ISSN 2007-9737

DCQSH: Dynamic Conflict-Free Query Scheduling in Heterogeneous Networks during Emergency

V. Ramasamy [1], B. Gomathy [2], Joy Lal Sarkar[3], Chhabi Rani Panigrahi[4], Bibudhendu Pati[4], Abhishek Majumder[3]

[1] Park College of Engineering and Technology,
Department of Computer Science and Engineering,
India

[2] KPR Institute of Engineering and Technology,
Department of Computer Science and Engineering,
India

[3] Tripura University,
Department of Computer Science and Engineering,
India

[4] Rama Devi Women's University,
Department of Computer Science,
India

{researchrams, bgomramesh, joylalsarkar, panigrahichhabi, patibibudhendu, abhi2012}@gmail.com

Abstract. There can be disasters such as tsunami, fire-related incidents, etc. in several ways. Mobile devices and the cloud occupy a significant position in connectivity and relief operations in these circumstances. This would be more important an efficiently performing query facility in mobile devices in a crisis situation. To achieve the mentioned facility, a Dynamic conflict-free query scheduling approach for heterogeneous networks during the emergency situation (DCQSH) is suggested in this paper. DCQSH is specifically built to schedule queries for the heterogeneous communication networks. DCQSH's key feature would be that it can optimize the query efficiently and often operates with complex tasks and adjusts the query rate without rebuilding the existing transfer schedule. DCQSH operates within heterogeneous networks, as it could accommodate the condition where the mobile devices become low energy-efficient on the networks. The experimental findings reveal that DCQSH outperforms in a heterogeneous scenario in terms of its relation to baseline algorithms. MATLAB framework was utilized to validate the simulation performance.

Keywords. Disasters, mobile device, networks, parallel,distributed.

1 Introduction

In 2023 there would be a growth in the number of mobile device connections will be 29.3 billion and mobile users will be 5.6 billion as per official report on the Internet given by Cisco for 2018-2023 [1]. The findings reveal that mobile devices have become the best option for disaster response. The unusual environment activity create crisis situation which annoys the human value and their existence [2]. By the year 2015, a record of 14,795 earthquakes occurred worldwide, harming human being's life as well as resources etc [2]. The essential contact among respondents as well as defaults / impacted locations plays a major role in normalizing the emergency. Respondents can, for example, act immediately by realizing the correct scenario of the evacuation site [4].

The innovative technology for Mobile Cloud Computing (MCC) provides a chance to tackle these conditions [4, 5, 6]. For MCC mobile devices are essential. There are many tasks that have to be handled with mobile devices, but they have limited energy and storage capacity [7, 8, 9, 10, 11, 12, 13, 14, 15, 16]. By using various social media, users post photos and videos, etc. Hence, the utilization of battery capacity by mobile devices to execute programs and meet the user's requirements is important [17, 18, 20, 21, 22, 23, 24, 25, 26].

As a benefit of MCC, mobile devices could offload and run those power-hungry programs in the cloud and only return back the outcomes. This strategy allows mobile devices to reduce both energy usage and resource capacity. In this case, MCC will contribute to improving power efficiency and handling limited infrastructure issue by executing the full program in the cloud and submitting only the outputs to the indented mobile device [9]. This helps mobile devices to retain electricity.

If the user saves photos and photographs, videos etc. in the cloud and return back when needed, it can reduce the mobile device resource issue. The mobile resource issue can be minimized as the user transfer images and photographs, recordings etc into the cloud and return back as appropriate.The key focuses of the Cloud Torrent [29], the Cloudlets [30], the Clone Cloud [31] etc. are a cloud-based efficient resource delivery [32] and performance interfaces with lower energy usage.

A Specific variants of researches are still performed regarding systems towards a limited data rate correlated with emergencies [47]. However, WSNs can help large-scale data-rate implementations in the fields of disaster, climatic conditions, health and welfare, etc. [48, 49]. Also wellness and disaster response systems often create huge network loads, as they want to monitor the faster level of data transfer from any mobile device [48]. Therefore the massive number of mobile devices accessible in geographical crises scenarios creates massive network loads. The same period, mobile apps relevant to emergencies create a huge number of queries where environmental anomalies have arisen, such as a fire incident, flood, earthquake, etc. A feasible framework is therefore required to manage a wide range of queries in crisis areas through various applications. The query service processing mechanism collects the necessary WSN information based on the queries submitted and builds them up to help enhance the success and cost - effectiveness of the execution on a network basis [50, 51]. In this article, we suggest the DCQSH model to fulfill the criteria for the huge data scale as well as conflict-free query scheduling.

This paper covers the following layout: Section 2 outlines the works which are relevant. Section 3 provides an overview of the problem. The network structure is explained in Section 4. Section 5 discusses our proposed methods for the scheduling of conflict-free queries. The mathematical formulation is defined in section 6. The experimental setup is described in section 7. Section 8 presents the details of the simulation along with the analysis of results. The paper conclusion presented in Section 9.

1.1 Motivation

The nearby mobile devices ought to provide a high volume of data communication in the event of a crisis scenario [48, 49]. To achieve a high volume of data communication, a simultaneous query execution strategy combined with conflict-free query scheduling must be enabled. Many mobile devices in the communication network have limited capacity to complete the query because of the lower strength of the batteries. Thus, an optimum query scheduling strategy needs to be developed and the target device discovery method is necessary that could be utilized in a heterogeneous situation and supports the scheduling of a query in a conflict-free manner. Some studies facilitate conflict-free query scheduling, however, those doesn't perform well in heterogeneous network scenarios.

An emergency management system using mobile cloud computing (EMC^2) [46] was presented in our earlier work. Primarily, EMC^2 detects obtainable mobile devices around the region with a mobile probing service (MPS) and rank them through a mobile ranking service (MRS). Then $E2M$ [19] determines the required mobile device among them by optimizing the mobile devices rank and network load scores. This expands our previous work. DCQSH is configured to perform with heterogeneous networks, in which certain devices are less energy-intensive and facilitate conflict-free query scheduling.

2 Related Work

There are limited work relevant to disaster response [3, 5, 6]. Authors in [3, 5] introduced their research in the form of a disaster response by suggesting Cloud Probing Service (CPS) for finding the right cloud [32] with an Anchor Point (AP) and Cloud Ranking Service (CRS). The Network Probing Services (NPS) was also suggested to pick the right network. This research was performed by utilizing the Amazon Online Service as a public cloud and rescue vehicle as a local cloud. In this research, 3G / WiFi networks have become substantial, but mobile devices may lose network coverage in disaster zones.

For the critical scenario, few android based applications like disaster Alert, Hurricane Hound and Storm Eye, etc. were introduced [33, 34]. Authors in [35] implemented an emergency monitoring android application by choosing the ideal path for diverse geological locations using genetic algorithm. Mobile devices with lack of energy is a common scenario in emergencies [36, 37, 44, 46] and little research work was expressed related to efficient energy management for the mobiles [27, 28, 29]. Remote sensing approach lowers the energy consumption of mobiles for varied activities [17]. In early stage prediction of CPU time of mobiles using decision making techniques may lowers the battery usage of the mobiles [35]. Lyapunov optimization technique [38] based eTime [41] is cut down the mobile device energy utilization and this technique offloads the task dynamically [39].

Mobile to cloud offloading whether minimizes the battery power or not is checked by the work carried in the paper [40]. Markov decision based scheduling concepts for wireless network and rigid computing based scheduling technique for mobiles were proposed in [40]. Method-level computing offloading of Thinkair approach [8] was used to audit the scalability of cloud and mobile offloads. But the task was run in mobile alone due to this approach was not recognizing the accessible public sources. A public cloud of Amazon web services and android based offloading architecture called mCloud was presented in [26] for the cloud with heterogeneous options. The social media information [42] of specific patients, various institutions and government, etc. were combined with shared cloud for efficient sharing of this disaster information by the public, NGOs, volunteers and rescue teams by using PDAs and cell phones for economical rehabilitation activity [2].

Wireless Intelligent Sensor Networks (WISN) based Smart Cloud Evacuation System (SCES) was proposed by [43] collects the real time disaster information by using cell phones and sensor arrays and kept it in integrated cloud storage for effective analysis and reply to the public in an easy way to leave the critical location on time. Emergency Medical Service (EMS) was a centralized cloud oriented pervasive approach which bridges the hospitals, health care units, ambulance assistance and patients for efficient access of critical healthcare data by recognized users in a standard format to make cost effective treatment on time [44]. An integrated smart phones, sensors and cloud oriented smart city based emergency management approach proposed in [45] helps to guide the First Responders (FRs) in an effective way of rescue operation by capturing the emergency area and FRs migration and place it in the cloud with efficient analysis.

The conflict related communication protocols need to provide assistance for the real time based applications [52]. Few of the protocols which are related to contention normally manages the clogging [53]. But those protocols are not working with the applications which has huge data rate alike emergency and healthcare monitoring etc. TDMA

ISSN 2007-9737

protocol accomplish surprising latency related to contention oriented protocols.

Some protocols discussed in [52, 53] are effective for single-hop networks and which is not suitable for multi-hop networks. The Earliest Deadline First (EDF) oriented MAC protocol used like a precedence oriented protocol with seven frequencies to avert the intervention of the transmission channel [54]. Precedence oriented Real-time Query Scheduling (RTQS) approach described in [55]. RTQS keeps various arrangements to schedule the queries. But, it follows minimal inter-free duration to execute two high latency successive queries.

The authors of [56] presented an arbitrarily dispersed schedule method named DRAND for wireless ad-hoc networks, which were TDMA-based but not support large data loads. A DRAND based DCQS was addressed in [57] for the high level of query scheduling. Point-to - point data communication time delay studies were examined in [58] amongst sensor devices utilizing WirelessHART network.

3 Problem Statement

An emergency respondent/user with mobile device would like to handle the urgent queries in crisis scenarios, he needs to prefer the optimal mobile device. The mobiles with extremely poor battery capacity and certain mobiles with adequate battery balance are possible in an emergency condition. The challenge for the respondent/user seems to be selecting the right mobile from the mobile devices accessible. Mobile devices can produce several queries at once in crisis scenarios.

All devices is not adequate to handle the queries because of poor battery capacity. In this case, the query needs to be processed using the appropriate mobile device. EMC^2 can choose the cost effective target device for processing the data according to our prior work [46]. The difficult job of the mobile device even after choosing the optimal target device, it needs to handle the requested queries in a conflict-free way.

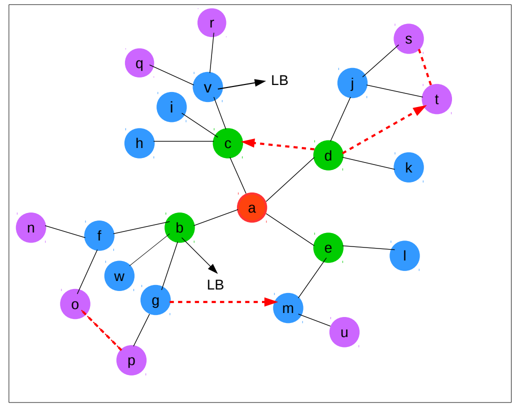


Fig. 1. A sample ICH tree (here, LB denotes low battery)

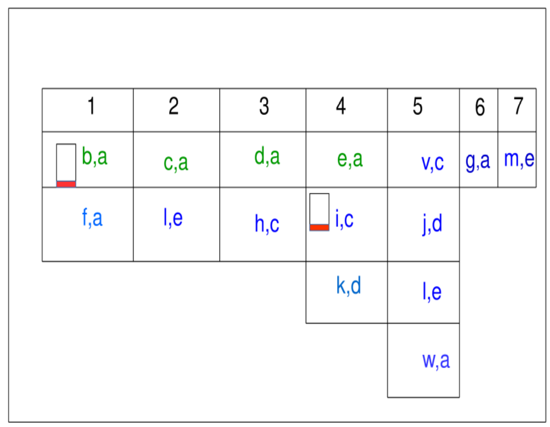


Fig. 2. Time chart for ICH tree as in Fig. 1

4 Network Model

Mobile devices with the low battery capacity are distributed via DCQSH and are intended to perform conflict-free query scheduling. The exchange of data through mobile devices having poor battery capacity is however a complex process. For example, device A is trying to transfer information to device B, but B doesn't often collect information from A due to the lower battery capacity. DCQSH aims here to enable alternative appropriate device to acquire information from A and prevent the conflicting circumstance of device B.

In addition, we provided a Interference-Communication Heterogeneous (ICH) tree and its corresponding time chart in Fig. 1 and Fig. 2 respectively. Thus by Fig. 1, a direct line connection is described in ICH tree for the effective connection between sender and receiver devices. This enables every device to collect information between every devices. The interfering connection represented as a dotted line in the tree often indicates that some other neighboring communication may be hindered by a data connection among the sender and the receiver devices. The single direction of connectivity given with an arrow link whereas multi-direction of connectivity given without an arrow link inside the tree. The interaction of $\overrightarrow{ba}$ and $\overrightarrow{le}$ should be conflict-free communications like ($ba \parallel le$), so (1) e, l, a and b must be parallel, and (2) la and be must not communicate with one another.

As per Fig. 1, the devices b, c, d, e inside the ICH tree, device b has been the highest rank and scheduled to initiate connection to device a. However, it possesses low battery capabilities and couldn't interact with the device a. In slot 1 in Fig. 2, therefore, device f, a sub device of device b, initiates a connection with the device a. The c, d and e devices are collectively reserved for slots 2 , 3 and 4 to avoid conflict situations. At periods 5 and 6, the other sub devices w and g of b device shall be configured to ensure conflict-free communications towards the device a. Owing to the fact that slots 2, 3 and 4 are reserved to interact with devices a specifically for c, d and e. Now, i has reduced battery power and couldn't interact with c with the scheduled time slot 4. However, it is not possible to allocate the device v at time slot 4 alternative of the device i because of interruption edge $\overrightarrow{dc}$. Therefore time slot 5 is reserved for v.

5 Conflict-free Query Scheduling

Numerous emergency systems such as fire protection, detection of earthquakes, environmental tracking, etc. are typically quite common and creates lots of queries. In order to avert conflicts with these queries the scheduler must retain an accurate tracking of the run queue as well as

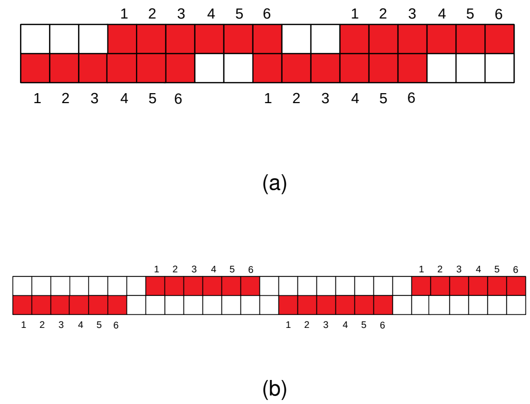


Fig. 3. (a) An example of conflict table. (b) An example of conflict-free table

release queue regarding the last query starting time in the run queue and releasing time of a new query in the release queue. Since the *Conflict point* (*C-point*) arises when a release queue is emitted a new query until the present query operation is not finished in the run queue. Therefore the release queue would then emit the newest query to run queue subsequently to the *C-point*. It gives the existing query one position ahead in the run queue and avoids any conflict.

Fig. 3a and Fig. 3b is showing the example conflict and conflict-free tables as well. Each query has a total of 6 time frames. According to Fig. 3a, the conflict occurs whenever 2 different queries are performed concurrently as per in the conflict table. However, as per in conflict-free table shown in Fig. 3b the conflicts do not occur. Since present query is minimum one time frame ahead of succeeding query. The scheduler therefore tracks the C-point every time in order to prevent conflict operation. In addition, the query size for single and multi-class is not always an identical one. The query sizes are distinct. However the scheduler needs to maintain records of the size of the query, too, each time.

Theorem 4.1 *DCQSH provides conflict-free query schedules in every time frames.*
Proof. As a consideration, let $\mathfrak{q}_1$ and $\mathfrak{q}_k$ queries be performed at the same time frame. Since $\mathfrak{q}_1$ and $\mathfrak{q}_k$ are assumed to be conflict-free, they will avoid

the C-point according to DCQSH. In contrast, the scheduler wants to perform any of the queries at most once in order to avoid conflicts and thus $(s_1 - s_k) \geq (k-1).\mathcal{C}(x) \geq \mathcal{C}(x)$. Here, $\mathcal{C}(x)$ is release time of the queries $\mathfrak{q}_1$ and $\mathfrak{q}_k$ and $k \geq 2$ as well as the time frames of the $\mathfrak{s}_1$ and $\mathfrak{s}_k$. DCQSH thus maintains conflict-free query schedule process.

Theorem 4.2 *The maximum query rate for* $\frac{1}{\mathcal{S}(\mathfrak{n}).\mathcal{C}(x)}$ *remains with DCQSH. Here, S(n) be the frame length in seconds.*

Proof. As per DCQSH, in $\mathcal{C}(x)$ time frames the queries are released to avoid conflicts. Therefore, $\frac{1}{\mathcal{S}(\mathfrak{n}).\mathcal{C}(x)}$ is the maximal query rate.

6 Mathematical Formulation

A very critical job would be to transfer query requests from sender to receiver devices due to several ranges of mobile devices that have poor battery capacity in a disaster environment. The battery will not be replaced quickly under such situations. The enhanced scheduling of emergency queries may be effective in the diversified circumstance. Let's take $\mathcal{Q}_\mathfrak{i}$ list has $\mathfrak{i}$ amount of queries obtained by the device referred in Eqn. (1):

$$\mathcal{Q}_\mathfrak{i} = \mathfrak{q}_1 + \mathfrak{q}_2 + ... + \mathfrak{q}_\mathfrak{i}. \quad (1)$$

The conflicts will be eliminated when the device collects the queries beyond a C-point. Consequently, Eqn. (2) measures the total device load:

$$\mathcal{T}^{w,\mathfrak{i}} = \sum_{\mathfrak{p}=1}^{\mathfrak{i}} \mathfrak{q}_\mathfrak{p} - \left((\mathfrak{i}-1)\mathcal{C}(x) - (\sum_{\mathfrak{p}=1}^{\mathfrak{i}-1} \mathcal{S}_\mathfrak{p}) \right). \quad (2)$$

There the scheduler scheduled the query S time frames prior to those of the following query.

Let $\mathcal{W}_d$ be known to be the worse time delay determined by Eqn. (3):

$$\mathcal{W}_d = (\mathcal{C}(x) - 1) + \sum_{\mathcal{H}} \frac{\mathcal{W}_d}{\mathcal{P}_\mathcal{H}}.\mathcal{C}(x). \quad (3)$$

Thus, $\mathcal{H}$ denotes a relatively high order query that is the maximum constrained time span for $\frac{\mathcal{W}_d}{\mathcal{P}_\mathcal{H}}$ and $\mathcal{P}_\mathcal{H}$. Eqn. (4) measures the average latency of the total queries:

$$\mathcal{T}(w_d, \mathfrak{i}) = \left(\sum_{\mathfrak{p}=1}^{\mathfrak{i}} \mathfrak{q}_\mathfrak{p} - \left((\mathfrak{i}-1)\mathcal{C}(x) - (\sum_{\mathfrak{p}=1}^{\mathfrak{i}-1} \mathcal{S}_\mathfrak{p}) \right) \right) \left((\mathcal{C}(x) - 1) + \sum_{\mathcal{H}} \frac{\mathcal{W}_d}{\mathcal{P}_\mathcal{H}}.\mathcal{C}(x) \right). \quad (4)$$

Let $\mathfrak{n}$ number of devices inside the disaster zone as specified in Eqn. (5) be taken into consideration:

$$\mathcal{N}(\mathfrak{s}, \mathfrak{n}) = \mathfrak{n}_1 + \mathfrak{n}_2 + \mathfrak{n}_3 + ... + \mathfrak{n}_\mathfrak{n}. \quad (5)$$

Each mobile device does have the maximum battery power in the initial stage as presented in Eqn. (6):

$$\mathcal{E}(\mathfrak{s}, \mathfrak{n}) = max(\mathfrak{e}_1) + max(\mathfrak{e}_2) + max(\mathfrak{e}_3) + ... + max(\mathfrak{e}_\mathfrak{n}). \quad (6)$$

According to DCQSH, some mobiles have the lower battery capacity, as well as some are battery efficient. Therefore, Eqn. (7) calculates the overall battery capacity for mobile devices:

$$\mathcal{E}(max(\mathfrak{e}), min(\mathfrak{e})) = max(\mathfrak{e}_1) + (max(\mathfrak{e}_2) - \alpha) + (max(\mathfrak{e}_3) - \beta) + ... + max(\mathfrak{e}_\mathfrak{n}). \quad (7)$$

When i amount of queries executed using α, β amount of energy, the mobile devices attain a lower battery power stage. Eqn. (8) measures the total amount of consumed power:

$$\mathcal{E}_\mathfrak{s}(\mathcal{N}) = \Bbbk^c \left(\sum_{\mathfrak{p}=1}^{\mathfrak{i}} \mathfrak{q}_\mathfrak{p} - \left((\mathfrak{i}-1)\mathcal{C}(x) - (\sum_{\mathfrak{p}=1}^{\mathfrak{i}-1} \mathcal{S}_\mathfrak{p}) \right) \right). \quad (8)$$

Now, the $\Bbbk^c$ is often the consumed mobile devices' battery capacity to execute queries. By considering DCQSH, lower-power devices shall send their queries to certain nearby devices.

Afterwards, energy used by newer devices is measured in Eqn. (9) as i amount of queries are executing:

$$\mathcal{E}_\mathfrak{s}(\mathfrak{elg}) = \Bbbk_\mathfrak{c}^{\mathfrak{elg}} + \Bbbk^\mathfrak{c} \left(\sum_{\mathfrak{p}=1}^{\mathfrak{i}} \mathfrak{q}_\mathfrak{p} - \left((\mathfrak{i}-1)\mathcal{C}(x) - (\sum_{\mathfrak{p}=1}^{\mathfrak{i}-1} \mathcal{S}_\mathfrak{p}) \right) \right) \quad (9)$$

ISSN 2007-9737

In this scenario, the $\mathbb{k}_{\mathfrak{c}}^{\mathfrak{elg}}$ is the battery capacity that the new device still has until the lower capacity task is applied. Therefore, the remaining energy of every device is determined with Eqn. (10) after the distribution of the query:

$$
\begin{aligned}
\mathcal{R}(\mathfrak{e}, \mathfrak{n}) &= max(\mathfrak{e}_1) \\
&- \left(\mathbb{k}^{\mathfrak{c}} \sum_{\mathfrak{p}=1}^{\mathfrak{i}} \mathfrak{q}_{\mathfrak{p}} - \left((\mathfrak{i}-1)\mathcal{C}(x) - (\sum_{\mathfrak{p}=1}^{\mathfrak{i}-1} \mathcal{S}_{\mathfrak{p}}) \right) \right) \\
&+ (max(\mathfrak{e}_2) - \alpha) + (max(\mathfrak{e}_3) - \beta) \\
&+ max(\mathfrak{e}_4) - ((\mathbb{k}_{\mathfrak{c}}^{\mathfrak{elg}} \\
&+ \mathbb{k}^{\mathfrak{c}} \left(\sum_{\mathfrak{p}=1}^{\mathfrak{i}} \mathfrak{q}_{\mathfrak{p}} - \left((\mathfrak{i}-1)\mathcal{C}(x) - (\sum_{\mathfrak{p}=1}^{\mathfrak{i}-1} \mathcal{S}_{\mathfrak{p}}) \right) \right))... + \mathfrak{e}_{\mathfrak{n}}.
\end{aligned} \tag{10}
$$

Now, $\mathfrak{e}_4$ is the succeeding appropriate device in this situation where a device is subject to poor energy consumption.

7 Experimental Setup

We have used the MATLAB framework to implement the proposed DCQSH approach. We also configured a network limit of $802.11b$ including a 2 Mbps transfer rate to enhance the data transmitting capacity. This architecture has been adapted and developed in certain health care service applications [48]. Our approach has installed 100 devices, $800\ m\ *\ 800\ m$ network area and $80\ m\ *\ 80\ m$ grid distribution. Randomly, the devices had been distributed throughout the grid.

8 Results and Discussion

We have conducted experiments of queries relying on mono and multiple classes. In addition, our suggested DCQSH system was compared to baseline algorithms such as CQS [59], CETM [60], DCQS [57] and DRAND [56]. DCQS is based on conflict-free query planning and DRAND takes TDMA strategies into account. CQS is a single class based approach. As per heterogeneity based comparison, the baseline algorithms degrade its performance, but DCQSH works well compared to baseline approaches.

We found that DCQSH performs well compared to baseline approaches with respect to battery efficiency, query finishing rate and query latency. In Fig. 4, X and Y-axis illustrate the query latency Q_l and query rate Q_r respectively. The latency time to respond about the query status is more of the baseline algorithms. The explanation towards the higher latency would be that hopping devices need to wait up to the entire frame size. In addition to the minimal-energy environment, the mobile devices were unable to relay the information to their parent mobiles in terms of baseline method. This will not help for the rescue process in the emergency scenario on time. The proposed approach takes very less time duration for its response.

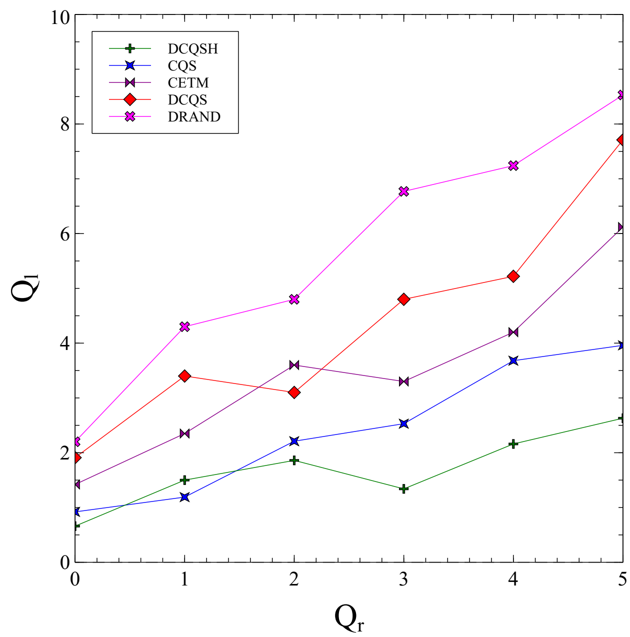
Fewer devices had been assumed in our experimentation as lower energy. The baseline algorithms were consuming more battery power while executing multiple queries. We also found that baseline strategies loosened to incredibly low efficiency by through low-energy devices had been maximized. The power usage per data frequency E_c is represented in Y axis as well as query frequency Q_r represented in X axis in Fig. 5. The proposed DCQSH approach saves more battery power while executing many queries. This will helps to process many queries in emergency situations and to do many useful operations.

In Fig. 6, the query finishing frequency Q_c mentioned in the Y axis as well as query frequency Q_r mentioned in X axis. The baseline algorithms take more time to complete the queries which were submitted to them. It is not sufficient enough in the case of emergency scenario.

The proposed DCQSH approach completes the query execution in very less time duration in terms of baseline algorithms.

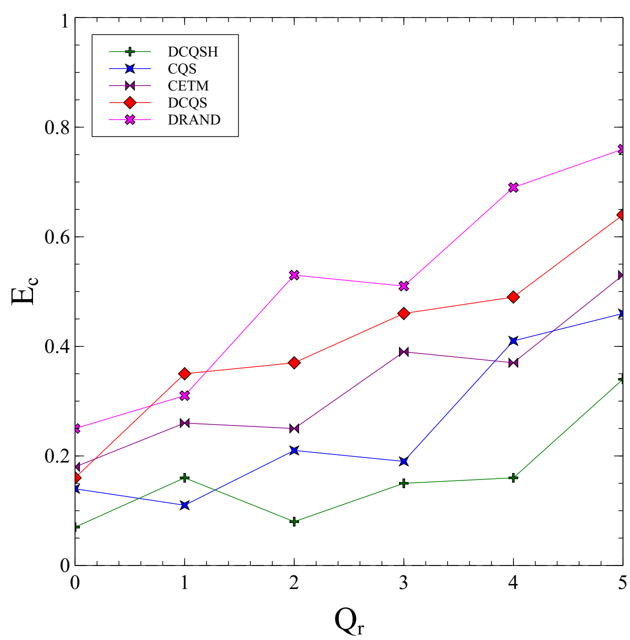
9 Conclusion

The DCQSH approach is aimed at enhancing the query efficiency and often operates for variable queries and the variations in the query frequency, without rebuilding the transfer pattern in heterogeneous networking scenarios. The exchange of data through mobile devices having a poor battery capacity is however a complex process. A routing tree is designed to address this



(a)

Fig. 4. Query frequency in proportion to the delay of the query



(a)

Fig. 5. Query frequency per data frequency in terms of power usage

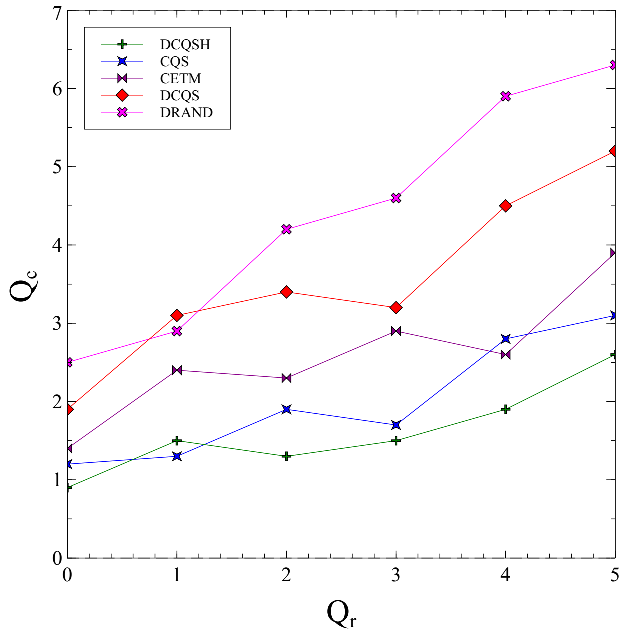


Fig. 6. Query frequency regarding its completion frequency of the query

issue. A conflict-free table is also presented for scheduling the queries to avert the conflicts when multiple queries are trying to execute in a single time moment.

We evaluated by comparing DCQSH with baseline algorithms such as CQS, CETM, DCQS and DRAND. The findings from the simulation reveals that DCQSH outperforms in heterogeneous networks in terms of baselines. We found that DCQSH performs well compared to baseline approaches with respect to battery efficiency, query finishing rate and query latency.

In addition, the DCQSH assessment reveal that any slot in DCQSH is capable of providing single and multi-class based conflict-free query scheduling. We plan to enhance DCQSH in our future work with preference based query request.

References

1. **CISCO (2015).**https://www.cisco.com/c/en/us /solutions/collateral/service-provider/visual-networking-index-vni/vni-forecast-qa.html.

2. **Li, J., Li, Q., Khan, S.U., Ghani, N. (2011).** Community-based cloud for emergency management. Proceedings of 6th International Conference on System of Systems Engineering, pp. 55–60.

3. **Ramchurn, S.D., Wu, F., Jiang, W., Fischer, J.E., Reece, S., Roberts, S., et al. (2015).** Human agent collaboration for disaster response. Journal of Autonomous Agents and Multi-Agent Systems, Vol. 30, No. 1, pp. 1–30.

4. **Mitra, K., Saguna, Åhlund, C. (2014)**. A mobile cloud computing system for emergency management. IEEE Cloud Computing, Vol. 1, No. 4, pp. 30–38.

5. **Mitra, K., Saguna, S., Åhlund C., Lulea, D.G. (2015)**. M^2C^2: A mobility management system for mobile cloud computing. IEEE Wireless Communications and Networking Conference (WCNC), pp. 1608–1613.

6. **Bi, H., Gelenbe, E. (2014)**. A Cooperative Emergency Navigation Framework Using Mobile Cloud Computing. Information Sciences and Systems, pp. 41–48.

7. **Sanaei, Z., Abolfazli, S., Gani, A., Buyya, R. (2014)**. Heterogeneity in mobile cloud computing: Taxonomy and open challenges. IEEE Communi-

cations Surveys and Tutorials, Vol. 16, No. 1, pp. 869–876.

8. **Kosta, S., Aucinas, A., Hui, P., Mortier, R., Zhang X. (2012)**. ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. Proceedings of the 31st IEEE International Conference on Computer Communications, pp. 945–953.

9. **Zhou, B., Dastjerdi, A.V., Calheiros, R.N., Srirama, S.N., Buyya, R. (2015)**. A context sensitive offloading scheme for mobile cloud computing service. Proceedings of IEEE 8th International Conference on Cloud Computing, CLOUD'15, pp. 869–876.

10. **Panigrahi, C.R., Sarkar, J.L., Pati, B., Bakshi, S. (2016)**. E^3M: An energy efficient emergency management system using mobile cloud computing. IEEE International Conference on Advanced Networks and Telecommunications Systems, ANTS'16, pp. 1–6.

11. **Bahl, P., Han, R.Y., Li, L.E., Satyanarayanan, M. (2012)**. Advancing the state of mobile cloud computing. Proceedings of the 3rd ACM Workshop on Mobile Cloud Computing and Services, pp. 21–28.

12. **Zhou, B., Dastjerdi, A.V., Calheiros, R.N., Srirama, S.N., Buyya, R., Zhang, W., et al. (2015)**. Just-in-time code offloading for wearable computing. IEEE Transactions on Emerging Topics in Computing, Vol. 3, No. 1, pp. 74–83.

13. **Panigrahi, C.R., Pati, M.T.B., Sarkar, J.L. (2015)**. EEOA: Improving energy efficiency of mobile cloudlets using efficient offloading approach. Proceedings of 9th IEEE International Conference on Advanced Networks and Telecommunications Systems (IEEE ANTS), pp. 1–6.

14. **Pati, B., Sarkar, J.L., Panigrahi, C.R., Debbarma, S. (2018)**. ecloud: An efficient transmission policy for mobile cloud computing in emergency areas. Progress in Intelligent Computing Techniques: Theory, Practice, and Applications, pp. 43–49.

15. **You, C., Huang, K., Chae, H. (2016)**. Energy efficient mobile cloud computing powered by wireless energy transfer. IEEE Journal on Selected Areas in Communications, Vol. 34, No. 5, pp.1757–1771.

16. **Cai, Y., Yu, F.R., Bu, S. (2016)**. Dynamic operations of cloud radio access networks (c-ran) for mobile cloud computing systems. IEEE Transactions on Vehicular Technology, Vol. 65, No. 3, pp. 1536–1548.

17. **Priyantha, B., Lymberopoulos, D., Liu, J. (2010)**. Eers: Energy efficient responsive sleeping on mobile phones. ACM PhoneSense'10.

18. **Oliver, E., Keshav, S. (2010)**. Data driven smartphone energy level prediction. University of Waterloo, Technical Report CS-2010-06.

19. **Ramasamy, V., Gomathy, B. (2020)**. E2M: An efficient emergency management system. Arab. J. Sci. Eng.

20. **Heiser, G., Carroll, A. (2010)**. An analysis of power consumption in a smartphone. Proceedings of the 2010 USENIX conference on USENIX annual technical conference, pp. 21.

21. **Park, J., Yu, H., Lee, E. (2012)**. Resource allocation techniques based on availability and movement reliability for mobile cloud computing. Proceedings of the 8th International Conference on Distributed Computing and Internet Technology, pp. 263–264.

22. **Zhang, W., Wen, Y., Member, S., Wu, D.O. (2015)**. Collaborative Task Execution in mobile cloud computing under stochastic wireless channel. IEEE Transactions on Wireless Communications, Vol. 14, No. 1, pp. 81–93.

23. **Lin, C.H., Hsiu, P.C., Hsieh, C.K. (2014)**. Dynamic backlight scaling optimization: A cloud-based energy-saving service for mobile streaming applications. IEEE Transactions on Computers, Vol. 63, No. 2, pp. 335–348.

24. **Khan, A.R., Othman, M., Xia, F., Khan, A.N. (2015)**. Context-aware mobile cloud computing and its challenges. IEEE Cloud Computing, Vol. 2, No. 3, pp. 42–49.

25. **Zhou, B., Dastjerdi, A.V., Calheiros, R.N., Srirama, S.N., Buyya, R. (2015)**. mCloud: A context-aware offloading framework for heterogeneous mobile cloud. EEE Transactions on Services Computing, Vol. 1374, No. 99, pp. 1–14.

26. **Pereira, O.R.E., Rodrigues, J.J.P.C. (2013)**. Survey and analysis of current mobile learning applications. ACM Computing Surveys (CSUR), Vol. 46, No. 2, pp. 1–35.

27. **Verma Rajesh, Pati Bibudhendu, Panigrahi Chhabi, Sarkar Joy, Mohapatra Subhashish (2018)**. M2C: An Energy-Efficient Mechanism for Computation in Mobile Cloud Computing pp. 697–703.

28. **Pati, B., Sarkar, J.L., Panigrahi, C.R. (2017)**. Ecs: An energy-efficient approach to select cluster-head in wireless sensor networks. Arabian Journal for Science and Engineering, Vol. 42, pp. 669–676.
29. **Nurmine, J.K.,Kelényi, J. (2010)**.CloudTorrent - Energy-efficient BitTorrent content sharing for mobile devices via cloud services. Proceedings of 7th IEEE Conference on Consumer Communications and Networking Conference, pp. 646–647.
30. **Satyanarayanan, M., Bahl, P., Caceres, R., Davies, N. (2009).** The case for VM-based cloudlets in mobile computing. IEEE Pervasive Computing, Vol. 8, No. 4, pp. 14–23.
31. **Chun, B.G., Ihm, S., Maniatis, P., Naik, M., Patti, A. (2011).** CloneCloud: elastic execution between mobile device and cloud. Proceedings of Sixth Conference on Computer Systems, pp. 301–314.
32. **Mollah, M.B., Azad, A.K., Vasilakos, A. (2017).** Secure data sharing and searching at the edge of cloud-assisted internet of things. IEEE Cloud Computing Vol. 4, No. 1, pp. 34–42.
33. **Kwak, J., Kim, Y., Member, S., Lee, J. (2015).** DREAM : Dynamic resource and task allocation for energy minimization in mobile cloud systems. EEE Journal on Selected Areas in Communications, Vol. 33, No. 12, pp. 2510–2523.
34. **Missionmode (2015).** http://www.missionmode.com/15-disaster-and-crisis-apps-for-android/.
35. **Fajardo, J.T.B., Oppus, C.M. (2010).** Implementation of an Android-Based Disaster Management System. Proceedings of the 9th WSEAS international conference on Electronics, hardware, wireless and optical communications, pp. 126–130.
36. **Rahman, M.A., Hossain, M.S. (2017).** A location-based mobile crowdsensing framework supporting a massive Ad hoc social network environment. IEEE Communications Magazine, Vol. 55, No. 3, pp. 76–85.
37. **Rahman, M.A., Hossain, M.S., Hassanain, E., Muhammad, G. (2018).** Semantic multimedia fog computing and lot environment: slistainabilfly perspective. IEEE Communications Magazine, Vol. 56, No. 5, pp. 80–87.
38. **Hailes, S. (1998).** Power conservation strategy for mobile computers using load sharing. Mobile Computing and Communication Review, Vol. 2, pp. 44–51.
39. **Huang, D., Wang, P., Niyato, D. (2012).** A dynamic offloading algorithm for mobile computing. IEEE Transactions on Wireless Communications, Vol. 11, No. 6, pp. 1991–1995.
40. **Kumar, K., Lu, Y.h. (2010).** Cloud computing for mobile users: can offloading computation save energy?. IEEE Computer, Vol. 43, No. 4, pp. 51–56.
41. **Shu, P., Jin, H., Wen, F., Qu, Y., Li, B. (2013).** eTime: energy-efficient transmission between cloud and mobile devices. IEEE Infocom, pp. 195–199.
42. **Andersson, K., Granlund, D., Ahlund, C. (2007).** M4: Multimedia mobility manager: A seamless mobility management architecture supporting multimedia applications. Proceedings of 6th International Conference on Mobile and Ubiquitous Multimedia, Vol. 284, pp. 6–13.
43. **Qiu, M., Ming, Z., Wang, J., Yang, L.T., Xiang, Y. (2014).** Enabling cloud computing in emergency management systems. IEEE Cloud Computing, Vol. 1, No. 4, pp. 60–67.
44. **Poulymenopoulou, M., Malamateniou, F., Vassilacopoulos, G. (2012).** Emergency healthcare process automation using mobile computing and cloud services. Journal of Medical Systems, Vol. 36, No. 5, pp. 3233–3241.
45. **Palmieri, F., Ficco, M., Pardi, S., Castiglione, A. (2016).**: A cloud-based architecture for emergency management and first responders localization in smart city environments. Computers and Electrical Engineering, Vol. 56, pp. 810–830.
46. **Ramasamy, V., Gomathy, B., Sarkar, J.L., Panigrahi, C.R., Bibudhendu, P., Abhishek, M. (2020).** EMC^2: An emergency management system using mobile cloud computing. IET Networks, Vol. 2, No. 9, pp. 64–73.
47. **Bertoldo, N.A., Hunter, S.L., Fertig, R.A., Laguna, G.W., MacQueen, D.H. (2005).** Development of a real-time radiological area monitoring network for emergency response at Lawrence Livermore National Laboratory. IEEE Sensor Journal, Vol. 5, No. 4, pp. 565–573.
48. **Windmiller, J.R., Wang, J. (2013).** Wearable electrochemical sensors and biosensors: A review. Electroanalysis, Vol. 25, No. 1, pp. 29–46.
49. **Rodgers, M.M., Pai, V.M., Conroy, R.S. (2015).** Recent advances in wearable sensors for health monitoring. IEEE Sensor Journal, Vol. 15, No. 6, pp. 3119–3116.
50. **Sarkar, J.L., Panigrahi, C.R., Pati, B., Das, H. (2015).** A novel approach for real-time data

ISSN 2007-9737

management in wireless sensor networks. Proceeding of 3rd International Conference on Advanced Computing, Networking and Informatics, Vol. 2, pp. 599–607.

51. **Mall, R. (2007).** Real time systems. Theory and practice. Pearson Publication, London.

52. **Facchinetti, T., Almeida, L., Buttazzo, G.C., Marchini, C. (2004).** Real-time resource reservation protocol for wireless mobile ad hoc networks. Proc. of IEEE Int. Real-Time Systems Symp. (RTSS), pp. 382–391.

53. **Li, H., Shenoy, P., Ramamritham, K. (2004).** Scheduling communication in real-time sensor application. Real-Time and Embedded Technology and Applications Symposium (RTAS), pp. 10–18.

54. **Caccamo, M., Zhang, L.Y., Sha, L., Buttazzo, G. (2002).** An implicit prioritized access protocol for wireless sensor network. Proceeding of IEEE Real-Time Systems Symposium, pp. 39–48.

55. **Chipara, O., Lu, C., Roman, G. (2013).** Real-time query scheduling for wireless sensor networks. IEEE Transactions on Computers, Vol. 62, No. 9, pp. 1850–1865.

56. **Rhee, I., Warrior, A., Min, J., Xu, L. (2006).** DRAND: Distributed randomized TDMA scheduling for wireless ad hoc networks. MobiHoc, pp. 190–201.

57. **Chipara, O., Lu, C., Stankovic, J., Roman, G. (2011).** Dynamic conflict-free transmission scheduling for sensor network queries. IEEE Transactions on Mobile Computing, Vol. 10, No. 5, pp. 734–748.

58. **Saifullah, A., Xu, Y., Lu, C., Chen, Y. (2011).** End-to-end delay analysis for fixed priority scheduling in Wireless HART networks. Proceeding of IEEE Real-Time and Embedded Technology and Applications symp., pp. 13–22.

59. **Panigrahi, C.R., Sarkar, J.L., Pati, B., Verma., R. K. (2016).** CQS: A conflict-free query scheduling approach in wireless sensor networks. 3rd International Conference on Recent Advances in Information Technology (RAIT), pp. 13–18.

60. **Panigrahi, C.R., Sarkar, J.L., Pati, B. (2018).** CETM: A conflict-free energy efficient transmission policy in mobile cloud computing. Int. J. Commun. Netw. Distrib. Syst., Vol. 20, No. 2, pp. 129–142.

Article received on 26/07/2020; accepted on 23/10/2020.
Corresponding author is V. Ramasamy.