Optimal Gateway Selection in Sensor-Cloud Framework for Health Monitoring

Sudip Misra*, Samaresh Bera*, Ayan Mondal*, Reena Tirkey*,
Han-Chieh Chao†, and Samiran Chattopadhyay‡
* School of Information Technology, Indian Institute of Technology, Kharagpur, 721302, India,
Email: {smisra, samaresh, ayanmondal, reenat}@sit.iitkgp.ernet.in
† Department of Computer Science and Information Engineering, National Ilan University,
I-Lan, Taiwan, Republic of China, Email: hcc@niu.edu.tw
‡ Department of Information Technology, Jadavpur University, Salt Lake, Kolkata, India,
Email: samirancju@gmail.com

Abstract

Sensor-cloud computing is envisioned to be one of the key enabling technologies for remote health monitoring. Integration of sensed-data into cloud applications in sensor-cloud will help in real-time monitoring of patients over geographically distributed locations. In this paper, we study the optimal gateway selection problem in sensor-cloud framework for real-time patient monitoring system by using a zero-sum game model. In the proposed model, a gateway acts as the first player, and chooses the strategy based on the available bandwidth, while a user request acts as the second player, and follows the strategy chosen by the first player. We evaluate the execution time for selecting the optimal gateway through which the sensed-data will be fetched to the cloud platform. In addition, we show how user requests are serviced by the gateways to access data from cloud platform optimally. We also show that by using the proposed approach, the execution time decreases, thereby helping in forming a reliable, efficient, and real-time architecture for health monitoring.

Index Terms

Sensor network, Cloud Computing, Sensor-cloud, Zero-sum game, Health monitoring

I. Introduction

A sensor-cloud architecture conceptually integrates cloud infrastructure with sensor networks, thereby enabling real-time monitoring of data-intensive applications (such as health-care) that are typically spread over geographically distributed locations [1]. Sensor networks are popularly used for deploying health-related applications such as monitoring patients with blood sugar, blood pressure, and sleep-activity pattern monitoring [2], [3]. In such application, the health center takes necessary decisions according to the sensed data from patients. It is a difficult task to monitor the health-status remotely, when a patient moves randomly. So, an efficient computing mechanism is necessary to monitor the health-status of patients when they are mobile. The data-intensive, time-varying requirements of the sensor networks can benefit from the intricate integration of the computational and storage resources offered by the cloud computing applications for big-data processing [4]. Thus, sensor-cloud platforms are increasingly become popular in health care.

Standalone sensor networks have some inherent challenges [2], [5], [6], which can be undertaken by sensor-cloud infrastructures.

1) *Data management:* One of the major problems of sensor networks is data management. Storing and accessing sensed-data from sensor networks is efficient with sensor-cloud.
2) *Resource utilization:* The cloud computing infrastructure provides resource sharing facility among the users. This information sharing mechanism can be implemented around the sensor network environments, and thus, same data can be accessible from geographically distributed regions at the same time.
3) *High utility cost:* Some analysis such as detecting faulty sensors, removing erroneous readings, and fusing noisy measurements from several sensors is challenging and expensive. The sensor-cloud infrastructure is a cost-effective approach, where the existing cloud platform can be used.

A sensor-cloud architecture is conceptualized with a combination of sensor networks and cloud applications [5], [7]. Due to the limitations in sensor network such as limited bandwidth, storage, and memory, sensed-data from the sensor nodes are processed through a cloud infrastructure in order to have reliable, cost-effective and real-time monitoring facilities. Cloud applications are able to collect and process huge data that are generated from sensor nodes through the gateways on the both sides — cloud gateways and sensor gateways. In such a publish/subscribe system, sensor-cloud provides a platform for the execution of services that operate on sensed data and also satisfying and ensuring the derived trust and security requirements. A sensor-cloud enables users to easily gather, access, process, and search for a large number of data stored in the cloud infrastructure using the computational and storage devices [8]. The dynamic behaviors of sensor-cloud infrastructure facilitate automatic furnishing of its services as required by the users [9], [10]. In this paper, we design a framework for optimal-gateway selection in an efficient, reliable, and cost-effective manner in a sensor-cloud, that will help for health monitoring applications.

The rest of the paper is organized as follows. In Section II, we briefly present the literature review related to this work. Section III presents the system model for the sensor-cloud framework. We propose a solution for optimal-gateway selection in Section IV. Some results are discussed in Section V. Finally, we conclude the paper with few research directions in Section VI.

II. Related Work

Several works are done in the past years for health monitoring using sensor-cloud infrastructure [9], [11]–[20]. In the existing health monitoring systems prior to sensor-cloud, patients' health-data are collected by the workers (such as nurse, doctors) manually. This manual system leads to a cost-expensive, error-prone, and increased latency, which are unsuitable for real-time monitoring.

To address these issues, the authors in [9] discussed a solution for cost-effective, real-time health monitoring system. In such a scenario, the health information is fed into a cloud environment, so that information is available from anywhere to the patients and also to the health-care center. Thus, manual data collection issues have been addressed by the authors for real-time, always-on, and automated health monitoring systems. However, in such a solution model, delay-optimized gateway selection mechanism is required for mapping sensed-data with cloud applications.

In Ref. [11], a pub-sub based model is proposed by the authors in order to integrate cloud applications with sensor network. The authors represented publish and subscribe model as pub-sub model. According to the authors, users register their information to the *SaaS* application through the publish model. In the subscribe model, appropriate subscribers are found for each application using event matching algorithm. In such a model, published data is available through the existing Web-services, and is accessible only to the subscribed customers. They used pub-sub broker to utilize the cloud infrastructure as *SaaS* to monitor, process, and deliver the events to the users. However, this model is not autonomous. Periodic predictions are required to be calculated for the purpose of cost-optimization.

In Ref. [14], the authors proposed a fast and flexible information dissemination system for automatic publish/subscribe mechanism. In such a system, class-group index matching algorithm is proposed for minimizing delay to receive subscribed content, and predictions are automatically calculated. However, performance degradation takes place when the number of customers is increased.

Aoki et al. [15] proposed a deep sensor-data aggregation mechanism for reducing congestion in network traffic for fast response real-time data collection. The authors focus on the issues related to cloud applications such as latency, limited bandwidth, long distance transmission for real-world implementations using WAN architecture. To address these issues, they divide WAN architecture into two parts — *front-side* and *back-side*. The *back-side* is the same as today's network architecture, and *front-side* is located between the WAN and real-world part for handling real-world data streams.

In [16], the authors propose a secure multicast sensor-cloud application based on the combination of group-key and time-key (CoGKTK) mechanisms instead of broadcast mechanism. The group-key (GK) is used for the group of users that satisfy the data arriving from the sensor-cloud. The time-key (TK) is used for optimization of the key updating from joining or leaving of users. The authors show that using multicast method, it is possible to reduce the computation and response time.

The authors in [17] proposed a hybrid framework for monitoring patient health status using a sensor-cloud. They demonstrated the benefits of using sensor-cloud architecture for patient health status monitoring.

The authors in [21]–[23] studied the sleep-wake cycle of the patients in a nursing home. The authors proposed a sleep activity pattern monitoring (SAPM) system for real-time health monitoring in a local nursing home. They also implemented a cost-effective circadian rhythm monitoring system and tested the mechanism on trial data collected from a nursing home. Biswas et al. in [21] also claimed that their SAPM system works well for real-time health monitoring.

In [24], the authors proposed a cost-effective, versatile clinical tool for mobile voice monitoring to track the daily condition of vocal system. They use wearable sensor for data collection, and smart phone application platform for storing the real-time data. Using the smart phone application and the wearable sensors, the authors proposed that the treatment of hyper-functional vocal disorders can be effectively enhanced for providing real-time feedback to facilitate the healthy vocal function.

It can be inferred from the existing works that most of the existing works addressed the issues related to mapping sensor network with cloud applications for scalability and availability of the real-time monitoring system. However, still, an optimal gateway selection mechanism is needed to be addressed to establish an efficient, reliable, and cost-effective monitoring system and to minimize the service-delay in a dynamic sensor-cloud relationship.

III. System Model

A sensor-cloud uses the publish/subscribe (pub/sub) model for receiving data from sensor network, and for distributing that data to the interested cloud applications. All these intermediate operations are done by the pub/sub broker that act as an intermediary entity between the sensor network and the cloud platform. We address the execution delay problem for optimal gateway selection in sensor-cloud framework for the real-time monitoring systems.

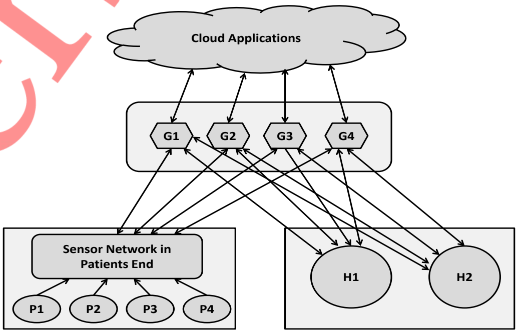


Fig. 1: A schematic view of a health monitoring system using sensor-cloud applications

In Figure 1, a health monitoring system using sensor-cloud application is shown. In the Figure 1, $G1$, $G2$, $G3$, and $G4$ are the gateways between the sensor network and the cloud platform. $P1$, $P2$, $P3$, and $P4$ represents the patients in different areas. The sensed-data from the sensor network are fed into the cloud environment with the help of the gateways. All the health-care centers, $H1, H2, H3$, and $H4$, can access the real-time health-data cloud through the gateways, and according to that, take decisions.

Let, N_G be the set of gateways and N_R be the set of requests from the users. So, our objective is to choose an optimal gateway $G_i \in N_G$ for each $R_i \in N_R$, where, $i \in N$, (N is the set of natural numbers) such that all the requests, N_R, are mapped according to their priorities. After selecting a gateway, G_i, it is included into a candidate gateway set, C_i, for the request, R_i, and considered as the candidate gateway for optimal gateway selection strategy. There may be more than one candidate gateway for a request, R_i. In such a scenario, we also consider the relative bandwidth, i.e., after selecting a gateway, G_i, for a request, R_i, the bandwidth $B_{G(i)}$ of the gateway is reduced during the time interval for serving the request.

A gateway, $G_i \in N_G$, is selected for any request, R_i, as an optimal gateway. After serving the request, R_i, the gateway bandwidth, $B_{G(i)}$, is reduced by a constant value, say c_x. So, the new relative bandwidth, $B_{G(i)} - c_x$, is considered to serve another request, R_j. Our aim is to choose the optimal one from the set of candidate gateways, C_i, for transferring the user requested data to the sensor-cloud framework. For example, suppose we have two requests, R_1 and R_2 , and two gateways, G_1 and G_2. Request R_1 can be served by either G_1 or G_2, and request R_2 can be served only by G_1. Now, let us consider that G_1 has higher bandwidth than G_2. In such condition, R_1 is served first as it comes first and served by the gateway, G_1, as it has higher bandwidth. After serving the first request, gateway G_1 does not have enough bandwidth to serve the request, R_2. So, in this circumstance, a proper gateway selection mechanism is needed to be addressed to serve all the requests optimally.

IV. Solution Approach

A. Game Formulation

For transferring the user requested data to the sensor-cloud, we use a two-player zero-sum game [25]–[27] for optimal gateway selection for the reduction of delay. A game is said to be zero-sum, if for any outcome, the sum of the pay-offs to all players is zero, and can be expressed as:

$$\sum_i U_i = 0 \qquad i = 1, 2, ..., N. \tag{1}$$

In such a zero-sum game model, the first player employs the strategy under the feedback information, while the second player utilizes the strategy under the information structure of the first player. The problem is to serve the user requests using the appropriate gateway in terms of bandwidth to reduce the delay for serving requests. In such a scenario, we consider the two-player game between the gateway and user-request. In such a game model, the gateway acts as a leader and the user-request acts as the follower. Now, if the strategies of two the players are the same, the value of the pay-off functions for the players are identical. The first player, i.e., the gateway chooses one strategy, and then the second player, i.e., the user, chooses the strategies over first player's strategy.

Definition 1. *Let, Z be a game defined by $(N, (A_i), (U_i))$, where N is the number of players, A_i is the set of pure strategies of player i, and U_i is the corresponding pay-off for the player i. Then Z is called a zero-sum game iff [28]:*

$$\sum_{i=1}^{n} U_i(\vec{a}) = 0, \forall \vec{a} \in A \tag{2}$$

In our two-player zero-sum game model, $N = 2$. So, Equation (2) may be written as:

$$U_1(a_1, a_2) = -U_2(a_1, a_2), \forall a_1 \in A_1, a_2 \in A_2 \tag{3}$$

B. Nash Equilibrium

In the proposed model, we show that there exists Nash Equilibrium, and also it follows the properties below [28]:

1) All players exhibit the same value of pay-off at Nash Equilibrium points.
2) For a two point Nash Equilibrium, it is possible to replace the strategy of one player in the first point by the strategy of the same player at another point.

This can be formally written, as shown in Theorem 1.

Theorem 1. *Let, Z be the two-player zero-sum game defined by $((A_1, A_2), \pi)$. Let, (τ_1, τ_2) and (σ_1, σ_2) be the two Nash Equilibrium points for the game, Z. Then*

1) $\pi(\tau_1.\tau_2) = \pi(\tau_1, \sigma_2) = \pi(\sigma_1, \tau_2) = \pi(\sigma_1, \sigma_2)$
2) *Both (σ_1, τ_2) and (τ_1, σ_2) are Nash Equilibria of Z.*

Proof (Part I): (σ_1, σ_2) is a Nash Equilibrium. Therefore, the first player, the gateway, plays for maximizing the pay-off, and, thus, we have

$$\pi(\sigma_1, \sigma_2) \geq \pi(\tau_1, \sigma_2) \tag{4}$$

We have also another Nash Equilibrium point as well, i.e., (τ_1, τ_2). For this Nash Equilibrium point, the second player, user request, plays for minimizing the pay-off, and, thus, we have

$$\pi(\tau_1, \sigma_2) \geq \pi(\tau_1, \tau_2) \tag{5}$$

After combining Equations (4) and (5), we get

$$\pi(\sigma_1, \sigma_2) \geq \pi(\tau_1, \sigma_2) \geq \pi(\tau_1, \tau_2) \tag{6}$$

TABLE I: Zero-sum game model

	Gateway	
Request	(0, 0)	(1, -1)
	(-1, 1)	(0, 0)

Similarly, we get the inequalities

$$\pi(\sigma_1, \sigma_2) \leq \pi(\tau_1, \sigma_2) \leq \pi(\tau_1, \tau_2) \tag{7}$$

Finally, from Equations (6) and (7), we can prove that

$$\pi(\tau_1 . \tau_2) = \pi(\tau_1, \sigma_2) = \pi(\sigma_1, \tau_2) = \pi(\sigma_1, \sigma_2) \tag{8}$$

Proof (Part II): We observe that (σ_1, σ_2) is a Nash Equilibrium point of the two-player game. From Equation (8), we have

$$\pi(\alpha_1^{'}, \sigma_2) \leq \pi(\sigma_1, \sigma_2) = \pi(\tau_1, \sigma_2), \qquad \forall \alpha_1^{'} \in A_1 \tag{9}$$

and

$$\pi(\tau_1, \alpha_2^{'}) \geq \pi(\tau_1, \tau_2) = \pi(\tau_1, \sigma_2), \qquad \forall \alpha_2^{'} \in A_2 \tag{10}$$

The combination of Equations (9) and (10) implies that (τ_1, σ_2) is a Nash Equilibrium as well.

In the same way, we can prove that (σ_1, τ_2) is also a Nash Equilibrium.

In Table I, we show the zero-sum game model, where the gateway and the request are the first and the second players, respectively. The gateway chooses the strategy based on its available bandwidth to maximize the pay-off to 1. With the chosen strategy of the gateway, the request follows it to get the service. Thus, the players, gateway and request, maximize their pay-off by choosing (1, -1) strategy.

At the beginning of the game, we consider the pay-off for gateways and requests as zero. In our proposed approach, all the gateways try to maximize their pay-off from 0 to 1. At the same time, all the requests try to get service from the gateways as soon as possible in order to reduce the delay. Thus, we consider the pay-off for the serviced request as -1. After servicing all the requests, the sum of the pay-off of requests and gateways is zero, and thus, follows the zero-sum game approach, i.e.,

$$\sum_{t=0}^{T} U_G + \sum_{t=0}^{T} U_R = 0 \tag{11}$$

where, U_G and U_R are the utilities of gateway and request, respectively.

*1) **Utility of gateways**:* We consider the gateway utility function as $U_G(S, B_{G(i)}, M, \alpha, \beta)$, where,

(i) $B_{G(i)}$ is the gateway bandwidth value.

(ii) M is the $(N_G \times N_R)$ matrix.

(iii) α and β are the decision variables of the two players.

So, for the first player (i.e., the gateway) the strategy is as follows:

$$max[U_G(B_{G(i)(0)}, M_0, \alpha, \beta)] = \sum_{t=0}^{T-1} U_{G(t)}(B_{G(i)(t)}, M_t, \alpha_t, \beta_t) + U_{g(T)}(S_T, B_{G(i)(T)}, M_T, \alpha_T, \beta_T) \quad (12)$$

*2) **Utility of user requests**:* We consider the utility function of the second player (i.e., the user request) to be denoted as $U_R(r_p, M, \alpha, \beta)$, where,

(i) r_p denotes the request priority.

(ii) M is the $N_G \times N_R$ matrix. In the matrix M, the matrix value is 1, if the corresponding gateway and request status is unity.

(iii) α and β are the decision variables of the two players.

For the user request, the utility function is defined as follows:

$$max[U_R(r_{p(0)}, M_0, \alpha_0, \beta_0)] = -\sum_{t=0}^{T-1} U_{R(t)}(r_{p(t)}, M_t, \alpha_t, \beta_t) + U_{M(T)}(r_{p(T)}, M_T, \alpha_T, \beta_T) \quad (13)$$

Algorithm 1 Algorithm for optimal gateway selection in sensor cloud

Inputs: Number of gateways (N_G), user request set (N_R), request ID ($R_{i(id)}$), request status (0 or 1), request priority ($R_{priority}$).

Output: Optimal gateway, $N_{G(opt)}$, for request ID, $R_{i(id)}$

```
 1: if (request value $\geq priority_{threshold}$) then
 2:     Count the number of high priority requests
 3:     Create status matrix based on the status of requests and gateways
 4:     Check which gateway value satisfies the range for each request value
 5:     Using utility factor make a matrix to check which gateway satisfy which requests
 6:     Count the number of gateways, which can satisfy the requirement of the requests
 7:     if the number of satisfied gateway $\geq 2$ then
 8:         Find optimal-gateway comparing the relative bandwidth of the gateways.
 9:     else
10:         Serve the request ($R_i$) using that gateway
11:     end if
12:     After serving each request, decrease the selected gateway bandwidth value, $B_{G(i)} - c_x$.
13:     Repeat steps $3 - 10$ for low-priority requests.
14: end if
```

TABLE II: Simulation Parameters

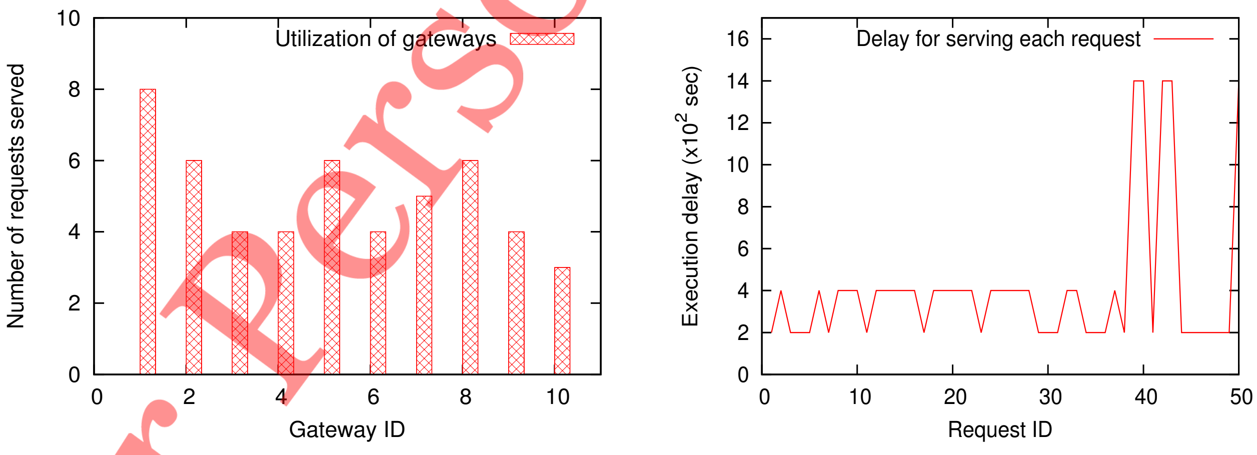
Parameter	**Value**
Maximum number of requests at a time	2000
Maximum number of gateways	50
Request priority	0 or 1
Gateway bandwidth	$\leq$ 40 unit
Requested bandwidth	$\leq$ 15 unit

C. *Algorithm for optimal gateway selection*

Our proposed algorithm follows the 'first come first serve' approach with priority-based service. The algorithm finds out the high priority requests among the active ones, and serves those requests that come first. We consider that the proposed framework supports multiple gateways for each request. When the gateway receives user requested data to serve, the system first checks the gateway value to ascertain whether the gateway satisfies the request or not, instead of storing the data into cloud storage first. If the gateway does not satisfy the gateway value, then the data is stored in the cloud storage to minimize the delay. Then, the optimal gateway is selected for serving the request.

V. Results and Discussions

We used MATLAB to simulate the proposed solution. Table II shows the different parameters used for simulation. We evaluated the performance of the proposed solution in three ways: (a) utilization of gateway, (b) delay for serving the requests, and (c) overall comparison of gateway utilization and service delay.



(a) Utilization of gateways

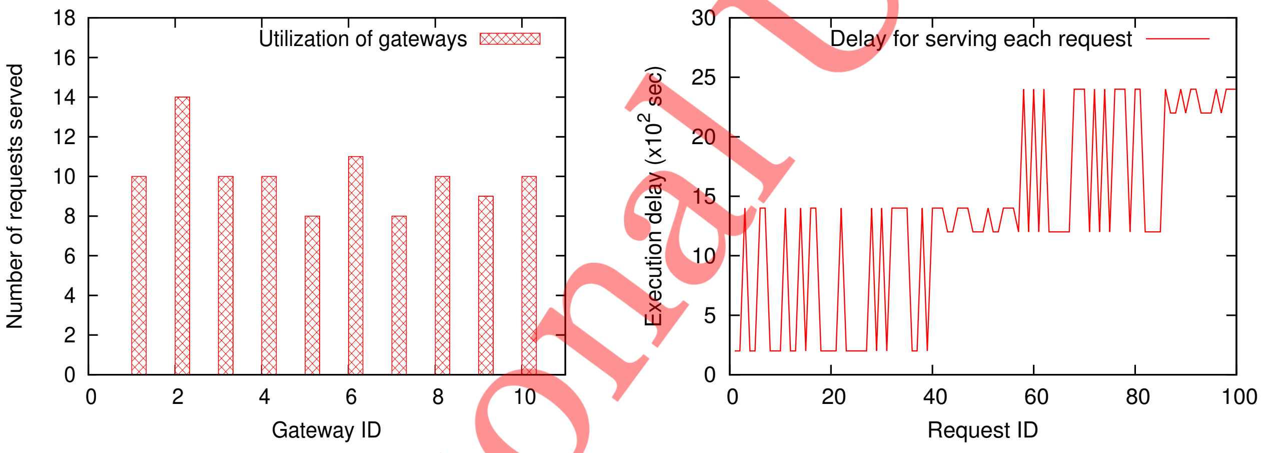
(b) Requests served with delay

Fig. 2: 50 requests are served by 10 gateways

We calculate computational delay for each requests which are serviced by the gateways. We observe that the high priority requests are serviced first, according to the proposed framework. Among the first priority requests, they are serviced on a ‘first come first serve’ basis. We calculated the delay for each request ID, when 10 gateways are available. Here, all the requests are serviced with optimal delay.

In Figure 2(a), we show the number of requests served by the gateways, when 50 requests are served by 10 gateways. We observe that all the gateways are optimally utilized according to the number of requests. In Figure 2(b), we show the results of the service delay while 50 requests are served by 10 gateways. All the high priority requests are serviced first, and, thus, delay is lower than the low priority requests.

In Figure 3(a), the utilization of gateways is shown for different gateway ID with the number of requests served, when we consider 100 requests and 10 gateways are available for serving the requests. We observe that all the gateways are optimally utilized to serve the requests, according to their available bandwidth. We also observed the changes in service delay while 10 gateways are available for 100 requests, as shown in Figure 3(b). We see that the delay increases compared to that in Figure 2(b).



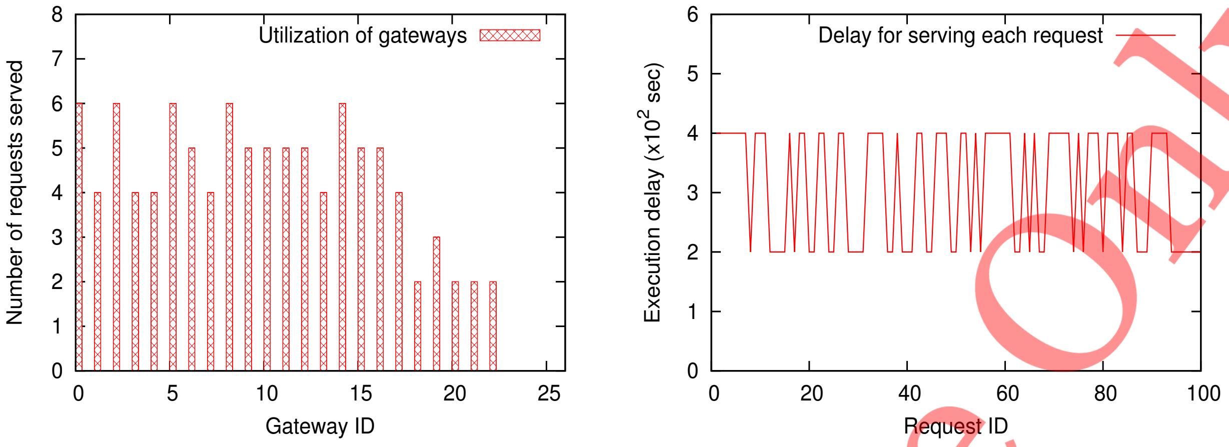
(a) Utilization of gateways

(b) Requests served with delay

Fig. 3: 100 requests are served by 10 gateways

Figure 4(a) presents the utilization of gateways for 100 requests, when 25 gateways are considered to serve the requests. In this case as well, we observe that the gateways are utilized optimally with an increase in the number of gateways in comparison with Figure 3(a), according to the available bandwidth. In Figure 4(a), we see that since there are no remaining requests to be served, some of the gateways (such as $23, 24, 25$) are not used. In Figure 4(b), we show the waiting time for each request to be served.

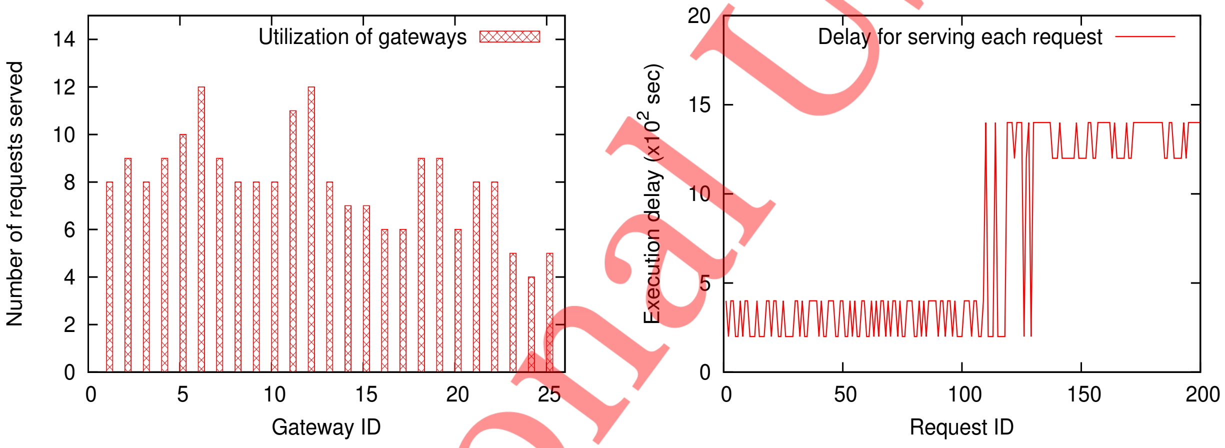
Figure 5(a) shows the utilization of gateways when 25 of these are considered for serving 200 requests. We observe that all the gateways are utilized for serving the requests unlike as seen in Figure 4(a). In Figure 5(b), with an increase in the number of requests and gateways, we calculated the delay for each request wait time to get service from the gateways. In Figure 5(b), we show the results, when 200 user requests are serviced by 25 gateways. We observed that with an increase in the number of gateways, delay is reduced compared to that in



(a) Utilization of gateways

(b) Requests served with delay

Fig. 4: 100 requests are served by 25 gateways



(a) Utilization of gateways

(b) Requests served with delay

Fig. 5: 200 requests are served by 25 gateways

Figure 3(b), where 100 requests are serviced by 10 gateways.

Overall comparison: Finally, we compared the number of requests served with fixed number of gateways in Figure 6, and the delay for each request to get the service in Figure 7.

We compared the number of requests served by each gateway in our proposed algorithm, while the total number of gateways is fixed (see Figure 6). We observe that each individual gateway serves more number of requests with an increase in the total number of requests. Each gateway serves requests optimally to reduce the delay.

In Figure 7, we show the results of comparison when each request is serviced by different number of gateways having different waiting time. Here, we can also observe that with an increase in the gateway number, the waiting time is reduced optimally, as shown in Figure 7.

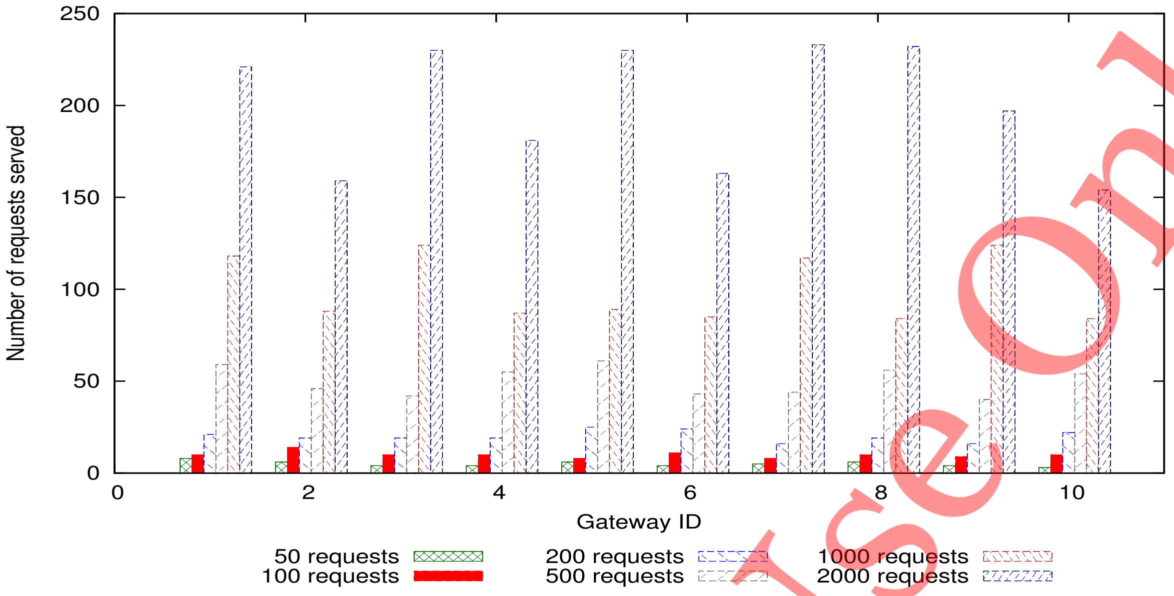


Fig. 6: Comparison of gateway utilization with the number of requests served

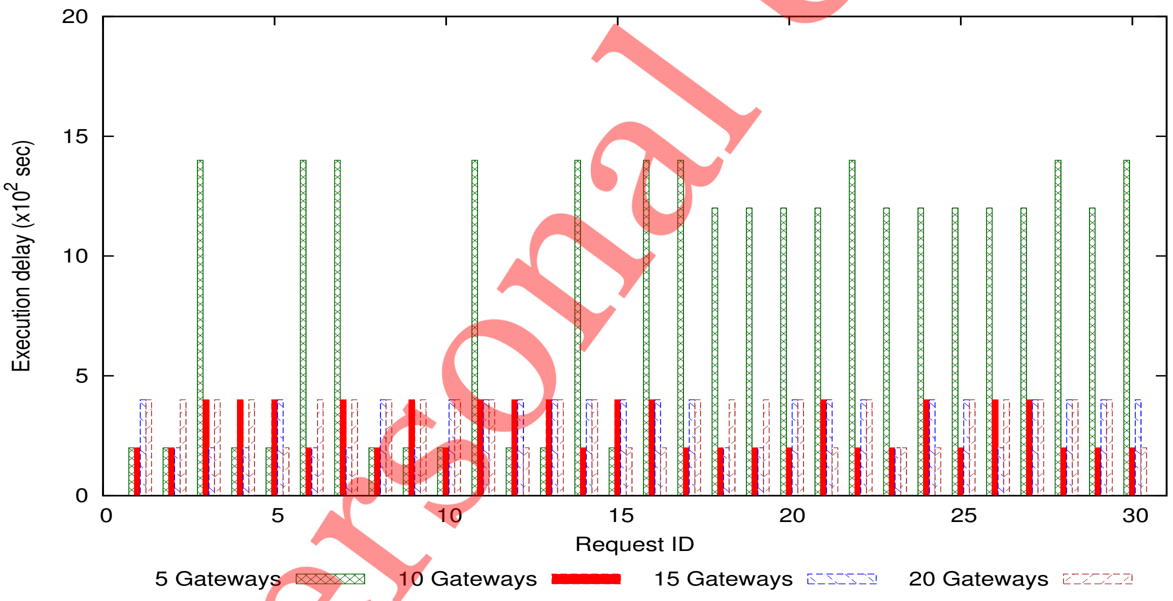


Fig. 7: Comparison of delay with the number of gateways

VI. Conclusion and Future Work

In this paper, we proposed a solution to the optimal gateway selection problem in sensor-cloud. We modeled the problem as a delay optimization problem in the sensor-cloud architecture. Based on the gateway selection problem, we have shown how the user requests can be mapped to the optimal gateway and serviced through the sensor-cloud environment. We observed that our proposed framework works well for delay optimization. We considered request priorities for gateway selection, i.e., all the high priority requests are served first. The future extension of this work includes how the requests can be serviced by the gateways more effectively in order to have a reliable, and efficient, real-time monitoring system.

References

[1] D. Irwin, P. Shenoy, E. Cecchet, and M. Zink, “Resource management in data-intensive clouds: Opportunities and challenges,” in *Proceedings of IEEE Workshop on Local and Metropolitan Area Networks (LANMAN)*, Long Branch, NJ, May 2010, pp. 1 – 6.

[2] A. Alamri, W. S. Ansari, M. M. Hassan, M. S. Hossain, A. Alelaiwi, and M. A. Hossain, “A Survey on Sensor-Cloud: Architecture, Applications, and Approaches,” *International Journal of Distributed Sensor Networks*, vol. 2013, pp. 1–18, 2013.

[3] S. Misra, V. Tiwari, and M. S. Obaidat, “LACAS: Learning Automata-Based Congestion Avoidance Scheme for Healthcare Wireless Sensor Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 4, pp. 466 – 479, May 2009.

[4] P. V. Krishna, S. Misra, D. Joshi, and M. S. Obaidat, “Learning Automata Based Sentiment Analysis for Recommeder System on Cloud,” in *Proceedings of the 2013 International Conference on Computer, Information and Telecommunication Systems (CITS)*, Piraeus-Athens, Greece, May 2013.

[5] M. Yuriyama and T. Kushida, “Sensor-Cloud Infrastructure - Physical Sensor Management with Virtualized Sensors on Cloud Computing,” in *Proceedings of IEEE International Conference on Network-Based Information Systems (NBiS)*, Takayama, Sept. 2010, pp. 1–8.

[6] U. Varshney, “Pervasive healthcare and wireless health monitoring,” *Mobile Networks and Applications*, vol. 12, no. 2-3, pp. 113–127, 2007.

[7] S. K. Dash, J. P. Sahoo, S. Mohapatra, and S. P. Pati, “Sensor-Cloud: Assimilation of Wireless Sensor Network and the Cloud,” *Advances in Computer Science and Information Technology. Networks and Communications*, vol. 84, pp. 455–464, 2012.

[8] K. T. Lan, “Whats Next? Sensor+Cloud?” in *Proceeding of the ACM International Workshop on Data Management for Sensor Networks*, 2010, pp. 978 – 971.

[9] C. O. Rolim, F. L. Koch, C. B. Westphall, J. Werner, A. Fracalossi, and G. S. Salvador, “A Cloud Computing Solution for Patients Data Collection in Health Care Institutions,” in *Proceedings of IEEE International Conference on eHealth, Telemedicine, and Social Medicine, ETELEMED*, 2010, pp. 95–99.

[10] A. Mondal and S. Misra, “Dynamic Coalition Formation in a Smart Grid: A Game Theoretic Approach,” in *Proceedings of the* 3^{rd} *IEEE International Workshop on Smart Communication Protocols and Algorithms (SCPA), colocated with IEEE ICC 2013*, Budapest, Hungary, June 2013, pp. 1–5.

[11] M. M. Hassan, B. Song, and E.-N. Huh, “A framework of sensor-cloud integration opportunities and challenges,” in *Proceedings of International Conference on Ubiquitous Information Management and Communication (ICUIMC)*, 2009, pp. 618–626.

[12] Y.-S. Yen, H.-C. Chang, and R.-S. Chang, “Routing with adaptive path and limited flooding for mobile ad hoc networks,” in *Proceedings of International Symposium on Wireless Pervasive Computing*, Jan. 2006.

[13] W.-M. Chen, C.-S. Li, F.-Y. Chiang, and H.-C. Chao, “Jumping ant routing algorithm for sensor networks,” *Computer Communications*, vol. 30, no. 14-15, pp. 2892–2903, Oct. 2007.

[14] M. M. Hassan, B. Song, and E.-N. Huh, “A dynamic and fast event matching algorithm for a content-based publish/subscribe information dissemination system in Sensor-Grid,” *Journal of Supercomputing*, vol. 54, no. 3, pp. 330–365, 2010.

[15] H. Aoki, H. Shikano, M. Okuno, Y. Ogata, H. Miyamoto, Y. Tsushima, T. Yazaki, and S. Nishimura, “Cloud Architecture for Tight Interaction with the Real World and Deep Sensor-Data Aggregation Mechanism,” in *Proceedings of IEEE International Conference on Software, Telecommunication and Computer Networks (SoftCOM)*, 2010, pp. 280–284.

[16] T.-D. Nguyen and E.-N. Huh, “An efficient key management for secure multicast in Sensor-Cloud,” in *Proceedings of International Conference on Computers, Networks, Systems, and Industrial Engineering (ACIS/JNU)*, 2011, pp. 1–9.

[17] S. Mohapatra and K. S. Rekha, “Sensor-Cloud: A Hybrid Framework for Remote Patient Monitoring,” *International Journal of Computer Applications*, vol. 55, no. 2, pp. 1–11, October 2012.

[18] C.-F. Lai, Y.-M. Huang, J. H. Park, and H.-C. Chao, “Adaptive Body Posture Analysis for Elderly-Falling Detection with Multisensors,” *IEEE Intelligent Systems*, vol. 25, no. 2, pp. 20 – 30, March-April 2010.

[19] C.-F. Lai, S.-Y. Chang, H.-C. Chao, and Y.-M. Huang, “Detection of Cognitive Injured Body Region Using Multiple Triaxial Accelerometers for Elderly Falling,” *IEEE Sensors Journal*, vol. 11, no. 3, pp. 763 – 770, March 2011.

[20] Y.-S. Yen, H.-C. Chao, R.-S. Chang, and A. Vasilakos, “Flooding-limited and multi-constrained QoS multicast routing based on the genetic algorithm for MANETs,” *An International Journal on Mathematical and Computer Modelling*, vol. 53, no. 11-12, pp. 2238–2250, June 2011.

[21] J. Biswas, J. Maniyeri, K. Gopalakrishnan, L. Shue, P. Eugene, H. Palit, F. Y. Siang, L. L. Seng, and L. Xiaorong, “Processing of wearable sensor data on the cloud - a step towards scaling of continuous monitoring of health and well-being,” in *Proceedings of IEEE Annual International Conference on Engineering in Medicine and Biology Society (EMBC)*, Buenos Aires, Aug. - Sept. 2010, pp. 3860 – 3863.

[22] J. Biswas, J. Maniyeri, S. Louis, and P. Yap, “Fast Matching of Sensor Data with Manual Observations,” in *Proceedings of IEEE Annual International Conference on Engineering in Medicine and Biology Society, (EMBC)*, Minneapolis, MN, Sept. 2009, pp. 1675 – 1678.

[23] J. Biswas, M. Jayachandran, L. Shue, K. Gopalakrishnan, and P. Yap, “Design and Trial Deployment of a Practical Sleep Activity Pattern Monitoring System,” in *Proceedings of International Conference on Smart Homes and Health Telematics, (ICOST)*, Tours, France, June-July 2009.

[24] D. Mehta, M. Zaartu, S. Feng, H. Cheyne, and R. Hillman, “Mobile Voice Health Monitoring Using a Wearable Accelerometer Sensor and a Smartphone Platform,” *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 11, pp. 3090 – 3096, Nov. 2012.

[25] X.-Y. Zhang, “A note on dynamic zero-sum games,” in *Proceedings of Chinese Conference on Control and Decision (CCDC)*, 2009, pp. 4758 – 4760.

[26] Z. Fang, Y. Liu, S. Liu, and H. Chen, “Functional model for zero-sum game and its matrix solution concerning uncertain information,” in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Seoul, Oct. 2012, pp. 17 – 23.

[27] S. B. Akers, “Threshold logic and two-person, zero-sum games,” in *Proceedings of the Second Annual Symposium on Switching Circuit Theory and Logical Design (SWCT)*, Detroit, MI, USA, Oct. 1961, pp. 27 – 33.

[28] G. Owen, *Game Theory*. Academic Press, New York, 1982.