

# An Energy Aware Strategy for Distributed Cache Discovery with Stale Page Reduction



Y.J.Sudha Rani, M.Seetha

**Abstract:** With the immense growth in the field of mobile communication, a good number of complex applications are now available for the mobile devices. The complex applications are made available to comply with the client demand for the higher performing and higher capable applications, which can be accessible from any locations and any devices. Thus, the application developers have attempted to make highly scaled applications to be deployed on mobile communication devices. The larger applications have higher demand for memory and processing capabilities. Thus, making the similar infrastructure available on the mobile computing environments was always a challenge. Nonetheless, with the availability of distributed computing architecture, the bottleneck for the computing capabilities for these complex applications can be handled. Nonetheless, the memory capabilities for the applications must be addressed more sophisticated manner using distribution of the memory and sharing of the data. Henceforth, distributed caching came under existence. A conveyed cache is an augmentation of the customary idea of cache utilized in a solitary district. A conveyed cache may traverse various servers with the goal that it can develop in size and in value-based limit. It is for the most part used to store application information living in database and web session information. One of the most popular technique for making the cache available is to perform cache discovery operations in the network. A number of parallel research attempts are made to identify the accurate place in the network to create or build the distributed cache network. However, the most of the parallel research attempts are criticised for considering single dimensions for cache discovery as few of the work focuses on distance, few of the work focuses on density and some of the works focuses on page replacement policies applicable on mobile computing environments as MANETs or WSN. Henceforth, the demand of the research is to consider multiple parameters for cache discovery and build a framework to automatically define the cache distribution. Hence, this work proposes a novel architecture or framework to detect the cache distribution based on distance, stale page reduction mechanism and finally the energy optimization. The outcome of the research is to automate the recommendation of cache discovery and increase the network life time by 90% compared to the existing methods for cache discovery. In order to handle the complex processing of the proposed algorithms, this work deploys machine learning methods to reduce the time complexity.

**Index Terms:** Cache Management, Stale Page, MANET Caching, Distributed Cache, Cache Data Pattern Analysis, Cache Stale Data Replacement, Energy Aware Caching

## I. INTRODUCTION

The enhancement in the capacitive allocation of the data for any MANET architecture has become the most vital scope for the research. Bounded by the limitations of the mobile communication device, the primary objective is to increase the data storage capacity for the mobility devices. Also, the mobile devices suffer from frequent disconnection from the network due to the higher distribution of the networks. Henceforth, the researchers have aimed to increase the cache cooperation by deploying various methods for cache data management.

The notable outcome from the research method proposed by G. Kauret al. [1] have furnished the demand of mobility awareness in the cache management strategies. The location awareness of the mobility devices cannot be stable for all the regions or for all routing types or for all application demands, hence the cache management must be location specific. The work of K. Tassum et al. [2] have demonstrated the benefits from location dependent cache management schemas. The location dependencies must also be considered for the affinity of the applications deployed on the network. The work by D. Lee et al. [3] ensures this principle.

Henceforth, considering the recent research factors, it is the demand for the research to address the following problems:

- Firstly, Analyse the distributed caching demands for routing algorithms in MANETs.
- Secondly, Propose a novel distributed caching framework with stale page reduction mechanism and reduction of framework complexity.

Thus, this work builds the solution towards these identified problems.

The rest of the work is organized such that in the Section – II, the fundamental distributed cache discovery method is analysed, in the Section – III, the recent research outcomes are analysed, in the Section – IV the problem is mathematically formulated, the Section – V defines the novel proposed algorithms, in the Section – VI, the results obtained from the proposed algorithms are discussed and finally, the conclusion of the research is presented in the Section – VII.

## II. FUNDAMENTALS OF DISTRIBUTED CACHE DISCOVERY

In this section of the work, the fundamental strategies for cache discovery for MANET architecture is discussed. MANET may work as independent style, or they can be the piece of the bigger web.



Revised Manuscript Received on November 30, 2019.

\* Correspondence Author

Y.J.Sudha Rani\*, Research Scholar, JNTUH- Hyderabad, Telangana, India

Dr. M.Seetha, HOD, Professor, G Narayanamma Institute of Technology and Science Hyderabad, Telangana, India

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

They structure profoundly powerful self-governing topology with the nearness of one or various distinctive handsets between hubs. The primary challenge for the MANET is to prepare every gadget to consistently keep up the data required to appropriately course traffic. MANETs comprise of a shared, self-framing, self-recuperating system MANET's around 2000-2015 commonly impart at radio frequencies (30MHz-5GHz). Considering the mobility factor, the allocation of the mobility of data, it is a prime concern for the MANET architects to configure the dynamic storage in terms of cache memories.

The basic cache discovery principle is discussed here. Assuming the set of nodes,  $N$ , is configured in the network and every node in the network can be considered as  $n_i$ , thus, the relation can be formulated as:

$$N = \sum_{i=1}^{\varepsilon} n_i \text{ (Eq. 1)}$$

Where,  $\varepsilon$  denotes the number of nodes in the network. Also, assuming the set of applications deployed,  $A$ , in the network and each application can be denoted as  $a_i$ , the relation can be established as,

$$A = \sum_{i=1}^{\eta} a_i \text{ (Eq. 2)}$$

Where,  $\eta$  denotes the number of applications in the network. Further, if few of applications,  $k$ , are having affinity towards few numbers of nodes,  $r$ , then the relationship can be represented as,

$$\sum_{i=1}^k a_i \Rightarrow \sum_{i=1}^r n_i = \mathfrak{R} \text{ (Eq. 3)}$$

Where,  $\mathfrak{R}$  denotes the set of relevant nodes in the network. The affinity of the applications must be towards the data available in the network, where the data items can be represented as  $d_i$ , thus the relation between the data and the applications can be represented as,

$$\sum_{i=1}^k a_i \Rightarrow \prod_k d_i \text{ (Eq. 4)}$$

Again, these data items can be further available on few selected nodes in the network.

$$\prod_k d_i \subset \text{data\_nodes}(\sum_{i=1}^{\varepsilon} n_i) = \mathfrak{S} \text{ (Eq. 5)}$$

Where,  $\mathfrak{S}$  denotes the set of nodes with data items available in the network.

In the ideal situations, the set of relevant nodes must be identical to the set of nodes with data items in them and can be formulated as,

$$\mathfrak{R} \in \mathfrak{S} \text{ (Eq. 6)}$$

Nonetheless, these sets are not always identical, and the data must be searched in the nearby nodes as well for the cache discovery.

Assuming, the nearby nodes,  $s$ , holding the relevant data is also available in the network, thus the final formulation can be presented as,

$$\mathfrak{R} \in \mathfrak{S} + s \text{ (Eq. 7)}$$

The access to the nearby nodes can always be challenged by distances and traversing the network nodes causes energy deprecation.

Hence, reducing the nodes or finding the nodes which contains major part of the data can significantly reduce the

time complexity and reduce energy consumptions for of cache discovery.

Thus, the objective of the modern research is to reduce the number of relevant nodes or build the cache over lesser number of nodes which can contain all the required data.

In further section of this work, the strategic solution to this problem is furnished.

### III. OUTCOMES FROM THE PARALLEL RESEARCHES

In this section of the work, the recent research outcomes are analysed.

In an exceptionally delegated framework, flexible centre points talk with each other using multi bounce remote associations. There is no stationary structure, for instance, base stations. Each centre in the framework in like manner goes about as a switch, sending data bundles for various centres. A central test in the arrangement of Ad hoc Networks is the improvement of a dynamic controlling show that can beneficially find courses between two giving centre points. Many controlling shows have been proposed till date anyway none of them are adequately successful to perform well in all circumstances. The guiding show must have the ability to remain mindful of the abnormal state of centre movability that changes the framework topology as often as possible unquestionably and whimsically. The openness of negligible exertion PCs and palmtops with radio interfaces have restored the energy for the field of unrehearsed frameworks.

In the recent research application dependencies or the application affinity is given highest priority. The work by G. Shanmugarathinam et al. [4] have showcased the thread-based models for building the consistent cache support for the network. This work is partially criticised for not making the architecture location aware, which might have improved the consistency. Through, the fundamental challenged faced by the research proposal of Anandharaj et al. [5] was successfully addressed in that work. The similar recommendation were also made by P. T. Joy et al. [6] and conveyed the standard bottlenecks of this research.

In the other direction K. Kottursamy et al. [7] have proposed to focus on the affinity of the data, rather than the location or the application. This proposal has intrigued the question of progressive or adaptive algorithms for cache managements in MANET architecture. The work by R. Tiwari et al. [8] have demonstrated the use of adaptive algorithms for cache management.

Finally, the cache replacements strategies came into the adaptation by various researchers. The replacements of cache data can significantly reduce the efforts to make sustainable cache. The work by H. Chavan et al. [9] have introduced the cache replacement policies for MANETs. With this initiation, the work on cache replacements have started and a good number of research attempts can be observed. The work by C. Li. [10] establishes the policies for cache updation by inclusion. This policy is briefly criticized for limitation in the cache sizes on the MANET nodes. The case study depending on this proposal was carried out in the work by J.H. Lee et al. [11]

for the e-commerce applications and achieves higher performances with the selective inclusion of the data. These comparative analyses were carried out by C. Chakravorty et al. [12].

Nonetheless, the existing architectures for the cache replacement policies are highly complex and a newer method is needed to reduce the complexity.

Hence, in the next section of this work, the problem is formulated.

#### IV. PROBLEM FORMULATION

In this section of the work, the problem is formulated using the mathematical model.

Stale information is an antique of storing, in which an item in the cache isn't the latest adaptation focused on the information source. To dodge stale information, actualize a fitting cache locking methodology. Cache locking controls when forms read or compose an article. Contingent upon how you arrange it, cache locking decides if a procedure can peruse or compose an article that is being used inside another procedure. A well-overseen cache makes your application increasingly proficient. There are not very many cases in which you turn the cache off completely, in light of the fact that the cache lessens database get to and is a significant piece of overseeing object personality.

**Lemma:** Reduction of the stale pages in the network nodes can improve the cache discovery.

**Proof:** Firstly, assuming that each data item, D, can be broken into two parts as required data  $D_R$ , and the stale data  $D_S$ , thus the relationship can be formulated as,

$$D \rightarrow \sum_{i=1}^n D_R + \sum_{i=1}^m D_S \quad (\text{Eq. 8})$$

Where n and m denote the number of data items for required and stale data items respectively.

Assuming the set of nodes, N, is configured in the network and every node in the network can be considered as  $n_i$ , thus, the relation can be formulated as:

$$N = \sum_{i=1}^k n_i \quad (\text{Eq. 9})$$

Where, k denotes the number of nodes in the network. Further assuming that the required data items needs  $t_1$  and stale data items needs  $t_2$  number of nodes to be completely allocated, then the following formulations can be realized,

$$D_R = K * t_1 = n / t_1 \quad (\text{Eq. 10})$$

And,

$$D_S = K * t_2 = m / t_2 \quad (\text{Eq. 11})$$

Henceforth, each node in the network can contain fewer parts of the required and stale data items as represented here,

$$n_i \rightarrow K / t_1 + K / t_2 \quad (\text{Eq. 12})$$

Further, the allocation relation between the required and the stale data items can be realized from Eq. 10 and Eq. 11 as,

$$D_R * D_S = K * t_1 * K * t_2 \quad (\text{Eq. 13})$$

Further,

$$D_R = (2K * 2t_1 * t_2) / n \quad (\text{Eq. 14})$$

Henceforth, from Eq. 12, if the stale part of the data items can be removed from the node data, then each node can contain more required data items as,

$$n_i \rightarrow K / t_1 + (3K * 2t_1 * t_2) / n * t_2 \quad (\text{Eq. 15})$$

Thus, it is to be realized that with the reduction in the stale data items, more required data can be allocated into each node in the network and with lesser number of nodes the distributed cache can be formed.

Henceforth, with this principle built, in the next section of the work, the proposed algorithm for stale page reduction is furnished.

#### V. PROPOSED ALGORITHM & FRAMEWORK

In this section of the work, the proposed algorithms are furnished.

Firstly, for a successful stale cache data item discovery it is the prime requirement to identify the data item demand of the upcoming routing requirements. Hence, the first algorithm is designed to identify the routing patterns.

Algorithm - 1: Data Demand Pattern Analysis Algorithm (DDPA)
Step - 1. Accept the list of nodes N[]
Step - 2. For each node n[i]
a. Check the routing table T[i]
b. For each T[i]
i. Accept the list of data items D[]
ii. For node N[i]
1. Check for the availability of D[i]
2. If D[i] is present
a. Preserve the D[i]
3. Else
a. Mark D[i] under routing pattern R[]
Step - 3. Report R[]

The routing table information structure contains all the data important to advance an IP information bundle toward its goal. Each routing table passage portrays the accumulation of best ways to a specific goal. When sending an IP information parcel, the routing table passage giving the best match to the bundle's IP goal is found. The coordinating routing table passage at that point gives the following bounce towards the parcel's goal. OSPF likewise accommodates the presence of a default course. At the point when the default course exists, it coordinates all IP goals.

Algorithm - 2: Stale Page Replacement and Cache Discovery Algorithm (SPR-CD)
Step - 1. Accept the application set as A[]
Step - 2. For each item as A[i]
a. Build the routing path with node set N'[]
b. For each N'[i]
i. Check for data items with R[]
ii. If R[] is present in the node data
1. Mark as No-Stale
iii. Else
1. Check for data items not in N'[] from R[]
2. If data items to be replaced > 50%
a. Discard the node
3. Else
a. Replace the data items not in R[n]
c. Build the final node list without discarded list with distance measures
Step - 3. Report final node list as N1[] as Distributed Cache

# An Energy Aware Strategy for Distributed Cache Discovery with Stale Page Reduction

In PC preparing, on the off chance that a processor changes the estimation of an operand and, at that point, at a consequent time, gets the operand and acquires the old instead of the new estimation of the operand, at that point it is said to have seen stale information. On a uniprocessor, stale information can't go on without serious consequences. It would imply that the processor disregarded essential assumptions regarding its own conduct. On shared memory multiprocessors, be that as it may, it is viewed as adequate for machines to create stale information on operands shared between procedures. For such operands, the desire is that projects will avoid potential risk to keep stale information from being seen.

Further, the results obtained from the proposed deployed framework is highly satisfiable and discussed in the further section of the work.

## VI. RESULTS AND DISCUSSION

In this section of the work, the results obtained from the deployed algorithms are analysed. The results are discussed in few sections as node availability in terms of distance, stale page replacement success and energy optimization due to the proposed algorithm.

### A. Node Distance Analysis

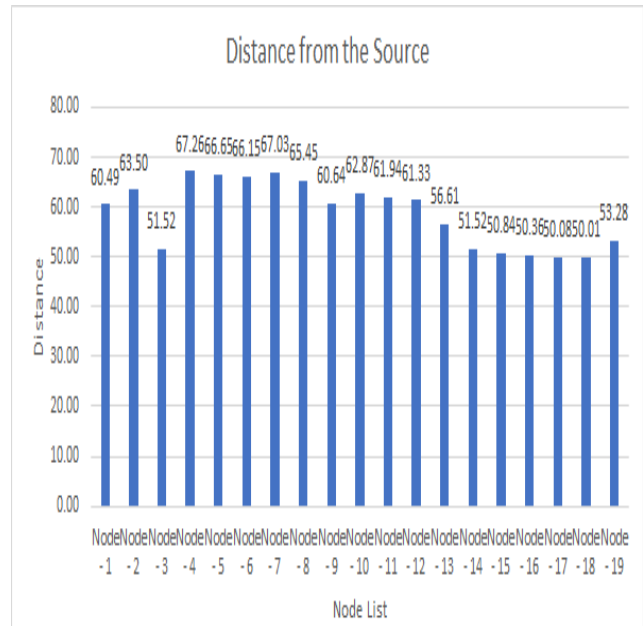
During the initial setup of the network, the distance is measured from the source node for each node in the network [Table – 1]:

**TABLE I**  
NODE DISTANCE ANALYSIS

Node Name	Distance from the Source (Meter)
Node - 1	60.49
Node - 2	63.50
Node - 3	51.52
Node - 4	67.26
Node - 5	66.65
Node - 6	66.15
Node - 7	67.03
Node - 8	65.45
Node - 9	60.64
Node - 10	62.87
Node - 11	61.94
Node - 12	61.33
Node - 13	56.61
Node - 14	51.52
Node - 15	50.84
Node - 16	50.36
Node - 17	50.08
Node - 18	50.01
Node - 19	53.28
Node - 20	0.00

Here, it is natural to realize that Node – 20 being the source node can showcase zero distance.

The result is visualized graphically here [Fig – 1].



**Fig. 1 Distance Vector Analysis**

Further, based on the distance measure, as calculated threshold 58.82 meters most closely situated nodes are selected for initial cache verifications [Table – 2].

**TABLE II**  
NODE SELECTION BASED ON DISTANCE VECTOR

Node Name	Node Selection Status
Node - 1	Selected
Node - 2	Selected
Node - 3	-No-
Node - 4	Selected
Node - 5	Selected
Node - 6	Selected
Node - 7	Selected
Node - 8	Selected
Node - 9	Selected
Node - 10	Selected
Node - 11	Selected
Node - 12	Selected
Node - 13	-No-
Node - 14	-No-
Node - 15	-No-
Node - 16	-No-
Node - 17	-No-
Node - 18	-No-
Node - 19	-No-
Node - 20	-No-

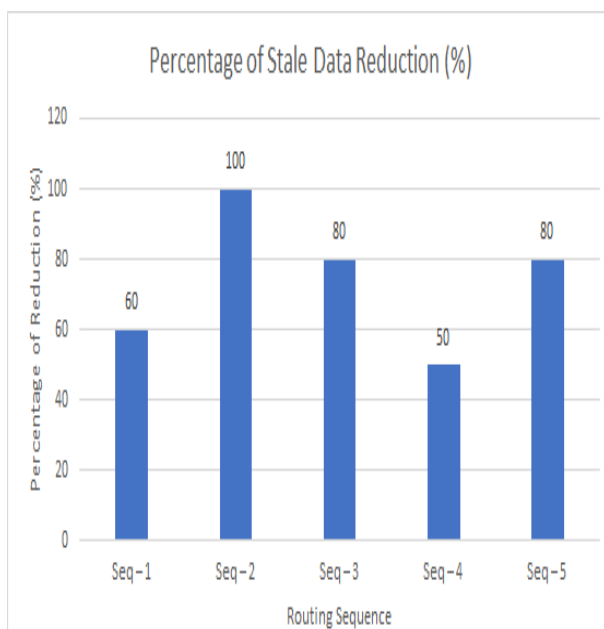
### B. Stale Data Item Replacement Analysis

The stale data items replacement algorithm result is analysed here [Table – 3]:

**TABLE III**  
STALE PAGE REPLACEMENT RESULT

Routing Sequence	Data Items Replaced	Percentage of Stale Data Reduction (%)
Seq - 1	Cache Replaced: 7 Cache Replaced: 1	60
Seq - 2	Cache Replaced: 3 Cache Replaced: 4 Cache Replaced: 5 Cache Replaced: 6 Cache Replaced: 1 Cache Replaced: 7	100
Seq - 3	Cache Replaced: 7 Cache Replaced: 1 Cache Replaced: 4 Cache Replaced: 3	80
Seq - 4	Cache Replaced: 4	50
Seq - 5	Cache Replaced: 0 Cache Replaced: 1 Cache Replaced: 3 Cache Replaced: 0	80

The result is visualized graphically here [Fig - 2].



**Fig. 2** Stale Data Reduction Analysis

**C. Energy Absorbing Value Analysis**

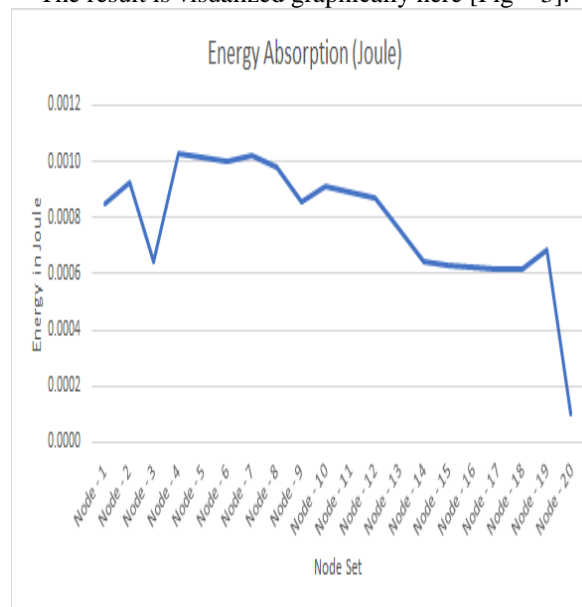
Due to the additional algorithms, the absorption of the energy is increased, and the analysis is furnished here [Table - 4]:

**TABLE IV**  
ENERGY ABSORPTION ANALYSIS

Node Name	Energy Absorption (Joule)
Node - 1	0.0009
Node - 2	0.0009
Node - 3	0.0006
Node - 4	0.0010
Node - 5	0.0010
Node - 6	0.0010
Node - 7	0.0010
Node - 8	0.0010
Node - 9	0.0009
Node - 10	0.0009
Node - 11	0.0009

Node - 12	0.0009
Node - 13	0.0008
Node - 14	0.0006
Node - 15	0.0006
Node - 16	0.0006
Node - 17	0.0006
Node - 18	0.0006
Node - 19	0.0007
Node - 20	0.0001

The result is visualized graphically here [Fig - 3].



**Fig. 3** Energy Absorption Analysis

**VII. CONCLUSION**

In the mobile device communication, the limitation of computing or processing capabilities and the storage or data sharing capabilities have initiated the research in various domains of study. Building the cooperative cache is always been a challenge to the research. Nonetheless, a wide range of research attempts were made to reduce the complexity of the framework works. Nevertheless, the existing research attempts were highly criticised for higher energy consumption and low flexibility. This research identifies the bottleneck of cache replacements and proposes novel schema for identification and replacement of stale cache. A passed-on cache is a growth of the standard thought of cache used in a single locale. A passed-on cache may navigate different servers with the objective that it can create in size and in esteem based farthest point. It is generally used to store application data living in database and web session data. A standout amongst the most famous system for making the cache accessible is to perform cache disclosure activities in the system. Henceforth, this research proposes a novel strategy for Data Demand Pattern Analysis for understanding the data item demand and making the cache replacement an intelligent policy. Further this work also deploys a Stale Page Replacement and Cache Discovery Algorithm for making the page replacement possible in order to accommodate more demanding pages, which ensures a better performing MANET architecture with reduced energy consumption.



## REFERENCES

1. Kaur, G., Saini, J.: Mobility aware cache management in wireless environment. In: International Conference on Methods and Models in Science and Technology, India, pp. 291–297 (2010)
2. Tassum, K., Syed, M., Damodaram, A.: Enhanced-location-dependent caching and replacement strategies in mobile environment. *Int. J. Comput. Sci. Issues* 8(2), 160–167 (2011)
3. Lee, D., Chin, S., Min, J.: Efficient resource management and task migration in mobile grid environments. *J. Commun. Comput. Inf. Sci.* 78(1), 384–393 (2010)
4. Shanmugarathinam, G., Vivekanandan, K.: Thread based cache consistency model for mobile environment. *Int. J. Comput. Sci. Commun.* 2(2), 315–318 (2011)
5. Anandharaj, G., Anitha, R.: A Distributed cache management architecture for mobile computing environments. In: International Advance Computing Conference, India, pp. 642–648 (2009)
6. Joy, P.T., Jacob, K.P.: A comparative study of cache replacement policies in wireless mobile networks. In: Proceedings of the Second International Conference on Advances in Computing and Information Technology (ACITY) vol. 176, pp. 609–619 (2012)
7. Kottursamy, K., Raja, G., Saranya, K.: A data activity-based server-side cache replacement for mobile devices. In: Proceedings of the International Conference on Artificial Intelligence and Evolutionary Computations in Engineering Systems, pp. 579–589 (2016)
8. Tiwari, R., Kumar, N.: An adaptive cache invalidation technique for wireless environments. *Telecommun. Syst.* 62(1), 149–165 (2016)
9. Chavan, H., Sane, S.: Mobile database cache replacement policies: LRU and PPRRP. In: International Conference on Computer Science and Information Technology, India, pp. 523–531 (2011)
10. Li, C., Wang, D., Zhang, X.: Enhanced adaptive insertion policy for shared caches. In: International Conference on Advanced Parallel Processing Technologies, Berlin, pp. 16–30 (2011)
11. Lee, J.H., Kwon, S.J., Chung, T.S.: ERF: efficient cache eviction strategy for e-commerce applications. In: International Conference on Mobile and Wireless Technology, pp. 295–304 (2018)
12. Chakravorty, C., Usha, J.: Cache management issues. In: Mobile Computing Environment, International Journal of Mobile Network Communications & Telematics (IJMNCT), vol. 2, No.1, February 2012.