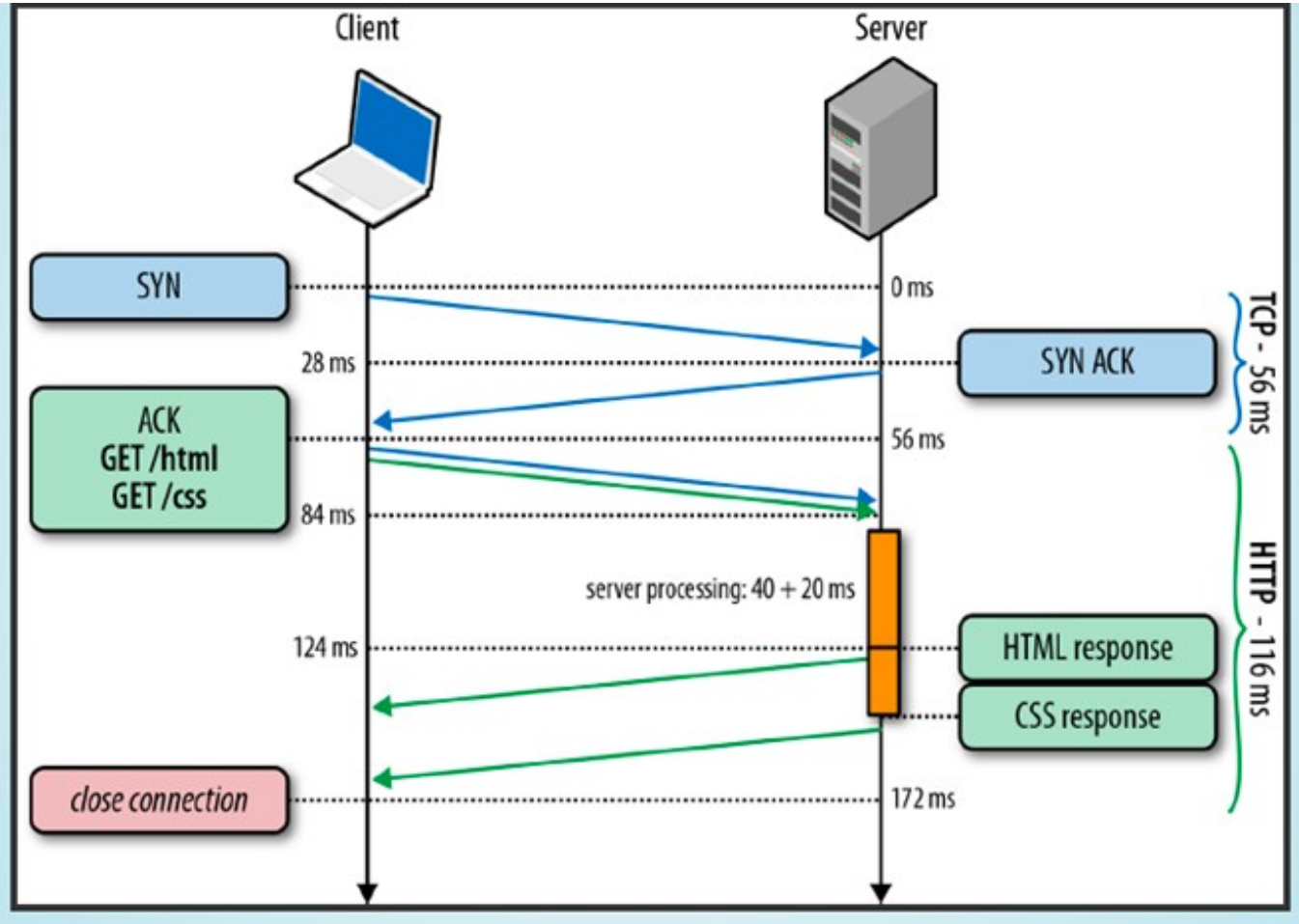
Application Layer Protocols : CoAP

Dr. Bibhas Ghoshal

IIIT Allahabad

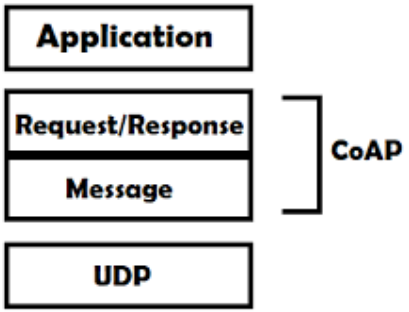
HTTP for IoT


Client
Server
SYN
0 ms
28 ms
SYN ACK
TCP - 56 ms
ACK
GET /html
GET /css
56 ms
84 ms
server processing: 40 + 20 ms
124 ms
HTML response
CSS response
HTTP - 116 ms
close connection
172 ms

Constrained Application Protocol (CoAP)

A specialized web transfer protocol for use with constrained nodes and constrained networks in Internet of Things

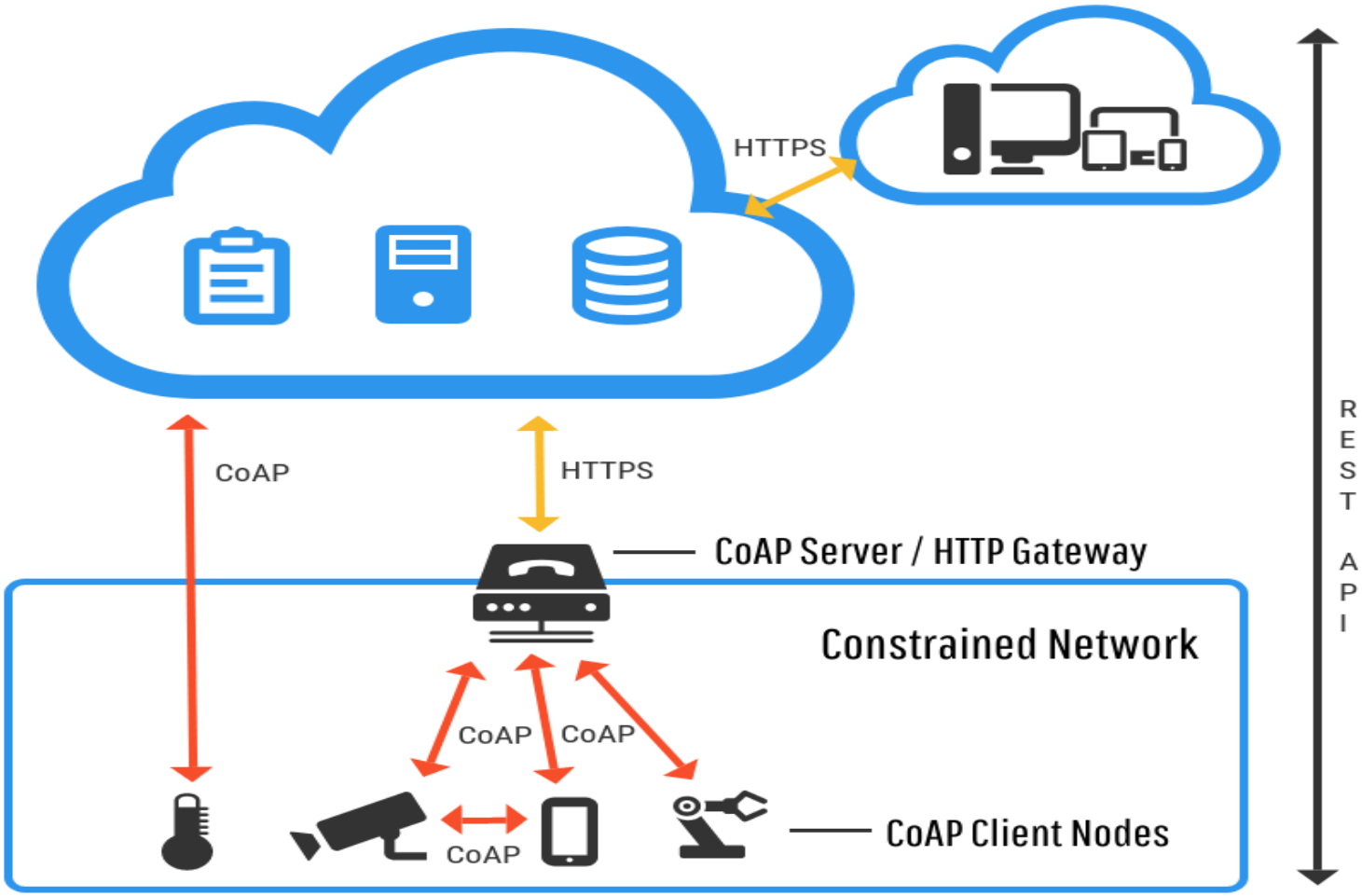
- **CoRE , IETF group**
- **Uses both request/response and publish/subscribe model**
- **Proposed Standard : RFC 7252**
- **Lightweight fast HTTP**
- **Designed for manipulation of simple resources on constrained node networks**
- **CoAP is designed to use low power sensors to use RESTful services while meeting power constraints.**



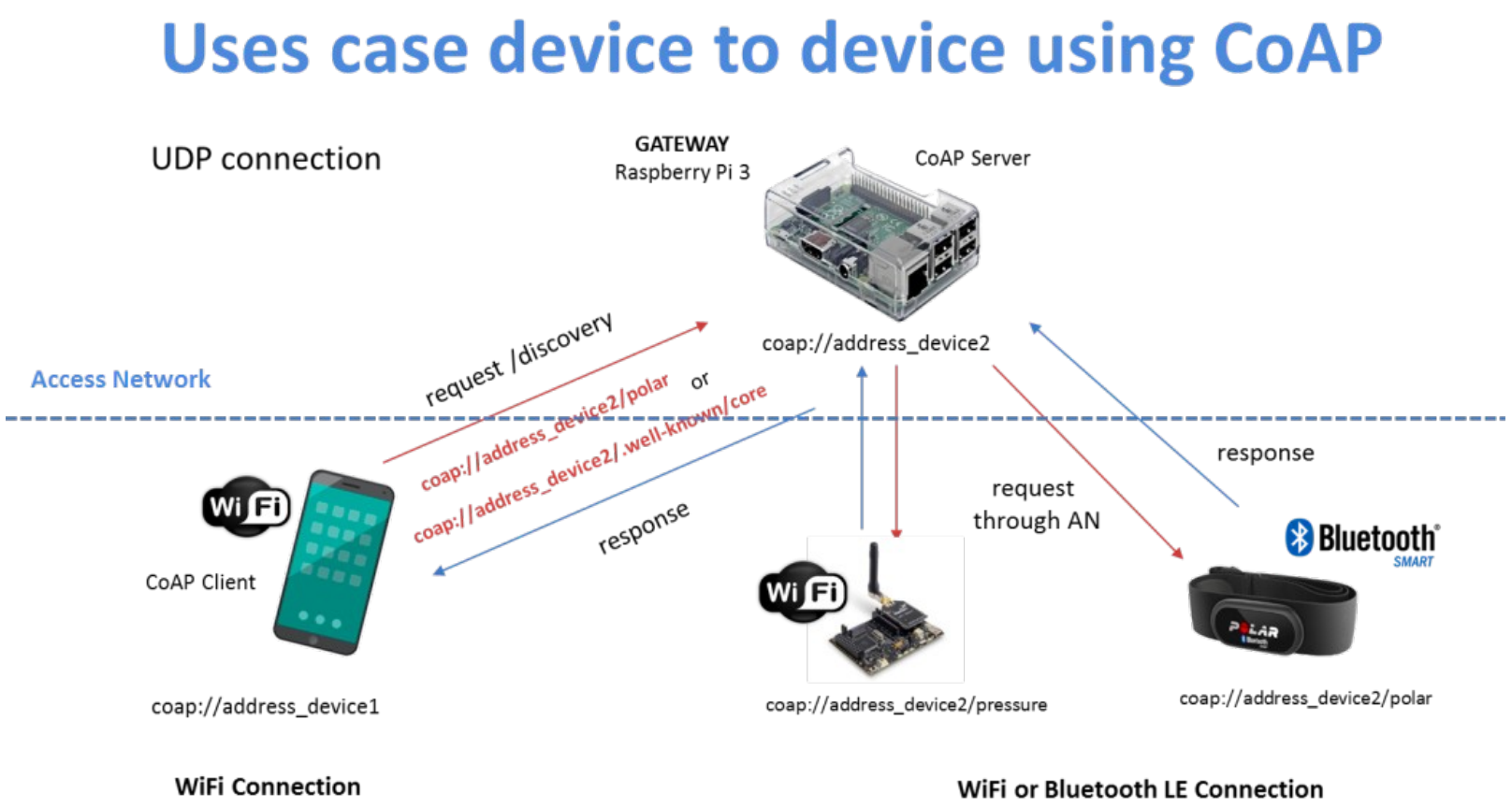
CoAP

- **Built over UDP, instead of TCP (commonly used with HTTP) and has lightweight mechanism to provide reliability**
- **CoAP Architecture has two sublayers :**
  - Messaging – reliability and duplication of messages
  - Request/Response – communication
- **CoAP Messaging Modes :**
  - Confirmable
  - Non-confirmable
  - Piggyback
  - Separate

CoAP Architecture



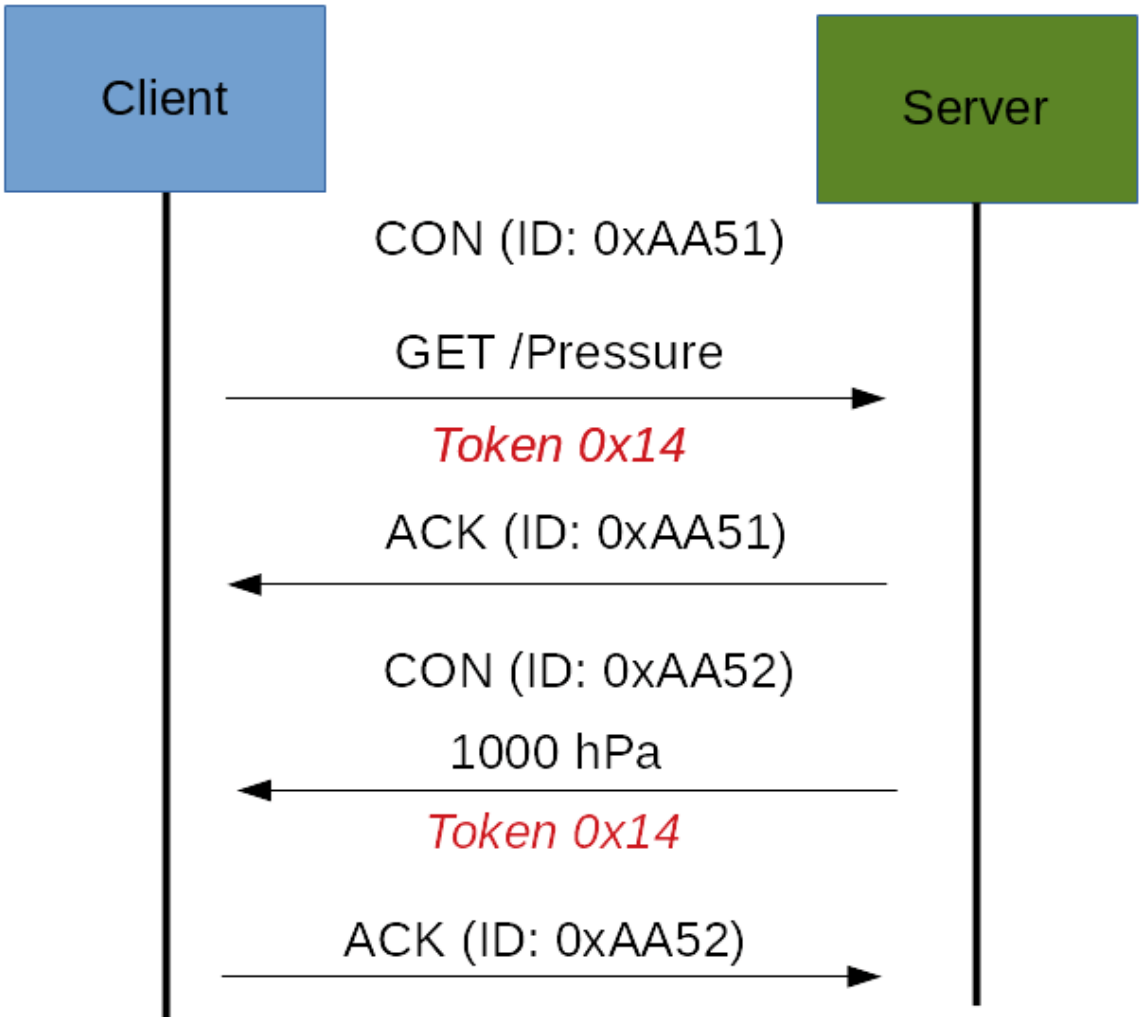
Device to Device Using CoAP



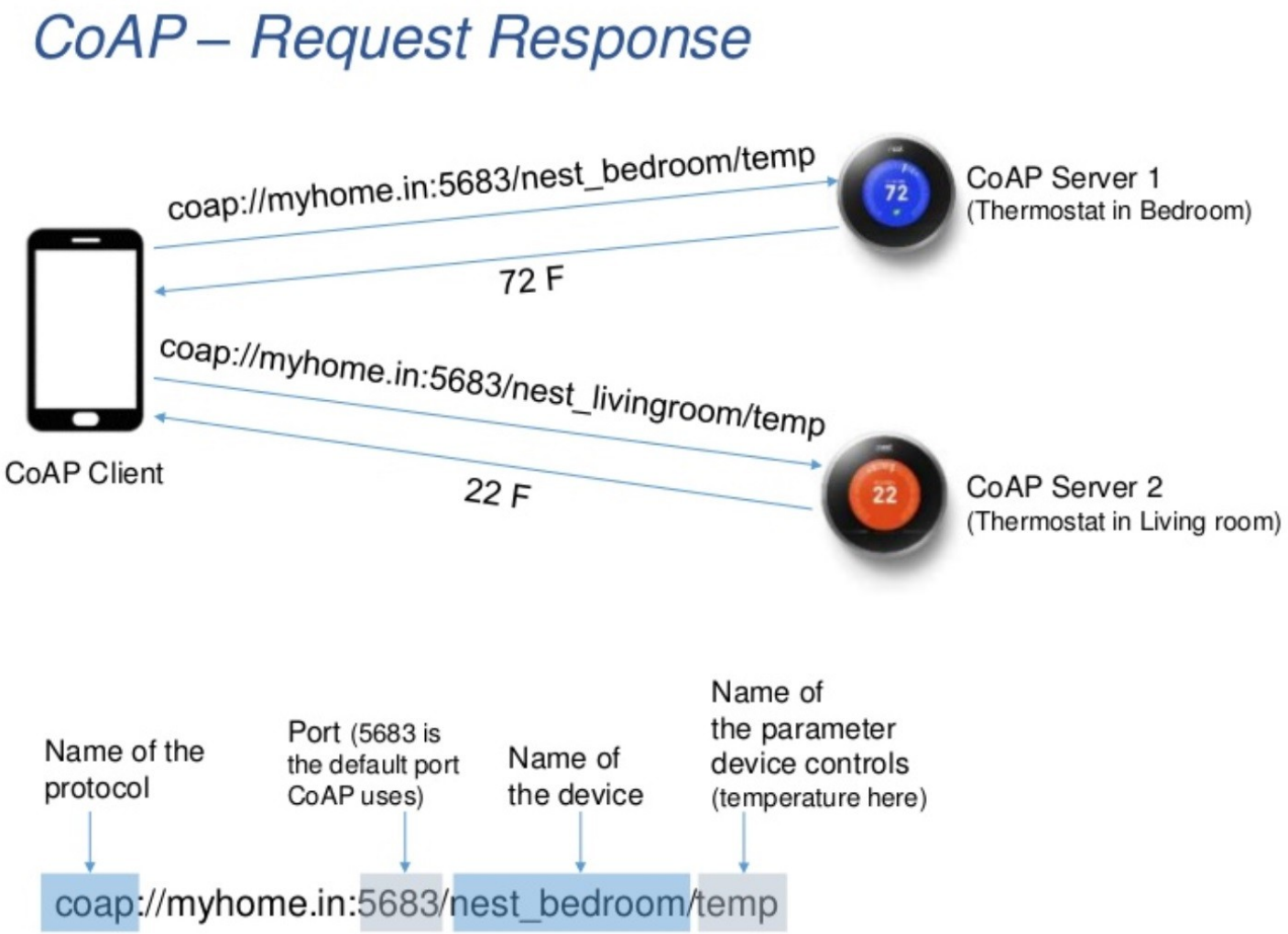
CoAP Message Types

- Confirmable and non-confirmable modes represent reliable and unreliable transmissions
- Other modes are used for request/response
- Piggyback is used for client/server direct communication where server sends its response directly after receiving message i.e. within acknowledgement message
- Separate mode is used when server response comes in a message other than acknowledgement
- CoAP uses GET, POST, PUT and DELETE messages

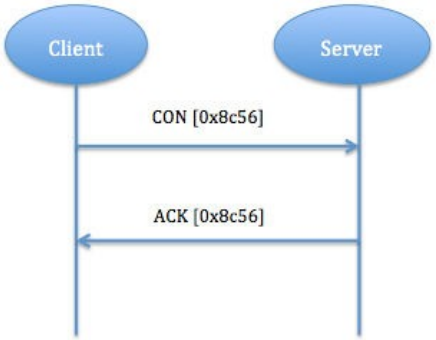
CoAP Request Response



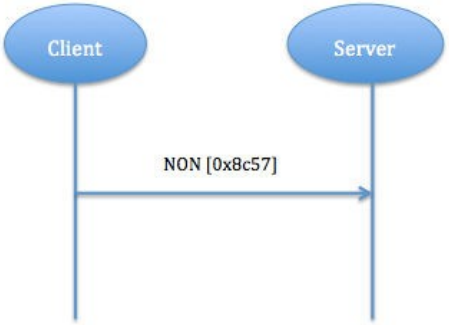
CoAP Request Response



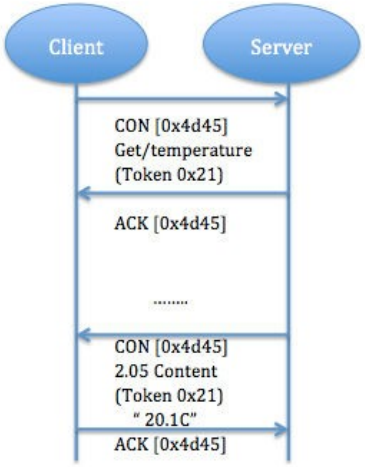
Confirmable and Non-Confirmable Mode



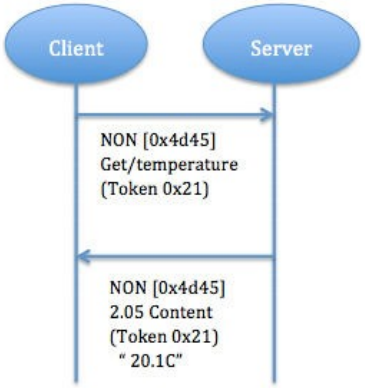
(a) Reliable Transmission



(b) Unreliable Transmission

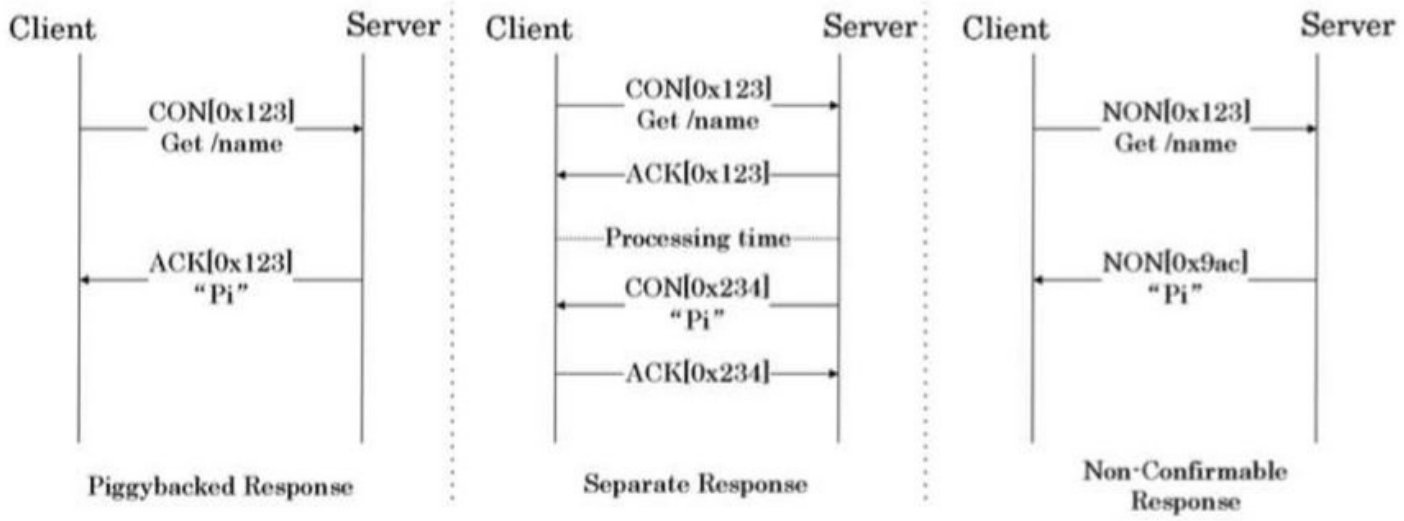


(c) Confirmable Request and Separate Confirmable Response

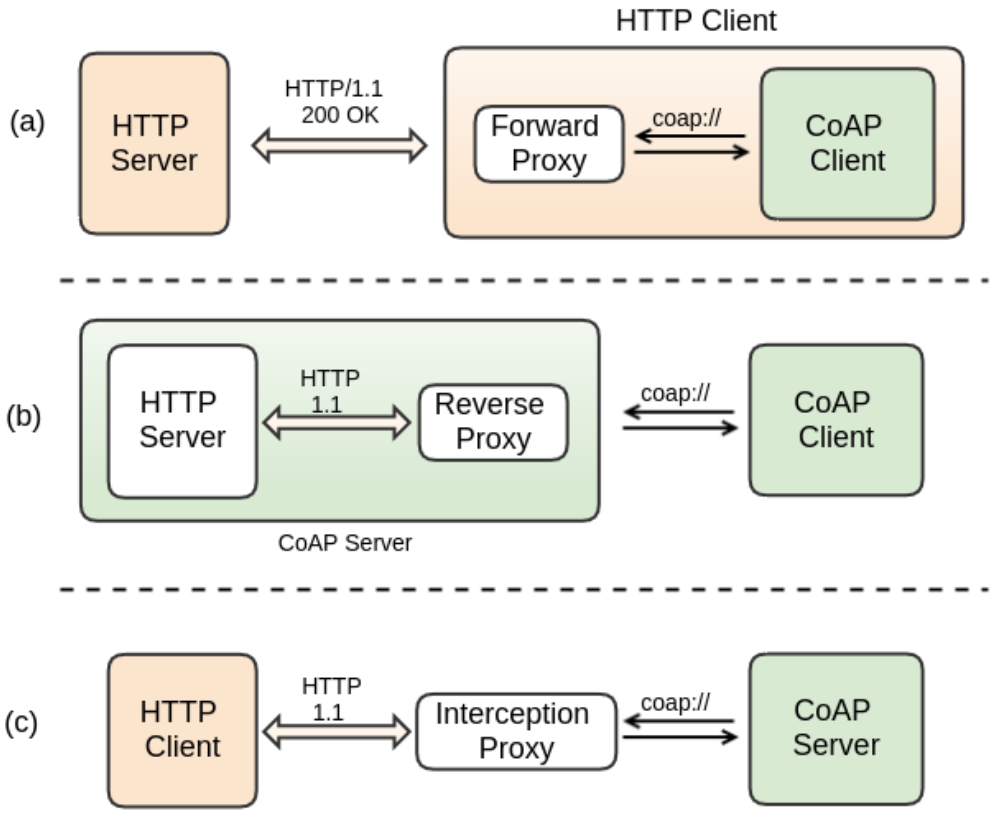


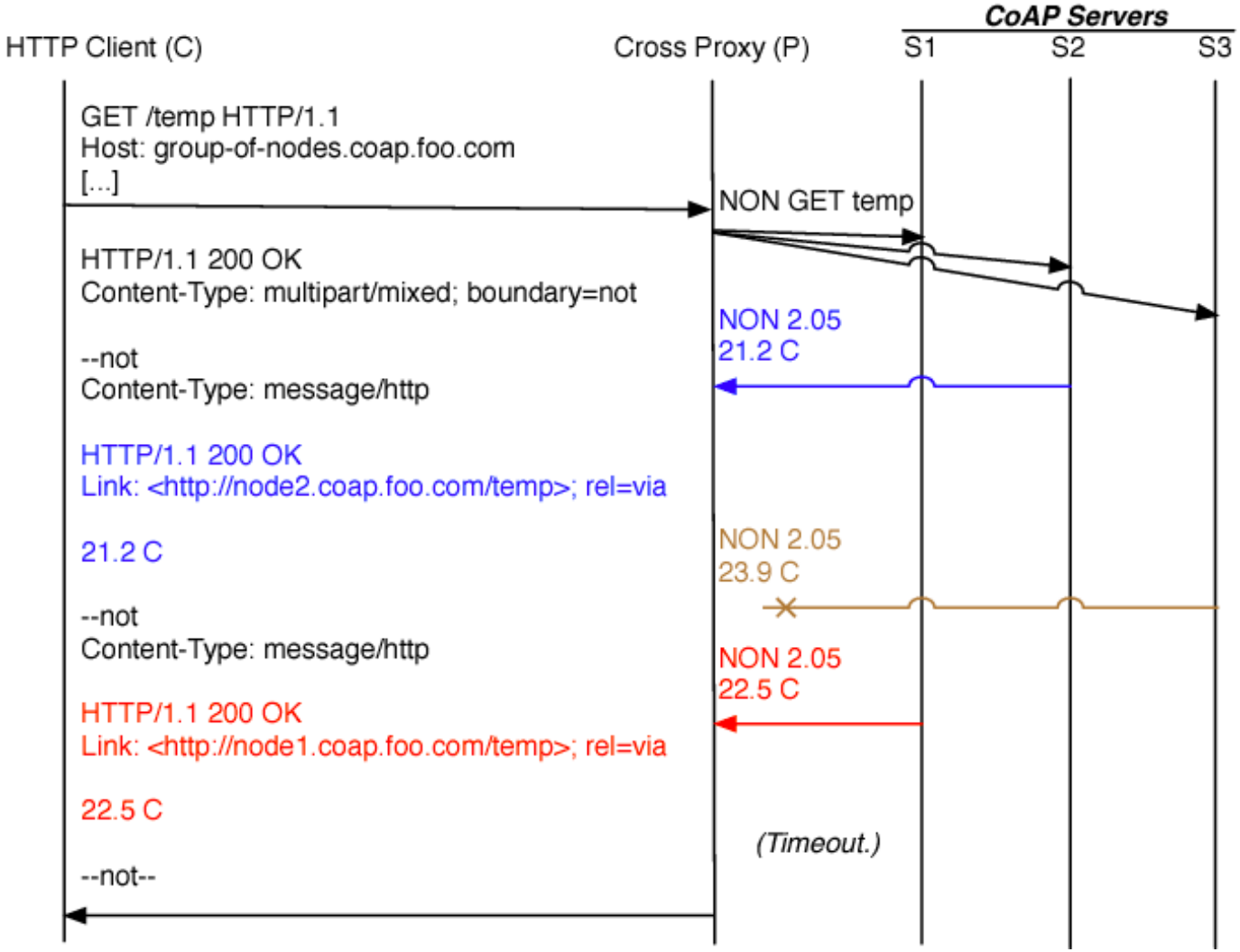
(d) Non-Confirmable Request and Non-Confirmable Response

CoAP Request and Response



CoAP-HTTP Proxy




HTTP Client (C)
Cross Proxy (P)
CoAP Servers
S1
S2
S3
GET /temp HTTP/1.1
Host: group-of-nodes.coap.foo.com
[...]
NON GET temp
HTTP/1.1 200 OK
Content-Type: multipart/mixed; boundary=not
NON 2.05
21.2 C
--not
Content-Type: message/http
HTTP/1.1 200 OK
Link: <http://node2.coap.foo.com/temp>; rel=via
21.2 C
NON 2.05
23.9 C
--not
Content-Type: message/http
NON 2.05
22.5 C
HTTP/1.1 200 OK
Link: <http://node1.coap.foo.com/temp>; rel=via
22.5 C
(Timeout.)
--not--

CoAP Functionalities

- URI
- GET/POST/PUT/DELETE
- Content type support – XML, JSON
- Built-In Discovery
- Multicast support
- Asynchronous message exchanges
- Designed to be extensible

CoAP Message Format

CoAP Message Format

Byte	0								1								2								3							
Bit	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Field	Ver		T		TKL				Code								Message ID															
	Token (if any, TKL bytes)																															
	Options																															
	1	1	1	1	1	1	1	1	Payload																							

Ver - Version (1)
T – Message Type (Confirmable, Non-Confirmable, Acknowledgement, Reset)
TKL- Token Length, if any, the number of Token bytes after this header
Code - Request Method (1-10) or Response Code (40-255)
Message ID – 16-bit identifier for matching responses
Token – Optional response matching token