Adaptive Data Caching for Provisioning Sensors-As-A-Service

Subarna Chatterjee, *Student Member*, *IEEE*, Sudip Misra, *Senior Member*, *IEEE*
School of Information Technology
Indian Institute of Technology
Kharagpur, 721302, India
Email: {subarna, smisra}@sit.iitkgp.ernet.in

***Abstract*—This work focuses on designing an optimal, and adaptive data caching mechanism to be implemented within sensor-cloud environment. Data caching within sensor-cloud is already explored in a previous work. However, the work lacked timeliness of data delivery, i.e., the response time of a query was large due to repeated caching. Further, the design of the work did not consider incoming sensor data within an unstable environment. To address these difficulties, we propose a revised scheme that modifies the existing caching architecture. This work models the data caching to be followed with the External Cache (EC) and the Internal Cache (IC). Results show that the new model considers both stable and unstable environment and accurately estimates the change of environment over time. Further, the experimental results depict that the timeliness of data is improved to 93.4% and the queries undergo comparatively lower response time thereby, justifying the real-time applicability of the proposed scheme.**

***Index Terms*—Caching, Sensor-cloud, Energy, Latency**

I. Background

Recent research has already acknowledged sensor-cloud infrastructure as a potential substitute for the traditional Wireless Sensor Networks (WSNs) [1]–[3]. Traditional WSNs have certain limitations inclusive of difficulties in scalability, vendor-specific designs, and heavy expenditures of ownership and maintenance [4]–[6]. To address the afore-mentioned issues, Yuriyama and Kushida proposed sensor-cloud infrastructure [7] after which several subsequent research works have acknowledged the platform for its improvised performance. Sensor-cloud infrastructure [7], [8] is a cloud-based infrastructure that virtualizes physical sensor nodes into virtual sensors and renders *Sensors-as-a-Service* (*Se-aaS*). Contemporary research has already explored different direction of sensor-cloud infrastructure [9]–[12] out of which one work, which is also one of our previous works, involves incorporating data caching mechanisms within sensor-cloud platforms [13]. The work proposes a dynamic and adaptive caching technique within sensor-cloud to achieve efficiency in virtualization. However, the proposed algorithm fails to achieve the timeliness of the data while serving the end-users.

In this work, we investigate the drawback of the afore-mentioned work and resolve it by proposing an enhanced data caching algorithm to be followed extensively within sensor-cloud platforms.

A. Motivation

Sensor-cloud infrastructure connects the physical world comprising of physical sensor nodes with the cyber world. The infrastructure allows the dissemination of sensor service to the common mass of people in a simple obtainable form. End-users of sensor-cloud are required to simply required to submit their application requirements through a web interface. The requirements are decoded in terms of physical sensor allocation. The allocated physical sensors are effectively grouped into virtual sensors and the aggregated information is transmitted to the end-users in terms of Se-aaS.

As mentioned earlier, in one of our prior works [13], we have already explored the motivation behind introducing data caching techniques within sensor-cloud. Although the work designed a unique cache-based sensor-cloud architecture and proposed a dynamic data caching algorithm, yet there were some technical limitations as follows:

(i) *Rationale behind the two cache units*: Firstly, although the work mentions the presence of two distinct cache structures, there is no explicit rationale behind this count, i.e., the reasoning supporting the requirement of the two cache units is missing. Further, the difference in the significance of EC and IC is also unclear.

(i) *Timeliness of data*: The work only achieves to extract and serve the end-users with 85.91% data timeliness, i.e., any query triggered at the 100^{th} time instant is fed with data of the 85^{th} time instant which means that the recency of the data is somewhat questionable.

(ii) *Delay Analysis*: Further, the work does not consider the delay incurred in the process of caching and re-caching which is extremely crucial to judge the performance of the algorithm.

(iii) *Unstable Environment Considerations*: Although the work achieves improved network performance, it fails to respond correctly during turbulent or unstable environmental conditions, i.e., when the change of the environment between consecutive time instants is greater than a threshold and varies hugely.

These limitations strongly motivated us to revisit the problem and propose alternative solutions to address the above difficulties.

978-1-5090-1925-0/16/$31.00 ©2016 IEEE

B. Contribution

This work proposes an enhanced, optimal, and adaptive data caching mechanism to be implemented within sensor-cloud platforms. The work focuses to design an accurate, and real-time data caching scheme with simultaneous considerations of the resource constraints of the sensor nodes. The contributions of the work are multifold. Firstly, the work identifies the difficulties with the existing caching techniques that are in use within sensor-cloud platforms. The work attempts to resolve these difficulties and develop an improved mechanism for efficient data caching. The work proposes significant changes in the positioning of the internal and the external cache units and justifies the rationale of their existence. Further, we model both the external and the internal cache units. For modeling of the external cache, the energy considerations of a sensor node are taken into account whereas, for modeling the internal cache, the accuracy of data provisioning is contemplated. We thouroughly study and analyze the performance of the proposed work in terms of its accuracy in the internal computations, its implications on network performance, the optimizations achieved in re-caching, and the effects in the timeliness of data delivery. Last, but not the least, unlike the existing work, this work successfully considers unstable environment thereby incorporating adaptiveness to deal with turbulence of data.

II. Proposed Architecture

In this Section, we propose a revised architecture of caching within sensor-cloud. The prior architecture propounded by us [13] is indicated in Figure 1(a). It is clearly observed that there are two caches – External Cache (EC) and the Internal Cache (IC). The EC directly captures the change of data from the physical environment and refreshes the content periodically. Therefore, the content of EC varies with the change of the physical environment. On the other hand, the IC receives the data from the EC and the change in the content of IC is dependent on the change of EC.

The proposed architecture, as shown in Figure 1(b), also indicates two cache blocks. Firstly, we observe that the position of EC is unchanged. However, the modeling of the EC will be significantly changed to address issues (ii) and (iv) of Section IA. The position of the IC is within a Virtual Machine (VM) allocated to an end-user. When an end-user E_1 requests for Se-aaS from a particular set of sensors for the first time, the data query is redirected to the EC. If the data has been previously cached, and the cache has not yet expired, the corresponding data is fetched from the EC and transmitted to the VM (VM_1) of E_1. The data received at VM_1 is further cached within IC_1 so that any subsequent query generated by E_1 involving data retrieval from the same set of sensors may be obtained from IC_1. Now, if another end-user E_2 wishes to access the sensor data from the same set of sensors, the corresponding data query is fed with cached data retrieved from EC subjected to the recency of the data.

III. Revised Model of the External Cache

In this Section we propose a revised model of the EC. Unlike the previous work, the primary difference in the model of the revised EC is its improved adaptive nature, i.e., the ability to incorporate sudden changes in the incoming data pattern of a sensor. From our previous work, we directly obtain that the current memory m of EC, at time t, $m_{EC}(t)$, is a k tuple, and is expressed as $m_{EC}(t) = \{(r_{s_i,1}, t_1), (r_{s_i,2}, t_2), ..., (r_{s_i,k}, t_k)\}$ where $(r_{s_i,j}, t_j)$ is the sensor data from sensor s_i reported as t_j time instant. The mean rate of change of the environment over the k time instants is obtained as,

$$\mathcal{M}_k = \frac{\sum\limits_{j=2}^{k} \mid m_{EC}(t).r_{s_i,j} - m_{EC}(t).r_{s_i,j-1} \mid}{\sum\limits_{j=2}^{k} m_{EC}(t).t_i - m_{EC}(t).t_{i-1}} \tag{1}$$

However, the proposed work formulates the expected rate of change of environment with a difference. We have,

$$E(\mathcal{M}_k) = \frac{\sum\limits_{j=2}^{k} \alpha_j \times m_{EC}(t).r_{s_i,j}}{\sum\limits_{j=2}^{k} m_{EC}(t).t_i - m_{EC}(t).t_{i-1}} \tag{2}$$

where α_j is a multiplicative factor to the j^{th} sensor data value. α_j is expressed as,

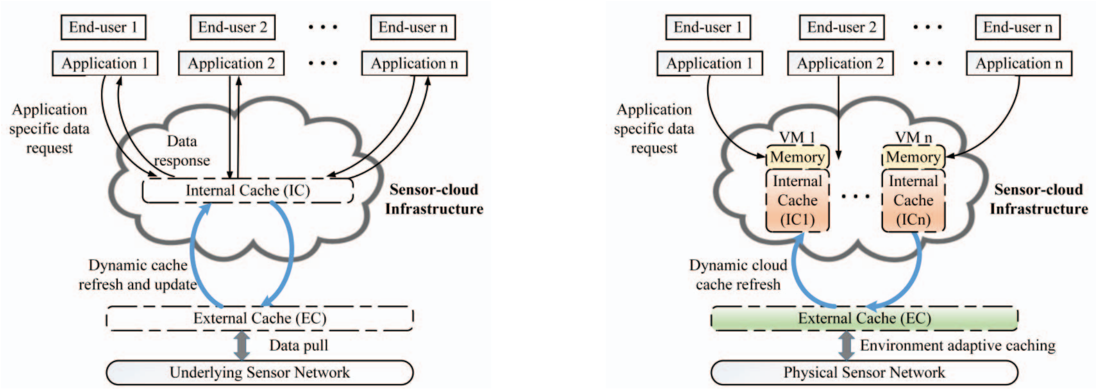
$$\alpha_j = \begin{cases} \alpha_{j-1}, \text{ if } \mid m_{EC}(t).r_{s_i,j} - m_{EC}(t).r_{s_i,j-1} \mid \leq r_{th} \\ \alpha_{j-1} + \frac{\left[m_{EC}(t).r_{s_i,j} - m_{EC}(t).r_{s_i,j-1}\right] \times \alpha_{j-1}}{m_{EC}(t).r_{s_i,j}}, \text{otherwise} \end{cases} \tag{3}$$

Thus, for k consecutive time instants, the magnitude of α will alter (or remain same subjected to conditions) as per Equation (3). The change of the sensed data at every time instant will affect the value of α which, in turn, directly affects the magnitude of $E(\mathcal{M}_k)$. We further re-formulate the equation of the cost incurred to extract non-cached sensor data from the physical environment. At a particular time, let us assume that $\mathcal{E}_{tr}$, $\mathcal{E}_s$, $\mathcal{E}_r$ be the respective energy expenditures due to transmission of a single data packet of size p bytes, sensing per unit time, and receiving a packet. Therefore, at time j, to receive n_t data packets from sensor s_i subjected to distinct d sensing events, the total energy cost incurred is as follows:

$$\mathcal{E}_j = pn_t\mathcal{E}_{tr} + (t_{j+1} - t_j)\mathcal{E}_s + pn_r\mathcal{E}_r \tag{4}$$

n_r is the number of packets transmitted to the sensor for data querying. Hence, we have, $\mathcal{E} = \sum\limits_{j=1}^{k} \mathcal{E}_j$. Thus,

$$\mathcal{E}_k = k\big(pn_t\mathcal{E}_{tr} + pn_r\mathcal{E}_r\big) + \sum_{j=1}^{k}(t_{j+1} - t_j)\mathcal{E}_s \tag{5}$$



(a) Existing architecture [13]

(b) Cache-enabled architecture

Figure 1: Previous and the proposed architectures for data caching in sensor-cloud

At k' time instant at which re-caching occurs, we have, $\mathcal{E}_{k'} = k'\left(pn_t\mathcal{E}_{tr} + pn_r\mathcal{E}_r\right) + \sum_{j=1}^{k'}(t_{j+1} - t_j)\mathcal{E}_s$, i.e.,

$$\mathcal{E}_{k'} = pk'\left(n_t\mathcal{E}_{tr} + n_r\mathcal{E}_r\right) + \sum_{j=1}^{k}(t_{j+1} - t_j)\mathcal{E}_s + (k' - k)\mathcal{E}_s \quad (6)$$

Now, to consider the cost incurred due to the delay of fetching the data from the EC to the VM, $\mathcal{E}_d$ is considered which includes the cumulative energy expenditure due to miscellaneous activities, e.g. computation in EC, packet propagation from EC to VM, and additional overhead, if any. Thus, for transmission of a single byte, Equation (6) can be rewritten as:

$$\mathcal{E}_{k'} = pk'\left(n_t\mathcal{E}_{tr} + n_r\mathcal{E}_r + \mathcal{E}_d n_t\right) + \sum_{j=1}^{k}(t_{j+1} - t_j)\mathcal{E}_s + (k' - k)\mathcal{E}_s \quad (7)$$

Now, the overall problem can be stated as, given the k time instants at which caching has occurred, i.e., EC had been refreshed and updated, we need to determine k' at which EC has to be updated subjected to energy constraints. Mathematically, we maximize $\Delta k = k' - k$ subject to,

$$\mathcal{E}_{k'} \leq \mathcal{E}_{th} \text{ and } E(\mathcal{M}_k) \leq \mathcal{M}_{th1} \quad (8)$$

where $\mathcal{M}_{th1}$ and $\mathcal{E}_{th}$ are the threshold minimum of the expected rate of change of the environment and the energy expenditure, respectively. Therefore, considering λ as the Lagrangian multiplier, our final multi-objective minimization problem is stated as:

$$\frac{1}{k' - k} - \lambda\Bigg[\left(pk'\left(n_t\mathcal{E}_{tr} + n_r\mathcal{E}_r + \mathcal{E}_d n_t\right) + \sum_{j=1}^{k}(t_{j+1} - t_j)\mathcal{E}_s \right. \\ \left. + (k' - k)\mathcal{E}_s - \mathcal{E}_{th}\Bigg] - \lambda\Bigg[E(\mathcal{M}_k) - \mathcal{M}_{th1}\Bigg] = 0 \quad (9)$$

Based on the value of λ, as obtained from Equation (9), the maximum value of k' is obtained satisfying the constraints as per Equations (8).

IV. Revised Model of the Internal Cache

Similar to the model of the EC, we obtain that the current memory of IC, at time t, $m_{IC}(t)$, is an m tuple, and is expressed as $m_{IC}(t) = \{(h_{s_i,1}, t_1), (h_{s_i,2}, t_2), ..., (h_{s_i,k}, t_m)\}$ where $(h_{s_i,j}, t_j)$ is the cache sensor data from sensor s_i obtained from EC at t_j time instant. The last caching was performed at m^{th} time instant. Thus, the mean rate of change of the data of the EC over the m time instants is obtained as,

$$\mathcal{M}_m = \frac{\sum_{j=2}^{k} \mid m_{IC}(t).h_{s_i,j} - m_{IC}(t).h_{s_i,j-1} \mid}{\sum_{j=2}^{k} m_{IC}(t).t_i - m_{IC}(t).t_{i-1}} \quad (10)$$

Similar to Equation (2), the expected rate of change of EC is,

$$E(\mathcal{M}_m) = \frac{\sum_{j=2}^{k} \beta_j \times m_{IC}(t).h_{s_i,j}}{\sum_{j=2}^{k} m_{IC}(t).t_i - m_{IC}(t).t_{i-1}} \quad (11)$$

where β_j is a multiplicative factor to the j^{th} sensor data value. β_j is expressed as,

$$\beta_j = \begin{cases} \beta_{j-1}, \text{ if } \mid m_{IC}(t).h_{s_i,j} - m_{IC}(t).h_{s_i,j-1} \mid \leq h_{th} \\ \beta_{j-1} + \frac{\left[m_{IC}(t).h_{s_i,j} - m_{IC}(t).h_{s_i,j-1}\right] \times \beta_{j-1}}{m_{IC}(t).h_{s_i,j}}, \text{otherwise} \end{cases} \quad (12)$$

where h_{th} is the threshold maximum for the change of the magnitude of the sensor data between two consecutive time instants. Now, considering that the accuracy of data provisioning of a particular sensor s_i at a time t, $\mathcal{A}_{s_i}(t)$, is proportional to the inverse of the mean square error of the data at EC and at IC. Therefore,

$$\mathcal{A}_{s_i}(t) = \left[\frac{1}{t}\sum_{j=1}^{t}\left(m_{EC}(t).h_{s_i,j} - m_{IC}(t).h_{s_i,j}\right)^2\right]^{-1} \quad (13)$$

If m' be the next instant for caching within IC, our goal is Maximize $(m' - m)$ subject to,

$$\left[\frac{1}{t}\sum_{j=1}^{t}\left(m_{EC}(t).h_{s_i,j} - m_{IC}(t).h_{s_i,j}\right)^2\right] < e_{th} \tag{14}$$

$$\text{and } E(\mathcal{M}_m) \leq \mathcal{M}_{th2} \tag{15}$$

where e_{th} is the threshold minimum for the means square error and $\mathcal{M}_{th2}$ is the threshold for the rate of change of EC. Thus, m' belongs to the solutions set of the following equation:

$$\frac{1}{m'-m} - \lambda\left[\frac{1}{t}\sum_{j=1}^{t}\left(m_{EC}(t).h_{s_i,j} - m_{IC}(t).h_{s_i,j}\right)^2 - e_{th}\right] - \lambda\left[E(M_m) - \mathcal{M}_{th2}\right] = 0 \tag{16}$$

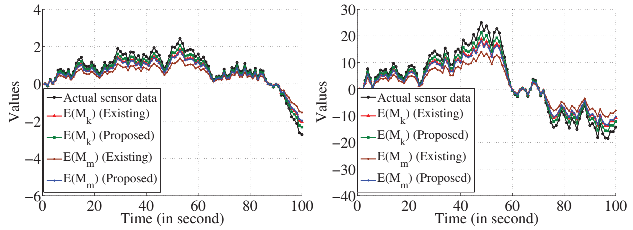
On solving the above Equation, we obtain the value of m' at which re-caching is required within IC.

V. Performance Evaluation

In this Section, we study and analyze the performance of the proposed scheme. The experimental set is illustrated in Table I. As we observe that we have revised the formulation for the expectation of the rate of change of data of the environment and the EC. Therefore, to analyze the change in the performance, we first study the comparative performance in estimating the expected values. Figure 2 shows the variation of actual sensor data from the expected values over time. A *stable* environment is one in which sensor data at two consecutive time instants vary within a very small range ($\pm 0.5\%$ in our case), i.e., the variation of sensor data is insignificant over time. In an unstable environment, the variation in the magnitude of sensor data at two consecutive time instants is reasonably high ($\pm 2\%$ in our case). We observe that in a stable environment, as indicated in Figure 2(a), the computation of expected values (by Equations (2) and (11)) are better compared to that of the existing work. Further, in Figure 2(b), we observe that unlike the previous work, the expectation of the values are more accurate in the proposed work. This is primarily because of the introduction of the multiplicative factor based on the change in the magnitude of the sensor data. To justify our results, we compute the Mean Square Error of the actual and the expected values, as shown in Figure 3. For both the EC (as indicated in Figure 3(a)) and IC (as indicated in Figure 3(b)), the error is significantly higher in case of the existing scheme and it reduces under the proposed scheme.

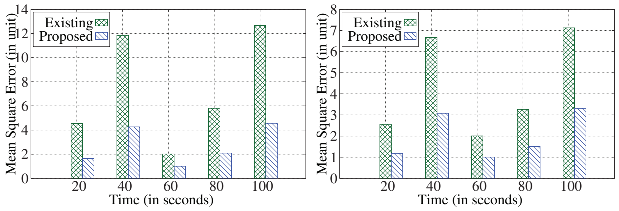
Table I: Experimental Setup

Parameters	Values
Deployment Area	1000 m ×1000 m
Deployment	Uniform, random
Number of nodes	50
Communication range	$[50, 100]$ m
Channel overhead	$[1, 5]\%$
Transmission energy	7 nJ/bit
Computation energy	5 nJ/sec
Number of query	$\{10, 50, 200, 500\}$
Number of time instants	100



(a) Stable Environment (b) Unstable Environment

Figure 2: Comparative study of accuracy in determination of the expected values in EC and IC under stable and unstable environmental conditions

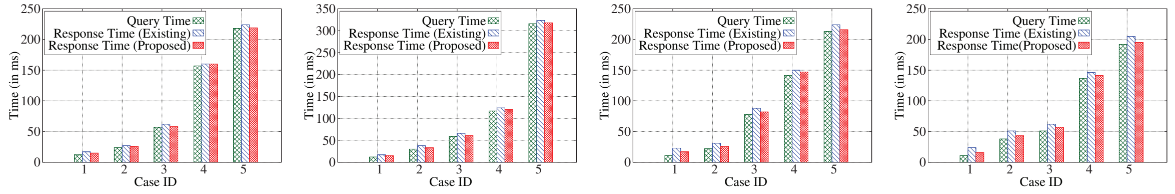


(a) External Cache (b) Internal Cache

Figure 3: Comparative study of the Mean square Error in computation of the expected values in EC and IC

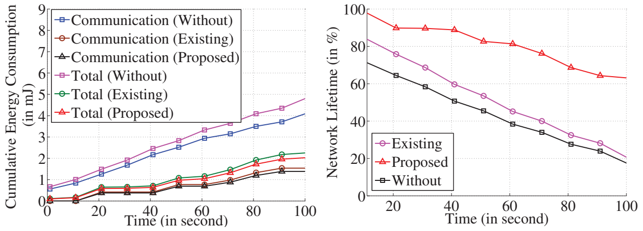
To analyze the effects on the network performance, we perform an experiment to study the energy consumption and the network lifetime as the network parameters. Figure 4 depicts the variation of the cumulative energy consumption and the network lifetime over time under three distinct scenarios – (a) without caching (b) caching under the existing scheme and (c) caching under the proposed scheme. Quite obviously, in absence of caching, the consumption of energy is significantly high as shown in Figure 4(a). However, due to accurate computation of the expected values, the energy consumption is satisfactorily reduced. This directly affects the network lifetime, as indicated in Figure 4(b). It is observed that the network lifetime is hugely increased with the existing and the non-caching scheme. Further, we also notice in Figure 5 that there is difference in the caching instant compared to the previous work. We observe that while caching with EC (in Figure 5(a)) and IC (in Figure 5(b)), the existing technique responded much after the change in the environment or the EC has reached the threshold. However, it is noticed that the existing scheme responds almost immediately with the change of the data thereby, inferring a better adaptability.

As previously mentioned in Section IA, the existing work did not analyze the delay incurred in the process of caching. To study this aspect, we perform an experiment, as shown in Figure 6, under multiple scenarios by varying the query count. For a particular query count we study 5 different cases and depict the response time to the queries. In Figure 6, the first bar depicts the time instant at which 10 queries were simultaneously triggered. The response time of all the queries



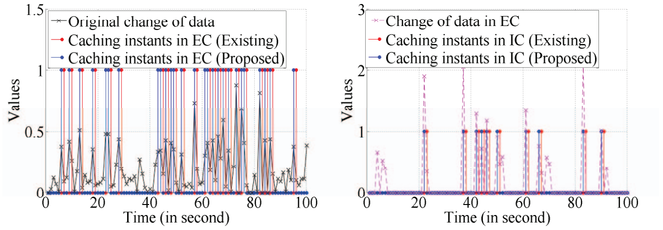
(a) Query count = 10 (b) Query count = 50 (c) Query count = 200 (d) Query count = 500

Figure 6: Comparative analysis of the response time under different query count



(a) Cumulative energy consumption (b) Network Lifetime

Figure 4: Comparative analysis of network performance



(a) Re-caching in EC (b) Re-caching in IC

Figure 5: Comparative analysis of the time instants for re-caching

under the existing and the proposed caching techniques are indicated by the second and the third bar, respectively. We observe that for query count 10 and 50 (Figure 6(a) and 6(b), respectively), the response time of the proposed scheme is marginally low compared to the existing scheme. However, with increased query count of 200 and 500 (Figure 6(c) and 6(d), the response time is notably low under the proposed scheme. The timeliness is observed to have increased from 85.91% to 93.4%. This is mainly due to accurate re-caching at correct time instants.

VI. Conclusion

This work examines the difficulties with the existing caching techniques and revises the mathematical modeling of the EC and the IC from the existing scheme. The work clearly reformulates the design of estimation of the expected values for the rate of change of environment. Results show that the proposed scheme estimates the expected values with a significantly reduced error compared to the existing scheme. The experiments also demonstrate enhancement in the network performance and the low response time of the proposed scheme thereby justifying its real-time applicability.

References

[1] K.-L. Tan, "What's next?: Sensor + cloud!?" in *DMSN*, ser. ACM International Conference Proceeding Series, D. Zeinalipour-Yazti and W.-C. Lee, Eds. ACM, 2010, p. 1.

[2] S. Madria, V. Kumar, and R. Dalvi, "Sensor cloud: A cloud of virtual sensors," *IEEE Software*, vol. 31, no. 2, pp. 70–77, 2014.

[3] S. Chatterjee and S. Misra, "Target tracking using sensor-cloud: Sensor-target mapping in presence of overlapping coverage," *IEEE Communications Letters*, vol. PP, no. 99, pp. 1–1, 2014.

[4] Y. Liu, Y. He, M. Li, J. Wang, K. Liu, and X. Li, "Does wireless sensor network scale? a measurement study on greenorbs," *IEEE Trans. on Parallel and Dist. Sys.*, vol. 24, no. 10, pp. 1983–1993, Oct 2013.

[5] H. He, Z. Zhu, and E. Makinen, "A neural network model to minimize the connected dominating set for self-configuration of wireless sensor networks," *IEEE Transactions on Neural Networks*, vol. 20, no. 6, pp. 973–982, June 2009.

[6] R. Rout and S. Ghosh, "Enhancement of lifetime using duty cycle and network coding in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 2, pp. 656–667, February 2013.

[7] M. Yuriyama and T. Kushida, "Sensor-cloud infrastructure - physical sensor management with virtualized sensors on cloud computing," in 13^{th} *International Conference on Network-Based Information Systems (NBiS)*, Sept 2010, pp. 1–8.

[8] 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," *Intl J. of Distributed Sens Netw*, p. 18, Nov 2013.

[9] S. Misra, A. Singh, S. Chatterjee, and M. Obaidat, "Mils-cloud: A sensor-cloud-based architecture for the integration of military tri-services operations and decision making," *IEEE Sys. J.*, pp. 1–9, 2014.

[10] S. Misra, S. Chatterjee, and M. Obaidat, "On theoretical modeling of sensor cloud: A paradigm shift from wireless sensor network," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–10, 2014.

[11] S. Chatterjee, R. Ladia, and S. Misra, "Dynamic optimal pricing for heterogeneous service-oriented architecture of sensor-cloud infrastructure," *IEEE Transactions on Services Computing*, 2015.

[12] S. Misra, A. Singh, S. Chatterjee, and A. K. Mandal, "Qos-aware sensor allocation for target tracking in sensor-cloud," *Ad Hoc Networks*, vol. 33, pp. 140 – 153, 2015.

[13] S. Chatterjee and S. Misra, "Dynamic and adaptive data caching mechanism for virtualization within sensor-cloud," in *IEEE ANTS*, Dec 2014, pp. 1–6.