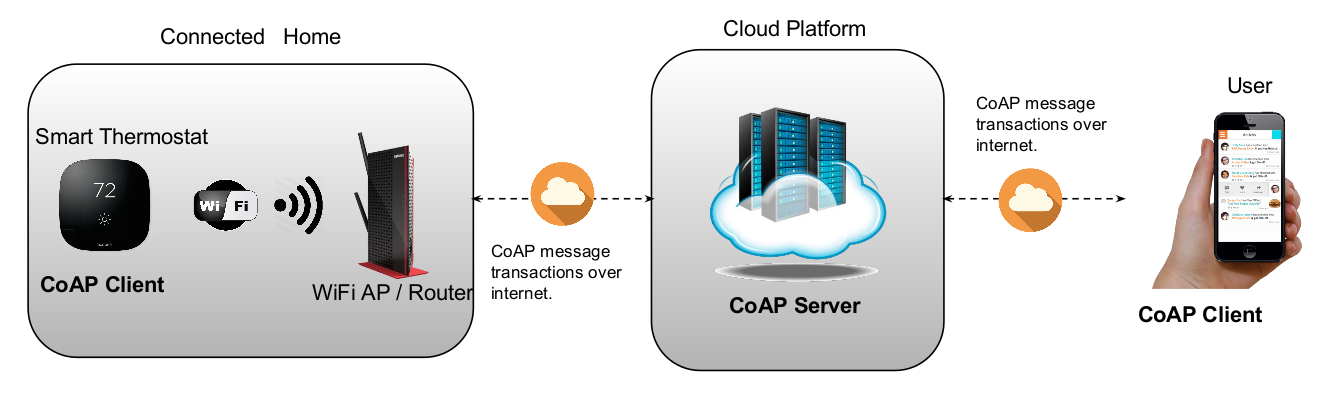
**Server CoAP**

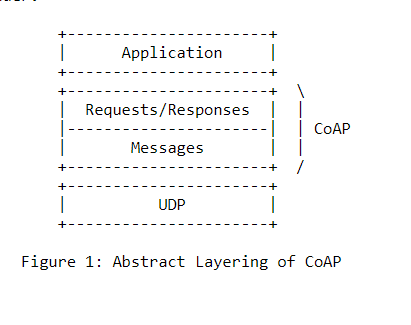
****

Protocolul **CoAP (Constrained Application Protocol)** este un protocol întâlnit la nivelul aplicației. Este un protocol software creat pentru a fi folosit de dispozitivele electronice simple care pot comunica între ele prin intermediul Internetului. Dezvoltatorii pot interacționa cu orice dispozitiv CoAP în același mod în care puteau comunica cu un dispozitiv care se bază pe tradiționalul REST (Representational State Transfer) utilizat de World Wide Web. Serverele pun resursele disponibile sub o adresă URL și clienții pot accesa aceste resurse utilizand metode cum ar fi GET, PUT, POST sau DELETE. Acest protocol este în particular util atunci când se dorește comunicarea între dispositive care necesită să fie controlate prin intermediul Internetului.

Este un protocol cerere-răspuns care urmărește modelul tradițional client-server. Principalele caracteristici alea acestui protocol sunt:

* Protocol software creat pentru a fi folosit de dispozitivele electronice simple
* Schimbul asincron de mesaje intre client si server
* Metodele GET, POST, PUT, DELETE

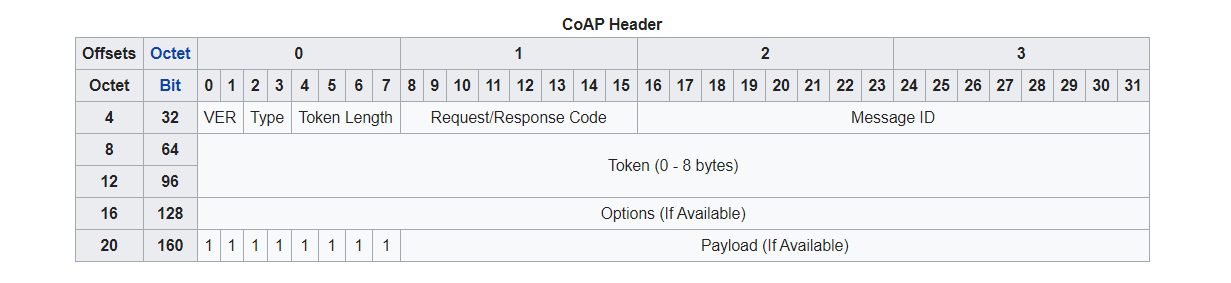
**Nivelul de abstractizare a protocolului CoAP**



**Modelul mesajelor**

Toate meseajele CoAP pot fi marcare că fiind verificate, neverificate sau confirmate.

Mesajele transmise sunt codificate binar .Mesajul incepe cu un header de 4 bytes urmat de un token value intre 0 si 8 bytes.



* V (Versiunea) este reprezentată prin 2 biți; versiunea curentă este 1, celelalte valori fiind rezervate pentru versiunile viitoare.
* T (Tipul) - indică dacă tipul mesajului pentru cele 2 tipuri de mesaje, cerere sau raspuns. Pentru **cerere**, acesta poate fi **0**, **confirmable**, si asteapta un mesaj de acknowledge de la server, sau **1**, **non-confirmable**, care nu asteapta un mesaj de confirmare de la server. Pentru **raspuns,** acesta poate fi 2, insemand ca mesajul de acknowledge este confirmat, respectiv 3, **reset,** indicand faptul ca mesajul a fost receptionat dar nu a putut fi procesat.
* Message ID este reprezentat prin 16 biți utilizați pentru detecția mesajelor duplicate si pentru a lega mesajele de tip acknowledgement/reset de cele confirmable/non-confirmable.Atat mesajele de tip raspuns cat si cele de tip cerere vor avea acelasi Message ID
* Token - poate fi reprezentat printr-un număr de biți cuprins între 0 și 8, acest câmp este utilizat pentru a corela cererile cu răspunsurile, este generat de client.

`

**Requests**

A CoAP request consists of the method to be applied to the resource,

the identifier of the resource, a payload and Internet media type (if

any), and optional metadata about the request.

CoAP supports the basic methods of GET, POST, PUT, and DELETE, which

are easily mapped to HTTP.

A request is initiated by setting the Code field in the CoAP header

of a Confirmable or a Non-confirmable message to a Method Code and

including request information.

**Responses**

After receiving and interpreting a request, a server responds with a

CoAP response that is matched to the request by means of a client-

generated token; note that this is different from the

Message ID that matches a Confirmable message to its Acknowledgement.

A response is identified by the Code field in the CoAP header being

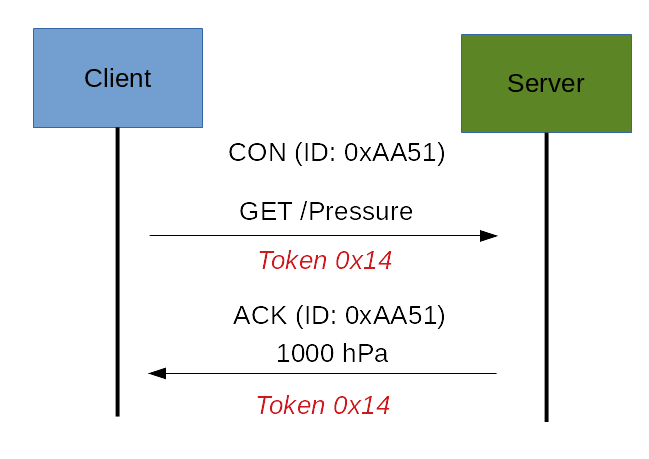
set to a Response Code. Similar to the HTTP Status Code, the CoAP

Response Code indicates the result of the attempt to understand and

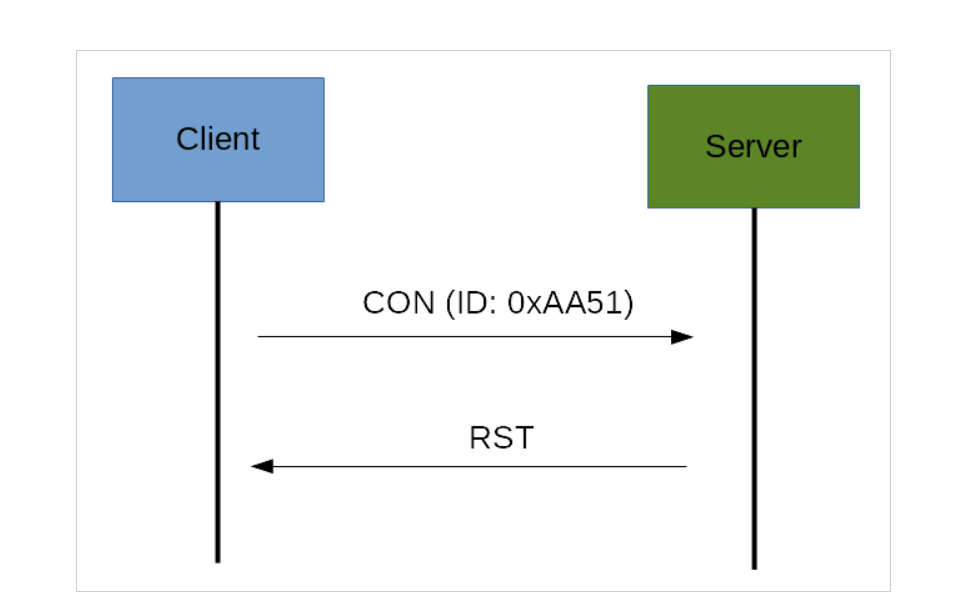
satisfy the request. The Response Code numbers to be set in the Code field of the

CoAP header are maintained in the CoAP Response Code Registry.

Model Cerere/Raspuns

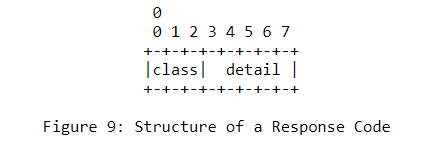


Daca cererea este trimisa utilizand confirmable message, serverul trimite inapoi un mesaj de acknowledge



Daca serverul intampina probleme la procesarea unei cereri, acesta trimite inapoi un mesaj de reset.

**Response Code**



The upper three bits of the 8-bit Response Code number define the

class of response. The lower five bits do not have any

categorization role; they give additional detail to the overall class.

There are 3 classes of Response Codes:

2 - Success: The request was successfully received, understood, and

accepted.

4 - Client Error: The request contains bad syntax or cannot be

fulfilled.

5 - Server Error: The server failed to fulfill an apparently valid

request.