A Jini Based Implementation for Best Leader Node Selection in MANETs

Monideepa Roy, Pushpendu Kar and Nandini Mukherjee

Abstract MANETs provide a good alternative for handling the constraints of disconnectivity and disruption of static communication infrastructure faced during emergency situations. In such frameworks, it sometimes becomes necessary to elect a leader node for the MANET. Electing a leader node in a MANET poses several challenges due to the inherent properties of mobility, resource constraints etc. of these ad hoc networks. Although there are at present some existing leader node selection algorithms, it is difficult to apply them to MANET based frameworks where multiple selection of leader nodes is allowed, because of various reasons. This paper presents an algorithm for the best leader node selection, along with a network wide searching technique that a client uses to search for a particular service in the network, both of which are especially suitable for these types of frameworks.

Keywords MANETs · Leader node selection · Jini

M. Roy (✉)
School of Computer Engineering, KIIT University,
Bhubaneshwar, Odisha, India
e-mail: monideepa.roy@gmail.com

P. Kar
Department of Computer Science and Engineering, Jadavpur University,
Kolkata, West Bengal, India
e-mail: pushpendukar@rediffmail.com

N. Mukherjee
Department of Computer Science and Engineering, Jadavpur University,
Kolkata, West Bengal, India
e-mail: nmukherjee@cse.jdvu.ac.in

N. Chaki et al. (eds.), *Computer Networks & Communications (NetCom)*,
Lecture Notes in Electrical Engineering 131, DOI: 10.1007/978-1-4614-6154-8_2,

1 Introduction

Leader election in a network is the process of electing a node uniquely as the leader of that system. There is a number of existing leader election algorithms which have been proposed for classical distributed systems. But they are not suitable for MANETs which do not require any infrastructure or central control nodes, since they have independent network organization, and dynamic topology. And the participating mobile devices in a MANET also suffer frequently from constraints of resources and loss of communication due to their mobility. So leader election is challenging in MANETs. There are several existing leader election algorithms that have been proposed for ad hoc networks also. But the assumption in most of the cases is that the nodes are homogeneous and only one node is selected as the leader for the network. However, in our proposed framework, there may be multiple leader nodes present simultaneously in the framework. So a service provider or client basically has to choose to register with the most suitable leader node from among the multiple leader nodes within the range. In this paper we have presented an algorithm for a service provider or a client to register with the most suitable leader node based on factor called the threshold value of a leader node, the calculation of which is given in Sect. 4. This paper also presents a network wide search process which is performed by the clients while searching for a particular service. The rest of this paper is organized as follows: Sect. 2 explains the justification for proposing this algorithm, Sect. 3 presents the related work in this area. Section 4 presents the Leader node selection algorithm and Sect. 5 describes the network wide search activity. Finally, Sect. 6 contains the conclusion and future work.

2 The Need for Best Leader Node Selection

The standard definition of the leader election problem for static networks is that: I. Eventually there is a leader and II. There should never be more than one leader. Since in mobile ad hoc networks, the topology may change frequently, so, in such circumstances, it is necessary to reallocate the task of a leader node to another competent node. The existing leader election algorithms for MANETs assume that all the nodes are homogeneous. However, our work focuses on a mobile Grid framework which is composed of heterogeneous nodes (laptops, PDAs and other types of mobile devices) with different capabilities. Furthermore, complications arise because partitions can occur in an ad hoc network due to mobility of nodes. In such a case, some applications require that every component of the partition must have a unique leader. The mobile Grid framework which we had presented in our earlier work, is MANET—based and uses the Jini network technology, where the nodes are classified as being of three types: leader nodes, client nodes and service provider nodes. The nodes that are capable of being designated here as the leader nodes, are actually the nodes which host the Jini lookup service. A lookup service acts as repository of services, where a proxy object of each registered service of a service provider is stored with a set of attributes which define the service. The service provider nodes are mobile devices which have some services to offer and register with a leader node. The registration consists of acquiring a lease [1] from the lookup service and letting the lookup service know what services they have to offer (if any). The duration of the lease is to be granted to a service provider is decided by the lookup service based on various parameters. The clients also register with any one of the leader nodes to avail a service. If a requested service is not provided by the leader node that the client has registered with, then that leader node retrieves details of similar services from nearby leader nodes. This framework allows multiple leader nodes to be accommodated within a network and with overlapping ranges. Therefore if a service provider or client is within the range of multiple leader nodes, the proposed Jini [2] based algorithm is applied to select the most suitable leader node it should register with.

3 Related Work

While there are a number of leader election algorithms for traditional networks, they do not work in the highly dynamic environment found in mobile networks. This is primarily because these solutions to the leader election problem assume a static topology or assume that topological changes stop before an election starts or assume an unrealistic communication model such as a message-order preserving network. Leader election algorithms for mobile ad hoc networks have been proposed also [3–5]. But they have certain shortcomings for which they are not suitable for MANET based frameworks [6]. Firstly, these algorithms are designed to perform random node elections. Secondly, the existing algorithms assume that all the nodes in the network are homogeneous (i.e. having the same configurations) and the election is made from all the participating nodes in the network. Thirdly, most existing algorithms assume at most one leader node in the network. Because of the above factors, unlike existing work on leader election we present a leader node selection algorithm in a mobile wireless setting. In our framework [6], some nodes which have relatively higher resource and computational power are pre-designated as leader nodes when they join the network. Therefore all the participating nodes may not be eligible for the election. This algorithm calculates the minimum threshold value for each leader node. Since our framework supports multiple leader nodes, this algorithm chooses the most suitable node for each service provider. In fact we are dealing with the subsequent scenario of when a service provider or a client has to register with a leader and is faced with multiple options of leader nodes. In this case the selection of the best leader node for a service provider/client is not done from all the leader nodes in the network, but only from among the leader nodes which are within signal range of the service provider/client.

4 The Leader Node Election Algorithm

We consider any predefined operational area. Within this area a number of service providers, clients and leader nodes are deployed. The calculations are based on the assumption that the mobile devices do not leave the boundary of the total operational coverage area but may move around within it for the duration of time the application is running. So at any time *t* the position coordinates of the device can be determined. The pattern of movement of the nodes is assumed to be the Random Waypoint Mobility Model [7]. In this mobility model a node moves from its current location to a new location by randomly choosing a direction and speed in which to travel. The new speed and direction are both chosen from predefined ranges, [*Vmin;Vmax*] and $[0;2\pi]$ respectively. Each movement in the Random Waypoint Mobility Model occurs in either a constant time interval *t* or after a constant distance traveled *d*, after which there is a pause *t*, and at the end of which a new direction and speed are calculated and changed. To determine the position coordinates of any node at any point of time we use the following calculations: Suppose initially a node is at any point (x_0, y_0). It travels for time *t* seconds, at a random angle θ_1 with velocity v_1, after which it again changes direction. The new position (x_1, y_1) of the Mobile Node (MN) will now be:

$$\mathrm{x_1 = x_0 + v_1 \cdot t \cdot \cos\theta_1} \tag{1}$$

$$\mathrm{y_1 = y_0 + v_1 \cdot t \cdot sine\,\theta_1} \tag{2}$$

We continue this position calculation process to track further movements of the MN within the network. So at any time *t* the position of the device is determined by the formula:

$$\mathrm{x_n = x_{n-1} + v_n \cdot t \cdot \cos\theta_n} \tag{3}$$

$$\mathrm{y_n = y_{n-1} + v_n \cdot t \cdot sine\,\theta_n} \tag{4}$$

Considering a network, where there are multiple leader nodes, service providers and clients,

1. The threshold value t is calculated and stored by each leader node in the network, based on battery life (b) of the leader node, the number of nodes registered with the leader node (n), and the distance from a registered node (d), as:

$$\mathrm{t = w_1 \cdot b(w_2 \cdot n\, w_3 \cdot d)} \tag{5}$$

2. If a new leader node enters into the network, then if it finds any other leader node within range, it registers with it. The new leader node also allows any unregistered service provider or client node to register with it.

3. If a new service provider or client node joins the network, then it searches for a leader node to register with. If there are multiple nodes within range, then it selects the leader node with the minimum positive threshold value.
4. In case a leader node moves out of range, then the nodes that were registered with it follow Step 3 to register with a new leader node.

Each leader stores its threshold value, which is recalculated and updated every time a new node registers with it. The threshold value is calculated as:

$$t = f(b, n, d) \tag{6}$$

$$\text{i.e.}\quad t = w_1 \cdot b(w_2 \cdot n\, w_3 \cdot d) \tag{7}$$

where w_1, w_2 and w_3 are the respective weights to be assigned to the corresponding parameters. Here all the weights are assumed to be 1. The following calculations are used to determine the threshold value of a leader node:

1. For determining the battery life 'b', of each leader node, four values are considered:

Value	Range of remaining battery life
1	If the remaining battery life is >80 % of total capacity
2	If the remaining battery life is within 60–79 %
3	If the remaining battery life is within 30–59 %
−1	If the remaining battery power is <30 %

   To avoid registration with a node which has a remaining battery life of less than 30 %, the value of b and the function t have been chosen in such a way that it should lead to a negative threshold value if the battery life goes below 30 %.
2. The machine load 'n' is denoted to be the number of nodes registered with a leader node at any point of time. So it is considered that $n =$ number of service provider or client nodes registered with a leader node at any point of time.
3. The distance of each leader node from a service provider/client is denoted as 'd' and is measured using the standard Euclidean method to calculate the distance between any two points. The calculations of the positions of the service provider are done using the Random Way-point Mobility Model, as mentioned earlier in this section.

After the initial deployment, every subsequent entry or exit of the participating nodes that occurs will trigger the calculation of the threshold value of all the current leader nodes. Depending on this threshold value the service provider and client nodes will choose to register itself on a leader node among the multiple neighboring leader nodes. Algorithm 1 is the proposed algorithm.

```
for each node type = "leader node" {
    if new node registered then update {
        b← remaining battery life
            n ← number of registered nodes
            d ← distance from sp or client node
            threshold value t = b.w1(n.w2+d.w3)  } }
        Case 1: If (node entry= "service provider") then  {
          Receive  multicast message  to  discover  new
          lookup service within its neighbor
          If more than one leader node within range then
              find  leader  node  with  minimum  positive  thre
              shold value t and register with it } }
        Case 2: If (node entry = "client") then {
          Search for a lookup service.
              If more than one leader node within range,
              Then {find  leader  node  with  minimum  positive
                threshold value and register with it } }
         Case 3: If (node entry = "leader node") then {
            if within range of another leader node, then {
              registers with it then {
                    allows unregistered service providers to
                    register on it }}}
        Case 4: If (node exit = "leader node") then  {
        Service providers registered with this leader node will
        start to discover another leader node;
        Receive multicast message to discover new lookup ser
        vice within its neighbor;
        if more than one leader node within range, then  {
              register with leader node with minimum posi
              tive threshold value}}}
```

5 The Network Wide Search

Finally in this section, we present the process of the network wide search that a client performs when it searches for a service. The dynamic and self-configuring nature of Mobile Ad Hoc networks lead to formation of arbitrary topologies in the network [8]. The nodes are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Here the service provider sends a multicast registration request to every other node in the network to discover leader nodes. If it finds one or more such leader node, then it registers on a suitable leader node based on the minimum threshold value as described in the Leader Node Selection algorithm in Sect. 5. Now when the client needs to avail a particular service, it needs the service proxies of the registered services from the lookup service of the leader node on which that service provider providing the service is registered. But the client may not be within the range of communication of the leader node on which the required service is available. So a network-wide search for the service is performed here by utililizing the multi-hop nature of MANETs. The client sends its search request to the leader node within its direct communication range and with which it is registered. If the service is not available on the lookup service of this leader node, then this request is propagated by the concerned leader node to other leader nodes in the network as well using multi- hop. In Jini the client search request initiates with the default *ttl* (time to live) value 15. Each propagation of this search request by a node will decrease the *ttl* value by one. When the *ttl* value becomes 0 the search request will not proceed any further. This default *ttl* value can be changed by setting the appropriate property in the Jini system. The client search request may match with the service proxies on multiple lookup services. All these matched service proxies on the lookup services are unmarshalled by the client for further use. Figure 1 represents network wide searching in this system.

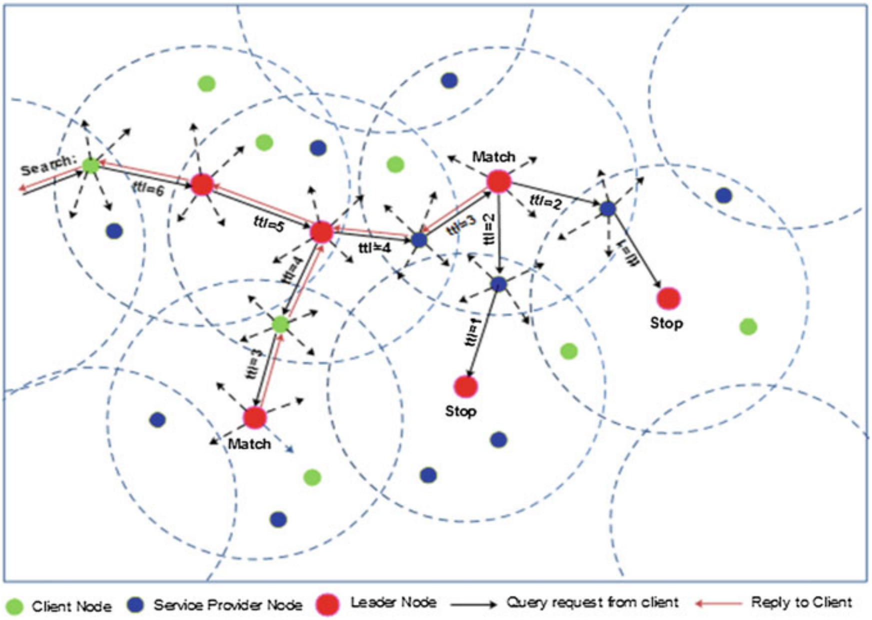

Fig. 1 The network wide searching

6 Conclusion and Future Work

This application has been tested within a limited range using 7–8 nodes. In future we need to test this framework on a larger real time area and with the number of nodes scaled up. However to be implemented on a larger scale, the implementing agencies need to follow uniformity in the design of the GUIs and other compatibility issues need to be smoothened out. The framework can be adapted to other scenarios also with minor modifications in the fields. Additionally the weights of all the parameters for determining node suitability have been taken to be 1. In future the weights need to be fine tuned and set to get more accurate results.

References

1. Kevin B, Kevin M, Scott R (2003) Self-adaptive leasing for JINI. In: PerCom 2003:539–542
2. Zhu F, Mutka MW, Ni LM (2005) Service discovery in pervasive computing environment. IEEE Pervasive Comput 4:81–90
3. Muhammad MR, Abdullah-Al-Wadud M, Chae O (2008) Performance analysis of leader election algorithms in mobile ad hoc networks. Int J Comput Sci Netw Secur 8(2):379–388
4. Zhang G, Kuang X, Chen J, Yu Z (2009) Design and implementation of a leader election algorithm in hierarchy mobile ad hoc network. In: Proceedings of 4th international conference on computer, science and education
5. Lee SS, Muhammad RM, Kim CG (2007) A leader election algorithm within candidates on ad hoc mobile networks. In: Embedded software and systems: third international conference, ICESS 2007
6. Roy M, Kar P, Mukherjee N (2010) An enhanced service framework in MANETs for application in emergency services. In: Proceedings of international conference on intelligent network and computing
7. Camp T, Boleng J, Davies V (2002) A survey of mobility models for ad hoc network research. Wirel Commun Mobile Comput (WCMC): Special Issue Mobile. Ad Hoc Netw 2(5):483–02
8. Roy M, Kar P, Mukherjee N (2010) Determining JINI leasing time limits using the random waypoint mobility model in mobile ad hoc networks. In: Proceedings of international conference on networking and information technology

http://www.springer.com/978-1-4614-6153-1

Computer Networks & Communications (NetCom)
Proceedings of the Fourth International Conference on
Networks & Communications
Chaki, N.; Meghanathan, N.; Nagamalai, D. (Eds.)
2013, XXIII, 865 p., Hardcover
ISBN: 978-1-4614-6153-1