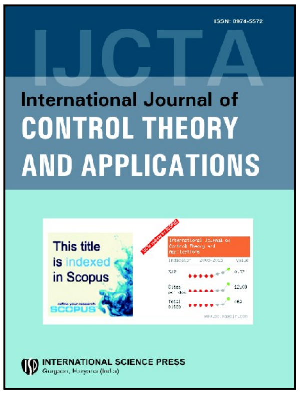

International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 9 • Number 49 • 2016

Satisfying the Latency Requirements of IoT Systems

Ashok Kolli, Vijaya Perumalla, Martin K M and B. Seetha Ramanjaneyulu

Dept. of Electronics and Communication Engineering, VFSTR University, India-522213, E-mail: ashok.kolli13@gmail.com

Abstract: An Internet of Things (IoT) system integrates the information coming from heterogeneous sensor devices, to other devices through Internet. Transmitting the data within the deadline is important in some IoT systems that deal with critical monitoring and control applications. This paper proposes a Message Scheduling method, based on Short Processing Time for improving the latency values of the devices. It makes use of Message Broker architecture. Assigning priorities to each Broker messages and scheduling them with proposed algorithm makes the given IoT system to have improved latency values and energy-efficient while offering message stability also, in IoT environment. Here sensor devices are divided into IoT sub units to overcome the limitation of the number of direct links to a single working element. Each IoT sub unit in the system has a Broker that acts as information exchange center.

Keywords: Internet of Things (IoT), Quality of Service (QoS), Service Oriented Architecture (SOA) and Short Processing Time (SPT).

1. INTRODUCTION

Internet of things is defined as "Things having identities and virtual personalities operating in smart spaces using intelligent interfaces to connect and communicate within social, environmental, and user contexts". IoT introduces the concept of linking all physical things in an environment that is based on Internet for communication [1]-[2]. Service Oriented Architecture (SOA) provides an architectural approach that enables the deployment and management of real-world services running on physical devices. Satisfying the Quality of Service (QoS) requirement is one of the dominant and challenging factors that need to be addressed in order to achieve the designed goals of IoT system. The role of WSNs is to collect the information from surrounding environment. The concept of it is discussed in [1], where WSNs are part of an Internet of Things and different integration approaches to Internet through the gateways are illustrated. The system where the IoT devices are directly connected to the server as described in two-tier architecture, is efficient when the system has few sensor nodes. In the three-tier architecture, the sensor devices are connected to server through the gateway [4]. It prevents the number of direct links to single working element.

In Fig. 1, the sensor devices are connected to the server through the intermediate device called broker. The broker in the IoT system acts as message exchanger. Here the IoT system is divided into IoT sub units. Each sub unit comprises of a broker and sensor devices. Every message broker in the IoT sub unit collects the sensed environment information from their sensor nodes and sends it to the server. Queuing technique is applied to calculate the traffic intensity of the messages and accordingly adjust the messages in the order.

Latency is one of the important QoS parameters in IoT systems, especially when the data needs to be delivered to the destination within the stipulated time. Improving the latency parameter of IoT devices is considered in this work. Related works are discussed in Section-2. Architecture of the system for implementing the proposed mechanism is presented in Section-3. It presents the proposed scheduling mechanism, along with the discussion of the importance of scheduling in latency improvement. Section-4 is on the performance analysis of the proposed system. Section-5 concludes the paper.

2. RELATED WORK

The IoT introduces "The concept of linking all physical things in an environment based on Internet for communication among them. The authors in [2] discussed about the requirements and challenges in IoT. The authors in [3] and [5] discussed about the QoS aware IoT layer architecture. In the above, the IoT system was divided into three layers such as sensing, networking and application layers respectively and designed layer wise.

Sourav Kumar Dhar et al. [6] proposed interference aware scheduling of Sensors in the IoT enabled health-care monitoring system. This system reserves a slot of time to every message and if the message does not fit in between the time slots of other sensor messages, then the information must be divided into several parts to nullify the inference among the sensors.

Wajia Naveed et al. [7] proposed a model for QoS in IoT, in which the messages from the sensors are classified into low and high priorities and use the push-out buffer scheme to pause the low priority messages when high priority message arrives. The authors in [9] propose a QoS aware algorithm for link scheduling in wireless networks. They make some enhancements on the distributed matching algorithm to improve the QoS aspects like Delay and Throughput performance. The authors in [8] proposed a scheduling for IoT-enabled waste collection in smart cities. [10], [11] discuss about the improvement in service rate of IoT devices, by manipulating the message queues and implementing intermediaries. They classify the messages coming from sensor devices to broker as Low, Normal and High priority messages and scheduling each type of messages to get the Message stability in IoT system. This approach is considered in this work also.

3. PROPOSED WORK

(A) System Architecture

Fig. 1 shows the architecture considered for the proposed scheduling of the system, that comprises of IoT subunits. Here sensor devices are divided into IoT subunits to overcome the limitation of the number of direct links to a single working element. Each IoT sub unit in the system has a broker that acts as an information exchange center. In the system, the messages from sensor nodes are classified into High Priority (HP), Normal Priority (NM) and Low Priority (LP) types. The broker in each IoT sub unit maintains three queues for HP, NM and LP messages respectively. In the traditional two-tier architecture, multiple sensor devices/ things connected to a single server provide multiple direct links to a single working element. To prevent this problem we divide the system into small units which are called IoT subunits based on clustering-based approach. Here each subunit comprises of message broker and sensor nodes. Every time one node in the IoT subunit acts as broker, it acts like a message exchanger. The broker is responsible for collecting sensed data from the sensing devices in the subunit. Each subunit broker buffers the requests from the clients and sends the responses to it from the sensor nodes. For IoT system, we adopt the SOA concept to make the system flexible and extendable [14].

In each IoT subunit, the sensor nodes send heterogeneous messages through broker to the base station. These messages can have different attributes like expiry time, size, priorities etc. Priorities of messages are considered in this work to decide on further processing of them. So, the messages of sensor devices having higher priority are given preference by broker while forwarding them to clients.

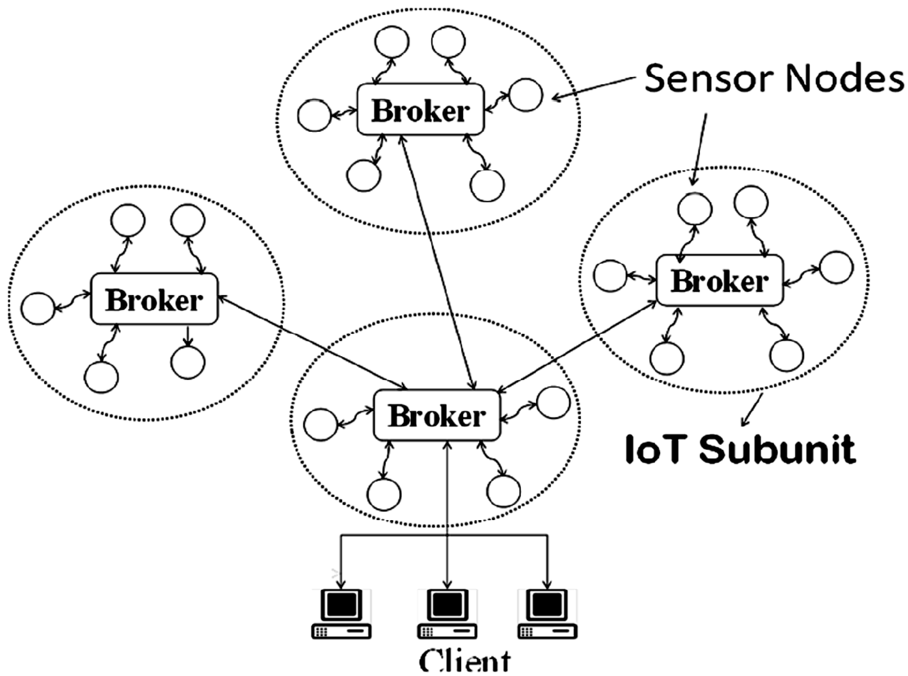

Figure 1: Architecture of the IoT System Considered

In this work, messages from the different sensors are classified into three types based upon the priorities of applications that make use of these sensors. The class having real time applications is named as High Priority (HP). The Normal Priority (NM) and Low Priority (LP) classes are for non-critical and non-emergency applications. After classifying the incoming messages from sensor nodes, the three classes HP, NM and LP are served based on the preemptive scheduling following the SPT rule. By doing so, the waiting time of the each Message also gets reduced, thereby making the system energy-efficient also.

(B) Message Scheduling Mechanism

In practical IoT systems, ‘n’ sensor nodes which are distributed randomly forms a group called as IoT subunit. The ‘n’ different sensor nodes are sending ‘m’ different types of messages to the designated broker in its subunit. The sensor nodes in each subunit have the same energy to send messages to the broker. In the IoT system considered, each subunit sensor nodes send different types of messages to client through the broker. Each message from the sensor node are to be delivered within deadline, otherwise it may lose its importance [4]-[5]. The uncertainty of network traffic from sensor nodes to broker makes it difficult to adjust the messages. We use an M/M/1 queue to model arrivals and services of all the heterogeneous messages from the sensor nodes to the broker in each IoT subunit. Each subunit broker is responsible for collecting the messages from the sensor nodes. The sensor nodes are sending different messages at different rates. Each broker in the subunit has three queues corresponding to three types of messages HP, NM and LP respectively. The proposed scheduling runs in each broker.

If the service rate and arrival rate for n^{th} message for n={1,2,...,m }are represented by μ_n and λ_n respectively, then the traffic intensity of each message is obtained by

$$\rho_n = \frac{\lambda_n}{\mu_n} \tag{1}$$

The average waiting time of each message in their respective class of priority queue is taken as an indication of system effectiveness. The total traffic intensity from class 1 priority to class 'm' priority is

$$\rho = \sigma = \Sigma_{n=1}^{m} \frac{\lambda_n}{\mu_n}$$

$$\Sigma_{n=1}^{m} \frac{Str_n}{Treq_n},\ \rho < 1 \tag{2}$$

Here we are dealing with 3 classes, so m=3. Now the waiting times corresponding to these three classes are:

$$W_Q(1) = \frac{\Sigma_{n=1}^{3} \frac{\lambda_n}{\mu_n}}{(1-\sigma_1)} \tag{3}$$

$$W_Q(2) = \frac{\Sigma_{n=1}^{3} \frac{\lambda_n}{\mu_n}}{(1-\sigma_1)(1-\sigma_2)}$$

$$= \frac{WQ(1)}{(1-\sigma_2)} > W_Q\ (1) \tag{4}$$

$$W_Q(3) = \frac{\Sigma_{n=1}^{3} \frac{\lambda_n}{\mu_n}}{(1-\sigma_1)(1-\sigma_2)(1-\sigma_3)}$$

$$= \frac{WQ(2)}{(1-\sigma_3)} > W_Q \tag{5}$$

So from the deduction of (3), (4) and (5), we get

$$W_Q\ (1) < W_Q\ (2) < W_Q\ (3) \tag{6}$$

From (6), we get that the waiting time for the low priority queue is more than the waiting time for high priority queue. It indicates that a lower priority results in longer waiting time in the queue. In this paper, we represent HP, NM and LP as high, normal and low priority queues.

For every message from a sensor node to a broker, there is a successful transmission time and service request period time. Service time is defined as the successful transmission time taken by the broker to send the message to a nearby broker in a multihop way. The request period time is the rate at which the messages are coming into the broker queue. Priorities are assigned to messages which are coming from sensor nodes to broker, to be served in the broker queue. By analyzing the traffic intensity of every class message we can calculate waiting time.

Message Scheduling Algorithm

The messages are coming from sensor nodes/devices. These are the information like smoke, humidity, temperature etc. By properly ignoring the unimportant data that are coming from the sensor nodes improves the overall performance of the message transmission in IoT system [6]-[10]. In the IoT subunit, the 'n' different sensor nodes are sending 'n' heterogeneous messages to the broker. Each message from sensor node represented as $M_n(Str_n, Treq_n, P_n)$ with successful transmission time of the request (Str_n), requesting time period $(Treq_n)$ and priority (P_n). The arrival rate and the service rate of the message are represented as

$$\lambda_n = \frac{1}{Treq_n} \; and \; \mu_n = \frac{1}{Str_n} \tag{7}$$

The traffic intensity of message is represented as

$$\rho_n = \frac{\lambda_n}{\mu_n} = \frac{Str_n}{Treq_n} \tag{8}$$

In the proposed scheduling algorithm, the three queues at broker level revives three types of messages and are placed in their respective queue. Next check the traffic intensity of each class message, if the traffic intensity is greater than or equal to 1, then requesting periods of the messages must be scheduled and altered by proposed scheduling algorithm until the traffic intensity of the system is less than 1. In the proposed algorithm, initially we assume that the minimum requesting period is represented as the minimum requesting period of all type of devices at system startup. They should support the Requesting period greater than or equal to minimum requesting period of device. The time of request period will progressively increase by the ratio of minimum request period. The traffic intensity of each type of message will be recalculated by the new value of the requesting period, and the property set of all types of messages is rearranged according to the new traffic intensity value. Then, the check on the overall traffic intensity will be made again and so on. By the time loop ends, system traffic intensity will become < 1 making the system stable. It is described below.

Algorithm: Proposed Message Scheduling for QoS in each IoT sub unit

---

1. The n^{th} device Message property $M_n(Str_n, Treq_n, P_n)$
2. if P_n=1
3. M_n moves in P_1 queue
4. elseif P_n=2
5. M_n moves in P_2 queue
6. elseif P_n=3
7. M_n moves in P_3 queue
8. end
9. $Treq_n$= $Rmin_n$
10. if M_n in P_1 queue ñ>1
11. Go to loop Label
12. elseif M_n in P_2 queue ñ>1

13. Go to loop Label
14. elseif M_n in P_3 queue ñ>1
15. Go to loop Label
16. end
17. Loop :
18. **while** $\rho > 1$
18. Rearrange M_n in descendant $Treq_n$ order
19. for $k = 1$ to m
20. $Treq_k = Treq_k + \frac{R\min_k}{2^k}$
21. $\rho_k = \frac{Str_k}{Treq_k}$
22. end
23. $\rho = \Sigma_{n=1}^{m} \frac{Str_n}{Treq_n}$
24. end

The proposed scheduling algorithm reduces waiting time of messages to get its turn to service. The sensor nodes in the IoT system are battery powered. When every message from sensor node to broker, it dissipates some energy. If the broker is busy with providing service to another device, the device will have to wait until response from the broker occurs. This waiting time affects the battery life of the sensor or device. So the proposed scheduling algorithm improves the battery life also as it reduces the waiting time of message in the system.

4. SIMULATION SCENARIO

(A) IoT service response time

In the proposed IoT system, the broker in each subunit receives the requests from the client and sends responses that are coming from the sensor nodes/devices. Each broker maintains three types of queues to serve the three different priority messages HP, NM, and LP. For simulating this scenario, Matlab-Simulink software is used. The service time of the IoT system without scheduling is shown in Fig-2. As can be observed, the service time increases when more messages are coming. Here we are showing service time of IoT system for 30 messages and the waiting times of messages are increasing with message count or time. The average service time for 30 seconds without scheduling is reaching approximately 10 units as shown in Fig. 2.

After scheduling of each message as per the proposed method, the average waiting time is reduced considerably as shown in Fig. 3. The average service time found in 30 seconds is approximately 0.7 units.

The proposed scheduling algorithm decreases the average waiting time of the messages that are coming from the sensor nodes to IoT client through the broker. So the less waiting time of the messages in the system increases system effectiveness and also reduces the dissipation energy of the sensor devices to transfer the messages.

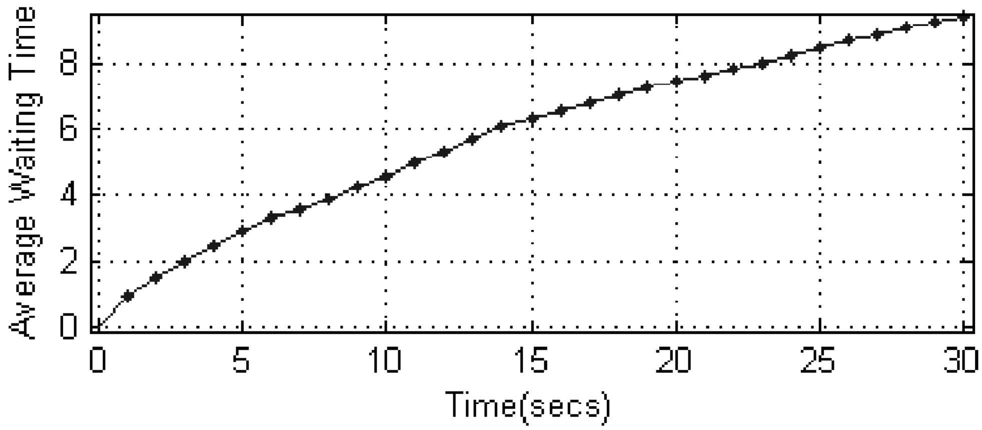

Figure 2: Latency values without scheduling

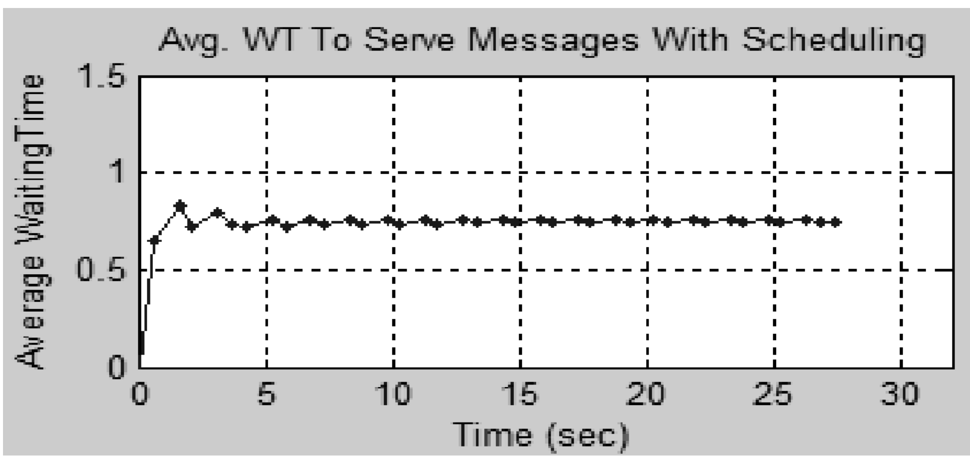

Figure 3: Latency values with scheduling

(B) Different types of messages Average Service Time analysis

In our case, we divided the messages in three types HP, NM, and LP respectively based on priorities. The high priority message waits for less time in the system. The lower waiting time of high priority messages in the system maintains the higher service rate. Fig-4 shows the service time of the different class messages. The

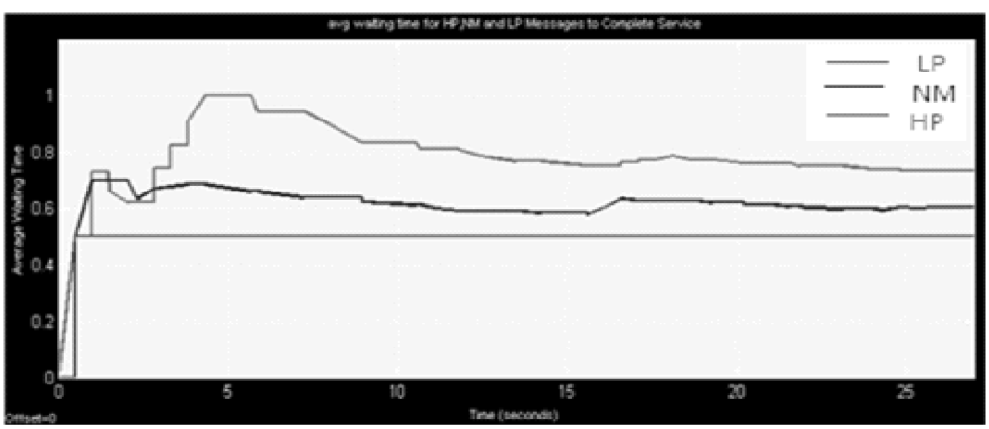

Figure 4: Waiting Times for Messages of Different Priorities

average waiting time for 30 seconds goes down as shown in the figure. It is 0.8 units for low priority messages and 0.5 units for high-priority messages.

5. CONCLUSION

A Message Scheduling algorithm that improves the latency values of IoT system is presented in this work. It consists of brokers to collect messages from the sensor nodes of different priorities and transmit them to the destination. The proposed scheduler maintains three queues for three classes of messages coming from the sensor nodes. We use the priority queue model to rearrange the messages. In our proposed scheduling algorithm, it evaluates the traffic intensity to determine the order of requests by following the SPT rule. The simulation results show that high-priority messages are transmitted with least delays and the low-priority messages suffer higher delays. Improvement in the overall latency of the system is also observed.

REFERENCES

[1] Delphine Christin, Andreas Reinhardt, Parag S. Mogre, R. Steinmetz. “Wireless Sensor Networks and the Internet of Things: Selected Challenges”, 8th GI/ITG KuVS Fachgesprch “Drahtlose Sensornetze”, 2009. pp. 31-33.

[2] L. D. Xu, W. He and S. Li, “Internet of Things in Industries: A Survey,” in IEEE Transactions on Industrial Informatics, vol. 10, no. 4, pp. 2233-2243, Nov. 2014.

[3] R. Duan, X. Chen and T. Xing, “A QoS Architecture for IOT,” 2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing, Dalian, 2011, pp. 717-720.

[4] A. Gluhak, S. Krco, M. Nati, D. Pfisterer, N. Mitton and T. Razafindralambo, “A survey on facilities for experimental internet of things research,” in IEEE Communications Magazine, vol. 49, no. 11, pp. 58-67, November 2011.

[5] L. Li, S. Li and S. Zhao, “QoS-Aware Scheduling of Services-Oriented Internet of Things,” in IEEE Transactions on Industrial Informatics, vol. 10, no. 2, pp. 1497-1505, May 2014.

[6] S. K. Dhar, S. S. Bhunia and N. Mukherjee, “Interference Aware Scheduling of Sensors in IoT Enabled Health-Care Monitoring System,” 2014 Fourth International Conference of Emerging Applications of Information Technology, Kolkata, 2014, pp. 152-157.

[7] I. Awan, M. Younas and W. Naveed, “Modelling QoS in IoT Applications,” 2014 17th International Conference on Network-Based Information Systems, Salerno, 2014, pp. 99-105.

[8] T. Anagnostopoulos, A. Zaslavsy, A. Medvedev and S. Khoruzhnicov, “Top — k Query Based Dynamic Scheduling for IoT-enabled Smart City Waste Collection,” 2015 16th IEEE International Conference on Mobile Data Management, Pittsburgh, PA, 2015, pp. 50-55.

[9] A. Ghiasian and H. Saidi, “QoS aware distributed matching algorithm for link scheduling in wireless networks,” 2009 International Conference on Ultra Modern Telecommunications & Workshops, St. Petersburg, 2009, pp. 1-6.

[10] S. Abdullah, K. Yang “An Energy-efficient Message Scheduling Algorithm in Internet of Things Environment”. 9th International Wireless Communications and Mobile Computing Conference (IWCMC), 2013, pp: 311-316.

[11] Jenq-ShiouLeu, Chi-Feng Chen, and Kun-Che Hsu “Improving Heterogeneous SOA-Based IoT Message Stability by Shortest Processing Time Scheduling” Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan.

[12] Dr. JánosSztrik “Basic Queueing Theory” University of Debrecen, Faculty of Informatics. János Sztrik, “Basic Queueing Theory”, GlobeEdit, OmniScriptum GmbH & Co, KG, Saarbrucken, Germany (2016), ISBN 978-3-639-73471-3.

[13] D. Gross, J. F. Shortle, J. .M. Thompson, C. M. Harris, “Fundamentals of queuing theory”, 4th Edition, Wiley, 2013.

[14] S. Abdullah and K. Yang, “A QoS aware message scheduling algorithm in Internet of Things environment,” 2013 IEEE Online Conference on Green Communications (Online Green Comm), Piscataway, NJ, 2013, pp. 175-180.