN:f6 ea, /living_room/temperature
24360	212.944038	10.79.1.51	10.79.1.35	CoAP	60	ACK, MID:296, Empty Message
24882	218.526117	10.79.1.35	10.79.1.51	CoAP	56	CON, MID:297, 2.05 Content, TKN:f6 ea, /living_room/temperature
24900	218.570133	10.79.1.51	10.79.1.35	CoAP	60	ACK, MID:297, Empty Message
25530	223.390820	10.79.1.35	10.79.1.51	CoAP	56	CON, MID:298, 2.05 Content, TKN:f6 ea, /living_room/temperature
25538	223.492133	10.79.1.51	10.79.1.35	CoAP	60	ACK, MID:298, Empty Message
25859	227.073035	10.79.1.35	10.79.1.51	CoAP	56	CON, MID:299, 2.05 Content, TKN:f6 ea, /living_room/temperature
25867	227.172178	10.79.1.51	10.79.1.35	CoAP	60	ACK, MID:299, Empty Message
26416	231.806623	10.79.1.35	10.79.1.51	CoAP	56	CON, MID:300, 2.05 Content, TKN:f6 ea, /living_room/temperature
26419	231.882306	10.79.1.51	10.79.1.35	CoAP	60	ACK, MID:300, Empty Message
27101	237.144432	10.79.1.35	10.79.1.51	CoAP	56	CON, MID:301, 2.05 Content, TKN:f6 ea, /living_room/temperature
27104	237.206921	10.79.1.51	10.79.1.35	CoAP	60	ACK, MID:301, Empty Message
27368	242.313971	10.79.1.35	10.79.1.51	CoAP	55	CON, MID:302, 2.05 Content, TKN:f6 ea, /living_room/temperature

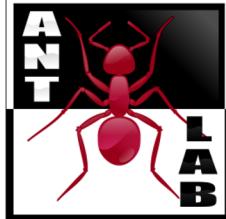


# HTTP vs CoAP

USB 10/100/1000 LAN: en7

No.	Time	Source	Destination	Protocol	Length	Info
+ 347484	2211.246666	10.79.1.51	10.79.1.35	HTTP	156	GET /temperature HTTP/1.1
+ 347487	2211.247448	10.79.1.35	10.79.1.51	HTTP	207	HTTP/1.0 200 OK (text/html)
347802	2215.001648	10.79.1.51	10.79.1.35	CoAP	72	CON, MID:2772, GET, End of Block #0, /living_room/temperature
347803	2215.002517	10.79.1.35	10.79.1.51	CoAP	55	ACK, MID:2772, 2.05 Content, End of Block #0

► Internet Protocol Version 4, Src: 10.79.1.35, Dst: 10.79.1.51  
► Transmission Control Protocol, Src Port: 5000, Dst Port: 50926, Seq: 18, Ack: 91, Len: 141  
► [2 Reassembled TCP Segments (158 bytes): #347486(17), #347487(141)]  
▼ Hypertext Transfer Protocol  
  ▼ HTTP/1.0 200 OK\r\n    [Expert Info (Chat/Sequence): HTTP/1.0 200 OK\r\n\r\n      [HTTP/1.0 200 OK\r\n\r\n      [Severity level: Chat]  
      [Group: Sequence]  
      Response Version: HTTP/1.0  
      Status Code: 200  
      [Status Code Description: OK]  
      Response Phrase: OK  
      Content-Type: text/html; charset=utf-8\r\n\r\n    ► Content-Length: 5\r\n    Server: Werkzeug/0.16.0 Python/3.8.0\r\n    Date: Thu, 23 Jan 2020 10:52:59 GMT\r\n    \r\n    [HTTP response 1/1]  
    [Time since request: 0.000782000 seconds]  
    [Request in frame: 347484]  
    [Request URI: http://10.79.1.35:5000/temperature]  
    File Data: 5 bytes  
  ► Line-based text data: text/html (1 lines)



# Some CoAP filters

---

- coap.code
- coap.length
- coap.opt.observe
- coap.opt.uri\_path
- coap.opt.uri\_port
- coap.opt.uri\_query
- coap.payload
- coap.payload\_length
- For a more detailed reference:  
<https://www.wireshark.org/docs/dref/c/coap.html>



# No challenge today