ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization)
Vol. 3, Issue 4, April 2015

EABPF: Efficient Algorithm to Overcome Battery Power Failures in Mobile Ad Hoc Networks

Thota Subhashini, Reeshma K
M.Tech Student (CNE), Dept of ISE, The Oxford College of Engineering, Bangalore, India
Assistant Professor, Dept of ISE, The Oxford College of Engineering, Bangalore, India

ABSTRACT: Wireless mobile devices such as high-end cell phones, PDAs, portable gaming devices, tablet PCs etc. all having wireless networking capabilities have a tremendous increase in recent years. These devices when used in Mobile Ad Hoc Networks (MANETs) may extend their capabilities, e.g., to reach the Internet when no Wi-Fi base stations are within range, or to have communication with each other over multiple hops when no other networking infrastructure is provided. Major problem of continuous participation in a MANET is energy consumption which is the main characteristic of MANETs. Wireless devices mentioned are battery powered and since energy is a rare resource and the devices are mobile, optimization of energy consumption in MANETs has greater impact as it directly corresponds to lifetime of networks. In cooperative MANETs, data replication is done on different mobile devices in order to improve system's availability. A master node is assigned to act as a coordinator for the data copies that are shared among the mobile nodes. When this master node failure occurs, another node has to be elected to replace the failed one. Since mobile devices have a limited battery power, the master node may fail at any time. Moreover, current master node election protocols in MANETs employ a notable wireless communication overhead which consumes a considerable amount of battery power. An Efficient algorithm to overcome battery power failures in MANETs(EABPF) is proposed to replace the exhausted master node before its battery dies with a healthy node with much less communication overhead thus overcoming the battery power failures in MANETs .

KEYWORDS: Mobile Ad Hoc Networks, Replication in MANETs, Master node election.

I. INTRODUCTION

A MANET that stands for Mobile Ad Hoc Network is a set of nodes in mobile which can communicate via message passing over wireless links [1]. Nodes having same transmission range can communicate directly with each other. If it is not the case, they communicate via other intermediate nodes [4, 12]. In Many mobile applications, mobile users share information known as collaborative work such as rescue operations at a disaster site, collaborative multimedia applications, multi-user games and robotic pets entertainment, and collaborative white board in a university campus. For such applications, improving data availability, accessibility, and consistency plays crucial role.

The most practical technique to achieve a high degree of data accessibility and availability in MANET is Replication, which distributes copies of data (caches) on various mobile devices in the wireless network. Guaranteeing consistency among replicated data implies avoiding conflicts during updating of data. For preserving data consistency, a master or control node is assigned that acts as a coordinator for the data update requests which are shared. If the master node fails, another node should be elected to act as a new master. Various master election algorithms [3, 5, 6, 19, 20] are employed to elect a new master and replace the failed master node. The new elected master must acquire shared data objects' information from other nodes. However, wireless communication overhead got incurred due to messages exchange which helps to preserve data consistency. Limited characteristic of battery power of mobile devices in MANETs may cause master node failure at any time. Hence, saving battery power in mobile devices is very much necessary. Current master node election approaches in MANETs incur a notable overhead in wireless communication. In [16], the conclusion was that wireless communications consume about 70 % of the battery power. Hence, master node election is not an efficient solution for power consumption in mobile environments. In this paper, we propose an efficient algorithm to overcome battery power failures in MANETs (EABPF), to replace an exhausted master node

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

having power shortage by measuring its remaining battery power, before it fails. EABPF replaces the master node that got exhausted with a node that is healthy when its remaining battery power reaches a threshold value that is predefined. Very less communication overhead is incurred with this EABPF algorithm. Hence, this approach has two major contributions: 1) Early detection of power failure thus reducing the chance of master node outage, 2) Increasing the lifetime of mobile nodes by saving the power when compared to that consumed by traditional master node election algorithms.

The rest of this paper is organized as follows: section II presents a literature survey and discusses related work, section III describes the system architecture that supports the idea of master node replacement algorithm, section IV introduces the proposed approach, section V evaluates the performance of this approach through simulation experimental results, finally section VI gives our conclusions and suggestions for future work. This paper extends our preliminary work introduced in [14].

II. LITERATURE SURVEY

In general failures of master node can be of two types which may occur due to predictable or unpredictable events in MANETs. Unpredictable failures are the failures that cannot be known or predicted beforehand. Examples of unpredictable failures include hardware/software failure, network signal loss and disconnection due to mobility. Predictable failures are those which can be estimated or monitored .Examples of predictable failures include excessive energy consumption, unbalanced CPU utilization, or shortage of available memory. When master node failure occurs, a new master should be elected for maintaining data consistency and accessibility. Various algorithms study this master election problem in MANETs [1, 9, 10, 21, 22].We can divide these algorithms into two main categories. We describe them below:

Under normal operation, a simple heartbeat monitoring technique for detection of node failure is adopted by the first category of algorithms .Mobile nodes exchange simple “alive” messages continuously to indicate that nodes are functioning properly. If the master node’s heartbeat gets ceased for a predefined set of time period, then an election algorithm is employed. These algorithms are presented in [8], and are classified as Non-Compulsory and Compulsory protocols. These protocols are not realistic as they require information exchange by nodes in order to elect a master node to act as a coordinator for the shared data copies. A Routing algorithm called Temporally Ordered Routing Algorithm, TORA [15] for routing the data in wireless mobile ad hoc networks is the basis for the algorithms presented in [10, 22, 1] in which node adjustment is acquired by employing a locally maintained variable, called the height. This local variable points to the master node, in a decrementing manner over a Directed Acyclic Graph (DAG). Another work [21] is presented for other master node election algorithm. Classical termination-detection algorithm for diffusing computations by Dijkstra and Scholten [2] is the basis for this algorithm. In this algorithm, each and every node should always maintain information dynamically about its neighborhood that is mobile .The master node which is a coordinator sends periodic heartbeat messages to other nodes. The absence of such messages at a particular node for certain amount of predefined time indicates a departure from the master node and a diffusing computation is triggered at that particular node to elect a new master. The algorithm in [11] presented a consensus–based master node election algorithm. This algorithm elects a local extreme as a master node. This algorithm can also be tuned to the global extreme of the network. In the global extreme of the network, all the nodes are visited instead of the majority.

Algorithms based on a game theory using Volunteers ‘Dilemma [9, 17] are employed in second category. In this, a replica sends a “ping” message to the master node, whenever it wants to update data copies. If there is no response from the master node within a time window (tw), the master node election algorithm is invoked. Otherwise, a message consisting of set of metrics indicating its health such as remaining battery life and the frequency of complete disconnection from the network are sent to the master node by the replica. Upon receiving all of the metrics, the master node would compute an ordered list of best candidates to take over as the new master node from all replicated nodes, and it should go off line. The new master node would then broadcast out its nearest neighborhood list to the replicating node. In the event that the master node should go offline the replicating nodes would ping the neighborhood list in order. The top one that responds would be the new master node.

We note that in the first category, there is relatively high communication overhead due to the large number of messages exchanged. In [18], the numbers of messages estimation is between 128 and 176 messages for a set of 64 nodes in wireless network. This wireless communication overhead consumes considerable amount of the battery power of the mobile devices in MANETs .On the other hand ,communication overhead is reduced in second category , but no precautionary action is taken until the master node fails. In Recent paper [13] Master Shift Replication (MSR), has

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

been proposed for managing the planned master node outages without using traditional master node election protocol. MSR protocol shifts the master role from the current master to another replica that is scheduled, when a planned outage is about to occur. Experiments showed that MSR protocol reduces significantly number of communication messages to N+3 messages in which N is the number of nodes or replicas. The work proposed in this paper is inspired from MSR protocol and it is applicable in mobile environments.

III. SYSTEM ARCHITECTURE

EABPF system architecture includes three major components as shown in Fig. 1: local object cache, clients and replicas. A local object cache is a set of data objects that are uniquely identified in memory and are replicated at each replica in the system. A client is the one that acts as an interface between applications and replicas .Through this , an application can add a new data object, update an existing one, or receive notifications on the objects. A transaction, Ttid (Otid, Mtid), can be defined as a procedure employing a set of methods called Mtid = {m1,…, mk}, in order to modify the states of a subset of objects Otid = {o1,..,oz}, in which tid refers to transaction ID. A procedure known as transaction procedure (TransProc) is sent by the application through the client interface to the replica and it is executed there. A client communicates through API with the replica such as add_object (Oi, Class), where Oi is a unique object identifier and Class is the class definition code, and trans (TransProc).

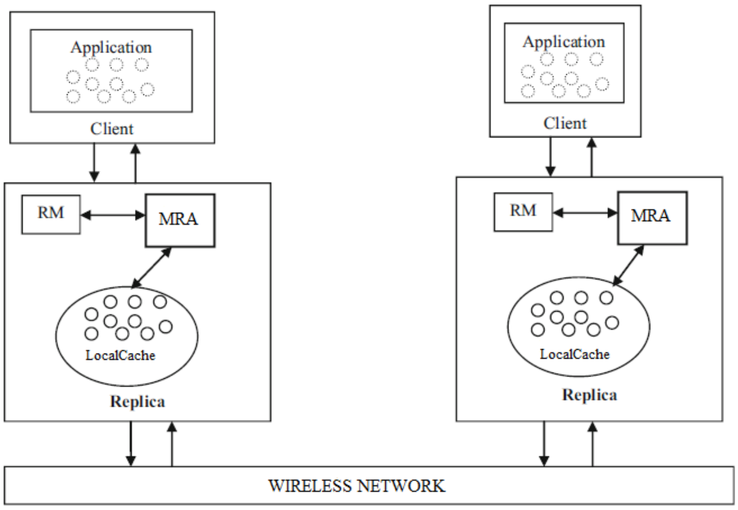

Fig 1: System Architecture

When a transaction call is received by the replica, it extracts the objects belonging to TansProc and try to reserve these objects through the master node replica. The master node controls object access permission via a lock_table which is a 3-field table containing the locked object ID, the ID of the replica requested to update/lock the object, and the active transaction ID. The total energy of nodes is spent in following modes: (1) Transmission Mode (2) Reception Mode (3) Idle Mode and (4) Overhearing Mode.

These modes of power consumption are described as:-

- Transmission Mode

  A node is said in transmission mode when it sends data packet to other nodes in network. These nodes require energy to transmit data packet, such energy is called Transmission Energy (Tx), of that nodes. Transmission energy is depended on size of data packet (in Bits), means when the size of a data packet is increased the required transmission energy is also increased. The transmission energy can be formulated as:

  $Tx = (330*Plength)/2*10^{6}$ And

  $P_T = Tx / T_t$

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

Where Tx is transmission Energy, P_T is Transmission Power, T_t is time taken to transmit data packet and Plength is length of data packet in Bits.

- Reception Mode

When a node receives a data packet from other nodes then it said to be in Reception Mode and the energy taken to receive packet is called Reception Energy (Rx). Then Reception Energy can be given as:

$$R_x = (230 * \text{Plength})/2*10^6$$

And

$$P_R = R_x / T_r$$

Where R_x is a Reception Energy, P_R is a Reception Power, T_r is a time taken to receive data packet, and Plength is length of data packet in Bits.

- Idle Mode

In this mode, generally the node is neither transmitting nor receiving any data packets. But this mode consumes power because the nodes have to listen to the wireless medium continuously in order to detect a packet that it should receive, so that the node can then switch into receive mode from idle mode.

Despite the fact that while in idle mode the node does not actually handle data communication operations, it was found that the wireless interface consumes a considerable amount of energy nevertheless. This amount approaches the amount that is consumed in the receive operation. Idle energy is a wasted energy that should be eliminated or reduced. Then power consumed in Idle Mode is:

$$P_I = P_R$$

Where P_I is power consumed in Idle Mode and P_R is power consumed in Reception Mode.

- Overhearing Mode

When a node receives data packets that are not destined for it, then it said to be in over-hearing mode, and it may consume the energy used in receiving mode. Unnecessarily receiving such packets will cause energy consumption. Then power consumed in overhearing mode is:

$$P_{over} = P_R$$

Where P_{over} is power consumed in Overhearing Mode and P_R is power consumed in Reception Mode.

In our algorithm, a transaction may be active (under processing), committed (finish processing), or aborted (cancel processing) at the replica. A notification is used to refer a response to a request from replica to client such as trans_abort and trans_commit or from replica to another replica such as ok_lock/ko_lock. In this algorithm, replica is the crucial component in the system and it is uniquely identified by a parameter replica reference Rm. Replica reference Rm has two main roles: 1) It executes client's requests and notifies clients on the results of these requests e.g., it manages and coordinates transactions issued from the connected client(s); 2) the master node replica owns the whole local object cache, it gives privileges or permission to lock/unlock object(s). We assume the following:

- Each client is connected to single replica.
- Asynchronous message communication is used by replicas in a Mobile Ad hoc Network(MANET)
- A replica does not crash and even the network does not crash as the replicas work in a fault free environment.
- Replicas have a resource manager (RM) to notifies them when a predefined battery level is reached

IV. PROPOSED APPROACH

Though there are many drawbacks, master node election appears to be the only way to recover from unpredictable failures for master node. Predictable (measurable) master node failures can be treated in a much efficient way to avoid drawbacks in election. Battery power consumption rate is a measurable performance metric for reflecting the health of the node. Hence, it can be monitored at the master node in order to achieve an early replacement of the master node before its battery power goes off. Thus, leadership is transferred to a new master node in the presence of the old (exhausted) master node. The proposed EABPF algorithm introduced in this section is an extension of the traditional master backup approach having the advantage of avoiding election in case of master node power failure by transferring the leadership to another node when the master node's battery power reaches a predefined threshold. The proposed algorithm is described in the following steps and is illustrated in Fig. 2 shown below.

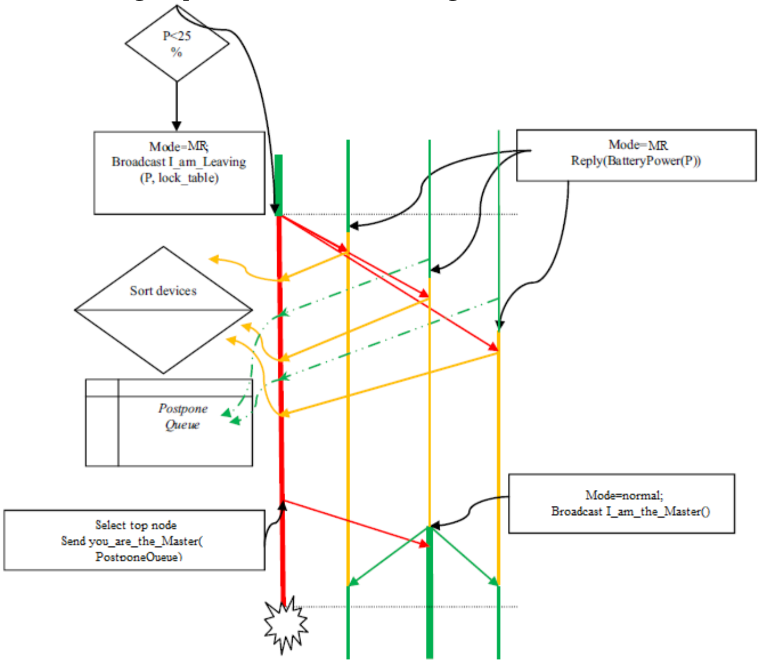

Fig 2: Master Replacement Algorithm

1. The master node receives notification from Resource Manager (RM) when its battery power level reached a predefined threshold of 25 %.
2. There are two different modes in this algorithm. Normal mode and Master Replacement mode. The master node changes its mode from Normal mode to Master Replacement (MR) mode. Then it broadcasts the message "I am Leaving (P,LockTable)" message to all replicas. Where P is the battery power level and LockTable is the lock table of the master node.
3. During lock table transfer time (LTTrans), all new requests (i.e., UpdateDataObject(X)) received by the master node from other replicas are stored in a PostponeQueue and can be handled later by the new master node.
4. Once the message "I_am_leaving (P,LockTable)" is received , the battery power is measured by replica and then it changes its mode from Normal mode to MR mode, and responds to the master node by sending a message "batteryPower(P)". It is obvious that in MR mode replicas cannot issue new requests to the master node.
5. Master node collects batteryPower (P) values from all replicas and then sorts them in descending order.
6. Master node will then select the replica at the top of the list having the highest battery power to be the new master.
7. The ex-master node sends message (you_are_the_master(PostponeQueue)) to the new master
8. The new master node then executes the postponed requests appropriately until the PostponeQueue is empty. Now at this point, it changes its mode from MR to Normal.
9. The new master node will now announce itself to all other nodes by broadcasting a message "I am the master ()".
10. Finally replicas change their mode from MR mode to Normal mode. The number of communication messages needed for master node replacement can be determined as follows: one broadcast message "I_am_Leaving (P, LockTable)" with (n-1) reply messages "batteryPower (P)". Thus the first round takes n messages. For the second round, there is only one message "you_are_the_master (PostponeQueue)" with one broadcast message "I am the master ()". Hence, the total number of messages is n+2 where n gives the total number of nodes. For

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

a set of 64 nodes, EABPF needs 66 messages which are much less than the number of messages estimated in [18].

We can summarize the properties of EABPF algorithm as follows:

- At any time, there is a unique master node.
- The new master will start working only when the old master node stopped (handshaking by delivering the lock table and the postpone queue)
- This algorithm is fault tolerant. If the master node failure occurs before it completes the replacement process, other nodes will select a new master node by exchanging their battery powers. In this case, the loss is very less as only requests that are in the postpone queue will be lost.

V. RESULTS

The performance can be evaluated by building a prototype to represent both Master Backup Replication (MBR) approach and the Master Replacement Algorithm (MRA) approach. This prototype comprises of the system model described in section III and the application model. The application can be simply represented using a set of counters that are initiated to zero and are then incremented when a transaction is committed. The client can initially set the number of transactions submitted to the system and the network delay (i.e., the time taken by a transaction to be transferred from the client to a replica) at each run. The system architectural model consists of four replicas (MP1, MP2, MP3, MP4) and four clients (C1, C2, C3,C4) and are connected to the replicas MP1, MP2, MP3, MP4 respectively. An application creates four shared object counters (oc1, oc2, oc3, oc4) which are replicated at each replica. Each client issues 2,000 transactions having a rate of one transaction every 100msec that represents the delay of the network from a client to a replica. A transaction contains only one statement which increases the object counter by one meaning that C1 increments oc1, C2 increments oc2 and so on. The network transmission delay between replicas is set to 20 msec. Experiments were conducted to measure the number of messages exchanged between replicas in MBR and MRA. The total number of messages sent or received by a replica will not only measure the traffic network traffics, but also draws an indication on the amount of battery power consumption due to communication overhead. First, measurements are recorded for the MBR system as shown in Table 1 and Fig. 3.

Table 1 Distribution of Load in MBR

	MP1	MP2	MP3	MP4	Total
Number of lock requests	8000	0	0	0	8000
Number of update notifications	6000	6000	6000	6000	24000
Number of ok_lock replies	8000	0	0	0	8000
Total number of messages	24000	6000	6000	6000	40000

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

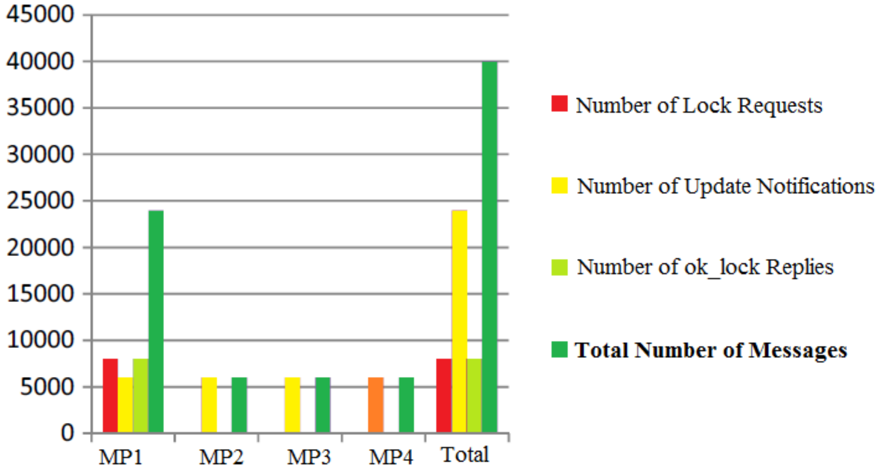

Fig 3: Distribution of Messages in MBR

A quick inspection for these measurements reveals that MP1 (the master) is loaded by 24,000/40,000=60 % of the total messages in the system, as a result,MP1's battery bleeds power rapidly and soon it will go down. The second experiment was conducted to evaluate and test the performance of MRA system under the same load condition (8,000 transactions). When the battery's power level of MP1 reaches the threshold of 25 %, MP1 is replaced by the MP2 which had the highest power level and MP1 is recharged. Similarly, when MP2 battery's power reaches 25 % MP2 is replaced by MP3 which had the highest power level and MP2 is recharged, and so on. Thereby, the system continues functioning without the need for election due to power failure. We also recorded measurements for the messages distribution in case of MRA system as shown in Table 2 and Fig. 4. These measurements show that almost the same number transactions have been executed by the system, however, the load distribution is improved and balanced among all replicas. Finally, we notice that only 110 transactions (1 %) were aborted during master node replacement.

Table 2 Distribution of Load in MRA

	MP1	MP2	MP3	MP4	Total
Number of lock requests	3160	1555	2056	1109	7880
Number of update notifications	5914	5914	5914	5961	23703
Number of ok_lock replies	3166	1555	2056	1109	7886
Total number of messages	12240	9024	10026	8179	39469

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

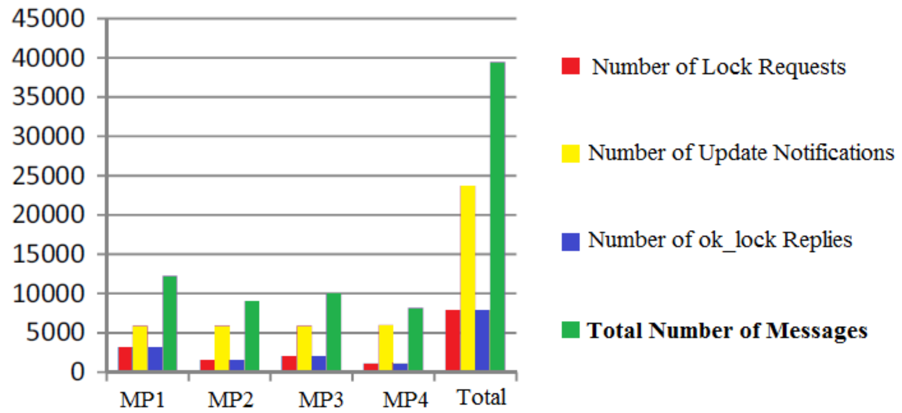

Fig 4: Distribution of Messages in MRA

VI. CONCLUSION AND FUTURE WORK

In this paper, a new algorithm is proposed for master node replacement in MANETs based on the battery power level of the nodes. The proposed algorithm replaces the exhausted master node with a healthy node by early discovery of it outage when its remaining battery power reaches a predefined threshold. This replacement can be accomplished with much less communication overhead. Therefore, our approach reduces the chance of master node outage by taking precaution of early detection of potential power failure and thus this saves the battery power consumed by traditional master node election algorithms due to communication overhead. In the future, this research can be continued by investigating other performance metrics in addition to battery power such as memory capacity and CPU power which is the main characteristics for Mobile ad hoc Networks survivability .We can select the next highest node in the sorted list, if the top node with highest battery power does not have sufficient memory capacity or CPU power.

REFERENCES

[1]. Melit L, Badache N (2011) An energy efficient leader election algorithm for mobile ad hoc networks. In:Programming and systems (ISPS), 2011 10th International Symposium on (pp 54–59). IEEE
[2]. Mostafa A, Youssef AE (2013) A leader replacement protocol based on early discovery of battery power failure in MANETs. In: IT convergence and security (ICITCS), 2013 International Conference on (pp 1–4). IEEE
[3]. Frederickson GN, Lynch NA (1987) Electing a primary in a synchronous ring. J ACM 34(1):98–115
[4]. Fernando N, Loke SW, Rahayu W (2013) Mobile cloud computing: a survey. Futur Gener Comput Syst 29: 84–106
[5]. EffatParvar M, Yazdani N, EffatParvar M, Dadlani A, Khonsari A (2010) Improved algorithms for leader election in distributed systems. In: Computer engineering and technology (ICCET), 2010 2nd International Conference on (vol. 2, pp V2-6). IEEE
[6]. Singh S, Kurose JF (1994) Electing "good" leaders. J Parallel Distrib Comput 21(2):184–201
[7]. Hatzis KP, Pentaris GP, Spirakis PG, Tampakas VT, Tan RB (1999) Fundamental control algorithms in mobile networks. In: Proceedings of the eleventh annual ACM symposium on Parallel algorithms and architectures (pp 251–260). ACM
[8]. Mostafa AM, Youssef AE (2013) A primary shift protocol for improving availability in replication systems. Int J Comput Appl 72(4):37–44
[9]. Vasudevan S, Kurose J, Towsley D (2004) Design and analysis of a leader election algorithm for mobile adhoc networks. In: Network Protocols, 2004. ICNP 2004. Proceedings of the 12th IEEE International Conference on (pp 350–360). IEEE
[10]. Vasudevan S, DeCleene B, Immerman N, Kurose J, Towsley D (2003) Leader election algorithms for wireless ad hoc networks. In: DARPA Information Survivability Conference and Exposition, 2003.Proceedings (vol 1, pp 261–272). IEEE .
[11]. Dijkstra EW, Scholten CS (1980) Termination detection for diffusing computations. Inf Process Lett 11(1):1–4
[12]. Derhab A, Badache N (2008) A self-stabilizing primary election algorithm in highly dynamic ad hoc mobile networks. IEEE Trans Parallel Distrib Syst 19(7):926–939
[13]. Guerraoui R, Rodrigues L (2006) Reliable distributed programming (vol. 138). Springer Verlag, Berlin
[14]. Masum SM, Ali AA, Bhuiyan MT-yI (2006) Asynchronous leader election in mobile ad hoc networks, Advanced Information Networking and Applications, AINA 2006. 20th International Conference on vol. 2, no., pp 5 pp 18–20
[15]. Park VD, Corson MS (1997) A highly adaptive distributed routing algorithm for mobile wireless networks.In: INFOCOM'97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution. Proceedings IEEE (vol. 3, pp 1405–1413). IEEE
[16]. Shirali M, Toroghi AH, Vojdani M (2008) Leader election algorithms: history and novel schemes. In:Convergence and Hybrid Information Technology, 2008. ICCIT'08. Third International Conference on (vol.1, pp 1001–1006). IEEE
[17]. Pering T, Agarwal Y, Gupta R, Want R (2006) Coolspots: reducing the power consumption of wireless mobile devices with multiple radio interfaces. In: Proceedings of the 4th international conference on Mobile systems, applications and services (pp 220–232). ACM

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

[18]. Shinohara M, Hara T, Nishio S (2007) Data replication considering power consumption in ad hoc networks.In: Mobile Data Management, 2007 International Conference on (pp 118–125). IEEE

[19]. Garcia-Molina H (1982) Elections in a distributed computing system. IEEE Trans Comput 31(1):48–59

[20]. Malpani N, Welch JL, Vaidya N (2000) Leader election algorithms for mobile ad hoc networks. In:Proceedings of the 4th international workshop on discrete algorithms and methods for mobile computing and communications (pp 96–103). ACM

[21]. Hirsch D, Madria SK (2011) A cooperative game theoretic approach for data replication in mobile ad-hoc networks. In: Collaborative computing: networking, applications and worksharing (CollaborateCom), 2011
7th International Conference on (pp 115–124). IEEE

[22]. Stoller SD (2000) Primary election in asynchronous distributed systems. IEEE Trans Comput 49(3):283–284

BIOGRAPHY

Mrs. Thota Subhashini a Student of Information Science and Engineering Department at The Oxford College of Engineering-Bangalore, affiliated to VTU pursuing M.Tech in Computer Networking and Engineering. She received her Bachelors of Engineering in Computer science and Information Technology Engineering from Nagarjuna Institute of Technology-Vijayawada affiliated to JNTU. She is currently working as a research assistant under the guidance of **Assistant Prof. K Reeshma.** Her research interests are Mobile Ad hoc Networks and Cloud Computing.

Mrs. K Reeshma has done her B.Tech in Computer Science and Engineering from College of Engineering Vadakara Affiliated to Cochin University of science and Technology and M.Tech in Computer Science and Engineering from AMC College of Engineering Affiliated to Visvesvaraya Technological University. She has worked at MES Engineering college for three year. She is currently working as the Assistant Professor in ISE Department of The Oxford College Of Engineering since 2 years. She has guided many M.Tech students in Computer Network Engineering. She has over all 5 years of teaching experience.