Wireless Personal Communications (2020) 112:2643–2662
https://doi.org/10.1007/s11277-020-07168-7

Data Replication in Mobile Edge Computing Systems to Reduce Latency in Internet of Things

N. Saranya[1] · K. Geetha[2] · C. Rajan[3]

Published online: 23 January 2020

Abstract

The progress in the development in the field of information technology has brought the Internet of Things (IoT) into existence to play a crucial role in our daily lives. There are interconnected sensors or devices that can both collect and also exchange various data among themselves by employing a modern network of communication as an infrastructure that has been connected by many millions of the IoT nodes. After this, there are various applications of the IoT that may be able to provide accurate and fine-grained services to the users. Using this as a strategy which can mitigate an escalation to the congestion of resources, Edge Computing is emerging as the new paradigm that solves the needs of localized computing and the IoT. The Mobile Edge Computing (MEC) has been emerging to handle the volume of data produced and this can reach a latency of demand of the IoT applications that are intensive in terms of computation. Even though the MEC has advanced in terms of latency of service and has been solidly investigated, the efficiency of data usage and security are not identified clearly. Replication of data is well suited for improving the time taken for a response, global traffic and data sharing as even at the time of server disconnection this can be done. In this work, efficient techniques of data replication for the Mobile Ad hoc Networks (MANET) like the simple and the random applications are evaluated for improving availability of data which considers all the issues that are related to the MANET like consumption of power, availability of resource, time taken for response and consistency management. The results of the experiment have shown that a random algorithm for replication can achieve a bandwidth that is better in terms of savings compared to a simple replication algorithm.

Keywords Internet of Things (IoT) · Edge computing (EC) · Mobile edge computing (MEC) · Data replication · Mobile Ad hoc networks (MANET) · Cloud computing · Simple replication algorithm and random replication algorithm

✉ N. Saranya
nsaranyann@gmail.com

1 Anna University, Chennai, India

2 Excel Engineering College, Komarapalayam, India

3 K S Rangasamy College of Technology, Tiruchengode, India

1 Introduction

Before the end of 2015, there is expected to mobile cellular subscriptions that are more than 7 billion corresponding to a rate of penetration of about 97% from 738 million in the year 2000 according to the ITU report. This only shows that mobile phones and tablets have exceeded the numbers of laptops and computers. Additionally, both intensive and ubiquitous usage of mobile phones has been accompanied with the evolution in the architecture of mobile network (2G/3G/4G/5G) that has an explosive growth in demand for services of a higher bandwidth (such as the video on demand) owing to the concept of "big bang" among entertainment applications and social networking. In spite of the evolution of the components that are Nano-technological in nature and the capability of storage of the portal devices, they do not have battery life or power of computing, thus permitting them to be able to perform in an effective manner. This requires the involvement of cloud computing. So, data centralization found in the core of these mobile networks can make the bandwidth of the solution intensive aside from its high latency [1].

The Mobile Ad hoc Network (MANET) is an autonomous collection of the mobile nodes to communicate over the wireless links that are bandwidth-constrained. The network is generally decentralised and activities such as the discovery of topology and message delivery will be executed by nodes. But the availability of data in the MANET is low owing to its topology being dynamic [2].

In the case of a distributed database system, the data are normally replicated in order to bring about an improvement to the dependability and the availability. The data are stored on the computers for the reduction of costs. But a reliable system does not always have a high level of availability. Another critical issue in replication of data is exactness of data (called data consistency) that is present in the systems. Failure of communication among two mobile nodes in the MANET may result in partitioning of the network in which the division is as isolated sub-networks that result in chances of an inconsistency of data. Additionally, there may be a slow response from some nodes that may appear as if the network is partitioned even if it is not so.

Further, there could be other issues in the MANET resulting in replication of data such as [3]:

- Power Consumption: a major issue in data replication is the limitation of a battery. In case the node has low power with many data items accessed, it can be drained out and may not provide services.
- Node mobility: the MANET's topology is arbitrary and is difficult to locate. So data for every node may not always be available.
- Availability of resources: as nodes in the MANET are portable, they have limited memory and replication of all items may not be possible.
- Response time: this refers to the time a client takes to access data from servers. If the server is located far away response time will automatically be higher to be able to service clients.
- Consistency Management [4]: one more issue in replication of data is the synchronization of replicas. If the replica is updated frequently, others may become invalid. And invalid accesses also consume power and require rollbacks that can become a major issue for nodes with poor resources.

Internet of Things (IoTs) is looked at as a new paradigm where both networking and computing capabilities have been embedded in a conceivable object. It makes use of these capabilities for the purpose of querying the object and its state for a change in case it is possible. While looking at it in common parlance, the IoTs are a new world in which all devices or appliances can be connected to the network. They may be used collaboratively for achieving complex tasks needing a high level of intelligence. For the purpose of interconnection and intelligence, the IoT devices have been equipped using embedded sensors, transceivers, processors, actuators and so on. It is an agglomeration of different types of technology that work in tandem. The sensors and the actuators are the devices that help in interaction with the physical environment. Both data storage and data processing may be done in the network or by using a remote server. The IoTs will find different applications in transport systems, automation of homes, environment monitoring, conservation of energy, social life, entertainment, education, fitness and health care [5].

Edge Computing (EC) has been gaining a lot of popularity here since the IoT is now very common in data processing found on the edge of the network. There are many approaches like the cloudlet, the MEC and fog computing that are providing some complementary solutions to cloud computing that can bring down processing of data on the network edge. In short, the EC is the term that is used to represent fog computing, the micro clouds, and the MECs. Power, computing and storage are taken to the edge of networks for increasing the availability and reducing the latency, and finally overcoming the bandwidth-hungry and delay-sensitive applications found near the source of data [6]. The MEC technology, that was introduced recently by the European Telecommunications Standards Institute (ETSI) Industry Specification Group (ISG) made on the MEC, has offered capabilities of cloud computing in the Radio Access Network (RAN) with information technology service and environment in the edge of mobile networks that was close to the subscribers. The environment was characterised by a latency that was ultralow, high bandwidth and real-time access to the information of radio network which was leveraged by the applications and the Quality-of-Experience (QoE) platforms of optimization. The Edge is the base station (evolved node B (eNB), the Radio Network Controller (RNC) and so on] where the data centres that are close to the network (situated at the aggregation points). The operators may be able to open RAN edge to the authorized third parties, thus, permitting them to deploy all innovative services and applications towards the subscribers of the mobiles and their vertical segments. The MEC has brought about a significant level of benefits not just for the operators but also to third parties and the Over-The-Top (OTT) companies having an opportunity to be able to run applications at the mobile network and its edge which is close to the mobile subscribers. Lastly, a common consumer can further experience an improved level of performance and some new services that are offered by this MEC system [7].

A MEC reference architecture will enable implementation of the applications of the MEC as the entities that are software-only running on the host of the MEC. A mobile edge platform may offer an essential functionality and environment needed for running this MEC application. These applications will run as the VM on the virtualization infrastructure and will be able to interact with their mobile edge platform for supporting some procedures that are related to the application of life-cycle. Also, this virtualization infrastructure has a data plane to execute various traffic rules received by a mobile edge platform thus routing traffic among external networks, local networks, and applications. A MEC host level management contains a mobile edge platform manager along with a virtualization infrastructure manager. The first one will manage the application life cycle with the rules and the requirements of applications that include authorization of services, resolving of conflicts, traffic rules and authorization of services. The latter will be responsible for the allocation, the management and the releasing of the visualized (computing, storing, and finally networking) resources belonging to the infrastructure of virtualization [8].

An operations support system will receive requests by user applications through a proxy of life-cycle management or by the operators and third-party customers that face the service portal. This operation support system will decide as to whether the requests have been granted. If so, they will be forwarded to a MEC orchestrator to be processed. This MEC orchestrator will be the core functionality and will maintain an overall view that was based on the MEC hosts deployed, the resources available, the MEC services and finally the topology. For the purpose of performance, the costs, the scalability, and the operator preferred deployments, the MEC will support various scenarios of deployment like the macro base station (eNodeB) site, found at the 3G RNC site, or a technology of multi-radio access and cell aggregation site, which is at a point of aggregation (known as the edge of its core network as in a new distributed data centre). The problem of network planning in the determination of where the MEC servers may be optimally installed among the sites available in order to identify a trade-off between the costs of installation and the Quality of Service (QoS) [9].

Identified are many benefits that are connected to the MEC which is now very promising to the Mobile Network Operators (MNOs), and the Application Service Provider (ASP), which is in addition and is also quite befitting the providers of content, the providers of middleware, IT, network equipment vendors and the OTT players. Consumption of power is, however, a primary concern there. The task of computation is sent to the external resource-rich systems for increasing User Equipment (UE) and its battery life. Additionally, the scalability and the reliability of the distributed virtual server is identified. In connection to the actors (the end users, the ASPs and the MNOs) the benefits of the MEC are [10]:

- The operators of the mobile network will be able to enable a RAN access to the vendors of third parties for deploying applications or services in a much more agile and flexible manner.
- The providers of application services were able to gain profits by the MEC enabled Infrastructure-as-a-Service (IaaS) platform which was at the network edge making the ASP services quite scalable with higher bandwidth and lower latency.
- The end users were able to experience a fast application in terms of computation by means of an offloading technique handled by the servers of the MEC in the RAN.

In the environment of the IoT there were some massive devices introducing a demand for the storage of data. The correctness of such sensitive data is quite imperative in areas of traffic or speed of smart vehicles, records of energy consumption or personal health data. So, these distributed storage systems involve the IoT environments in order to handle a massive demand and also ensure the accuracy of data. These distributed storage systems will be able to increase dependability and further extent Mean Time to Failure (MTTF) by means of using replication of data. Here, data will be divided into several pieces and every piece will have a fixed size with code blocks. The data pieces will have fixed overlaps with one another and the data that is stored on every piece will be reconstructed from the related pieces. EC-based storage will be a distributed storage system, and logically and physically well-distributed. So, with the EC-based storage assistance, any sensitive IoT data may be replicated and all data pieces stored in various geological locations. This can mitigate data loss to a significant level [11].

In order to overcome these issues, a simple and random algorithm for replication in the MEC system has been proposed. The latency of IoT is reduced by the proposed Data Replication methods. The rest of the investigation has been organized thus. Section 2 has discussed all related work in literature. The different methods employed are explained in Sect. 3. The experimental results have been discussed in Sect. 4 and the conclusion has been made in Sect. 5.

2 Related Works

Zhang et al. [12] had focused on the MEC applications in IoT and also addressed the efficiency of energy as performance offloading which was in terms of the end-user experience. In this connection, the work further presented a hierarchical MEC framework for the green and the low-latency IoT. It also deployed a theoretic approach for the offloading of computation for optimizing the service providers and their utility thus bringing down energy cost and time of task execution using smart devices. The numerical results proved that the scheme proposed had a significant level of enhancement in terms of latency and energy efficiency among MEC applications in IoT.

Satyanarayanan et al. [13] had described GigaSight, which was an Internet-scale repository pertaining to the crowd-sourced video content, that had very strong enforcement of the preferences of privacy and their access controls. This GigaSight architecture was a federated system belonging to the VM-based cloudlets which perform analytics in the video at the edge so as to reduce demand for the ingress bandwidth in the cloud. Denaturing is a reduction that is owner-specific to the fidelity to the content of the video and for preserving privacy. Content-based indexing that searches any other form of the cloudlet analytics is employed and the article is a special issue of these smart spaces.

Li et al. [14] had proposed an adaptive transmission architecture used with the Software Defined Network (SDN) and the EC for being employed to the Industrial Internet of Things (IIoT). After this, in accordance with the data streams that have various constraints of latency, the needs are divided into two different groups: (1) the ordinary and (2) the emergent stream. In a situation of a low deadline, there may be an algorithm of coarse-grained transmission path that is given by means of finding the paths meeting the constraints of time in the hierarchical IoT. Once this is done, by means of employing a Path Difference Degree (PDD), there is an optimum path of routing that is chosen. In a situation with a high-deadline, in case the coarse-grained strategy is well beyond this situation, there may be a fine-grained scheme that is adapted for establishing an effective path of transmission using the method of adaptive power. Lastly, the proposed strategy's performance gets evaluated using simulation.

Ning et al. [15] had considered cooperation in cloud computing and the MEC in the IoTs. It begins from the computation of a single user and its problem of offloading in which the resources of the MEC were not constrained. This was solved by using an algorithm known as the branch and bound. Once this was done, there was a formulation of a problem of multi-user computation offloading problem made in the form of a Mixed Integer Linear Programming (MILP) by means of taking into consideration the resource competition which was NP-hard. Owing to the complexity of computation of the problem which was formulated, there was an Iterative Heuristic MEC Resource Allocation (IHRA) based algorithm that was proposed in order to make a dynamic decision of offloading. The results of the simulation had demonstrated the algorithm to have outperformed all the existing schemes in terms of the latency of execution and efficiency of offloading.

Wang et al. [16] had brought about an introduction of a new and innovative framework known as the Knowledge-Centric Edge (KCE), for dynamic detection of the structure of the network and also manage resources of communication thus leveraging an insightful knowledge that was obtained from the Device-to-Device (D2D) communications that was found among the mobile users. Also, in order to support the network access that was trustworthy, and for meeting various needs of Quality of Service (QoS) in the architecture proposed, some of the Service Providers (SPs) were selected from the InPs using a new KCE server. In this way, the selected resources of the SP were dynamically allocated for meeting the needs of services by the communication techniques of D2D communication. The results of the simulation had demonstrated the proposed framework's effectiveness under various scenarios and their parameter settings.

Verma et al. [17] had proposed another load balancing algorithm that was efficient in an architecture based on the fog-cloud. This algorithm made use of the technique of replication to maintain data in the cog networks that brings down the dependency on the big-sized data centres. There was a comparison made to the current techniques of load balancing within the `cloud-based' infrastructure toward the `cloud-fog' duo which is shown. The primary goal was to balance the load using fog networks and keep the internet from getting too dependent on the cloud by means of having the available data closer to its user end.

For the purpose of this work, leveraging the potential of the techniques of virtualization that were light-weight and container-based was made by Farris et al. [18] who analysed this novel approach supporting the provisioning in the MEC environment. It also presented a new framework within which there was a service replication that was proactive for the stateless applications that are exploited for reducing the service migration time for all the applications that were stateless for meeting their requirements of latency. There were some promising results shown by the performance evaluation in connection to the classic and reactive migration of service. The future direction is to be improve the performance.

Orsini et al. [19] had presented a MEC or the fog computing in order to provide all necessary resources at its logical edge by means of including certain components of infrastructure for creating the ad-hoc mobile clouds. But the approach needs both replication and management of various applications of business logic which was in an untrusted, dynamic and unreliable environment. As a consequence, the work further presented another novel approach that permits the developers of mobile apps to be benefitted from the MEC features. Particularly, it also presented a new programming model to fit the developers of common apps for designing a new and elastic mobile application which was scalable.

Ni and Lin [20] had made an examination of the MEC architecture and had explored the potential of using the MEC for enhancing the analysis of data for the applications of the IoT and also for achieving security of data and efficiency of computation. More specifically, there was an introduction to its overall architecture with many promising IoT applications that were edge-assisted. After this, there was a study made to the privacy, security and the efficiency in order to enhance the computational efficiency for which the EC was used. This also included a secure aggregation of data, secure deduplication of data and a secure offloading of computation. Lastly, there were many interesting directions on the data analysis that was edge-empowered that had been presented for research in the future.

Chandrakala et al. [21] had proposed yet another method of replication for the MANETs aiming at attaining a high level of data availability using an optimum consumption of energy. This was based on the cost function and the trade-off that was between communication and the cost of consumption of energy that was considered. This replication of data was based on the cost function and a trade-off between communication and the cost of energy consumption which is considered. The mechanism of replica selection will identify a number of nodes that are suitable to be placed and will take benefit from the 1-hop cluster formation that is based on the coefficient of entropy. The utilization of energy will be kept minimal along with replication by means of controlling the actual number of replicas and by exploiting the structure of clusters. This algorithm is designed and simulated by using a Network Simulator (NS)-2. It is observed that it can control the consumption of energy efficiently and is also adaptive in relation to the replication rate. The time complexity is need to be reduced in future.

3 Methodology

Data replication is found to be extremely effective in improving the accessibility of data. At the same time, the mobile nodes may have some poor resources and may not be possible for the mobile nodes to manage to have replicas of every item. The section deals with a simple replication algorithm and also a random replication algorithm.

3.1 Simple Replication Algorithm

With shared data having a particular interest for the network nodes, this algorithm can duplicate and also uniformly distribute replicas. The phase is accomplished while creating newer data and when loading their first replica from a fixed server where the ad hoc network may be connected. Such distribution attempts at ensuring access to data for every node. The phase consists of dissemination of data and the replication of the same on a mobile network. There is a distance evaluation among replicas based on the actual number of hops that is accrued out. For the avoidance of redundancy, there should be not two neighbours without replicas of similar data. In the case of node B having two of the immediate neighbours who are A and C and they are not neighbours, it is interesting to find that B can retrieve replicas from different data on both A and C. This will optimize a global time response and will increase accessibility. However, connection failures cannot be ruled out. So for this phase, the data will be replicated based on the nodes that are separated using three different hops. In the case of a failure in connection happening between B and A, then B will retrieve data of A which is on the other node H at two different hops [22]. The algorithm further considers all existing nodes and the current network topology.

Principle: this makes use of a hop counter which is initialized to zero by a node the created or received data. After this, the node will initiate a diffusion to the neighbours of the message which is the Primary Creat (Node Add, D_k, and Hop Cpt) of the primary replica creation of data D_k. Every time the message comes in another node, the hop counter gets incremented by one or gets reinitialized to zero in case there is a replica in existence on the mode visited. The node receiving such a message will process it as:

- In case the counter is found to be equal to the three hops and in case the replica of data does not exist in the cache of a node or in one neighbour, such as in the case of node N_9 as in Fig. 1. Once the replica is created, the node will reinitialize its counter to zero and will send the creation message to neighbours.
- In case the counter is found equal to the three different hops simultaneously, there will be a data replica in one node. This means the “receiver” will initialize a hop counter to

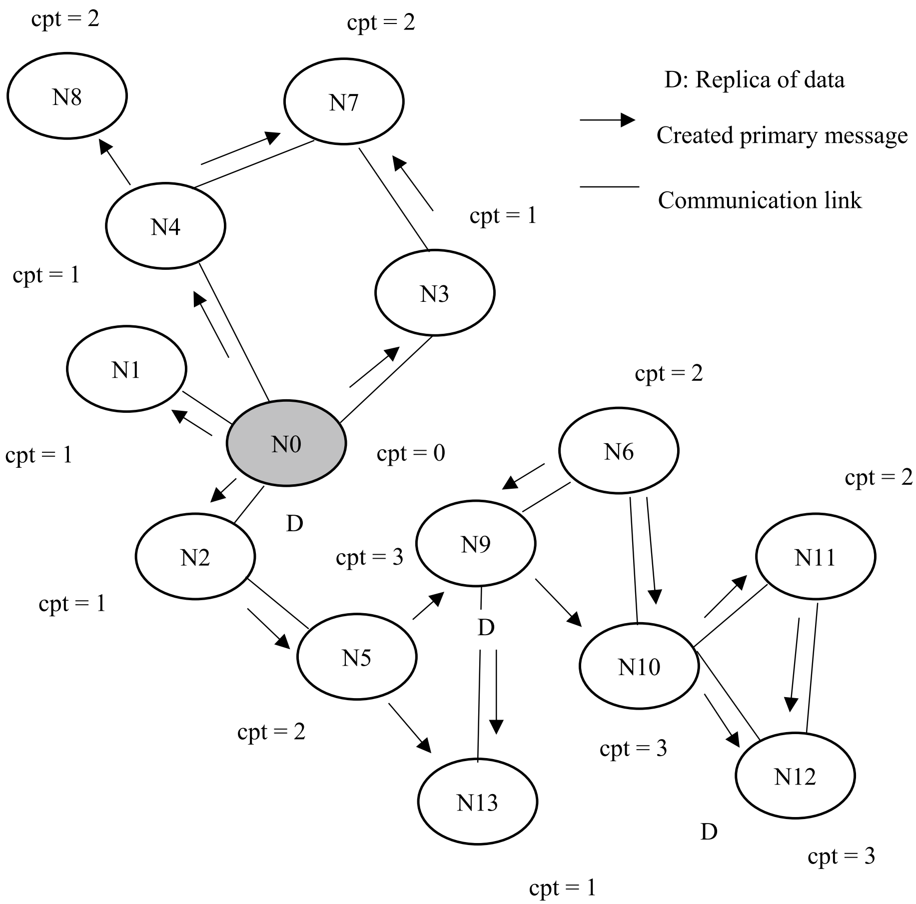

Fig. 1 Example of primary replicas generation

the one sending creation messages to the other nodes (the ones that do not send messages of Primary Creat) and not hold a replica.

For example, in Fig. 1, node N_{10} had received at the very first message Primary Creat of the N_6 and will increment its hop counter to 3. But the N_9 is a node neighbour of the N_{10} that holds data replica, then N_{10} will not create any replica of a similar data for limiting redundancies of the replicas on the neighbours. A message Primary Creat will be transmitted towards the N_{12} and N_{11}. In case the counter is found to be equal to both hops, it will not exist in the following links (the ones beside those where messages were received) so, the node is selected and the replica for node N_{18}.

A node receiving this for the second time will have to ignore it and when placing the primary replicas, if there is no space, a local data with a frequency of low access will be deleted. In Dynamic management of the replicas, the data acceded does not exist locally, there are high chances of it identifying the node neighbour. This can bring down the time of response and the passing band and also energy consumption. The workload will be repartitioned on this network uniformly and this will balance the energy expenses and bring down traffic. This space has been used optimally by means of avoiding the replicas of similar data. The replicas on the two neighbours will be of different data and the node is able to retrieve other data among neighbours. After this, there is an increase in the accessibility of data. The hops existing between the nodes will be used for this.

The primary role of the Primary Creat message is: (1) informing the nodes regarding the creation of new data on this network, (2) estimating the distance between the node and the replica, and (3) distributing data replicas on the same network.

Algorithm: 1/both functions and parameters used in this algorithm will be mainly:

- Primary Creat (Node Add, D_k, Hop Cpt): the message is sent for primary data replication has been found at its node level, Node Add. The Hop Cpt was the node counter that is covered since its last re-initialization in this counter. - PrimaryReplica (D_k): The function will locally create a data replica, set a counter Hop Cpt to zero and will broadcast messages of Primary Creat (Node Add, D_k, Hop Cpt) to the node neighbours of the local node identity. The node border will be a node that does not have the following link such as the N_8 (on Fig. 1). The node that preceding N_i is the node "sender" for a message Primary Creat to N_i. The node that follows node N_i will be a node "non-sender" of the message Primary Creat to the node N_i.

2/Algorithm of a primary data replication of D_k:

In case N_J creates another new data D_k

the Hop Cpt = 0; Node Add= N_J;

Send the Primary Creat (Node Add, D_k, Hop Cpt) to the neighbours;

endif;

In the case of the reception of Primary Creat (Node Add, D_k, Hop Nb) then

If the 1st reception, then the HopCpt = HopCpt +1;

If the (HopCpt =3) then

If (a replica of the D_k will not exist on neighbours) then the Primary Replica (D_k);

else

If (the replica of D_k will exist on any one neighbouring N_i) then the Hop Cpt = 1;

Node Add= N_i; Send the Primary Creat (the Node Add, D_k, Hop Cpt) to the neighbours

endif

endif

else

If (HopCpt =2) and the (this is the node having only a neighbour) then the Primary Replica (D_k);

else

Send the Primary Creat(the NodeAdd, D_k, Hop Cpt) to all the nodes

endif

endif

if not ignore this message received;

endif;

endif ;

3.2 Random Replication Algorithm

For the purpose of solving this problem using a method of simple replication, there were many duplications of replicas made along with a dynamic replication algorithm to eliminate duplication of replicas among the mobile nodes. Firstly, the method had determined the allocation of a replica in a similar way in which the simple replication was done. Access frequency of a data item, d, at a particular node, the number of times that d is accessed is termed as Access Frequency data items. This helps in deciding the replication prioritization. The Dynamic Access Frequency and Neighbourhood (DAFN) algorithm considers the frequencies of neighbours in order to avoid redundancies. Once there was a duplication of a replica of the data item that was between two of the neighbouring mobile nodes, the one with a lower access frequency to data item will change the replica to that of the other replica. As the status changes in accordance with the movement of mobile nodes, there is a dynamic method of replication that is executed for every relocation period. The algorithm for the method has been described as below [23, 24]:

1. During the period of relocation, every mobile will broadcast the node identifier and the information on the access of frequencies to their data items. Once all these nodes complete broadcasts received from node identifiers, each node will know the connected mobile nodes.
2. Every mobile node will preliminarily determine the allocation of replicas that are based on a method of simple replication.
3. For every set of nodes that are connected to one another, the following process will be repeated using the order of breadth-first search which was from a mobile node having a suffix (l) for node identifier (M_i) that was the lowest. At the time of a data item duplication either original or a replica that takes place between two of the neighbouring mobile nodes, and in case one among them is original, the node holding a replica will change to the other one. In case both are replicas, the node that has an access frequency to the item will be the highest among all the possible items.

During a period of relocation, the mobile node may not be able to connect to that of the other node that has either an original or replica of the data item that the node has to allocate. In such a case, memory space for the replica will be filled temporarily with the ones that are allocated from the earlier period of relocation but not currently chosen for the purpose of allocation. The replica allocated temporarily will be selected from the possible replicas that have access frequency to a replica (the data item) which is the highest. In case there is no replica that can be allocated, memory space continues to remain free. At the time of access of data items with the replica that is allocated is successful, the memory space will be filled using proper replica.

An example to execute a dynamic algorithm of replication within the environment shown by Table 1 is depicted in Fig. 2. This has a dark grey and rectangular replica allocated for the purpose of duplication of the replica. While looking at it in detail, there are some changes to the replica that take place between a combination of the neighbouring mobile nodes:

Table 1 Access frequencies to data item

Data	Mobile node					
	M_1	M_2	M_3	M_4	M_5	M_6
D_1	0.65	0.25	0.17	0.22	0.31	0.24
D_2	0.44	0.62	0.41	0.40	0.42	0.46
D_3	0.35	0.44	0.50	0.25	0.45	37
D_4	0.31	0.15	0.10	0.60	0.09	0.10
D_5	0.51	0.41	0.43	0.38	0.71	0.20
D_6	0.08	0.07	0.05	0.15	0.20	0.62
D_7	0.38	0.32	0.37	0.33	0.40	0.32
D_8	0.22	0.33	0.21	0.23	0.24	0.17
D_9	0.18	0.16	0.19	0.17	0.24	0.21
D_{10}	0.09	0.08	0.06	0.11	0.12	0.09

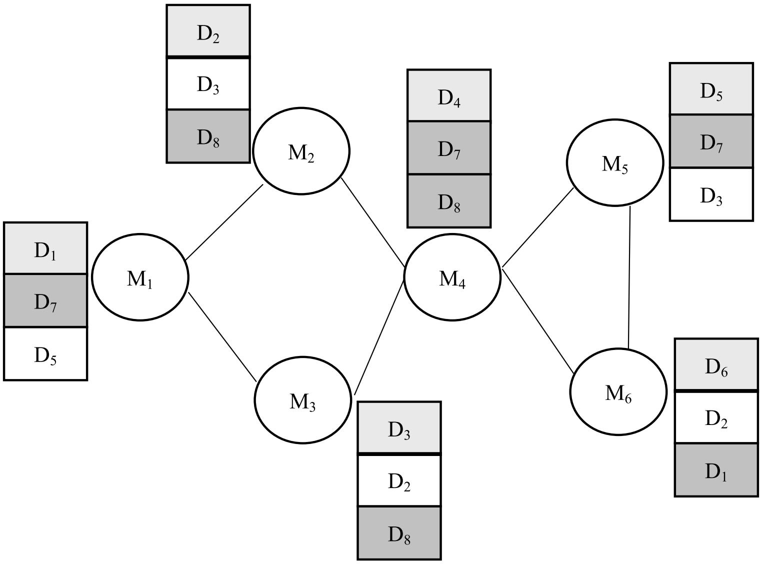

Fig. 2 An example of executing the DAFN method

$$
\begin{aligned}
&M_1 - M_2 : D_2 \rightarrow D_7(M_1), D_5 \rightarrow D_8(M_2) \\
&M_1 - M_3 : D_5 \rightarrow D_8(M_3) \\
&M_2 - M_4 : D_2 \rightarrow D_7(M_4) \\
&M_3 - M_4 : No\ duplication \\
&M_4 - M_5 : D_5 \rightarrow D_8(M_4) \\
&M_4 - M_6 : No\ duplication \\
&M_5 - M_6 : D_2 \rightarrow D_7(M_5), D_3 \rightarrow D_1(M_6)
\end{aligned}
$$

For this, the $M_i - M_j$ will denote a replica duplication that has been eliminated between two of the mobile nodes which are M_i and M_j, and the $D_k \rightarrow D_l(M_i)$ is the allocation of replica that has been changed from D_k to D_l at the node M_i. There are six types of replicas that have been allocated within the entire network in a method of simple replication where there are eight replicas that are allocated in the method of dynamic replication. This is owing to the dynamic replication algorithm eliminating duplication of the replica. The result of this is that the accessibility of data can be higher compared to the method of simple replication.

But, this method of dynamic replication algorithm will not be able to completely eliminate duplication of replica among the neighbouring nodes since the process of elimination by means of scanning the network which is based on a concept known as breadth-first search. In an example as per Fig. 2, the duplication of the replica of D_8 between both M_2 and M_4 and further between M_3 and M_4, and also D_7 between M_4 and M_5 is seen. In case the topology of the network changes at the time of executing the method, the reallocation of the replica may not be done at the mobile nodes over the links that are disconnected. The overheads and the traffic are found to be higher compared to the method of simple replication algorithm since for every period of reallocation, the mobile nodes will exchange information and also reallocate the replicas.

4 Results and Discussion

In this section, the simple replication algorithm and random replication algorithm are used. Experiments are carried out using 10–50 number of MEC nodes, 100–500 number of data to be replicated and 10–50 capacity. The bandwidth savings (MEC nodes, number of data to be replicated and capacity), as shown in Tables 2, 3 and 4 and Figs. 3, 4 and 5:

Table 2 Bandwidth Savings %

Number of MEC nodes	Simple replication algorithm	Random replication algorithm
10	30.61	32.65
20	35.71	38.76
30	41.83	44.88
40	48.96	52.03
50	57.13	61.2

Table 3 Bandwidth savings %

Number of data to be replicated	Simple replication algorithm	Random replication algorithm
100	36	38
200	41	45
300	48	52
400	57	60
500	66	71

Table 4 Bandwidth savings %

Capacity %	Simple replication algorithm	Random replication algorithm
10	42	46
30	52	57
50	57	61
70	71	77
90	81	87

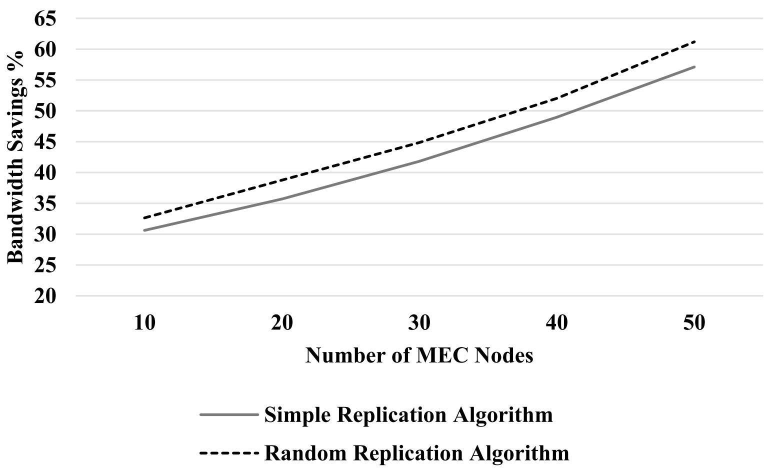

Fig. 3 Bandwidth savings %

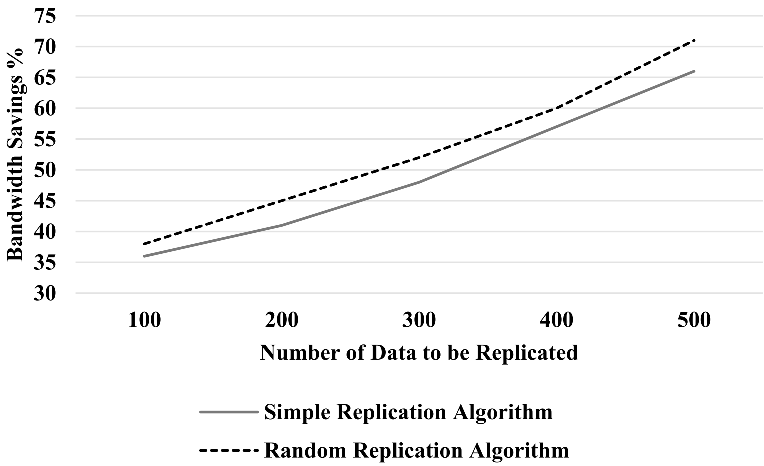

Fig. 4 Bandwidth savings %

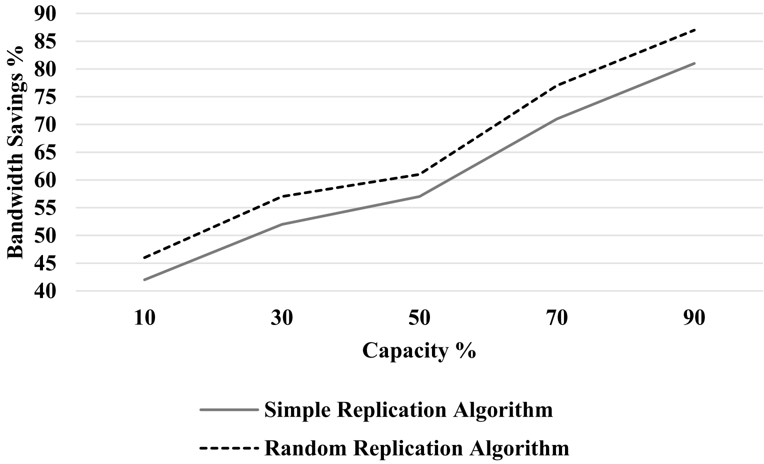

Fig. 5 Bandwidth savings %

From the Fig. 3, it can be observed that the random replication algorithm has higher bandwidth savings by 6.44% for 10 number of MEC nodes, by 8.19% for 20 number of MEC nodes, by 7.03% for 30 number of MEC nodes, by 6.07% for 40 number of MEC nodes and by 6.87% for 50 number of MEC nodes when compared with simple replication algorithm respectively (Tables 5, 6 and 7).

From the Fig. 4, it can be observed that the random replication algorithm has higher bandwidth savings by 5.4% for 100 number of data to be replicated, by 9.3% for 200 number of data to be replicated, by 8% for 300 number of data to be replicated, by 5.13% for 400 number of data to be replicated and by 7.29% for 500 number of data to be replicated when compared with simple replication algorithm respectively.

From the Fig. 5, it can be observed that the random replication algorithm has higher bandwidth savings by 6.44% for 10 capacity, by 8.19% for 30 capacity, by 7.03% for 50 capacity, by 6.07% for 70 capacity and by 6.87% for 90 capacity when compared with simple replication algorithm respectively.

From the Fig. 6, it can be observed that the random replication algorithm has higher data utilization by 6.6% for 10 number of MEC nodes, by 6.7% for 20 number of MEC nodes, by 4.24% for 30 number of MEC nodes, by 4.11% for 40 number of MEC nodes and by 7.64% for 50 number of MEC nodes when compared with simple replication algorithm respectively.

Table 5 Data utilization %

Number of mobile edge computing nodes	Simple replication algorithm	Random replication algorithm
10	33.59	35.88
20	40.04	42.8
30	47.07	49.11
40	54.83	57.13
50	62.95	67.95

Table 6 Cost savings %

Number of mobile edge computing nodes	Simple replication algorithm	Random replication algorithm
10	26.08	25.51
20	30.18	31.03
30	33.57	37
40	39.07	43.75
50	47.51	48.07

Table 7 Cost savings %

Number of data to be replicated	Simple replication algorithm	Random replication algorithm
100	30.46	32.08
200	33.26	35.82
300	39.59	41.79
400	45.12	50.57
500	55.67	61.59

From the Fig. 6, it can be observed that the random replication algorithm has higher cost savings by 2.21% for 10 number of MEC nodes, by 2.78% for 20 number of MEC nodes, by 9.72% for 30 number of MEC nodes, by 11.3% for 40 number of MEC nodes and by 1.17% for 50 number of MEC nodes when compared with simple replication algorithm respectively.

From the Fig. 6, it can be observed that the random replication algorithm has higher cost savings by 5.2% for 10 number of MEC nodes, by 7.4% for 20 number of MEC nodes, by 5.41% for 30 number of MEC nodes, by 11.4% for 40 number of MEC nodes

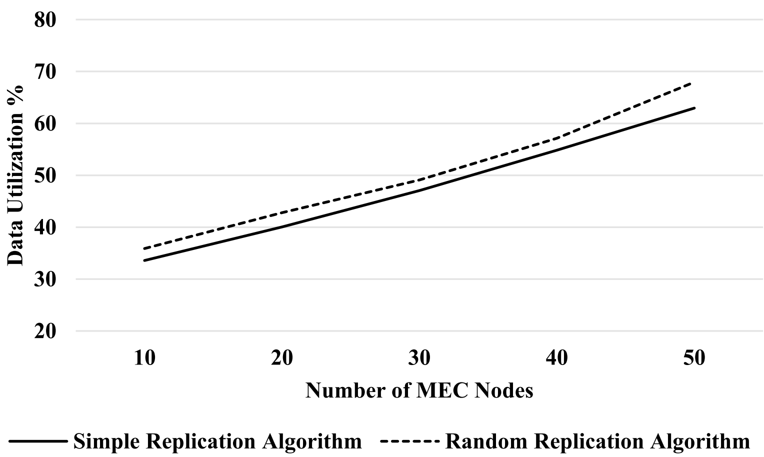

Fig. 6 Data utilization %

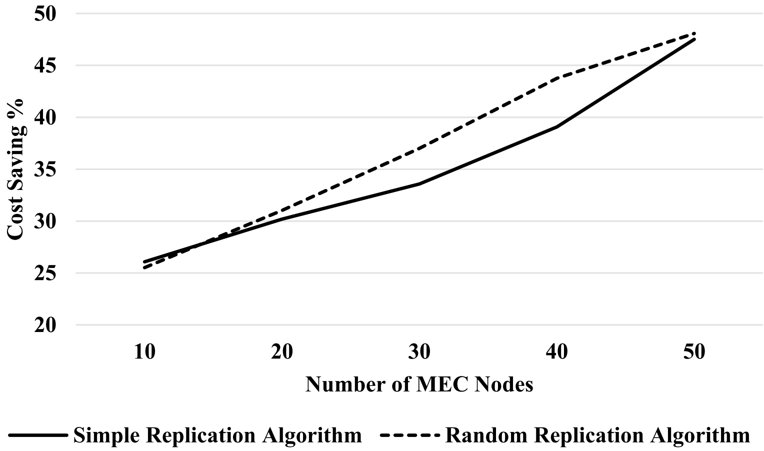

Fig. 7 Cost savings %

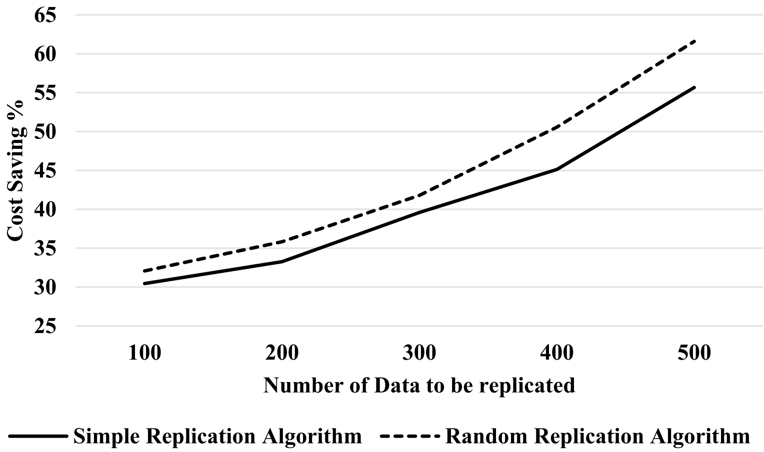

Fig. 8 Cost savings %

and by 10.1% for 50 number of MEC nodes when compared with simple replication algorithm respectively (Figs. 7 and 8).

5 Conclusion

The development of the IoT has made the EC an emerging solution to the challenges of managing millions of devices and sensors along with the corresponding resources needed. A major challenge that is associated with the 5G technology has been the RAN. Here, the MEC provides a real-time RAN information and using this helps in improving the QoE for the end users since the real-time RAN offers services that are context-aware. The EC platform also permits the edge nodes to respond to their service demands thus reducing the consumption of bandwidth and the latency of the network. The replication of data helps in improving response time for access requests and also helps in the distribution of load processing of requests. As access is normally carried out on the replica of data that is the nearest, global traffic will be decreased. For the purpose of this work, both simple and the random algorithms of replication have been proposed. The technique of replication has made the replication of data every effective based on the data item frequency of access, the network topology, and wireless link stability. It also improves the time taken for response and also maintains consistency. The results have proved the replication algorithm to have a higher savings in terms of bandwidth by about 6.44% for the 10 number of MEC nodes, by about 8.19% for the 20 number of MEC nodes, by about 7.03% for the 30 number of the MEC nodes, by about 6.07% for the 40 number of the MEC nodes and by about 6.87% for the 50 number of MEC nodes on being compared to the simple replication algorithm.

References

1. Salman, O., Elhajj, I., Kayssi, A., & Chehab, A. (2015). Edge computing enabling the Internet of Things. In *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)* (pp. 603–608). IEEE.
2. Gupta, N., Bafna, K., & Gupta, N. (2011). Efficient data replication algorithm for mobile Ad-hoc networks. In *International Conference on Information and Network Technology* (pp. 164–170).
3. Hara, T., & Madria, S. K. (2009). Consistency management strategies for data replication in mobile ad hoc networks. *IEEE Transactions on Mobile Computing, 8*(7), 950–967.
4. Sasaki, Y., Hara, T., Ishikawa, Y. (2019). Efficient framework for processing top-k queries with replication in mobile ad hoc networks. *GeoInformatica*, *23*(4), 591–620.
5. Sethi, P., & Sarangi, S. R. (2017). Internet of things: Architectures, protocols, and applications. *Journal of Electrical and Computer Engineering*, *2017*(1), 1–25.
6. Hassan, N., Gillani, S., Ahmed, E., Yaqoob, I., & Imran, M. (2018). The role of edge computing in Internet of Things. *IEEE Communications Magazine, 99,* 1–6.
7. Sabella, D., Vaillant, A., Kuure, P., Rauschenbach, U., & Giust, F. (2016). Mobile-edge computing architecture: The role of MEC in the Internet of Things. *IEEE Consumer Electronics Magazine, 5*(4), 84–91.
8. Rahmani, A. M., Gia, T. N., Negash, B., Anzanpour, A., Azimi, I., Jiang, M., et al. (2018). Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach. *Future Generation Computer Systems, 78,* 641–658.
9. Ai, Y., Peng, M., & Zhang, K. (2018). Edge computing technologies for Internet of Things: A primer. *Digital Communications and Networks, 4*(2), 77–86.
10. Mach, P., & Becvar, Z. (2017). Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials, 19*(3), 1628–1656.
11. Yu, W., Liang, F., He, X., Hatcher, W. G., Lu, C., Lin, J., et al. (2018). A survey on the edge computing for the Internet of Things. *IEEE Access, 6,* 6900–6919.
12. Zhang, K., Leng, S., He, Y., Maharjan, S., & Zhang, Y. (2018). Mobile edge computing and networking for green and low-latency Internet of Things. *IEEE Communications Magazine, 56*(5), 39–45.
13. Satyanarayanan, M., Simoens, P., Xiao, Y., Pillai, P., Chen, Z., Ha, K., et al. (2015). Edge analytics in the Internet of Things. *IEEE Pervasive Computing, 14*(2), 24–31.
14. Li, X., Li, D., Wan, J., Liu, C., & Imran, M. (2018). Adaptive transmission optimization in SDN-based industrial Internet of Things with edge computing. *IEEE Internet of Things Journal, 5*(3), 1351–1360.
15. Ning, Z., Dong, P., Kong, X., & Xia, F. (2018). A cooperative partial computation offloading scheme for mobile edge computing enabled Internet of Things. *IEEE Internet of Things Journal, 6,* 4804–4814.
16. Wang, R., Yan, J., Wu, D., Wang, H., & Yang, Q. (2018). Knowledge-centric edge computing based on virtualized D2D communication systems. *IEEE Communications Magazine, 56*(5), 32–38.
17. Verma, S., Yadav, A. K., Motwani, D., Raw, R. S., & Singh, H. K. (2016). An efficient data replication and load balancing technique for fog computing environment. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 2888–2895). IEEE.
18. Farris, I., Taleb, T., Flinck, H., & Iera, A. (2018). Providing ultra-short latency to user-centric 5G applications at the mobile network edge. *Transactions on Emerging Telecommunications Technologies, 29*(4), e3169.

19. Orsini, G., Bade, D., & Lamersdorf, W. (2015). Computing at the mobile edge: Designing elastic android applications for computation offloading. In *2015 8th IFIP Wireless and Mobile Networking Conference (WMNC)* (pp. 112–119). IEEE.
20. Ni, J., & Lin, X. (2019). Towards Edge-assisted Internet of Things: From security and efficiency perspectives. arXiv:1902.07094.
21. Chandrakala, C. B., Prema, K. V., & Hareesha, K. S. (2017). Effective data sharing in MANETS using cluster based replication. *Far East Journal of Electronics and Communications, 17*(5), 971–993.
22. Moussaoui, S., Guerroumi, M., & Badache, N. (2006). Data replication in mobile ad hoc networks. In *International Conference on Mobile Ad-Hoc and Sensor Networks* (pp. 685–697). Berlin: Springer.
23. Hara, T. (2001). Effective replica allocation in ad hoc networks for improving data accessibility. In *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society* (Cat. No. 01CH37213) (vol. 3, pp. 1568–1576). IEEE.
24. Samal, S., Mishra, M. R., Pati, B., Panigrahi, C. R., & Sarkar, J. L. (2018). A novel approach to avoid selfish nodes during allocation of data items in MANETs. In *Progress in Advanced Computing and Intelligent Engineering* (pp. 571–580). Singapore: Springer.

N. Saranya received the B.E. (CSE) from Anna University, Tamil Nadu in 2006 and M.E. (CSE) from Anna University, Tamil Nadu, in 2015. She worked as Assistant Professor in SASTRA University, Tamil Nadu. She is currently pursuing the Ph.D degree in the Department of Computer Science, Anna University Chennai. Her research interests include Wireless Sensor Networks, Internet of Things, Big Data Analytics, Datamining, Machine Learning Techniques.

Dr. K. Geetha holds a M.E. degree in Computer Science and Engineering from K. S. Rangasamy College of Technology, affiliated to Anna University of Technology Coimbatore, Tamil Nadu, India in 2010. Now She completed the Ph.D degree in Anna University, Chennai. She is currently working as a Professor in the Department of CSE, Excel Engineering College. She has 10 years of teaching experience. She has published 25 papers in various national and international journals. She is an active member of ISTE and CSI. Her Research interests include Mobile computing, Ad hoc Networks and Network Security.

Dr. C. Rajan received his B.E. degree in Computer Science and Engineering from SSN College of Engineering at University of Madras. Then he obtained his Master's degree in Computer Science. He have Completed the Ph.D degree in Anna University, Chennai. He is currently working as an Associate Professor in the Department of Information Technology, K. S. Rangasamy College of Technology. He has 16 years of teaching experience. He has published papers in 45 various national and international journals. His research interests Computer Networks, Soft Computing and Network Security.