A Model to Evaluate the Performance of IoT Applications

Edward Guillén, Jeisson Sánchez, Carlos O. Ramos

***Abstract*— the main characteristics of IoT implementations is the distribution of elements such as sensors, IoT hubs and electronic devices within different communications solutions distributed around the world. The synchronization of the elements and controls is a new challenge for IoT approaches and depends on parameters such as latency, jitter and control protocols. This paper shows a model to evaluate these parameters on IoT applications with geographically distributed elements. A model to correct measure deviations is also presented.**

***Index Terms*— time-request sensor, IoT model, IoT devices,**

I. Introduction

New technologies in computer science as Internet of Things –IoT, Cloud Computing, Big Data, among others have contributed to the creation of solutions in different fields of society. One of the main technologies in engineering research is IoT. IoT is considered as the network of embedded electronic devices and software to collect data [1]–[4]. However, the main feature of IoT architecture is the information exchange between multiple set of nodes in different locations. IoT is widely used in multiple applications such as are home monitoring, Tele-HealthCare, smart cities control, smart devices, smart vehicles and smart grid [5]. In figure 1 a statistical graph on most common subjects published for IoT applications are presented.

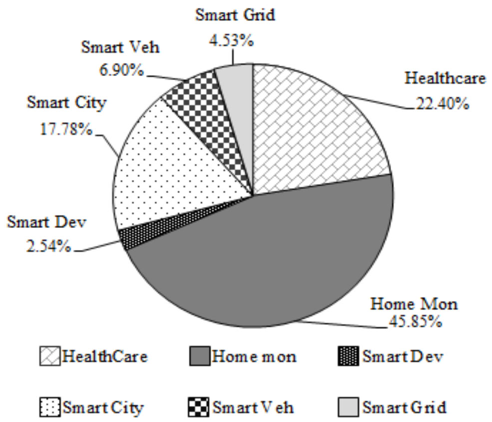

Fig. 1. The most common IoT applications from 2012

This work was supported by the Military University Nueva Granada with INV-ING 2114 Project with GISSIC research group.

E. Guillén, J. Sánchez and C. Ramos are with the Military University Nueva Granada, Cra 11 No. 101-80, Bogota, Colombia (e-mail: edward.guillen@unimilitar.edu.co,u1400881@unimilitar.edu.co, carlos.ramos@unimilitar.edu.co).

The design of an IoT Laboratory allows testing different devices and applications, within the resources analysis and its cost in real-word scenarios.

Different solutions with IoT systems are presented by Takpor et. al work [6]. Their work analyzes the student's performance on academic records based on their health condition. The study monitors university students with IoT devices as wearable and implement the Electronic Medical Records –EMR eHealth solution. Further with this solution other research projects with IoT architecture are focused on nature environment, algorithms to solve vehicular traffic problems, temperature control on industries and all the problems of the real world. However the solutions are not the unique feature with IoT-Cloud Computing systems. The efficiency of these systems is important to offer QoS and availability of the resources. Dakshayini and Guruprasad proposed a priority and admission control based on client service request [7]. The efficiency of provision is set up with a high throughput of cloud services. The main feature of the algorithm proposed is the highest precedence for paid user service-requests and cloud costs. Then, in 2016 Verma et. al. proposed an architecture to introduce a new policy for load balancing in Fog Computing [8]. Fog computing is one technology that only gives a response to the client in less time than Cloud Computing systems due to location. Real time streaming applications and IoT devices data communication is better for reliable and high speed transmission. The architecture has three layers to solve deadlines, execution time, data consistency and proper resource utilization. The load balancing is based on the required client resource between the layers. The algorithm reduces the bandwidth utilization compared to other algorithms.

In 2015, Capossele et. al. developed a IoT Laboratory to test serial communication factors as power transmission in a base station of radio, energy consumption and latency [9]. Additional Zhu et. al. analyzed the IoT relevance on Universities and School education for professional talent [10]. The main goal was to develop a laboratory to improve the knowledge and student's skills with IoT modules. IoT modules are based on the construction of different experiments in stages. As result of IoT laboratory model, multiple students will improve their knowledge on programming and electronics applied to different areas.

In this paper, we evaluate the IoT deployment costs based on computational time, physical and virtual resources, bandwidth, and protocol's behavior, among other features with the design of a scenario in an IoT platform. Finally to test multiple IoT electronic devices in the IoT laboratory is established with a global model to solve the requirements.

II. Materials and Methods

A. Proposed Algorithm

Initially, the model was designed to test simple parameters and the system overload. However, sensor response time for transmission phase is not fixed in test stages. In a majority of IoT networks, the performance is limited due to the number of sensor devices. Although different IoT devices have been tested in engineering laboratories or Industries, the computational costs and long time in processing are a challenge in general engineering.

The algorithm was programmed to consider multiple sensor features and different sensors. The sensor features are temperature, ultrasonic motion, current and other type of sensors. Nevertheless, the main feature of IoT algorithm is the scalability to study the operation of sensors in any application. For instance, the current sensor is configured as measure to control a Short Circuit –S/C in electronic systems. Also the current sensor could be configured as a counter measure of devices and optimizes the maximum and minimum capacity of the system. The ultrasonic motion sensor acquires features according to the stepper motor movement. The pseudo code of the proposed algorithm is presented below.

```
D: Distance
P: Path
S: Stop
R: Return
A: Advance
C: Current

Motion Sensor = active
Camera Sensor = active
Current Sensor = active

S = 0
C = 0
Máx D = 100 centimeter
if (P ≠ máx D)
  R ();
    D = 0 centimeter
    P = 0 centimeter
While(P=0 centimeter)
  if (P > máx D)
    error();
  elseif (P > D)
    P = P -D
    R (P value)
    C = 1
    S = 1
  else (P = D)
    A (P value)
    C = 1
    S = 1
close ()
```

The algorithm process is based on IoT architecture communication with a cloud broker. The proposed algorithm has a streaming module. The streaming server sends and receives data from camera, as video and photos. Streaming server provides services based on Real-Time Protocol –RTMP. The current sensor system is activated when the *C* flag has a value 1 in algorithm pseudo code. Also, the camera sensor record video when the stepper motor is in motion. The video control on streaming server determines the real-time response of IoT instructions messages.

B. Sensor Devices

Although the most common subjects published for IoT applications are home monitoring, smart vehicle and smart city, also there are healthcare applications. However, the main reason is test the algorithm with basic sensors in different areas. After the primary tests the algorithm is optimized to adjust any type of sensor and deploy different applications.

Current sensor verifies the hardware security, if the stepper motor is overload due to bad instructions or any object block up the path. Nevertheless, also it tests the excess operation from any electronic and mechanic element. Furthermore the temperature sensor and ultrasonic sensor is sending data at the same time with the current sensor. The temperature of the system is monitored with a solid state rheostat. The algorithm provides the control to work with the testbed laboratory multiple hours and different sensors.

The IoT algorithm architecture sends data to an IoT broker with time intervals. Time intervals are programmed according to the information of each sensor. A common value of ultrasonic sensor is distance. The distance only is published when the status of *D* changes. However the distance also is published during 5 minutes intervals time. While IoT broker receives messages on temperature topic per minute. Data is published in different devices and analytics modules process the information in statistical charts. The analysis is based on the condition of the sensors sending/receiving data. During the transmission different factors are studied such as the response time and delay of arriving messages.

III. Real Time Transport Protocol and Real Time Messaging Protocol

The protocol to control and analyze the response time in data transmission was the real time transport protocol –RTP. The algorithm was tested in different stages in remote location and local site. The basic operation of the test scenario was planned according to RFC-1889, *A Transport Protocol for Real-Time Applications* [11]. The control of data packet arrival is calculated by the interarrival jitter. The interarrival jitter is determined continuously with the equation 1. Where *Si* is the RTP timestamp from a *i* packet and *Ri* is the interarrival time in units of *i* packet. While *D* is the difference between RTP times from the *i* packet and *j* packet [11]-[12].

$$D(i, j) = (\mathrm{Rj} - \mathrm{Ri}) - (\mathrm{Sj} - \mathrm{Si}) = (\mathrm{Rj} - \mathrm{Sj}) - (\mathrm{Ri} - \mathrm{Si}) \qquad (1)$$

In order to test streaming video, the transmission was performed with Real Time Messaging Protocol –RTMP [13]. RTMP was selected testing a streaming transmission with an IoT camera. Open Source server provides the RTMP Url. The TCP port used is 5119. The connectivity test was developed in a programmable card with a simple camera of 1080p video.

IV. Test Application Scenarios

A. Ultrasonic and Temperature Scenario

The ultrasonic sensor is connected with the programmable card and installed over a platform geared with the stepper motor. While the stepper motor is moving on, the ultrasonic sensor is acquiring the distance during the time intervals. The sensor changes the values according to the instructions of stepper motor rotation. The maximum range of distance is 4 meters and the minimum range of distance is 2 cm for a correct operation. The sensor works with an echo and trigger pulse. The trigger starts with *10us* and will send out an 8-cycle burst of ultrasound.

The temperature sensor only tests the temperature of the stepper motor. In general the temperature is to control the operation and overload of the system. However, the system has additional devices that increase the temperature. To solve the problem of temperature, a fan was connected.

B. Current scenario

The current sensors are the security elements to close instructions. During the operation, the exceptions were programmed for mechanical errors. Additional the overload of the electronic system is interrupted. The current sensor has a range of 0-100A. Also has a work temperature range from 248 K to 343 K. The current of the system with the connected devices is 1300 mA approximately.

V. Analysis of Results

Initially the algorithm was adjusted only to test the ultrasonic sensor. Multiple data was collected from the motion sensor. Meanwhile the streaming server receipt video data from the camera. The camera is programmed with 1 seconds of time sleep between each frame. The video was captured during 10000 frames. RTMP timestamp was monitored with a protocol analyzer. The video streaming is configured to watch online in a webserver. The delay is 400 ms in average between the transmission and live video. Multiple sensors were activated during different stages of the operation. The overload increases the time delay when the algorithm is not configured. The main reason is the unnecessary transmission during death time of operation. By other hand the time request of the data packets is consolidated on the stop of current in stepper motor. The test was performed in remote location. In figure 2, response time, time delay and latency are presented according to video transmission and data communication. The response time is relative similar with the algorithm operation and normal operation. However, the time delay increase in normal operation due to multiple task to process in each sensor. Latency is processed based on the sum of delays average of video transmission. The latency values are in milliseconds.

Streaming transmission has 26 frames per second with 19500 KB of packet size approximately. The bitrate average was of 504 Kbits/second with a 1.02X transmission speed.

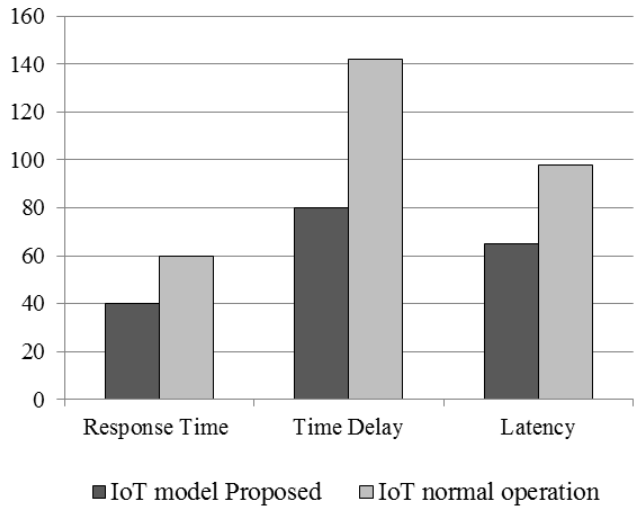

Fig. 2 Response time, time delay and latency in milliseconds

When the stepper motor begin the sequence and stop after the instructions, the coils remain the current and operation. That means the increasing temperature on stepper motor to generate a current overload. The sequence is changed to 1 value state per coil after the operation. The Stop instruction is the function to change the sequence. After this the algorithm decreases the current of the system in general.

Additional the time delay between local and remote location was tested with the IoT testbed laboratory. A high difference between the network connections increases the error average without proposed model.

The temperature was tested during the complete operation. The temperature decreases based on the stop sequence of the algorithm. In figure 3, the temperature is compared with the normal scenario and model proposed scenario. The temperature is in Kelvin scale. Additional the temperature was divided on 10 quarters to analyze the difference.

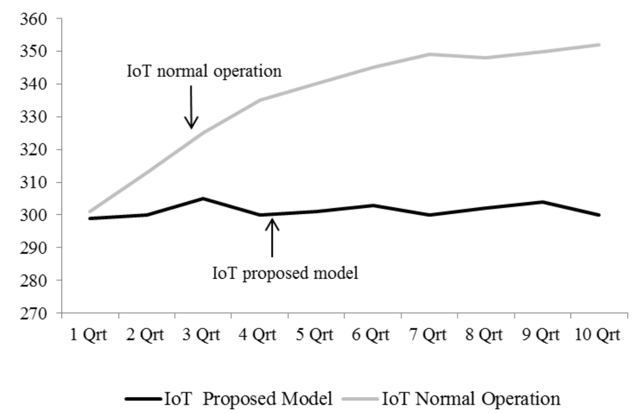

Fig. 3. Temperature Average on IoT proposed algorithm and normal operation

Hardware use is the minimum on together cases with and without proposed model. Nevertheless the energy consumption decreases with the model implementation. Also the virtual resources are optimized according to data storage on databases in virtual machines. The data is transmitted when the system is on operation; meanwhile the channel use is null.

ISBN: 978-988-14047-3-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

VI. Conclusion

The IoT model presented an increase of the optimal performance of different sensors connected with a programmable card. The optimization was according to time request on streaming, current consumption of the system and temperature control. The optimum operation during different hours validates the resistance of the proposed model. The transmission with the cloud storage makes the system robust to security vulnerabilities. Although the IoT testbed laboratory improves common integrating problems with the sensors and programmable card, only is the first stage of multiple tests.

By other hand the system was controlled in the same city as remote location. In future work different world locations are the approach to test these and other variables.

Acknowledgment

E. Guillén author thanks to Military University Nueva Granada for the labs used for the development of the project INV-ING-2114 with GISSIC research group.

References

[1] Dhumane, A., Prasad, R., & Prasad, J. (2016). Routing Issues in Internet of Things: A Survey. In Proceedings of the International MultiConference of Engineers and Computer Scientists (Vol. 1, pp. 16-18).

[2] Byun, J., Kim, S., Sa, J., Kim, S., Shin, Y. T., & Kim, J. B. (2016). Smart City Implementation Models Based on IoT Technology.

[3] Samad, T. (2016). Control Systems and the Internet of Things [Technical Activities]. IEEE Control Systems, 36(1), 13-16.

[4] Voas, J. (2016). Demystifying the Internet of Things. Computer, 49(6), 80-83.

[5] Lueth Lasse K. (2015). The 10 most popular Internet of Things applications right now [online]. Available on: https://iot-analytics.com/10-internet-of-things-applications/

[6] Takpor, T. O., & Atayero, A. A. (2015). Integrating internet of things and ehealth solutions for students' healthcare. In Proceedings of the World Congress on Engineering (Vol. 1).

[7] Dakshayini, D. M., & Guruprasad, D. H. (2011). An optimal model for priority based service scheduling policy for cloud computing environment. International Journal of Computer Applications, 32(9), 23-29.

[8] Verma, M., Bhardwaj, N., & Yadav, A. K. (2016). Real Time Efficient Scheduling Algorithm for Load Balancing in Fog Computing Environment.

[9] Capossele, A., & Spenza, D. (2015). Internet of Things Laboratory.

[10] ZHU, K., JIANG, W., LIU, T., & ZHANG, Y. Y. (2013). Research of IOT practice and innovation laboratory. Internet of Things Technologies, 7, 027.

[11] Audio-Video Transport Working Group, Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, DOI 10.17487/RFC1889, January 1996.

[12] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003.

[13] Parmar, H., & Thornburgh, M. (2012). Adobe's Real Time Messaging Protocol. Copyright Adobe Systems Incorporated, 1-52.