2008－MBL－46 (6)
2008／9／25

Two-Layer Distributed Service Placement Method on Mobile Ad-hoc Networks

ASAAD AHMED , ,[†1] KEIICHI YASUMOTO ,,[†1] NAOKI SHIBATA ,,[†2]
TOMOYA KITANI ,[†1] and MINROU ITO [†1]

Abstract The effectiveness of service provisioning in mobile ad hoc networks (MANETs) depends highly on the number and location of services deployed at various hosts. Existing approaches try to determine the optimal service position by using centralized or distributed methods. In centralized methods, knowledge of global topological and demand information to adapt the number of service instances to the current service demand is needed. Meanwhile, distributed methods do not take a complete view of dynamic change in the network, thus likely result in a high message overhead. We propose an efficient distributed service placement method for MANETs in which a service is placed and replicated on some of nodes to minimize the whole communication cost and the global service discovery overhead by handling service requests at a nearby service node. In our proposed method, the network nodes classify themselves into two categories: *static nodes (SNs)* and *mobile nodes (MNs)*. All *SNs* construct a stable multi-hop network and each *SN* maintains a group of the *MNs* in its vicinity that construct a zone. We propose a heuristic algorithm to find the best static node to act as a service provider to other nodes and also to compute and locate the near optimal number of replicas of the service. Through simulations, we confirmed that our method improves the performance of service provision by 55% and 64% in terms of the total communication cost and message overhead, respectively, compared with an existing method.

1. Introduction

The recent development of wireless communications enabled users to utilize useful services anytime and anywhere. Some of these services are supported by a dynamic model provided by Mobile Ad-hoc Networks (MANETs). MANETs consist of devices which establish temporary connections as they want to communicate, without any previously deployed infrastructure, thus cooperative behavior of all devices is needed to allow communication beyond the wireless coverage of a single node.

The effectiveness of service provisioning in mobile ad hoc networks (MANETs) depends highly on the number and location of services deployed at various hosts. The problem of determining the appropriate nodes in MANETs to act as server is referred to as the *service placement problem*. This is a challenging problem due to the dynamic nature of MANETs.

For a practical example, consider a directory service which is hosted on a node with an optimal position in the network in terms of overall routing hops to clients of the service. In case if the node hosting the service moves away from its current position, when to move the service and which of the nodes is the best suited to be the new host are determined by a service placement algorithm.

In this paper, we propose a new distributed service placement and service replication method to increase the effectiveness of service provision in MANETs. It is important to satisfy the following when considering an efficient service placement and service replication system on MANETs: (1) low communication cost, (2) low global service discovery overhead by handling service requests at a nearby service node, and (3) stability against dynamic behavior of nodes.

In order to achieve the above goals, we adopt the following ideas: (1) classify mobile nodes to static and mobile nodes depending on their speeds and (2) construct a stable network consisting of only static nodes and let each of static nodes manages part of mobile nodes that construct a zone. Our motivation in using this two-layer architecture is to decrease the message overhead in MANETs by constructing a stable network which contains static nodes only. These static nodes do not need to update their routing tables frequently. Meanwhile, the frequent update is needed between each static node and the mobile nodes in its zone. On the other hand, without two-layer architecture, all nodes in the network need this frequent update. In this case, the message overhead becomes very high.

Through simulations, we confirmed that our method improves the performance of service provision by 55% and 64% in terms of the total commu-

†1 Graduate School of Information Science, Nara Institute of Science and Technology
†2 Department of Information Processing and Management, Shiga University

nication cost and message overhead, respectively, compared with an existing service placement algorithm.

The rest of this paper is organized as follows: related work is introduced in section 2. Models, assumptions, problem definition, and proposed distributed algorithm are described in section 3. Simulations and results are introduced in section 4, and section 5 concludes the paper.

2. Related work

Service placement in MANETs is either associated with a byproduct of middleware research or an application of facility location theory. A heuristics based on information gathered from nodes neighboring to current service provider, is commonly employed in the middleware-related approaches[1),7),8),11)]. None of these algorithms supports truly distributed services. While facility location theory approaches[2),3),5),6),9)] either solve uncapacitated facility location (UFL) problem with traditional algorithms on a central node after collecting the necessary information from the network[3),6)] or use distributed iterative approximations[2),5),9)]. Clearly, the centralized approaches need knowledge of global topological and demand information to adapt the number of service instances to the current service demand and thus do not scale for MANETs. On the other hand, the distributed methods do not take a complete view of dynamic change in the network, thus likely result in high message overhead.

A service is defined as a software component located on one or several nodes of the network. These nodes reply to the service requests received from client nodes. If the same service component is located on several nodes, these components are called *service instances*. We can classify the existing algorithms proposed for service placement into centralized and distributed algorithms. In centralized algorithms, the service is executed in a centralized manner on one node, while in distributed algorithms, the service is executed distributively in the form of an adaptable number of identical service instances. In the next two subsections, we survey some of centralized and distributed algorithms that were proposed to solve the service placement problem.

2.1 Centralized algorithms

REDMAN middleware[1)]was described by Bellavista et al, aiming at supporting resource replication in dense MANETs. The placement of replicas depends upon a simple heuristic, the placement of the controlling entity and the replication manager, and it considers local network topology. When a node n which has become part of a dense network region, n begins the process of selecting which node to host the replication manager. Then, n starts a broadcasting query towards the topological direction in which the most-distant node of the dense MANET is located.

Another algorithm was presented by Oikonomou and Stavrakakis[10)], which adopts a policy for placing a single service in a MANET. Their approach, similar to hill climbing algorithms, is to iteratively migrate the service from its current host node to the neighboring node if the neighboring node achieves lower communication cost. The service remains at its current location if no such neighboring node exists.

2.2 Distributed algorithms

The problem of service availability was addressed by Wang and Li[7)] according to network partitioning due to node mobility. Their algorithm is to group nodes by their velocity vectors and to predict the event of that group moving out of the radio range of the other group, resulting in partitioning the network. If a service is provided by a single node to both mobility groups, a new service instance is created on one node in the mobility group that would leave without access to the service in the other group. The node that is currently hosting the service establishes, using algorithm that is running on it, which node in the leaving group should host the new service instance. Information about the locations and velocities of the leaving nodes is piggybacked on service requests. According to this information, the current host chooses the node to which the data required to provide the service can be moved before the network partition happens. Once the availability of a service instance with a higher unique identification number is detected, redundant service instances in the same mobility group shut down.

Sailhan and Issarny[11)] present an architecture for service discovery in MANETs built around a homogeneous deployment of cooperating service directories. Their goal is to minimize global service discovery overhead by handling service discovery requests locally at a nearby directory. In their method, any node without access to a directory broadcasts a query for available resources and network topology information to nodes within its n-hop neighborhood. Then, the initiating node selects a node for hosting a new directory. The main selection criteria is the expected coverage of the new directory in terms of number of neighbors and number of other directories in the vicinity. The node that best matches these criteria is then notified of the decision and initializes a new service directory by requesting data from existing directories.

Laoutaris et al.[6] propose a distributed algorithm to solve the Uncapacitated k-Median (UKM) and Uncapacitated Facility Location (UFL) problems. This approach means to make up for the lack of global knowledge of the whole topology by limiting the scope of the problems to the n-hop neighborhood of the nodes currently hosting a service instance. It assumes that exact knowledge of network topology and service demands are available for this area. The service demand of nodes outside this area is taken into account by mapping it to the outer nodes of the neighborhood. There may be multiple overlapping neighborhoods, since the approach considers multiple service instances, in this case they need to be merged and considered collectively. These steps are applied iteratively to all service instances until the set of relocated service instances is empty.

A brief survey of recent approaches to service placement in MANETs was introduced in[12].

In summary, the existing centralized methods need knowledge of global topological and demand information to adapt the number of service instances to the current service demand and thus do not scale for MANETs. The existing distributed methods do not take a complete view of dynamic change in the network and thus incur high message overhead. On the other hand the proposed method takes into account the dynamic change in the network topology over time and determines when the service needs to migrate and which node is best suited to be the new host. Also, the proposed method reduces message overhead by constructing a stable backbone network consisting of nodes with a speed lower than a certain threshold.

3. Service Placement Problem

3.1 Models and Assumptions

In every communication network it is the responsibility of the employed routing protocol to provide a mechanism for forwarding the data packets. In this paper, we assume that the routing protocol is capable of finding the shortest path between any two nodes. We assume that each node knows its location with GPS or other means and knows existence of all neighbor nodes in 1-hop and their locations. Also, we do not consider the cost to replicate services on every node.

The mobile nodes can have different behavior depending on the roles they play in the environment. Based on the mobility characteristics of nodes, we will classify them into two categories: *Static Node (Sn)* and *Mobile Node (Mn)*.

A *Static node (Sn)*: is defined to represent the nodes which are moving with a speed lower than a certain threshold. The pedestrians sitting at theater, cinema, or park to take a rest are examples of static nodes.

A *Mobile node (Mn)*: is defined to represent the nodes which are moving with a speed higher than the threshold.

With the consideration of the nodes classification, we construct a MANET of *Two-Layers*: *Static layer* which contains all *SNs* in the network and *Mobile layer* which contains all *MNs* in the networks. Every node can change its state from *Static* state to *mobile* state and vice versa based on the observed speed of the node. For example, a mobile node changes its state to static state if its speed is smaller than the threshold *SP* and a static node changes its state to mobile node if it moves to new location with speed no less than *SP*. In addition, letting $Nig(u)$ be a set which contains all nodes in the radio range of *u*, any new static node, *x*, must satisfy the following constraint:

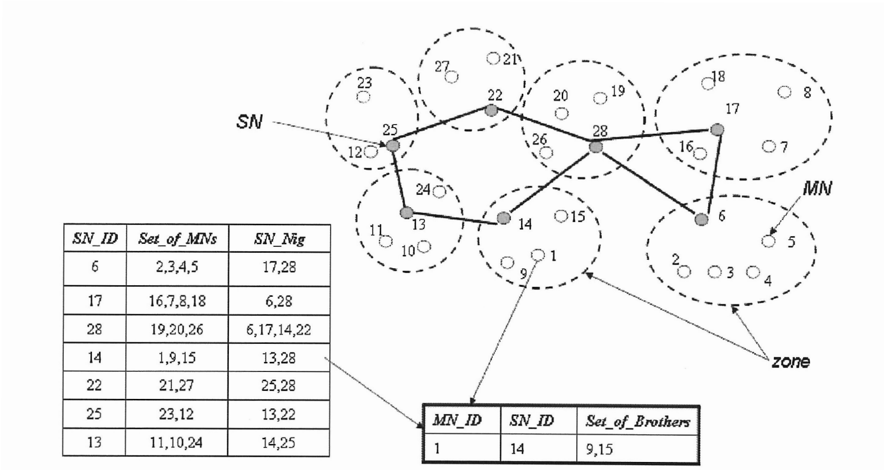
$$\| (u,x) \| > \| (u,v) \| \wedge \| (v,x) \| > \| (u,v) \| \quad (1)$$

where $u \in Nig(v) \wedge v \in Nig(u)$, and $\|\|$ is euclidian distance. This constraint regulates the number of static nodes in the whole network.

The network topology is represented by an undirected graph $G = (V, E)$, where V is the set of nodes and E the set of links among them. Let $d(v_i, v_j)$ denote the number of hops in the shortest path between v_i and v_j, Let $S = \{s_1, s_2, ..., s_k\}$ denote the set of services, and $Serd(s, v_i)$ the service demand originating (expected number of service requests) from node v_i for a service s per second.

We further make the following definitions:

- Let SN be the set which contains all static nodes in the network.
- Let MN_u be the set which contains all mobile nodes in the zone of static node u .
- Let $GSN = (SN, L)$ be the sub-graph which contains all static nodes in *SN* and all undirected edges between all nodes in SN.
- Let W_u be the set which contains a static



SN_ID	Set_of_MNs	SN_Nig
6	2,3,4,5	17,28
17	16,7,8,18	6,28
28	19,20,26	6,17,14,22
14	1,9,15	13,28
22	21,27	25,28
25	23,12	13,22
13	11,10,24	14,25

MN_ID	SN_ID	Set_of_Brothers
1	14	9,15

Fig. 1 Network Architecture: *Sn_Table*, *Mn_Record*, and Zones.

node u and all mobile nodes in zone of *u*, i.e. $W_u = MN_u \bigcup \{u\}$.

- Let $SG_u = (W_u, D_u)$ be the sub-graph which contains all nodes in W_u and all undirected edges between all nodes in W_u .
- Let $aggSerd(s, u, W_u)$ be an aggregate service demands for a service s from all nodes in W_u, and defined by $\sum_{\forall v_i \in W_u} Serd(s, v_i)$.
- Let $speed(u)$ be the speed of node *u* .

Hereafter, we consider the case that $|S| = 1$, for simplicity.

Based on the two-layer architecture, we formulate the service cost for the static layer as follows:

$$cost(GSN, s, v_j) = \sum_{\forall v_i \in SN} aggSerd(s, v_i, W_{v_i}) d(v_i, v_j) \quad (2)$$

where $v_j \in SN$ is the service provider.

We define the service cost for every static node u's zone as follows:

$$cost(SG_u, s, v_j) = \sum_{\forall v_i \in W_u} Serd(s, v_i) d(v_i, v_j), \quad \forall u \in SN \quad (3)$$

where $v_j \in W_u$ is the service provider.

3.2 Problem Definition

Definition 1: Given a graph $G = (V, E)$, $d(v_i, v_j)$, and $Serd(s, v_i)$. The problem is to select a subset $R \subseteq V$ to act as service nodes so as to minimize the total cost $Cost(V, s, R)$:

$$Cost(V, s, R) = \sum_{\forall v_i \in V} Serd(s, v_i) d(v_i, v_j) \quad (4)$$

where $v_j \in R$ is the service provider that is closest to v_i.

Based on our models and assumptions, by using (3) and definition 1, the objective function of the problem can be rewritten in the following form:

$$\textbf{minimize} \sum_{\forall u \in SN} cost(SG_u, s, v_j) \quad (5)$$

subject to (6), (7)

$$speed(u) \leq SP, \forall u \in SN \quad (6)$$

$$v_i \notin W_z, \forall z \in SN - \{u\}, \forall u \in SN, \forall v_i \in W_u \quad (7)$$

The expression (6) means that the speed of all nodes in static layer is less than or equal to *SP* and The expression (7) means that every mobile node belongs to only one zone.

3.3 Algorithm

The basic ideas of our approach are: (1) classify mobile nodes to static and mobile nodes depending on their speeds and (2) construct a stable network consisting of only static nodes and attach mobile nodes to *SN*-nodes. So, our approach consists of two phases: in the first phase, we find which static node among all static nodes acts as a service provider and in the second phase, we determine which node in every zone may host the *service instance (replica)*. Our objective is to find the set of service provider nodes *R* which minimizes the objective function (5). Below, we will describe our approach components:

1- ***Sn_Table***(**Fig. 1**): Each *Sn* has a table which consists of the following fields:
   - (a) ***Sn_ID***: ***ID*** of ***Sn***.
   - (b) ***Set_of_Mns***: The set of all the nearest mobile nodes to ***Sn***.
   - (c) ***Sn_Nig***: All ***Sn***'s static neighbor nodes within one-hop.

2- ***Mn_Record***(Fig. 1): Each ***Mn*** has a record which consists of the following fields:
   - (a) ***Mn_ID***: ***ID*** of ***Mn***.

(b) ***Sn_ID***: ***ID*** of the nearest static node.

(c) ***Set_of_Brothers***: The set of all mobile nodes which have the same ***Sn_ID*** of this ***Mn***.

All nodes in static layer construct an ad hoc network and every *Sn* maintains all mobile nodes whose nearest static node is *Sn* i.e. every static node with those *Mns* construct a zone. We assume that every *Mn* can get the nearest *Sn* from its 1-hop neighborhood nodes. The network architecture: *Sn_Table*, *Mn_Record*, and *Zones* are shown in Fig. 1.

To solve the problem based on the basic ideas our distributed algorithm consisting of three steps are described as follows:

1- Put the service s at a static node *u* which produces minimum $cost(GSN, s, u)$.

2- Compute numbers and locations of *service instances* of s by computing $cost(SG_u, s, v)$ $\forall u \in SN$ and $\forall v \in W_u$ if $M_u \neq \emptyset$ and there is no service in SG_u.

3- Put a *service instance(replica)* of s at $v \in W_u$ which produces minimum $cost(SG_u, s, v)$ $\forall u \in SN$.

The algorithm is summarized in algorithm(1) and associated parameters are described in Table 1.

Example:

To show the process of our algorithm consider MANETs with 14 nodes, a service, s, and each node is labeled by a pair $(ID, Serd)$ where *ID* is a node *ID* and *Serd* is a service demand for s per second as shown in **Fig. 2**. Let $SN = \{1, 6, 14\}$. In the first step by computing the service cost for each static node by using equation(2), we get:

$cost(GSN, s, 1) = 24$, $cost(GSN, s, 6) = 17$
$cost(GSN, s, 14) = 26$

so the service must be placed at node 6 and service cost for 6's-zone = 11.

In the second step by computing the service cost for every static node zone by using equation(3), we get:

For 1's-zone
$cost(SG_1, s, 1) = 7$, $cost(SG_1, s, 9) = 14$
$cost(SG_1, s, 10) = 8$, $cost(SG_1, s, 13) = 12$

For 14's-zone
$cost(SG_{14}, s, 7) = 7$, $cost(SG_{14}, s, 8) = 11$
$cost(SG_{14}, s, 11) = 15$, $cost(SG_{14}, s, 12) = 15$
$cost(SG_{14}, s, 14) = 9$

so the service instance must be placed at node 1 and 7 for 1's-zone and 14's-zone, respectively, and the total cost = 11+ 7+7 = 25.

Table 1 Algorithm parameters

Parameter	Discription
SN	the set of static nodes in the network
GSN	the stable network of static nodes
MN_u	the set of all mobile nodes in static node u's zone
W_u	$MN_u \cup \{u\}$
SG_u	the network of all nodes in W_u

Algorithm 1 Finding a set of service nodes *R*

```
j = 1, i = 1
minimumCost ← cost(GSN, s, SN(j))
R(i) ← SN(j)
j = j + 1
while j ≤ |SN| do
  CostV ← cost(GSN, s, SN(j))
  if minimumCost > CostV then
    minimumCost ← CostV
    R(i) ← SN(j)
  end if
  j = j + 1
end while
i = i + 1
for all u ∈ SN do
  if MN_u ≠ ∅ and no service in SG_u then
    j = 1
    minimumCost ← cost(SG_u, s, W_u(j))
    R(i) ← W_u(j)
    j = j + 1
    while j ≤ |W_u| do
      CostV ← cost(SG_u, s, W_u(j))
      if minimumCost > CostV then
        minimumCost ← CostV
        R(i) ← W_u(j)
      end if
      j = j + 1
    end while
    i = i + 1
  end if
end for
return R
```

4. Simulation Results

4.1 Simulation Environment

The QUALNET[13)] simulator was used to evaluate the performance of our distributed service placement method. Simulations were run for a duration of 100s. We considered an ad hoc network of mobile terminals with transmission range *t*=100m, where mobile terminals were randomly distributed over surface $S = 1500m$ x $1500m$, and the node

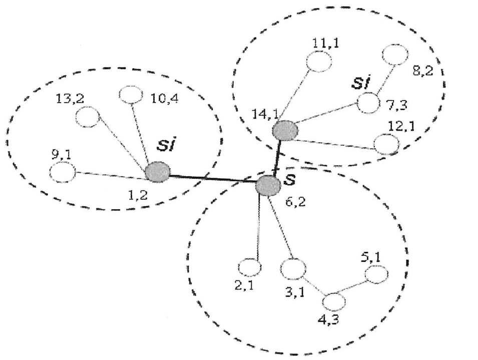


Fig. 2 TLDSP example, $N = 14$, service, s,and service instance, si.

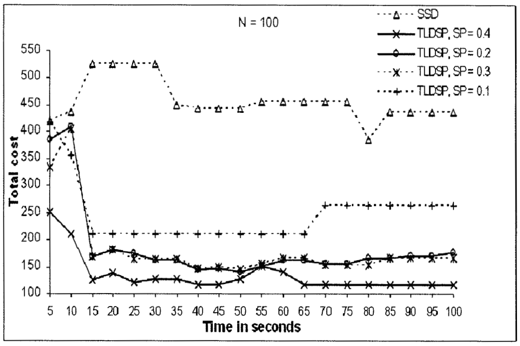


Fig. 3 Total Cost vs. Time for N=100

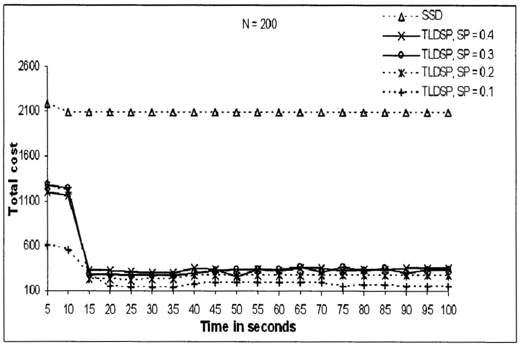


Fig. 4 Total Cost vs. Time for N=200

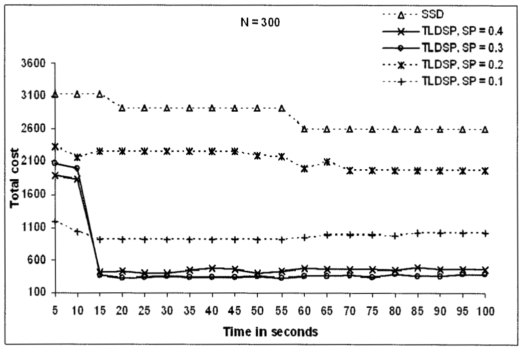


Fig. 5 Total Cost vs. Time for N=300

mobility is based on a random way point mobility model[4] with the minimum and maximum speeds of 0 and 1 *m/sec*, respectively. We implemented the proposed algorithm on top of IEEE 802.11b MAC protocol. In order to evaluate the effectiveness of our method, we compared our method with SSD[11] in terms of total cost, average number of services, and message overhead for different values of *SP*, threshold speed between static and mobile states.

4.2 Simulation Results

1- Total cost: We measured the total cost, which defined by objective function (5), over time when *SP* was changed from 0.1 to 0.4 *m/sec*, **Figs.**3 to 6 show the total cost of two methods over time for different number of nodes. the propsed algorithm for different values of *SP* kept the total cost as low as possible and decreased the total cost by 55% compared to the total cost kept by SSD algorithm. Also, when *SP* increased the total cost increased or decreased because the total cost depends on how many static nodes exist in the static layer, in other words, how many zones exist in the networks. Here, the construction of static layer depends on constraint (1).

2- Average number of services: We measured the average number of service instances when *SP* was changed from 0.1 to 0.4 *m/sec*, **Figs.**7 and 8 show the average number of services and average cost of two methods with different number of nodes, respectively. The proposed algorithm, for different values of *SP* can adapt number of services instances required to keep the total cost as low as possible. On the other hand, SSD method can not adapt the required number of services because SSD method does not take into account the service demands or how many hops exist between the service node and the client node.

3- Message overhead: We measured the message overhead ,as the total number of generated messages,when *SP* was changed from 0.1 to 0.4 *m/sec*, **Figs.**9 to 12 show the message overhead over time for different number of nodes. The message overhead for our method for different values of *SP* decreased by 64% than SSD method. Also, the message overhead for our method increased when

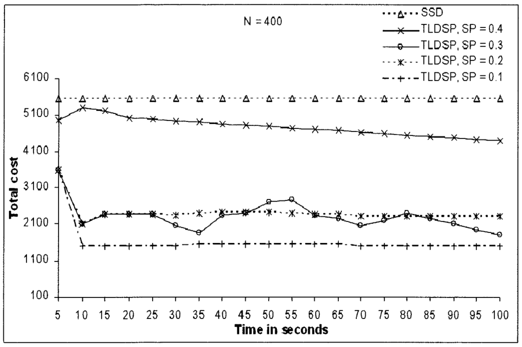


Fig. 6 Total Cost vs. Time for N=400

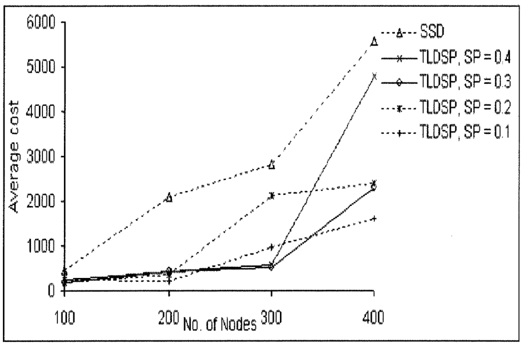


Fig. 7 Average Cost. vs. No. of Nodes

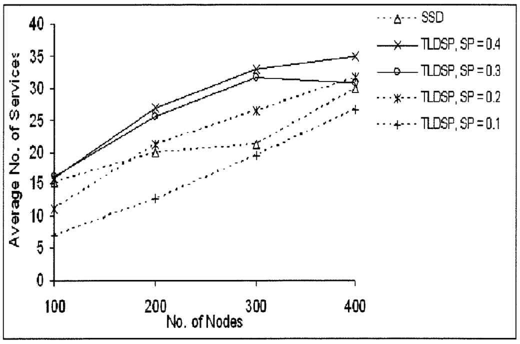


Fig. 8 Average No. of Service vs. No. of Nodes

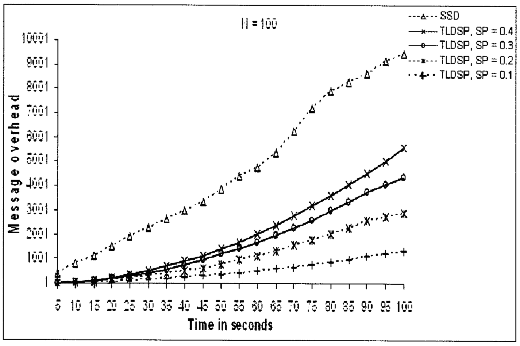


Fig. 9 Overhead vs. Time for N=100

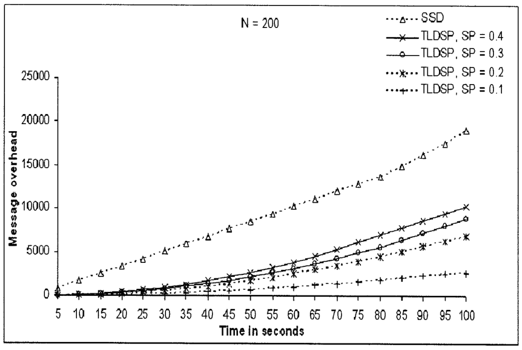


Fig. 10 Overhead vs. Time for N=200

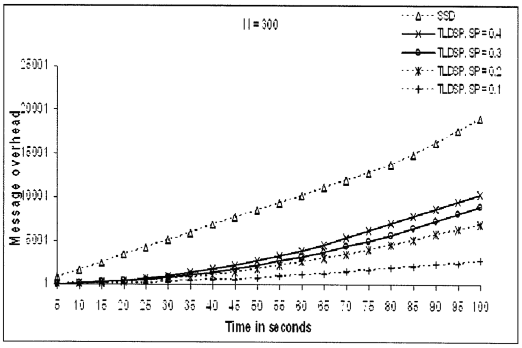


Fig. 11 Overhead vs. Time for N=300

SP increased because the number of static nodes increased when *SP* increased. Also, **Fig. 13** shows the average message overhead with number of nodes. This figure shows that our method for different values of *SP* is more scalable than the SSD method to maintain the network.

5. Conclusion and Future work

In this paper, a new distributed service placement and service replication method to increase the effectiveness of service provision in MANETs, was presented. Our approach classifies MANET into two layers: *Static layer* and *Mobile layer* based on the speed of nodes. All static nodes in the static layer construct a stable multi-hop network and each *SN* maintains a group of *MNs* that construct a zone. Our approach can adapt the locations and number of services required over time to keep the cost as

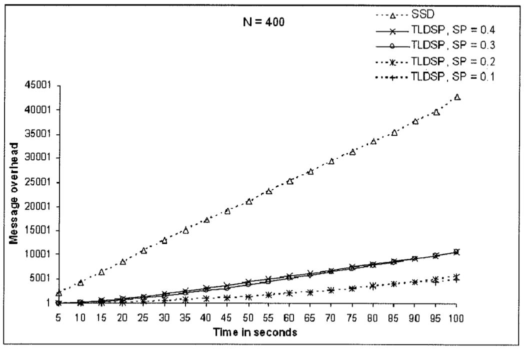


Fig. 12 Overhead vs. Time for N=400

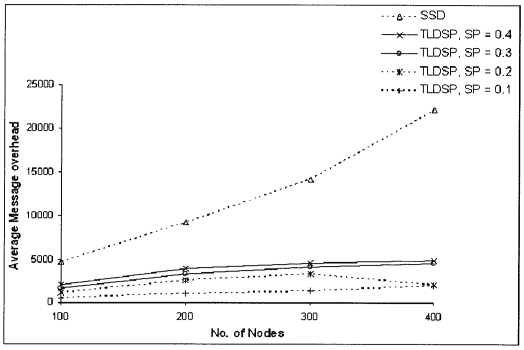


Fig. 13 Average Overhead vs. No. of Nodes

low as possible. Simulations demonstrated that our method improves the performance by 55% and 64% in terms of the total communication cost and message overhead, respectively compared with an existing service placement algorithms. In this paper, we did not treated power consumption and message overhead for service replication. In the future work, we will consider these metrics. Also, we will improve a distributed service placement for disjoint services and show how we can combine our method with service discovery protocols to increase the performance of service provision in MANETs.

References

1) Bellavista, P., C.A. and Magistretti, E.: REDMAN: An Optimistic Replication Middleware for Read-only Resources in Dense MANETs, *Pervasive and Mobile Computing*, Vol.1, No.3, pp.279–310 (2005).
2) Frank, C. and Romer, K.: Distributed Facility Location Algorithms for Flexible Configuration ofWireless Sensor Networks, *Distributed Computing in Sensor Systems(DCOSS '07)*, pp.124–141 (2007).
3) Furuta, T., S. M. I. F. S.A. and Miyazawa, H.: A New Clustering Algorithm Using Facility Location Theory for Wireless Sensor Networks, *Nanzan Academic Society(NANZAN)Technical Report* (2007).
4) Johnson, D. and Maltz, D.: Dynamic source routing in ad hoc wireless networks, *Mobile Computing*, pp. 153–181 (1996).
5) Krivitski, D., S.A. and Wolff, R.: A Local Facility Location Algorithm for Large-Scale Distributed Systems, *Grid Computing* (2006).
6) Laoutaris, N., S. G. O. K. S.I. and Bestavros, A.: Distributed Placement of Service Facilities in Large-Scale Networks, *Proc. of 26th Annual IEEE Conf. on Computer Communications*, pp.2144–2152 (2007).
7) Li, B. and Wang, K.H.: NonStop: Continuous Multimedia Streaming in Wireless Ad Hoc Networks with Node Mobility, *Selected Areas in Communications*, Vol.21, No.10, pp.1627–1641 (2003).
8) Liu, H., R. T. W. K. B.R. and Sirer, E.G.: Design and Implementation of a Single System Image Operating System for Ad Hoc Networks, *Mobile Systems, Applications, and Services*, pp.149–162 (2005).
9) Moscibroda, T. and Wattenhofer, R.: Facility Location: Distributed Approximation, *Proc. of 24th ACM Symp. on the Principles of Distributed Computing*, pp.108–117 (2005).
10) Oikonomou, K. and Stavrakakis, I.: Scalable Service Migration: The Tree Topology Case, *Proc. of 5th Annual Mediterranean Ad Hoc Networking Workshop* (2006).
11) Sailhan, F. and Issarny, V.: Scalable Service Discovery for MANET, *Pervasive Computing and Communications(PerCom '05)*, pp.235–244 (2005).
12) Wittenburg, G. and Schiller, J.: A Survey of Current Directions in Service Placement in Mobile Ad-hoc Networks, *Proc. of 6th Annual IEEE Intl. Conf. on Pervasive Computing and Communications(PerCom '08)* (2008).
13) Qualnet 4.0 Network simulator, *Scalable Network Technologies Inc.* available at http://www.scalable-networks.com.