Dynamic and Adaptive Data Caching Mechanism for Virtualization within Sensor-Cloud

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 proposes a dynamic, and adaptive caching mechanism for efficient virtualization in sensor-cloud — one of the first attempts in this direction. The work introduces both internal and external caching techniques to ensure efficiency in resource utilization of the underlying physical network. Conventional data transmission techniques involve periodic packet transmissions to the cloud-end. However, the rate of change of the physical environment may not be reasonably significant, thereby leading to redundant packet transmissions and inefficient utilization of network resources. The proposed caching mechanism is flexible with the varied rate of change of the physical environment. Results show that compared to the existing techniques, caching appreciably conserves network resources in terms of energy consumption and network lifetime, by 37.1%, and 48.43%, respectively. Experimental results also depict that using caching techniques, the data provisioned to the end-users through sensor-cloud are atleast 85.91% recent.**

***Index Terms*—Wireless Sensor Network, Sensor-cloud, Virtualization, Modeling and Simulation of sensor-clouds**

I. INTRODUCTION

Recent research has discovered sensor-cloud infrastructure as a potential substitute for the traditional Wireless Sensor Networks (WSNs) [1]–[3]. As defined in [4], [5], sensor-cloud infrastructure is a remote data management, and real-time data provisioning platform that thrives on virtualization of physical sensor nodes into virtual sensors, as per the demand of applications at the user-end. Data from the virtual sensors are stored, processed, and managed at the cloud end, thereby enabling the end-users to envision the physical sensor nodes as a service —*Sensors-as-a-Service* (*Se-aaS*), rather than as a typical hardware.

A. Motivation

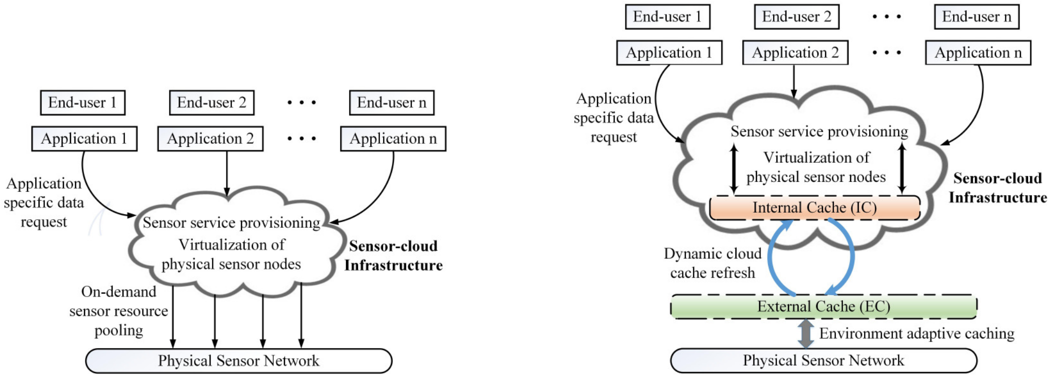
The architecture of sensor-cloud is three layered with the end-users at the top, the cloud at the middle, and the physical sensor networks at the bottom [4], as shown in Figure 1(a). The end-users request the sensor-cloud for the sensed information through an interface. The cloud, in turn, allocates the physical sensors, and subsequently forms the virtual sensor for serving the application. The data from the virtual sensors are transmitted over the Internet, and are fed to the respective applications of the end-users. In the proposed architecture, the sensor-cloud directly accesses the underlying physical nodes as per the application demand. Whenever a physical node becomes an active component of a virtual sensor, it is selected, and data is fetched from it. Thus, if a running application has a high rate of demand from its virtual sensor, the component nodes of the virtual sensor engage themselves in continuous data communications with the sensor-cloud. Thus, the continuous process of data retrieval and data logging occurs with time. However, in practical scenarios, the change in the environment may be sufficiently slow, thereby resulting in a negligible rate of change of data of the sensed attributes. For a small or moderate change in the environment, the sensed data remains almost unaltered. Thus, continuous transmission of the unaltered sensed data leads to unnecessary energy loss due to redundant transmission. Moreover, it takes a toll on the lifetime of the individual sensor nodes, as well as the network. The significance of data caching in sensor-cloud becomes important in this context.

B. Contribution

As mentioned above, sensor-cloud virtualizes the physical sensor nodes, and provisions Se-aaS. In real-life scenarios, the change of the physical environment in terms of sensor reading is imperceptible, when diagnosed very frequently, i.e. the change of the sensed data is generally negligible within a small time interval. The contribution of this work is to design an optimal caching mechanism within sensor-cloud to obtain resource efficiency in terms of energy, and network lifetime. The proposed data caching mechanism is dynamic, and is adaptive to the change of the physical environment, thereby preserving the accuracy of information, and conserving the network resources, simultaneously. The user requests for the sensed data are served from the cache when the change of the physical environment is gradual. The work determines an optimal caching interval beyond which fresh data is requested from the physical sensors, and cached again. Thus, the proposed solution can significantly reduce the expenditure of transmission energy, and enhance the network lifetime.

C. Organization of the paper

The rest of the paper is organized as follows. Section II highlights the prior work done so far. Section III illustrates the details of the proposed architecture. Section IV, and V presents the mathematical model of the two proposed cache



(a) Existing architecture

(b) Cache-enabled architecture

Figure 1: Existing and proposed architectures of Sensor-cloud

structures. Sections VI and VII demonstrate the theoretical evaluation, and the results of simulation, respectively. Finally, Section VII concludes the work.

II. RELATED WORK

This Section discusses the prior work done so far in this domain. Prior to the invention of virtualization of sensors in sensor-cloud, several works focused on the integration of sensor networks to a cloud platform [6]–[8]. In [9], Mendes *et al.* focused on the challenges that arise when two or more nodes with protocol differences (IEEE 802.15.3 and IEEE 802.15.4) communicate together. The work proposes a dynamic, cross-layer admission control scheme for cloud-based multimedia WSN. Mohamed, and Xuemin [10] propose a traffic analysis technique to locate hotspots. Cloud-based transmissions are used to ensure energy-efficiency and ensure optimized packet transmissions. Few works have also focused on dynamic gateway selection strategies [11], [12], or dynamic bandwidth management [13], to ensure load-balancing within cloud. Li *et al.* [14] contributed in analyzing an application scenario for dynamic decision-support for irrigation system using sensor-cloud integration.

The actual perspective of sensor-cloud infrastructure was presented in [4], [15], in which the virtualization of physical sensor nodes was proposed. Alamri *et al*. thoroughly surveyed the intrinsic concepts, the benefits, and the challenges associated with sensor-cloud [5]. Details of virtualization was presented by Madria *et al.* [3], in which the possible configurations of grouping of several physical sensors to one or more virtual sensors has been discussed. However, most of the works have primarily focused on the dogma and ideology of sensor-cloud.

Out of the very few works that have addressed the technical aspects of sensor-cloud, Chatterjee, and Misra [2] focused on a multi-organization application scenario for tracking multiple targets using the sensor-cloud infrastructure. Nguyen, and Huh [16] have focused on the security of data transmission in a sensor-cloud environment. Chandra *et al.* [17] addressed the benefits of using Ethernet enabled Arduino microcontroller for data communication at the sensor-cloud end. Bhunia *et al.* [18] addressed the problem of data acquisition from the underlying physical sensor nodes by fuzzification of the data. In all of the above works, the process of data transmission from the underlying physical networks to the cloud has been considered to be periodic, and continuous. However, as mentioned in Section I-A, the rate of change of environment is random, and inconsistent. The goal of this work is to achieve a double-caching mechanism that is dynamic, and adaptive to the change in the physical environment, thereby providing an efficient utilization of network resources, and simultaneously maintaining the accuracy of the provisioned data.

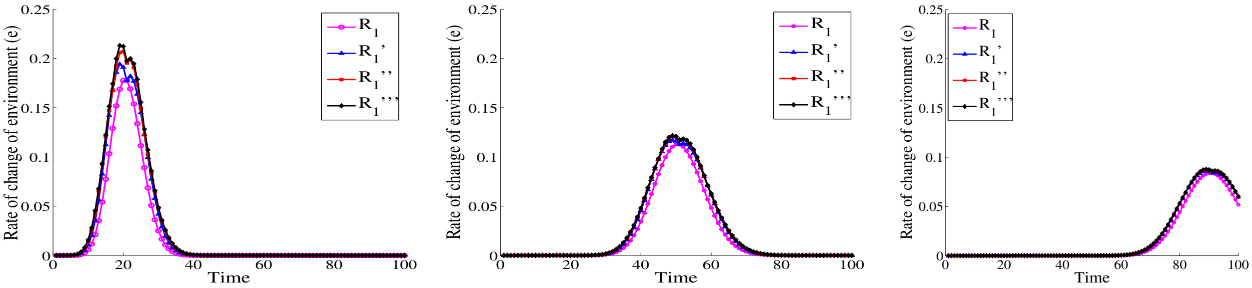
III. PROPOSED ARCHITECTURE

As mentioned in Section I-A, due to the direct access of communication of sensor-cloud with the underlying physical network, the problem of redundant sensed data transmission and logging persists. This, in turn, results in inefficient utilization of network resources. In this Section, we present the proposed cache-enabled architecture of sensor-cloud infrastructure to address the aforesaid problem.

As illustrated in Figure 1(b), we consider two caches —the *primary cache* or the *External Cache* (*EC*), and the *secondary cache* or the *Internal Cache* (*IC*). The EC is an interfacing cache that resides between the cloud, and the underlying network. EC is sensitive to the change in the physical environment, thereby dynamically updating and refreshing its content to retain synchronization with the current state of the physical environment. The working principle of EC is based on the expected rate of change of the environment. The IC, on the other hand, is located within the cloud, and is updated in coherence with the dynamism of the EC. IC obtains the information from EC and synchronizes it with the variable demand rate of the end-user applications.

IV. MODEL OF THE EXTERNAL CACHE

In this Section, we discuss the working principle of the EC. The goal of EC is to determine an optimal caching interval Δt, based on the history of the sensed data from a particular



(a) $R_1 \sim P(100, 20)$ (b) $R_1 \sim P(100, 50)$ (c) $R_1 \sim P(100, 90)$

Figure 2: Analysis of rate of change of physical environment with time

sensor s. Thus, given the last k readings $R_1 = \{r_1, r_2, ..., r_k\}$ at timestamps $T_1 = \{t_1, t_2, ..., t_k\}$, the goal is to find k', such that the caching interval $\Delta t = k' - k$ is maximum, subject to certain constraints.

Initially, we define some of the required metrics of the work.

Definition 1. The current memory m of EC, at time t, is a k tuple, where k is a pre-negotiated system value. m is expressed as,

$$m(t) = \{(r_1, t_1), (r_2, t_2), ..., (r_k, t_k)\}, r_i \in R_1, t_j \in T_1 \quad (1)$$

Thus, at a given time, the mean rate of change of environment, e, is obtained as,

$$e = \frac{\sum_{i=2}^{k} \mid m(t).r_i - m(t).r_{i-1} \mid}{\sum_{i=2}^{k} m(t).t_i - m(t).t_{i-1}} \quad (2)$$

Definition 2. The expected rate of change of environment, e, for a particular physical sensor, is based on the i^{th} degree of the rate of change of the environment, $1 \leq i \leq k$. Thus,

$$E(e) \propto \left(\frac{dR_1}{dT_1}, \frac{d^2R_1}{dT_1^2}, ..., \frac{d^kR_1}{dT_1^k} \right) \quad (3)$$

$$\Rightarrow E(e) = \sum c_i \frac{d^iR_1}{dT_1^i} \quad (4)$$

where c_i is a constant.

For the sake of justification of Equation (4), an experiment was performed for the 100 sensor readings, and the data were approximated to follow a Poisson distribution [19] with $n = 100$, and varied mean $\mu = (20, 50, 90)$, as shown in Figures 2(a), 2(b), and 2(c), respectively. It is observed that the rate of change of the environment is significant up to the 2^{nd} order, beyond which the rate of change is minimal. Hence, for the sake of simplicity, Equation (4) is revised as,

$$E(e) = c_1 \frac{dR_1}{dT_1} + c_2 \frac{d^2R_1}{dT_1^2} \quad (5)$$

$E(e)$ should be less than a threshold value $e_{threshold}$, beyond which re-caching should occur to maintain accuracy of data. Having estimated the expected rate of change of environment, we now design the constraint to satisfy energy efficiency of the underlying physical sensor network. Assuming α and β as the respective energy cost associated with per unit of communication (both transmission and reception), and per state transition, we infer that the cost $\mathcal{C}$ incurred for obtaining non-cached data directly from the physical network is,

$$\mathcal{C} = 2\alpha E_{tr} + \beta E_{st} \quad (6)$$

where E_{tr}, and E_{st} are the energy expended due to communication, and state transition, respectively. A factor of 2 is associated with E_{tr}, as it involves the transmission of a signal to the respective sensor and obtaining data packet from it. E_{st} is consumed mainly due to transition of the state of the node from passive (idle or not transmitting) to active (transmitting) [20]. The cumulative energy expenditure till the next time instant for caching (k') is expressed as,

$$\begin{aligned} f_2(k') &= \sum_{t_i=2}^{k'} \Big(2\alpha E_{tr} + \beta E_{st} + (t_i - t_{i-1})\gamma E_s \Big) \\ &= k'\mathcal{C} + \sum_{t=2}^{k} (t_i - t_{i-1})\gamma E_s + (k' - k)\gamma E_s \quad (7) \end{aligned}$$

where γ is the energy cost associated with per unit sensing activity, and E_s is the energy expended due to sensing. Now we formulate a multi-objective optimization problem as,

$$\begin{aligned} &\textit{Maximize } \Delta k = k' - k \\ &\text{i.e., } \textit{Minimize } f_1(k') = \frac{1}{\Delta k} = \frac{1}{k' - k} \quad (8) \end{aligned}$$

$$\textit{Minimize } f_2(k') \quad (9)$$

$$\text{subject to, } f_3(T_1) = c_1 \frac{dR_1}{dT_1} + c_2 \frac{d^2R_1}{dT_1^2} < e_{threshold} \quad (10)$$

Equation (10) accounts for the adaptability of EC. Thus, using the method of scalarization, and combining Equations (8) through (10), the resulting minimization problem is,

$$\textit{Minimize } L(k', T_1) = k'\mathcal{C} + \sum_{t=2}^{k} (t_i - t_{i-1})\gamma E_s + (k' - k)\gamma E_s$$

$$-\alpha_1\left(\frac{1}{k'-k}\right)-\alpha_2(c_1\frac{dR_1}{dT_1}+c_2\frac{d^2R_1}{dT_1^2}-e_{threshold}) \quad (11)$$

Therefore,

$$\frac{\delta L}{\delta k'}=\mathcal{C}+\gamma E_s+\frac{\alpha_1}{(k'-k)^2}=0, \quad (12)$$

$$\frac{\delta L}{\delta T_1}=\alpha_2\left[c_1\frac{d^2R_1}{dT_1^2}+c_2\frac{d^3R_1}{dT_1^3}\right]=0 \quad (13)$$

Using Equations (11) through (13), we obtain the optimal value of k'^{*}. Thus, EC defines a optimal caching interval $\Delta k^{*}=k'^{*}-k$ that minimizes the energy consumption of the physical nodes, and is adaptive to the dynamics of the physical environment. The schedule for re-caching commences at the end of every caching interval. Data requests, within the caching interval, are served from the EC itself.

V. MODEL OF THE INTERNAL CACHE

This Section illustrates the working model of the IC. The IC primarily handles the data requests from the user-applications, and decides to serve the data either directly from the cache or re-caches the data from the EC and then serves it. If the sequence of the data provisioned to the end-users at $\{t_i\}$ be d, $d=\{d_i\}, 1\leq i\leq p$. Initially, the first p readings from EC are directly fed into IC for preparing the history. Thus, the expected rate of change of EC, e', is given as,

$$E(e')=\frac{\sum_{j=2}^{k}\mid d_j-d_{j-1}\mid}{\sum_{i=2}^{k}t_i-t_{i-1}} \quad (14)$$

Definition 3. The mean accuracy $\hat{\mathcal{A}}$ of data provisioning is defined as the inverse of the Mean Square Error (MSE) of the sensor readings at EC and IC, evaluated for the previous j time instants. Thus, mean accuracy of a data at time t is expressed as,

$$\hat{\mathcal{A}}=\frac{j}{\sum_{i=t-j+1}^{t}(m(t).r_i-d_i)^2} \quad (15)$$

For accurate servicing of data, $\sum_{i=t-j+1}^{t}(m(t).r_i-d_i)^2\rightarrow r$, where r is an extremely small value, $r\neq 0$. Assuming that data has been cached within IC at time k, the minimization problem for IC is expressed as,

$$\textit{Maximize } (k''-k) \text{ i.e., } \textit{Minimize } g_1(k')=\frac{1}{k''-k} \quad (16)$$

where k'' is the next scheduled instant for caching within IC, subjected to the constraints,

$$\sum_{i=k''-j+1}^{k''}(m(k'').r_i-d_i)^2-r\simeq 0 \quad (17)$$

Thus, using Equations (16) and (17), the solution set for k'' is expressed as,

$$k\in\min_{k<h<g}\left\{\max\left\{\sum_{i=h-j+1}^{h}(m(h).r_i-d_i)^2\right\}\right\} \quad (18)$$

$$\sum_{i=g-j+1}^{g}(m(g).r_i-d_i)^2>r \quad (19)$$

After obtaining k'', the dynamic caching of IC can be executed by maintaining data provisioning accuracy, simultaneously.

VI. THEORETICAL ANALYSIS

Proposition 1. *If λ and $\hat{d}$ are, respectively, the non-uniform demand rate, and the data provisioning rate for p time instants, the mean accuracy $\hat{\mathcal{A}}$ has a lower bound $\hat{\mathcal{A}}_{min}$.*

Proof: A non-uniform demand sequence is characterized by, $\lambda=\{\lambda_j\}, 1\leq j\leq p$, where $\lambda_i-\lambda_{i-1}\neq\lambda_l-\lambda_{l-1}, i\neq l, 2\leq i,l\leq p$. We also have, $d=\{d_j\}, 1\leq j\leq p$. Let us assume that $\hat{\mathcal{A}}$ has no lower bound. Therefore, since θ^{th} time instant caching did not occur to preserve accuracy. Thus, using Equation (15), we get,

$$\sum_{i=t-j+1}^{t}(m(t).r_i-d_\theta)^2>\sum_{i=t-j+2}^{t+1}(m(t).r_i-d_\theta)^2 \quad (20)$$

Thus, for a very high value γ, at a particular time t',

$$\sum_{i=t-j+1}^{t}(m(t).r_i-d_\theta)^2\rightarrow\gamma \quad (21)$$

However, as $\gamma\gg r$, $\exists g$ as per Equation (19). Thus, caching must have occurred at least once to reflect the data of IC, as $d_{\theta+1}$. Naturally, $\{(m(t).r_i-d_\theta)^2\}$ is an increasing sequence till g, from we which infer that $\hat{\mathcal{A}}_{min}$ is bounded by a lower value. This concludes the proof. ■

Proposition 2. *Assuming k and k' as the previous, and the next instant of caching within EC, respectively, $\Delta k=k'-k$ always possesses a lower and an upper bound as Δk_{min}, and Δk_{max}, respectively.*

Proof: Δk is minimum, when $E(e)\gg e_{threshold}$, i.e. when the environment is highly changing . Thus,

$$\alpha_2(c_1\frac{dR_1}{dT_1}+c_2\frac{d^2R_1}{dT_1^2}-e_{threshold})\rightarrow h \quad (22)$$

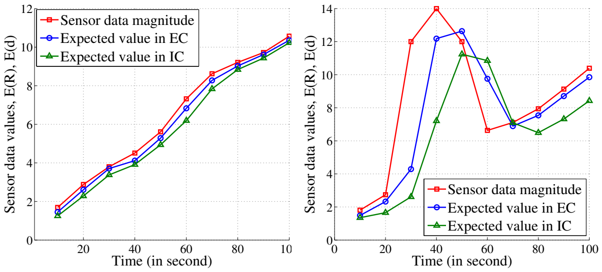
where h is a value of high magnitude, and Equation (22) becomes the dominant constraint. Thus, as per Equation (11), $L(k',T_1)$ will have its minimum value at k_{new}, $k_{new}\rightarrow k$. Thus, $\Delta k=k_{new}-k=\Delta k_{min}\simeq 0$. On the other hand, for a gradually changing environment, $c_1\frac{dR_1}{dT_1}+c_2\frac{d^2R_1}{dT_1^2}\ll e_{threshold}$. However, if $\{q_i\}$ be an increasing sequence, $\sum_{i=j}(q_i-q_{i-1})\gamma E_s>\sum_{i=j+1}(q_i-q_{i-1})\gamma E_s$. Thus, $L(k',T_1)$ obtains its maximum value at k_{new}, where $k_{new}-k\not\rightarrow 0$, and $\Delta k_{max}=k_{new}-k=b$, where b is of significant magnitude. This concludes the proof. ■

VII. PERFORMANCE EVALUATION

This Section presents the overall evaluation of the performance of the proposed dynamic adaptive caching mechanism. The experiments are performed in Matlab, and the experimental setup is illustrated in Table I.

Table I: Experimental Setup

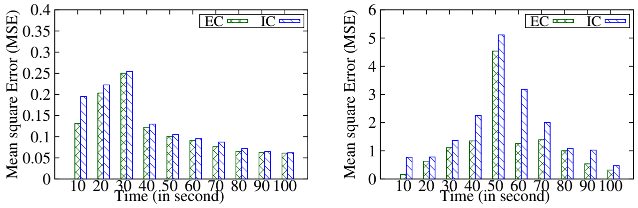
Parameters	**Values**
Deployment Area	500 m × 500 m
Deployment	Uniform, random
Number of nodes	100
Communication energy (E_{tr})	70 nJ/bit
Energy due to state transition (E_{st})	30 nJ
Sensing energy (E_s)	10 nJ/bit
Number of time instants	100



(a) Stable environment (b) Unstable environment

Figure 3: Study of the expectation of the sensed data

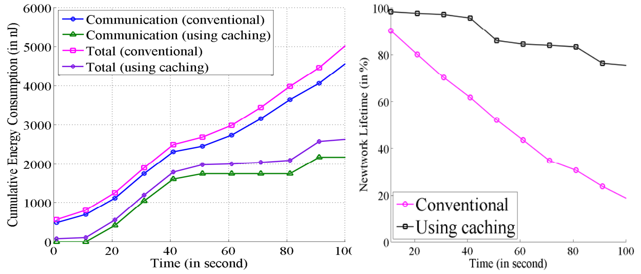
To justify the correctness of the expectation of e and e', as mentioned in Equation (5) and (14), respectively, an experiment is performed on randomized sensor readings, over 100 instants of time. The experiment is repeated for stable (gradually changing), and unstable (fast and sudden changing) scenarios of the physical environment, as shown in Figures 3(a), and 3(b), respectively. Figure 3(a) exhibits a gradual, and slow-paced change in the environment, thereby leading to close estimation of the sensed values in EC, and IC. On the other hand, Figure 3(b) shows the rate of change of the data for a turbulent environment leading to little deviations in the process of estimation.



(a) Stable environment (b) Unstable environment

Figure 4: Analysis of the Mean Square Error (MSE) in computation

The accuracy of computation, as given in Definition 3, is evaluated in terms of the computation of the MSE for the above two scenarios. Figure 4(a) clearly shows the MSE obtained for expecting the rate of change of environment in a stable condition. The error obtained for the first few time instants are initially high, due to the gradual learning or adaptiveness of the caching process, after which the error falls to a negligible value. For an unstable environment, as depicted in Figure 4(b), the MSE in expecting the change of the environment rises initially and the reduces with time.



(a) Analysis of energy efficiency (b) Analysis of network lifetime

Figure 5: Overall analysis of the network resources

The energy efficiency of the proposed caching mechanism within EC is studied in Figure 5(a) in terms of the cumulative energy consumption, $\mathcal{E}$, expressed as,

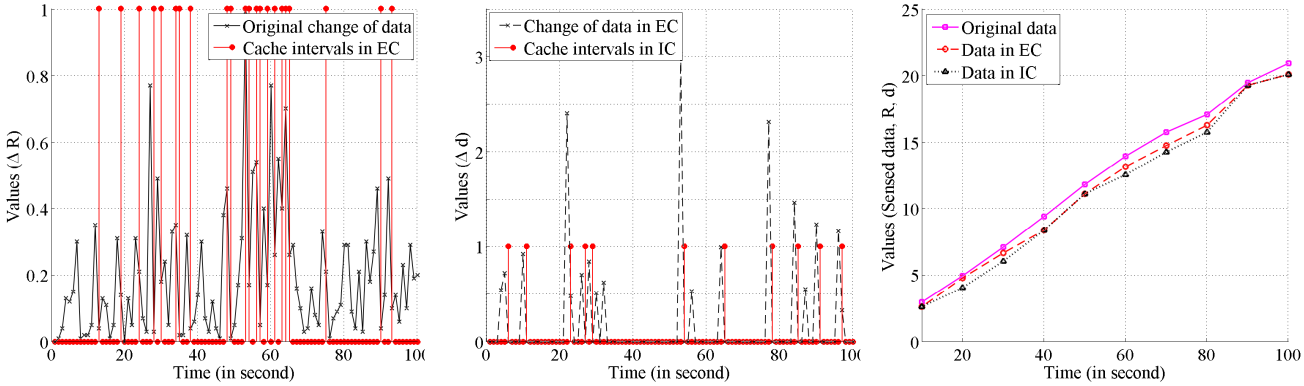
$$\mathcal{E}(j) = \sum_{t_i=1}^{j} \left(2\alpha E_{tr} + \beta E_{st} + (t_i - t_{i-1})\gamma E_s \right) \tag{23}$$

$\forall 1 \leq j \leq 100$. Conventional techniques generally follow a periodic data transmission scheme [21], thus, consuming more energy, compared to those that follow caching in which the transmission of the physical nodes is sensitive to significant change of the environment. Mathematically, the consumption of energy of individual sensor nodes is also reduced by 37.1%. This, in turn, enhances the network lifetime $\mathcal{N}$ of the nodes, as shown in Figure 5(b). $\mathcal{N}$ is computed as,

$$\mathcal{N}(t) = \frac{\mathcal{N}_{max} - \mathcal{E}(t)}{\mathcal{N}_{max}} \times 100\% \tag{24}$$

where $\mathcal{N}_{max}$ is assumed to be $6000nJ$. Figure 5(b) clearly indicates the improvement of the network lifetime by using caching mechanisms, compared to the conventional ones. The network lifetime was observed to increase by 48.43%.

Figure 6 examines the performance of the proposed caching techniques in EC, and IC. The stem plots in Figures 6(a), and 6(b) indicate the scheduling of caching in both EC and IC, respectively. A 0 indicates no caching, and a 1 indicates re-caching. Based on the substantial change in the readings of the physical sensors, the caching is dynamically performed within EC, as shown in Figure 6(a). The caching within IC depends on the rate of change of data within EC. However, the frequency of caching is much less in IC, compared to EC. Over 100 time instants, the deviation of the cached data was observed, compared to the original data, as shown in Figure 6(c). It is found that, due to caching, data within EC, and IC are respectively maximally deviated by 7.79%, and 14.09%, only. Therefore, in the worst case, the demand at $t = 100$ is served with a data of $t = 85.91$, thereby accounting for a minimum of 85.91% recentness of the provisioned data.



(a) Caching in EC

(b) Caching in IC

(c) Overall data deviation

Figure 6: Analysis of adaptiveness and dynamism of caching

VIII. CONCLUSION

This work introduces an adaptive caching mechanism to prevent the unnecessary depletion of network resources. The work proposes an external, and an internal cache that dynamically, and optimally cache the data based on the variable rate of change of the physical environment, thereby achieving reduction in redundant transmissions of data packets from the underlying sensor network to the sensor-cloud.

Future scope of the work will explore additional issues associated with the Quality of Service (QoS) parameters of sensor-cloud infrastructure. The examination of cost-effectiveness due to caching also induces research attention. Other aspects of sensor virtualization can also be considered as a relevant direction of future research.

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. 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.
[3] S. Madria, V. Kumar, and R. Dalvi, "Sensor cloud: A cloud of virtual sensors," *IEEE Software*, vol. 31, no. 2, pp. 70–77, 2014.
[4] 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.
[5] 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.
[6] 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 Systems Journal*, vol. PP, no. 99, pp. 1–9, 2014.
[7] R. Kumar and S. Rajalakshmi, "Mobile sensor cloud computing: Controlling and securing data processing over smart environment through mobile sensor cloud computing (mscc)," in *International Conference on Computer Sciences and Applications (CSA)*, Dec 2013, pp. 687–694.
[8] C. Srimathi, S.-H. Park, and N. Rajesh, "Proposed framework for underwater sensor cloud for environmental monitoring," in 5^{th} *Intl Conf on Ubiquitous and Future Netw (ICUFN)*, July 2013, pp. 104–109.
[9] L. Mendes, J. Rodrigues, J. Lloret, and S. Sendra, "Cross-layer dynamic admission control for cloud-based multimedia sensor networks," *IEEE Systems Journal*, vol. 8, no. 1, pp. 235–246, March 2014.
[10] M. Mahmoud and X. Shen, "A cloud-based scheme for protecting source-location privacy against hotspot-locating attack in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 10, pp. 1805–1818, Oct 2012.
[11] S. Misra and S. Chatterjee, "Social choice considerations in cloud-assisted WBAN architecture for post-disaster healthcare: Data aggregation and channelization," *Information Sciences*, no. 0, pp. –, 2014.
[12] S. Misra, S. Bera, A. Mondal, R. Tirkey, H.-C. Chao, and S. Chattopadhyay, "Optimal gateway selection in sensor-cloud framework for health monitoring," *IET Wireless Sensor Systems*, vol. 3, no. 4, December 2013.
[13] S. Misra, S. Das, M. Khatua, and M. Obaidat, "QoS-guaranteed bandwidth shifting and redistribution in mobile cloud environment," *IEEE Trans. on Cloud Computing*, vol. 2, no. 2, pp. 181–193, April 2014.
[14] C. Li, R. Dutta, C. Kloppers, C. D'este, A. Morshed, A. Almeida, A. Das, and J. Aryal, "Mobile application based sustainable irrigation water usage decision support system: An intelligent sensor cloud approach," in *IEEE SENSORS*, Nov 2013, pp. 1–4.
[15] M. Yuriyama, T. Kushida, and M. Itakura, "A new model of accelerating service innovation with sensor-cloud infrastructure," in *Annual SRII Global Conference (SRII)*, March 2011, pp. 308–314.
[16] M. V. Nguyen and E.-N. Huh, "An efficient key management for secure multicast in sensor-cloud," in *First ACIS/JNU Intl Conf on Computers, Netw, Systems and Industrial Engineering (CNSI)*, May 2011, pp. 3–9.
[17] A. Chandra, Y. Lee, B. M. Kim, S. Y. Maeng, S. H. Park, and S. R. Lee, "Review on sensor cloud and its integration with arduino based sensor network," in *International Conference on IT Convergence and Security (ICITCS)*, Dec 2013, pp. 1–4.
[18] S. Bhunia, J. Pal, and N. Mukherjee, "Fuzzy assisted event driven data collection from sensor nodes in sensor-cloud infrastructure," in 14^{th} *IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, May 2014, pp. 635–640.
[19] R. K. Lam, D.-M. Chiu, and J. C. Lui, "On the access pricing and network scaling issues of wireless mesh networks," *IEEE Transactions on Computers*, vol. 56, no. 11, pp. 1456–1469, 2007.
[20] W. Liu, L. Cui, and X. Niu, "EasiTPQ: QoS-based topology control in wireless sensor network." *Signal Processing Systems*, vol. 51, no. 2, pp. 173–181, 2008.
[21] Y.-L. Lai and C. J., "A cloud-storage RFID location tracking system," *IEEE Transactions on Magnetics*, vol. 50, no. 7, pp. 1–4, July 2014.