International Journal of Engineering Research And Management (IJERM)
ISSN: 2349- 2058, Volume-03, Issue-07, July 2016

Analysis of Power Consumption for CoAP of IoT and HTTP

Ms. Shruti L. Paramoji, Dr. Yerriswamy T

***Abstract*— CoAP is standard protocol used for IOT, an IOT stands for internet of things and CoAP stands for constrained application protocol. CoAP is also called as the light weighted HTTP protocol. CoAP is having semantics as that of HTTP protocol, that are Get, Post, Put and Delete. At the application layer CoAP runs. These semantics are works same as that in HTTP protocol. CoAP is the rest-ful state transfer protocol means it is low powered. Our aim is to demonstrate the CoAP and HTTP. Measuring the power consumption of both protocols, calculates this power consumption using formula and analysis that which protocol consumes more energy.**

***Index Terms*— CoAP, HTTP, Power consumption, IOT**

I. INTRODUCTION

From the past few years and now (in these days) the usage of IOT is increased because of its various advantages. The IOT reduce the work tension of the human. IOT is the global network allows many devices to connect. The user and things are connected through the Internet. . Through tis internet user can send the request to things and get back the response from those things. Like this Internet provides the communication between both user and things. The whole system works using many protocols in that CoAP is the one.

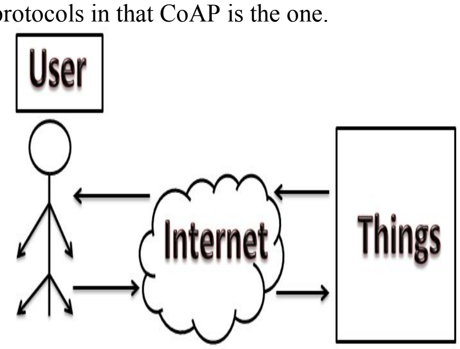

Figure: working of IOTs

The first section provides the information about CoAP. And the Next Session explained about the comparison of HTTP and CoAP. After next to this paper explains about analysis of CoAP and HTTP on measuring power consumptions.

Manuscript received July 13, 2016

Ms. Shruti L. Paramoji, Computer engineering, K.L.E Institute of Technology, Hubballi, Karnataka
Dr. Yerriswamy T, Electrical and Communications, K.L.E Institute of Technology, Hubballi, Karnataka

II. CoAP

This protocol is having the set of rules, these rules supports the IOTs. This protocol is having the similar client and server models as that of HTTP. The CoAP mainly lies on two layers that are lower layer is message layer and Request/Response layer. The application layer is the upper layer, provides the user interface layer. This layer is responsible for displaying the information to the user. The application layer is also referred as the abstraction layer that defines protocol and interface methods of interface used by end of the system. The Request/Response layer is responsible for communication of sending or receiving of request or response message. Message layer is implemented to deal with the user datagram protocol and it also performs the asynchronous exchange of message. This message layer model is having four types of messages that are Confirmable, Non-confirmable, Acknowledgment, and Reset.

A. Reliable message

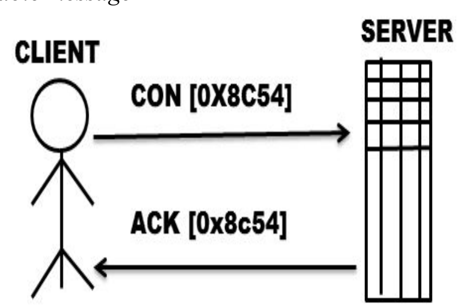

Figure: Reliable message transfer

In case of reliable message transportation the client can send the message (CON) until get an ACK with the message id same as the message sent (CON). It uses the some constant time out. The decreasing time of counting exponentially occurs in message transmission (CON). Suppose receiver fails to process the message then its reply will changes ACK to RST.

B. Unreliable message

In case of unreliable message transporting the client sending a message (NON). But it does not require ACK message instead it is having message ID for observing of resending. Suppose receiver fail to process the resending message then server responds reset message.

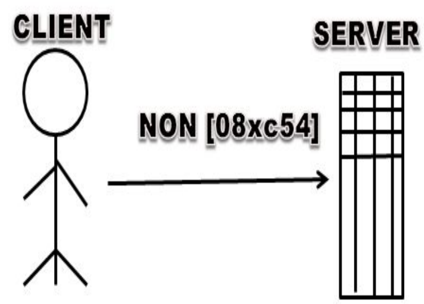

Figure: Unreliable message transfer

The request response layer model is containing Piggybacked Model, Separate Response Model, and Non-Confirmable Request/Reply Model.

C. Piggybacked

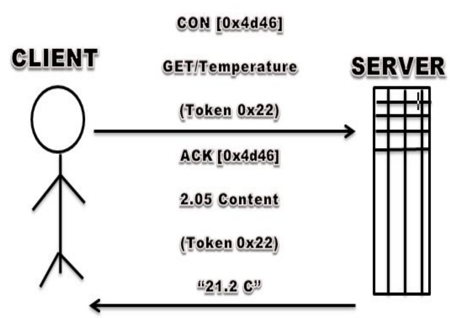

Figure: Successful reply of Get method

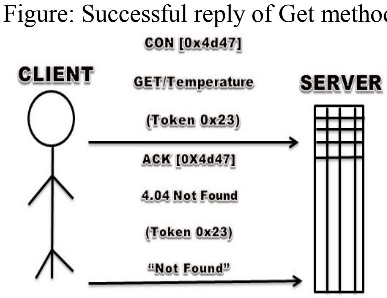

Figure: Unsuccessful reply of Get method

In case of piggybacked client sends the request message (CON or NON) the server receives the reply (ACK with CON) message. It is having two types of reply. For successful reply acknowledgement is having response code is detected by token. For failure reply acknowledgment is having failure response code (not found).

D. Separate response message

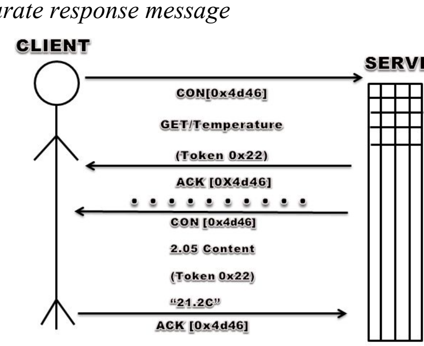

Figure: Get request with a Separate reply

In case of separate response the server receives the message (CON) and not capable of replay to the request. So it transfers ACK with empty if client again transfers the message. Suppose the server is ready to reply, it sends the new message of CON to client. Then client response to CON message with ACK. The CON message can be request/reply.

E. Non confirmable message

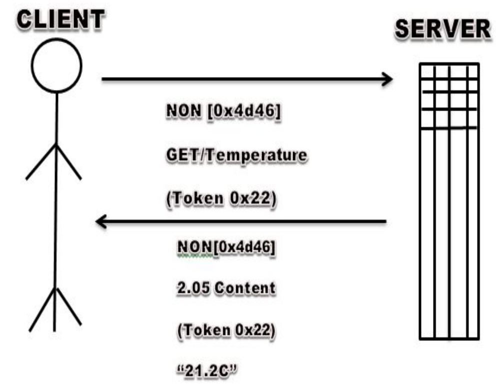

Figure: Nonconfirmable message transfer

In case of non-confirmable request and reply it is not as the piggy backed reply having message type CON, the client transfers NON type of message then server do not need to confirm, this server retransmits the message (NON) with reply.

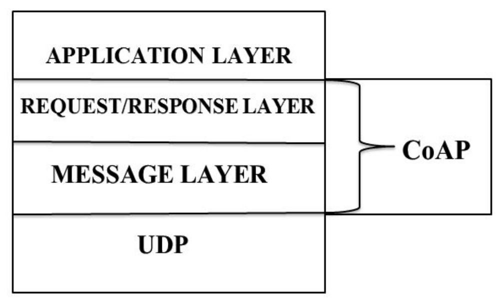

Figure: Architecture of CoAP

III. Comparison of CoAP and HTTP

CoAP is a network based protocol. It uses the same features as that of HTTP. CoAP uses the RESTful (Representation State Transfer) architecture. HTTP can also support the multicasting, low overhead. HTTP is the long term standards as compared with CoAP and can also allow small scripts to integrate variety of services and resources. HTTP is present at the level of application. HTTP is key point for IOT so this is using TCP for point to point communication. It is not suitable for constrained devices, and also not supports for push services notification. CoAP is employed on UDP. CoAP is implemented in such a way to access resources of internet. CoAP is optimized for datagram length to provide reliable communication. CoAP supports the GET, PUT, POST, and DELETE methods. CoAP supports IP based multicast that allows group communication of IOT. CoAP provides the resource discovery and retransmission mechanism in case unreliability of UDP.

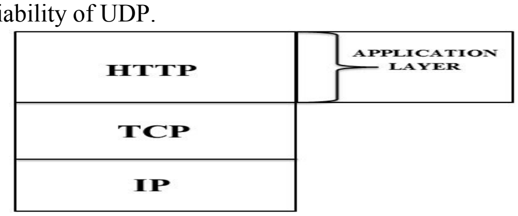

Figure: layers of HTTP

IV. MEASURING THE POWER CONSUMPTION OF COAP AND HTTP

For measuring power consumption of CoAP and HTTP we are conducting two experiments one for CoAP and another for HTTP. These experiments are conducted in Contiki OS and Cooja simulator. Cooja simulator is containing five different windows that are Network window, Simulation control, Notes, Mote output, Timeline. Network window is used to view the every nodes location of information. It is used for the information such as LED, mote ID and so on. The simulation control is having three options that are start, pause, reload, and step, these are used to execution of the example. In this step is used to execute the simulation example in step by step. And reload is used to reload the simulation example execution again from the beginning of execution. Note window is used for user to reference the user can write some important notes in this window for the later use. Mote output window is used for serial interface of motes. Simulation window is used to display the timeline of simulation. This simulation window is used to perform the events such as LED changes, channels changes, and log outputs so on. Other tools that are break point e. t. c. are also present in simulation window.

Contiki-2.7 can be easily to downloaded, in this 2.7 is version of Contiki OS. This Contiki-2.7 is having REST example. In this example we are going to add some functions to get the numerical values for measure the power consumption. That example is containing make file. If the WITH_COAP=1 is made in make file then that runs CoAP client and CoAP server of CoAP example. The server mote of this example provides numerical values, these values are used for calculation of power consumption on Transmit (TX), Receive (RX), Central Processing Unit (CPU), and Low Power Mode (LPM).

- TX this mode shows from the server nodes packets to transmit.
- RX this mode shows that packets transmitted from server nodes.
- CPU this mode occurs in server node that is containing the processing packets of CPU.
- LPM this mode shows that server node is in ideal.

The formula used for measuring power consumption is given below.

$$\text{Energy consumption} = \frac{(i1 - i2)*0.33*3}{32768/N}$$

Figure: power consumption formula

- i1, i2 = sequence number1 and sequence number2.
- 0.33 = constant values.
- 3 = constant values.
- 32768 = constant value (rtime) taken from number of ticks.
- N = it is a time added to source file for print profile in every n seconds.

To measure the energy consumption of CoAP server motes by taking IA time (Interarrival) 5 sec, 10 sec, and 15 sec. The simulation time 300 sec, 600 sec, 900 sec, 1200 sec is taken for every Interarrival time. The default port number in CoAP is 61616. The client access resources from the server periodical and prints the payload. All the values are taken from two motes.

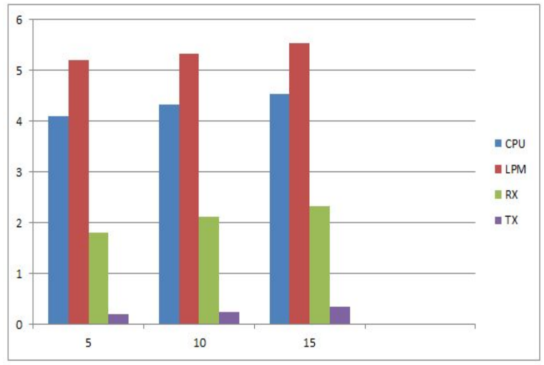

Figure: CoAP power consumption graph

To measure power consumption of example HTTP, WITH_COAP=0 is made in make file then that runs HTTP client and HTTP server of HTTP example. The default port number of HTTP is 8080. The server mote of this example provides numerical values, these values are calculated in four different operations that are explained above. The formula is used for measuring power consumption that is given above. The IA time and simulation time is taken same as that of CoAP experiment. That is 5 sec, 10 sec, and 15 sec of interarrival time. The simulation time 300 sec, 600 sec, 900 sec, 1200 sec. for every Interarrival time these simulation times are taken. This can also runs on two motes. The clients access the resources periodically from server and prints payloads. Mote provides the numerical output.

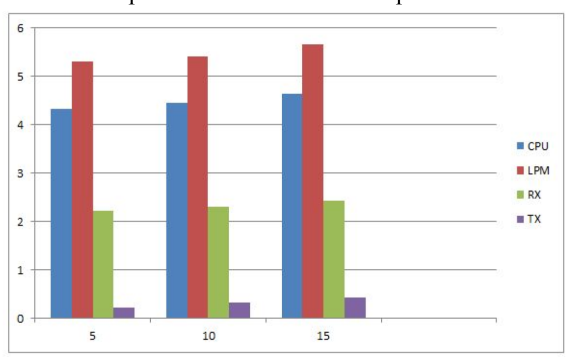

Figure: HTTP power consumption graph

The analysis graph of both CoAP and HTTP protocol for power consumption is given below. This graph displays that power consumption in HTTP is more as compared with CoAP.

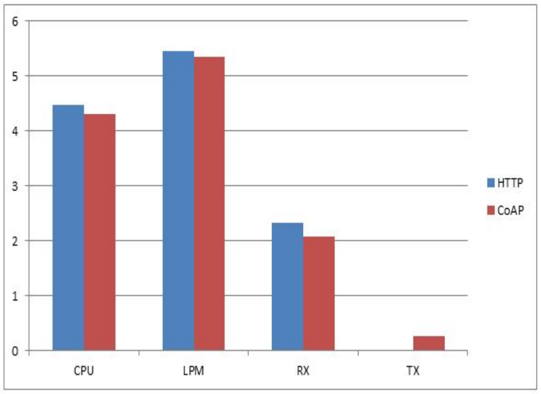

Figure: power consumption of both HTTP and CoAP

V. Conclusion

The CoAP and HTTP protocol both uses RESTful architecture so both are used to perform some similar characteristics. The CoAP consumes less power than HTTP protocol. So the response time taken in CoAP is less then HTTP protocol. For any request, CoAP response fast as compared with HTTP. CoAP is suitable for constraint devices, it also supports the IOTs.

References

[1] Z. Shelby, K. Hartke, C. Bormann. RFC7252. "The Constrained Application Protocol (CoAP)". June 2014.

[2] Glombitza, N, Pfisterer, D. Fischer, "LTP: An Efficient Web Service Transport Protocol for Resource Constrained Devices", June 2010

[3] Koojana Kuladinithi, Olaf Bergmann, Thomas Potsch Markus Becker, Carmelita Gorg, "Implementation of CoAP and its Application in Transport Logistics", April 2011.

[4] Maria Rita Palattella, Nicola Accettura, Xavier Vilajosana, Thomas Watteyne, Mischa Dohler," Standardized Protocol Stack for the Internet of Things", 2012.

[5] Castellani, A. P, "Architecture and Protocols for the Internet of Things: A Case Study", 2010.

[6] Guinard, D, "A Resource Oriented Architecture for the Web of Things", 2010