

Yet another Improvement in RSFT Algorithm in Mobile Grid

L. Anita Elizabeth, V.M. Sivagami

Dept of Computer Science, Dept of Information Technology

Sri Venkateswara College of Engineering

Sriperumbudur 602105, India

ABSTRACT

Grid computing is the virtualization and pooling of IT resources such as compute power, storage and network capacity, and so on into a single set of shared services that can be provisioned or distributed, and then redistributed as needed. Researchers integrate the mobile devices in the grid which results in Mobile Grid. The mobile agent paradigm has attracted many attentions recently but it is still not widely used. One of the barriers is the difficulty in protecting an agent from failure because an agent is able to migrate over the network autonomously. An important technique that speeds up data access in Mobile Grid systems replicates the data in multiple locations so that a user can access it from a site in his vicinity. It has been shown that data replication not only reduces access costs, but also increases data availability in many applications. Replication of data from primary repositories to other locations can be an important optimization step to reduce the frequency of remote data access. Fault tolerance is an important property in Grid computing as the dependability of individual Grid resources may not be able to be guaranteed. The Replica Supporting Fault-Tolerance (RSFT) algorithm is formulated to support QoS-aware replica placement, balancing the load of replicas and to reduce the communication cost in mobile grid environment using bottom-up dynamic programming approach. Then, a resource selection algorithm is integrated with bottom-up dynamic programming approach to support fault tolerance in mobile grid environment by considering the dynamic characteristics of mobile devices.

Keywords

Mobile grid; Mobile devices; Replication; QoS support; Fault-tolerance; Load balancing; RSFT Algorithm.

1. INTRODUCTION

Grid has the ability to aggregate large amount of computing resources which are geographically dispersed to tackle large problems and workloads as if all the servers and resources are located at a single site in a heterogeneous environment. The overall motivation for Grid is to facilitate the routine interactions of these resources in order to support large-scale science and Engineering. Alike the web provides uniform naming and access to documents via URLs, the grid provides uniform high-performance access to computational resources. The challenge for the grid infrastructure is to enable transparent access while allowing resource providers to keep or establish their own local policies. Recently grid computing is changing into wireless and mobile grid environment that considers user mobility and integrates mobile devices into wired grid. Previous mobile devices (e.g. PDA, laptop, cell phone etc.) were restricted to small amount of battery and poor performance, but capacity of the latest models is growing rapidly as much as that

of desktop computers. At the same time, users who use mobile devices are increasing. These reasons lead many researches to utilize mobile devices in grid. Compared to traditional (wired) Grids, mobile Grids have the advantage of exploiting the concepts of mobility and ubiquity in terms of being available any-time, any-where and by all means. Making use of wireless connection, the mobile Grid environment is available any-time and any-where.

The use of mobile devices in grid environments may have two interaction aspects: Devices are considered as the users of grid resources or as grid resource providers. Due to the limitation constraints on energy and processing capacity of mobile devices, their integration into the Grid as resource providers and not just consumers is very difficult. Because of the limited bandwidth and frequent disconnections in mobile grid environment significant work has already been done to improve the performance and reliability of mobile grid systems. Replication management in it provides local replicas for remote applications in order to quickly access and process remote data, avoiding a great deal of data transferring, improving the efficiency of data access and the capability of fault tolerance. In order to improve and maintain a high overall throughput of Grid jobs that access files, one of the major problems in Mobile Grid environment is optimal distribution and replication of data files.

In Mobile grid environment data replication technique is used to improve availability and access cost. In recent years, more and more works focused on the replica management in parallel and distributed systems. But most of them concerned on replica location, replica replacement, and consistency strategies of replica, or building infrastructures for replica management. In fact, replica placement is one of important challenges to improve performance and good placement strategies can result in significant performance gains. In replica-aware services, the servers in the system are aware of the locations of replicas.

Communication over the internet is not reliable. Host connected via the internet constantly fail and recover. The communications links go down any time. Due to high communication loads, link failure and the software bugs, transient communications and node failures are common in the internet. This leads to performance degradation of the job. Therefore reliability is an important issue in mobile grid. Without proper load balancing, all jobs may be allocated only to the stable resources such as physically fixed Grid components. It results in discriminating mobile devices with less performance but with the enormous number of population. It incurs not only the decrease of the Grid resources' utilization, but also the performance degradation due to the improper load-balancing. For this reason, fault tolerance is of vital importance in this new Mobile Grid paradigm; Mobile Grid computing is not expected to be fault free, despite the fact that individual technique such as fault

avoidance and fault removal may additionally be applied to its resources. Therefore, fault tolerance mechanisms need to be deployed to allow the Grid system to perform correctly in the presence of faults, enhancing it with the appropriate reliability.

A fault tolerant model for task scheduling in Mobile Grid systems [3] handles the additional tasks with a resource management scheme based on the knapsack problem formulation. A P2P discovery protocol in middleware architecture, MoGrid [4] is the central element which distributes tasks among the most resourceful devices. OMGR consists of mobility model, connected dominating set (CDS) election and mobile grid routing algorithm [5]. A price based distributed energy constrained resources allocation optimization algorithm [6] is proposed which aims to reduce energy consumption and also to improve the application utility in a mobile grid environment with a limited energy change, ensuring battery lifetime and also the deadlines of the grid application. IEEE 802.11 based mobile grid architecture [8] is proposed which discusses a generic node mobility prediction framework for mobile grid environment. This framework can be used to formulate cost effective job allocation schemes based on a predetermined pricing strategy at the Wireless Access Point to be distributed to the mobile nodes under it. Data replication issues are classified into three categories: Replica relocation, consistency management, and location management [9]. This paper investigates the QoS-aware replica placement problems for responsiveness is investigated and two classes of service models are proposed namely replica-aware services and replica-blind services [10]. Subject to server capacity and Quality-of-Service (QoS) constraints replication is placed in tree networks [11]. The client requests are known beforehand, while the number and location of the servers are to be determined. Replica placement issues are addressed to achieve improved performance for systems supporting mobile transactions. Requests and/or transactions issued by mobile clients may access multiple data objects and should be considered together in terms of replica allocation [12]. A fault model for Grid computing is constructed by refining the traditional distributed systems fault model; this is being used as a basis for fine-tuning a replication-based approach to fault tolerance [14]. A fault tolerant based mechanism is developed based on replication and voting frequencies [15] for secure mobile agent platform system (SMAPS).

In this paper a Replica supporting fault-tolerance (RSFT) algorithm is proposed to load balances the replicas, minimizes the communication cost and better Quality of Service support of mobile user. Also a resource selection algorithm is proposed to support fault-tolerance in case of any replica node failure.

2. PROBLEM STATEMENT

Figure 1 illustrates a mobile grid system as a tree network. It is based on a wireless network in which each cell consists of a number of mobile devices. Mobile devices residing in a cell of wireless networks are coordinated by a central entity that resides at the Access Point/Base Station, (BS) in order to perform a task. Mobile devices can be used as both resource consumers and resource providers to provide access to internet data resource.

The mobile users near a base station can directly access the Internet data resource, and hence can serve as gateways to the Internet data resource for other mobile users out of the coverage. Therefore, our mobile grid system allows mobile users to access Internet data resource either directly or by multi-hop wireless connections.

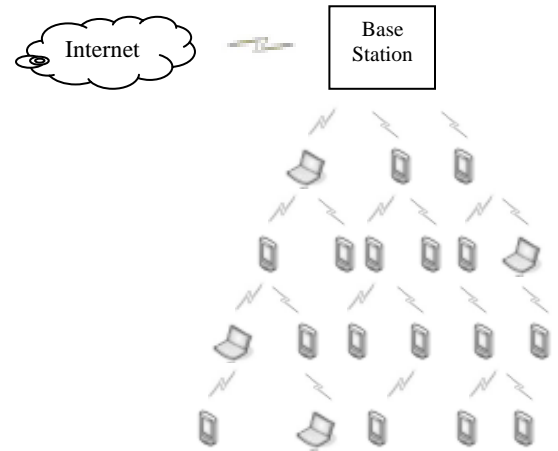


Figure 1. A Typical Mobile Grid System

In general clients' requests are to be satisfied by base station. The clients are known (their position in the tree, their number of requests and their QoS requirement), while the number and location of the servers are to be determined. A client is a leaf node of the tree, and its requests can be served by one or several internal nodes. Initially, there is no replica; the root server has the original copy of the database but cannot serve all clients directly, so a distribution tree is deployed to provide a hierarchical and distributed access to replicas of the original data. When a node is equipped with a replica, it can process a number of requests, up to its capacity limit. Nodes equipped with a replica, also called servers, can only serve clients located in their sub tree (so that the root, if equipped with a replica, can serve any client), this restriction is usually adopted to enforce the hierarchical nature of the target application platforms, where a node has knowledge of its parent and children in the tree. The rule is to assign replicas to nodes according to total number of request and its defined QoS.

An additional constraint is introduced, in order to guarantee Quality of Service (QoS). Responsiveness, which refers to how fast the users can access the requested information, is an important type of QoS requirements needed by a wide range of applications. Requests must be served in limited hops. A client would experience shorter access latency if a replica of the requested object is placed in its closer proximity. The QoS requirements are specified in the form of general distance metric. The service is guaranteed to the users by providing QoS in mobile grid networks. This is achieved by determining the QoS requirement for each node by base station. In a mobile grid, submitting jobs and receiving the results back is not straightforward, since power constraints and frequent disconnections are prevalent in wireless and mobile communications. Resources in a mobile grid for executing job need to be carefully chosen to support fault tolerance.

3. REPLICA SUPPORTING FAULT TOLERANCE

Consider Tree $T = (V, E)$ rooted at the base station, as shown in Figure 2 where node set V is the set of base station and other mobile devices and $E = V \times V$ is the set of wireless communication links. For each edge $(u, v) \in E$, a communication cost $d(u, v)$ is associated with it.

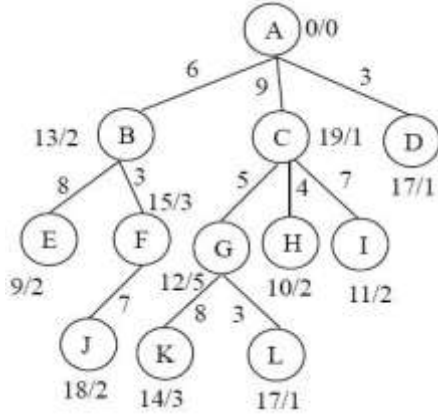


Figure 2. Model of Mobile Grid System

Each node $v \in V$ in tree has two weights $read(v)$ and $q(v)$ namely number of request and QoS requirement of that node respectively. And the request is served by the node named $r(v)$, which is the first node in the path from node v to root and there is a replica stored on it. The QoS requirement of node v is specified by $q(v)$: if the length of path from v to its $r(v)$ is no more than $q(v)$, we say the QoS requirement is satisfied; otherwise the QoS requirement is violated.

BS defines allowable QoS requirement $qos(v)$ and calculates the total number of request $w(v)$ for each node so as to find suitable replica nodes to balance the load, satisfy the QoS requirement of user and to reduce communication cost.

3.1 Computing Total Number of Requests - $w(v)$

If the mobile node is a leaf node of the tree, its $read(v)$ will be its $w(v)$. Else the sum of $read(v)$ of that node and its children will be its $w(v)$. Therefore, the dynamic programming equation of workload is

$$w(v) = \begin{cases} read(v) & \text{if } v \text{ is leaf} \\ \sum_{v_j \in ch(v), v_j \in S} w(v_j) + read(v) & \text{if } v \text{ is not leaf} \end{cases} \quad (1)$$

Table 1 shows total number of request $w(v)$ for each node using Equation (1).

Table 1. Total number of request $w(v)$ of each node

Mobile Nodes	read(v)	w(v)
BS	0	138
A	13	55
B	19	83
C	17	17
D	9	9

E	15	33
F	12	43
G	10	10
H	11	11
I	18	18
J	14	14
K	17	17

3.2 Communication Cost Without Replication - $cc(v)$

When the requests are served by BS, the total communication cost in mobile grid system can be given by

$$cc(v) = \sum_{v \in V} (read(v) \times \sum_{e_i \in path(v, r(v))} d(e_i)) \quad (2)$$

Table 2 shows communication cost $cc(v)$ for each node using Equation (2).

Table 2. Communication Cost of each node without replication

Mobile Nodes	w(v)	d(u,v)	cc(v)
BS	138	0	0
A	55	6	330
B	83	9	747
C	17	3	51
D	9	14	126
E	33	9	297
F	43	14	602
G	10	13	130
H	11	16	176
I	18	16	288
J	14	22	308
K	17	17	289

3.3 Defining QoS – $qos(v)$

To satisfy the QoS requirement of user, BS defines the allowable qos for each node in the tree. If the mobile node is a leaf node of the tree, its $q(v)$ will be its $qos(v)$. Else the minimum of the internal node and its children will be taken as its $qos(v)$, and whose dynamic programming equation is

$$qos(v) = \begin{cases} q(v) & \text{if } v \text{ is leaf} \\ \text{Min}(q(v), qos(v_j)) & | v_j \in ch(v) \text{ and } v_j \notin S \\ & \text{if } v \text{ is not leaf} \end{cases} \quad (3)$$

where $ch(v)$ - children node set of node v .

Table 3 shows defined $qos(v)$ for each node using Equation (3).

Table 3. Defined QoS $qos(v)$ of each node

Mobile Nodes	$q(v)$	$qos(v)$
BS	0	0
A	2	1
B	1	1
C	1	1
D	2	2
E	3	2
F	5	1
G	2	2
H	2	2
I	2	2
J	3	3
K	1	1

3.4 Finding Replica Node Set – S

The replica node set S , therefore, $S \subseteq V$ is found by the node which has $qos(v)$ as 0 and $w(v)$ which is greater than the threshold decided by base station at a particular instance according to total number of requests in the mobile grid. For an example, if maximum threshold for each node is 40, the mobile node which has total number of request greater than 40 will be chosen for replica node. Table 4 shows replica node set S , which satisfies the above two criteria.

Table 4. Replica Node set

Replica Node S	$qos(v)$	$w(v)$
BS	0	138
A	1	55
B	1	83
F	2	43

3.5 Communication Cost with Replication – $ccs(v)$

The replica nodes are found as shown in Table 4. A non – replica mobile node’s request need not be transferred till base station, and

its requests can be served by a replica node which is an internal node of the tree.

And now, the communication cost after replication to access data resource for each node is as shown in Table 5.

Table 5. Communication Cost of each node with replication

Mobile Nodes	$w(v)$	$d(u,v)$	$ccs(v)$
BS	138	0	0
A	55	0	0
B	83	0	0
C	17	3	51
D	9	8	72
E	33	3	99
F	43	0	0
G	10	4	40
H	11	7	77
I	18	10	180
J	14	8	112
K	17	3	51

The communication cost is greatly reduced, the load in the tree is balanced and Qos requirement is also satisfied. Table 6 below shows the difference in communication cost before and after replication.

Table 6. Communication cost without replication $cc(v)$ and with replication $ccs(v)$

Mobile Nodes	$cc(v)$	$ccs(v)$
BS	0	0
A	330	0
B	747	0
C	51	51
D	126	72
E	297	99
F	602	0
G	130	40
H	176	77
I	288	180
J	308	112
K	289	51

Since the mobile grid is a dynamic system, fault may occur to replica nodes due to unpredictable changes such as frequent movement of mobile node, drain of battery, network failure etc. This leads to performance degradation of the job. Therefore, the resource selection algorithm is adopted which is integrated with replica placement algorithm to support fault tolerance. When a replica node fails in a tree network, the children node of that failure node transmits request to the previous level parent node only if it satisfies the dynamic characteristics of mobile node else requests are served by base station.

When a replica node fails, children nodes check whether the previous level is a base station and requests are served by BS. Else, the children node checks whether the previous level is a replica node and also checks remaining battery power, movement pattern and performance of that replica node before assigning the request so as to support fault tolerance. For an example, if replica node F fails children nodes J and K checks whether the previous level parent is a replica node and checks the following conditions:

3.5.1 Remaining Battery Information

B_{i_c} - the current amount of battery before a job is assigned. B_i - the amount of battery after a job is processed. If the battery power is below threshold, a job isn't assigned. B_i is calculated using Equation (4). B_{i_p} - amount of battery to process a job.

$$B_i = B_{i_c} - B_{i_p} \quad (4)$$

If B_i is less than 10 percent of total amount of battery, job is not assigned.

3.5.2 Mobility Information

Job should not be assigned to a replica node if it has frequent movement, which is the probability of a replica mobile device v that keeps staying in a mobile grid network is greater than 40 percent by predicting path.

3.5.3 Performance Information

Performance information is those of CPU, memory, and storage. Performance information reflects dynamic status of a mobile device. If the load of the replica mobile node is too high, that is more than 70 percent of total capacity, the job is not assigned.

3.6 Communication Cost after Fault Tolerance

If the above three conditions are satisfied, requests from J and K are served by replica node B. Else base station serves the request. Table 7 shows the communication cost of each node if requests are served by base station and by Node B after the failure of replica node F. Changes in the communication cost occurs only to node J and K.

Table 7. Communication cost without replication $cc(v)$ and with replication $ccs(v)$

Mobile Nodes	$ccs(v)$	$cc(v)$
BS	0	0

A	0	0
B	0	0
C	51	51
D	72	72
E	99	99
G	40	40
H	77	77
I	180	180
J	308(if served by BS)	182(if served by Node B)
K	289(if served by BS)	136(if served by Node B)

If dynamic characteristics of node B are not considered in case of node F failure the children node submits job to Node B. If node B does not able to serve the request again, failure occurs in the tree. This leads to performance degradation of the job. Children nodes have to retransfer requests again to base station. In this case communication cost becomes higher. The sum of communication cost first assigned to node B and communication cost for reassignment to base station will be its total communication cost for nodes J and K.

The communication cost of this difference is clearly tabulated as shown below:

Table 8. Communication cost without and with Fault Tolerance support

Mobile Nodes	$ccs(v)$ without FT support	$ccs(v)$ with FT support
BS	0	0
A	0	0
B	0	0
C	51	51
D	72	72
E	99	99
G	40	40
H	77	77
I	180	180
J	490	308
K	425	289

The Replica Supporting Fault-Tolerance (RSFT) algorithm is as follows:

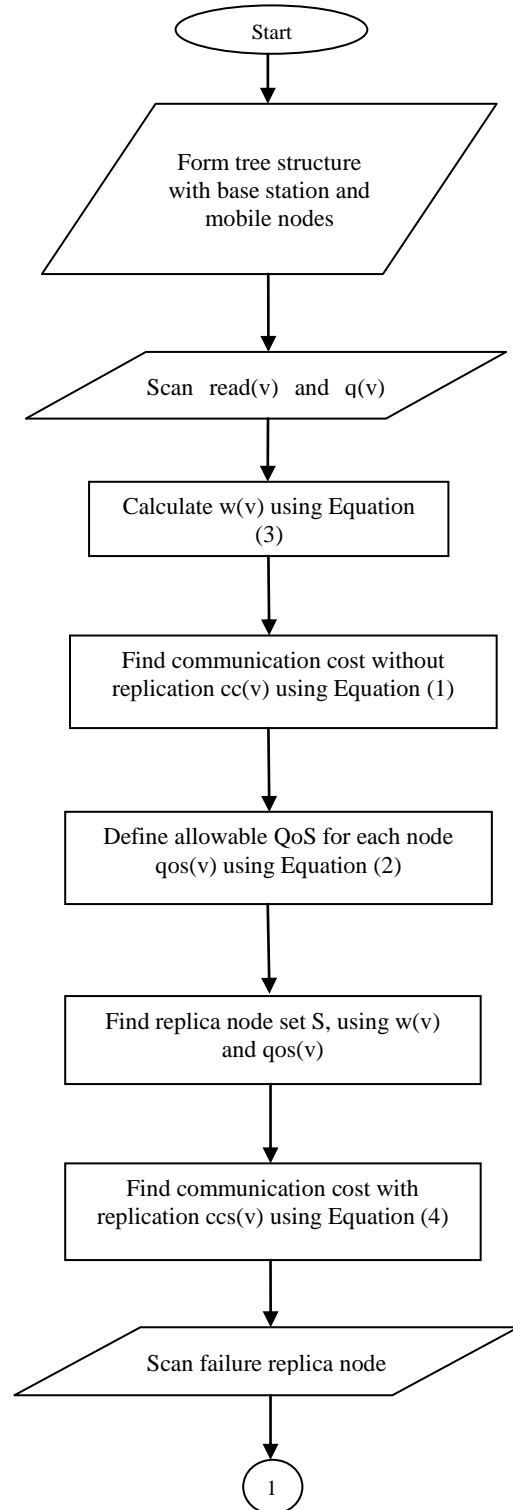
$l = \text{height}(T)$;

```

S = ∅;
while(l ≥ 0)
{
for each vi ∈ level(l) do
{
if(read(vi) > MW)
return ∅;
else
Computing qos (vi) and w(vi) by (3) and (1);
if (l ≠ 0 && qos (vi) = 0)
{
S = S U {vi};
qos (vi) = ∞;
w (vi) = 0;
}
//end if for QoS requirement
while (w (vi) > MW)
{
choose vj ∈ ch(v) and vj ∉ S with maximal w(vj);
S = S U {vj};
Recomputing qos (vi) and w (vi) by (3) and (1);
}
//end while for workload of vi
}
//end for each for nodes at level l
l--;
}
//end while for all levels
S = S U root (T);
for each v ∈ V (T) do
Computing cc(v, S) by equation (2);
return S;
if (s ∉ V)
{
collecting_information()
{
for all mobile devices in mobile grid
{
calculate the battery power of the mobile device (Bi);
if (Bi ≤ 10%) then Bi = 0;
if (Mobile user moves out current mobile grid) then Mi = 0;
calculate the performance of the mobile device (Pi);
if (Utilization rate ≥ 70%) then Pi = 0;
}
}
assignment_job()
{
for all mobile devices
{
allocate the job to mobile devices;
}
}
}
}

```

Algorithm 1. Replica Supporting Fault Tolerance



4. SYSTEM DESIGN – DATA FLOW DIAGRAM

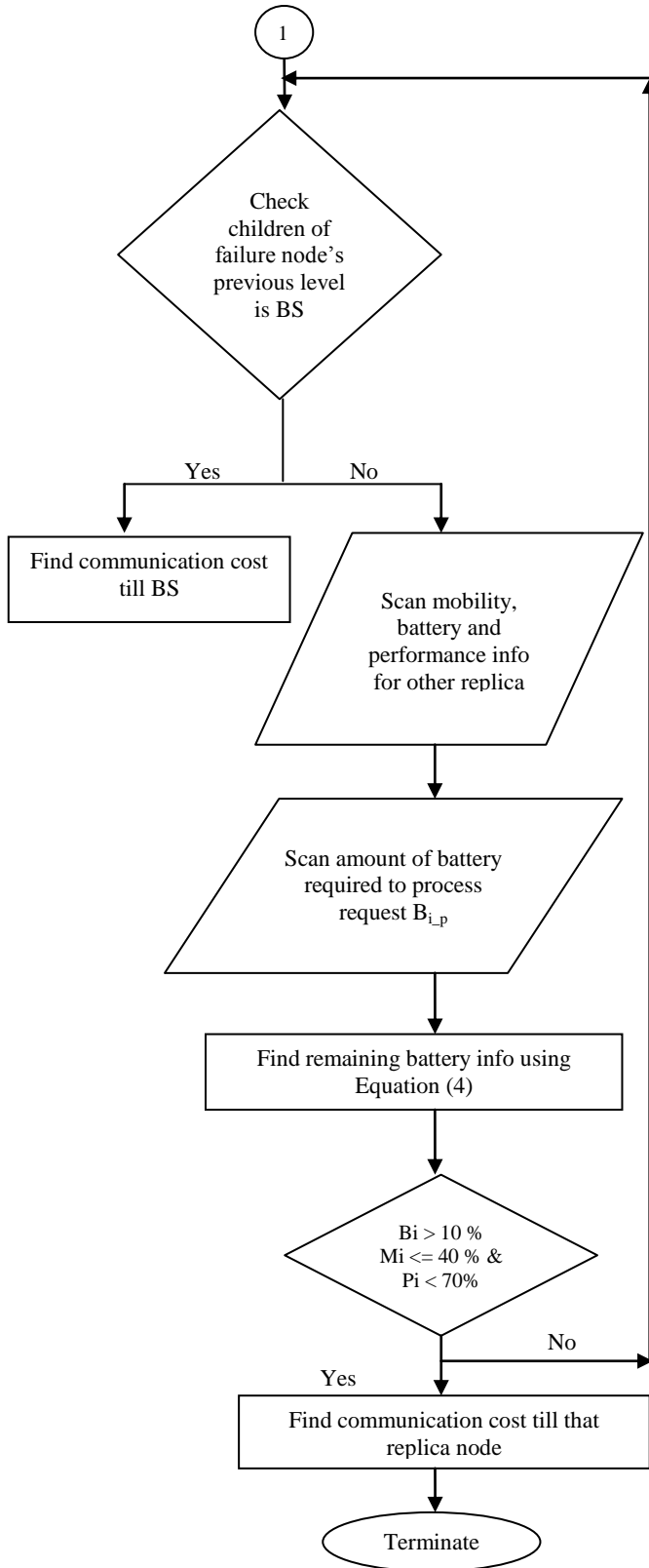


Figure 3. Overall system design

5. EXPERIMENTAL SETUP

To evaluate the performance of our algorithm, we conduct some simulation experiments based on a randomly mobile grid system. There are 9 nodes in our mobile grid system and the children number of each node is randomly from 0 to 3, read number of each node is a random distribution from 10 to 40, the communication cost of each edge is randomly from 5 to 15, and the QoS requirements of each node is a random value from 1 to 6.

The first experiment is the communication cost comparison between without replication and with replication as shown in Table 6. We can easily see that the communication cost of each node is clearly decreased. The second experiment is the communication cost comparison between without fault tolerance support and with fault tolerance support as shown in Table 8. And proposed algorithm (RSFT) has better performance than the existing algorithm.

6. ANALYSIS SECTION

In this section, we present the performance evaluation of our Replica supporting fault tolerance algorithm (RSFT). Figure 4 shows the experimental results of communication cost without replication and with replication clearly decrease as the number of replicas increases.

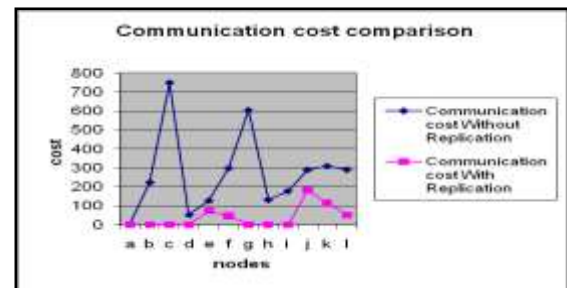


Figure 4. Communication cost comparison between without and with replication

Figure 5 shows the experimental results of communication cost without and with fault tolerance support and our algorithm (RSFT) has better performance than the existing algorithm (replica placement) without supporting fault-tolerance.

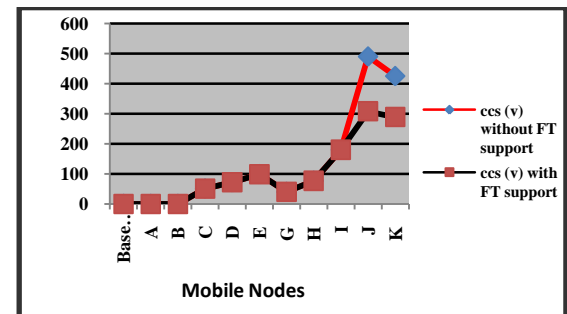


Figure 5. Communication cost comparison between without and with fault tolerance support

7. CONCLUSION

In this paper we introduce a Replica Supporting Fault Tolerance (RSFT) algorithm is proposed which reduces the total communication cost replicating data resource in more than one node, satisfies QoS requirement of mobile user, and balances the requested load in the mobile grid using bottom-up replica approach. The resource selection algorithm which is integrated with replica placement approach greatly reduces communication cost and supports fault tolerance in case of replica node failure by considering mobility, performance and battery information.

In future, assigning job to a mobile node even it has more frequent move from one mobile grid network to other will be considered. Also it includes long term data placement techniques that are likely to be integrated for making replica node optimal throughout the service duration.

8. REFERENCES

- [1] Xiong Fu, Ruchuan Wang, Yang Wang, Song Deng. "A Replica Placement Algorithm in Mobile Grid Environments". 2009 IEEE Computer Society.
- [2] SookKyong Choi, InSeock Cho, KwangSik Chung, Byungkwen Song, HeonChang Yu. "Group-based Resource Selection Algorithm Supporting Fault-tolerance in Mobile Grid". 2007 IEEE Computer Society.
- [3] Antonios Litke, Dimitrios Skoutas, Konstantinos Tserpes, Theodora Varvarigou. "Efficient task replication and management for adaptive fault tolerance in Mobile Grid environments". 2006 Elsevier.
- [4] Luciana dos S. Lima, Antonio T. A. Gomes, Artur Ziviani, Markus Endler, Luiz F. G. Soares, Bruno Schulze. "Peer-to-Peer Resource Discovery in Mobile Grids" 2005 ACM.
- [5] Di Wu, Ning Tong, Keqiu Li. "Mobile Grid Routing Algorithm in Mobile Ad Hoc Networks with Obstacles". 2006 IEEE Computer Society.
- [6] Li Chunlin, Li Layuan. "Energy constrained resource allocation optimization for mobile grids". 2009 Elsevier.
- [7] Wen-ying Zeng, Yue-long Zhao, Jun-wei Zeng, Wei Song. "Mobile Grid Architecture Design and Application". 2008 IEEE Computer Society.
- [8] Preetam Ghosh, Nirmalya Roy and Sajal K Das "Mobility-based Cost-effective Job Scheduling in an IEEE 802.11 Mobile Grid Architecture".
- [9] Takahiro Hara, Yamadaoka, Suita. "Data Replication Issues in Mobile Ad Hoc Networks". In Proceedings of the 16th International Workshop on Database and Expert Systems Applications (DEXA'05)
- [10] Xueyan Tang, Jianliang Xu. "QoS-Aware Replica Placement for Content Distribution". IEEE Transactions on parallel and distributed systems, vol. 16, no. 10, October 2005
- [11] Anne Benoit, Veronika, Rehn-Sonigo, Yves Robert. "Replica Placement and Access Policies in Tree Networks". IEEE Transactions on parallel and distributed systems, vol. 19, no. 12, December 2008
- [12] Manghui Tu, Peng Li, Liangliang Xiao, I-Ling Yen, Farokh B. Bastani. "Replica Placement Algorithms for Mobile Transaction Systems". IEEE Transactions on knowledge and data Engineering, vol. 18, no. 7, July 2006.
- [13] Yun Huang, Nalini Venkatasubramanian. "Mobile Data Overlay (MDO): A Data Placement Paradigm for Mobile Applications". International Conference on Mobile Data Management 2008 IEEE Computer Society.
- [14] Paul Townend, Jie Xu. "Fault Tolerance within a Grid Environment".
- [15] Rajwinder Singh, A.K.Sarje, Navdeep Kaur, Ramandeep Kaur. "Fault Tolerance Mechanism in Secure Mobile Agent Platform System". 2009 IEEE International Advance Computing Conference (IACC 2009)