

An Efficient Cluster Based Service Discovery Model for Mobile Ad hoc Network

Buvana M¹, Suganthi M²

¹Computer Science and Engineering, PSNACET
Dindigul, Tamilnadu, India.
bhuvana_beula@yahoo.co.in

²Thiyagarajar College of Engineering, Madurai
Tamilnadu, India.
suganthi@tce.edu.in

*Corresponding author: Buvana M

*Received July 22, 2014; revised October 6, 2014; revised October 30, 2014; accepted December 19, 2014;
published February 28, 2015*

Abstract

The use of web service has been increased rapidly, with an increase in the number of available services, finding the exact service is the challenging task. Service discovery is the most significant job to complete the service discoverers needs. In order to achieve the efficient service discovery, we focus on designing a cluster based service discovery model for service registering and service provisioning among all mobile nodes in a mobile ad hoc network (MANETs). A dynamic backbone of nodes (i.e. cluster heads) that forms a service repository to which MANET nodes can publish their services and/or send their service queries. The designed model is based on storing services with their service description on cluster head nodes that are found in accordance with the proposed cluster head election model. In addition to identifying and analyzing the system parameters for finding the effectiveness of our model, this paper studies the stability analysis of the network, overhead of the cluster, and bandwidth utilization and network traffic is evaluated using analytic derivations and experimental evaluation has been done.

Keywords: Ad hoc network, web services, service discovery, performance analysis.

1. Introduction

Service discovery is one of the important tasks for using MANET. This is because the Mobile ad hoc networks are formed to benefit the user with more number of services, and they should somehow try to identify the services. Mobile Ad-Hoc Networks (MANETs) [1] represent a system of wireless mobile nodes that can move freely and dynamically self-organized in random and short term network topologies and does not require wired backbones or centralized centers. Since it has no fixed infrastructure, it is easy to implement and organize at anytime and anywhere. All nodes are permitted to be mobile, and the group of such networks is time varying. Due to the mobility and ad hoc nature of the MANET applications can be handled over and done with dynamic mechanisms, such as service discovery. Towards that we proposed the cluster based service discovery model (CSDM) is simple enough to run on various devices such as printers, mobile phones; laptops are mutually connected with each other to automatically find services or resources and is shown to be energy efficient on a group of network. For discovering the services in a MANET, a new proposal has been designed to be energy efficient.

In this paper, we plan to give a solution for service discovery in ad-hoc network based on clustering using a weight calculation method with different factors. The set of cluster head nodes acts as a service repository for registering the services of the member nodes in the cluster. In this way we can attain low discovery time, since the service messages are exchanged only between the nodes from the service repository. Our goal is to reduce the energy consumed for the period of the discovery phase. The objectives of this paper are therefore: To form a clustered structure, for that each member node computes the weight using the factors such as connectivity, mobility, resources that need to improve the performance and to assign the cluster head node and calculates the distance between the cluster head and neighbor node to form a self-organized cluster and to find the topology changes due to the mobility. To offer the service requester the chance to find the services available in the network. For that each node listens the service information from adjacent node and stores the service information in its local cache entry for spreading advertisements to cluster head. And to prove that our CSDM gives the best performance to reduce the traffic during service discovery using some analytical derivations.

Since the practical point of view, we show that our model is lightweight. The rest of the paper is organized as follows. In Section 2, we review the previous work in service discovery. Section 3 describes the proposed model. Section 4 presents an optimization functionality of the model. The implementation details will be given in section 5. Analysis and experimental evaluation addressed in section 6. Section 7 describes the challenges that we faced during implementation. Finally we present the conclusion in section 8.

2. Related Work

Katsigiannis [1] demonstrate a service discovery algorithm based on the capability to select a service provider by capturing the following metrics, such as the energy of the service provider and connected path towards the destination. This algorithm is not a centralized. By adding too much signaling load overhead is increased. Mallah and Quintero [2], describes about light weight service discovery algorithm for ad hoc networks. This arranges sensibly selected (stable) nodes in the location of the service discovery procedure. Backbone nodes are

selected based on the power, speed. This will maintain the services. Other than backbone nodes free from maintaining the services. This method minimizes network load, success hit ratio of service request initiated and reduce the delay. This method does not think about separating the network.

The Weighted Clustering Algorithm (WCA) [3] to form a one hop clusters with one cluster head based on the weight metric. Nodes weights are calculated by the following factors, that are number of adjacent nodes, the total distance to all its adjacent nodes, mobility [4] [5] and remaining battery power [6]. This algorithm takes more time and reduces the scalability.

Clustering based [7] web service discovery is used to find the service based upon cluster. Services can be grouped based on operation, and / or input / output of the services. Discovering the appropriate cluster is difficult in cluster based method. If recognition of the type is hard or not feasible, then broadcasting method is used to discover the services of all types. So, the computation cost is very high if the system fails to obtain the exact category.

Adam Funk [8], discussed and, the classification of web service documents to ontology entities. He developed the system using information extraction with machine learning techniques. The limitation is performance of accuracy measurement model is very low.

Mamoun Mohamed [9] has demonstrated the ontological based indexing to enhance the accuracy of service discovery. The method ontology was represented to avoid lexical matching between terms from the query. The drawback of this work is irrelevant services may also retrieve.

Chakraborty, et.al [10] describes a protocol as Group Based Service discovery (GSD) is based on the concept of advertising their services in peer to peer manner and forwarding the request and the requests are not stored into the registry. The nodes are categorized into groups based on the parent - child tree format. Root node is called as service node. It introduces new metric called 'service discoverability' to discovering services. Due to the high mobility, the service has high chance of moving out the route.

Hassan Artail [11] describes the distributed architecture, where service directories act as a virtual backbone of locating and registering the services is proposed. Our system does not need to apply any condition to locate the head node, but we select the cluster head according to their weights.

Our proposed model extends the work to achieve the efficient service discovery on lightweight clustering method and it provides low maintenance overhead and discovery cost is low in the large scale network. We show the realistic and effective solution by our implementation on the real network.

3. Cluster Based Service Discovery Model

The proposed model is called as Cluster Based Service Discovery Model (CSDM). In CSDM, the model has three components there are service requestor node (SRN), service provider node (SPN), service repository called as a cluster head node (CHN). SPN can periodically distribute their information with service name, service type, service code, service description, service documentation, Exp_T to the CHN in the cluster. Our proposed model is designed for energy efficiency and scalability. The main theme of the network is self - organized cluster in which the cluster head is selected based on the node weight value (grade). The service discovery message is flooded over the CHNs which form a service repository for service registration.

3.1 Clustering Algorithm

The Clustering algorithm makes possible to the cluster construction and maintenance of a

clusterhead. Our algorithm contains the following feature:

- Decisions are made based on the adjacent node information to make the fast convergence in terms of topology changes.
- The algorithm constructs only small number of clusters.

Cluster is managed by a cluster head node. For identifying the clusterhead, each node is assigned a weight, named as node weight value (NWV), on behalf of the node's dynamics and available resources.

3.1.1 Cluster Setup

Based on the neighboring nodes local information the clustering algorithm is constructed. our CSDM has the following methods.

- Nodes that have the highest NWV among their adjacent nodes declare themselves and broadcast a *setHead* message declares their roles.
- After a node accepts the *setHead* message from its head; it finds the cluster membership and rebroadcast the *setHead* message.

3.1.2 Cluster Head Election method and Maintenance

Cluster head selection is invoked when the new node join in the cluster or moved from the cluster. Initially, each node broadcasts a beacon messages to notify its presence to the neighbours. First identify the cluster head node which manages the services on a cluster. On the other hand, perform a distributed basis setting up a relationship between the aforesaid central nodes (CHNs') on a mobile ad hoc network. Cluster head node has been defined for the use of service discovery, but the responsibilities of this node could go away from the service discovery, to act as a supervision node for service provisioning. Hence, the election and maintenance of the cluster head node within a cluster is vital. The election could be based on several different features that possible CHNs, known as nodes, have. Though not all have the same importance for taking on CHN responsibility and should be weighted accordingly to a decided range. The same cost function has to be used on every node and, for example, it includes a combination of values referring to connectivity, mobility, resources used by the node. As a result, the N will be the CHN with the higher weight. Offering a dynamic and easily reconfigurable method for the election and maintenance should be considered for the service discovery system setup. It is also important that CHN function transfer between the new and former nodes could also be performed.

The following method to select a CHN to check and satisfies the following criteria

- Highest possibility to reside for longer time within the cluster. since, cluster member nodes can change over time; and
- Minimum distance from the individual member nodes of the cluster.

We use a parameter called Node Weight Value (NWV) widely for finding a CHN. NWV indicates the node's appropriateness for the CHN node function. The rules for calculating NWV are applied at all the individual nodes and it is explained by the general formula (1)

$$\{N_n\} | F_X^n > \tau_X, \quad X = \{1/\alpha_1 \times R + \alpha_2 \times CTY + \alpha_3 \times MOB\} \quad (1)$$

Where $\{N_n\}$ the set of nodes are, F_X^n is the factors taken to compute the weight of an n^{th} node. τ_X is the threshold value for a factor X. Among the $\{N_n\}$ maximum weight is selected. That is,

if j^{th} node is selected then,

$$W_j = \max\{W_n\} | W_n = \sum \alpha_X F_X^n \quad (2)$$

The parameters, α_1, α_2 and α_3 depends on the weight factors for the particular applications they can be positive or negative in some cases. For example, α_3 can be negative, owing to the negative impact of mobility. However it is better to keep α_1, α_2 positive and summing to unity. The calculation of Connectivity (CTY), Mobility (MOB), and Resources (R) is explained below.

3.1.2.1 Connectivity (CTY)

An ad hoc network topology is modelled as an undirected graph G where $G = (C, E)$. C represents clusters in the network and E represents an edge which connects two nodes within a cluster. Within a cluster C_i , each cluster head node CHN_i has n spatially associated neighboring nodes. The nodes in C_i are represented by $SSN_{ij} = \{SSN_{ij}; j = 1 \dots k\}$, i.e., $N(C_i) = \{SSN_{ij} \in C_i \mid (SSN_{ij}, CHN_i) \in E\} \cup \{CHN_i\}$. The distance $D = (U, V)$ between two nodes u and v is the least number of edges on a path from v to u , i.e., hops. The i^{th} neighbourhood of a node v is the set of nodes $N_i(u) = \{v \mid v \in E, v \neq u, D(u, v) \leq i\}$. The next direct node is $N_1(u)$, which is the shortest neighbours of u . We represent the total number of direct neighbours as with $|N_1(u)|$ and we call it the degree of the node. So we define connectivity as C , which is specified by (2)

$$C(v) = \sum_{m=1}^K \frac{1}{n} \sum_n^{N_m(u)} \chi_n \quad (3)$$

Where K is the m^{th} adjacent node; $|N_m(u)|$ denotes the number of nodes in the m^{th} neighbourhood; and χ_j indicates the combination of transmission range and mobility of n^{th} node in m^{th} adjacent node.

3.1.2.2 Mobility (MOB)

Mobility MOB , is calculated taking into report the information belonging to the node with reference to its neighbours, thus it presents relative mobility characteristics. We use the metric proposed by Basu [12]. Using the ratio of received signal power (RP) between two successive MEMB packets for the estimation of relative mobility between the two nodes can be found. The relative mobility, $MOB_i^{rel}(j)$ at a node i by means of node j is found in (4) using [12]

$$MOB_i^{rel}(j) = 10 \log_{10} \frac{RP_{j>i}^{new}}{RP_{j>i}^{old}} \quad (4)$$

If $MOB_i^{rel}(j)$ is negative then the two nodes are moving away from each other, if it is positive then the nodes are moving towards each other. For a node i with all its neighbours $N_1(i)$ MOB_i is calculated using relative mobility metric $MOB_i^{rel}(j)$, where j is a neighbour of i ,

$$MOB_i = \text{var}_0 \{MOB_i^{rel}(j)\}_{j=1}^{N_1(i)} \quad (5)$$

A minimum value of MOB_i represents that node i is relatively less mobile node with respect

to its neighbour nodes. Thus the parameter, α_3 in equation (1), may have negative value so that node with less mobility should have a moderately higher node capacity to provide as a CHN. Subsequently we are taking into account comparative mobility here; we can say if an entire cluster moves the CHN never change. This is an important characteristic since we do not want to have frequent changes in CHNs.

3.1.2.3 Resources(R)

For available resources, R, we calculate it by taking into the memory size, battery power and CPU resources. We calculate available resources, R_i , for node i, by the formula below:

$$R_i = (\lambda_1 * MEM_i + \lambda_2 * CPR_i) * BP_i \quad (6)$$

The parameters, λ_1 and λ_2 are decided by the particular application, and MEM is the available memory. CPR indicates CPU resources and BP_i represents the amount of battery power left in node i . we calculate the BP by the formula below.

$$BP = \Sigma - (\alpha + \beta + \gamma) \quad (7)$$

Where, Σ , α , β , γ represents initial energy taken by the node, energy consumed in transmitting packets, energy consumed in receiving packets, and energy consumption in idle state respectively.

Nodes have to calculate and store the weight that summarizes their resource capabilities. The [Table 1](#) shows values for nodes within the single cluster.

Table 1. Example values of 7 nodes in the cluster

Node id	R			CTY	MOB	NWV
	MEM	BP	CPR			
N1	52	3	38	11	2	2.080741
N 2	42	2	20	13	2	3.21538
N 4	62	6	41	12	3	3.30324
N 6	47	7	22	10	4	3.20414
N 9	24	4	14	12	1	2.71316
N 11	66	4	56	13	0	2.6041
N 12	53	5	40	6	2	1.8043

The above table shows the value of nodes according to (2) (5) (6). After finding the value of nodes evaluate the Node Weight Value (NWV) of each node. It is determined by substituting these values on (1). At the end, we select the cluster head in our example N12 is a CHN1 it has the optimum value.

When a node enters into the cluster, they send the MEMB messages to all the nodes. After exchanging their MEMB messages the nodes weight is calculated. Compared the weight of CHN and newly joined node, if new node's weight is high compare to the existing head that node assigned as the CHN then it sends the cluster head assignment packet (CHAP) packet to all the nodes within the cluster and new node take over the job of existing CHN node. Otherwise the same CHN will continue the job.

The maintenance of CHN is carried out by checking periodic messages they send. As the CHN weight may vary and goes below a threshold level, it can stop the function as a cluster head and the cluster head selection process is activated, resulting on a new node taking the CHN role. This will lead to reduced network load within the cluster. In case a CHN moves away from the cluster, no further action can be carried out and the new one should gather all

the service information from the neighbouring nodes again. Now CHN is ready to fulfil the discoverers needs.

3.1.3 Awareness of Neighbouring Clusters

After identifying the cluster head the individuality of the cluster head node is propagated in the cluster through the broadcast message *setHead*. The cluster head node establishes the link with the neighbouring cluster with the help of the gateway node (GWN) which is placed in the border of the cluster. The message *setHead* is propagated to the nodes from the adjacent clusters the GWN stores the information and it will be broadcasted the adjacent node CHN information.

3.2 Service Discovery Model

The service discovery protocol employs the clustering algorithm to maintain a service repository of service descriptions. By this method we can reduce the communication cost, because the discovery message is exchanged only among the clusters.

A node selects its nearest CHN using the cluster head selection method as described in section 3.1. Every node that joins in the network and publishing or advertising their services with other nodes and registering their services with CHN as shown in [Fig. 1](#). The SSN (Service Subordinate Node) is either service requestor or service provider to advertise the services or query about the services depends on their needs. CHN maintains the list of all registered services contained with a description which is offered by SPN. CHN shares their service with other CHN whenever it needs using the cluster head selection method. A service requestor node (SRN) willing to utilize the service which is available in the network has to send the SREQ query that includes the service name, type or service code. CHN maintains the service list which contains all the available services in the particular cluster.

If the CHN service list service which is matched with the requested service name and type, it replies a service reply packet (SREP) contains the location (address) of all SPN nodes that offer the requested service. The Service Provider (SPN) is selected based on some criterion by the service requestor (SRN) node. The SPN will receive the service with service description, EXP_T.

In the cluster CHN provide the reply regarding the service details that will pass on through the more number of intermediate nodes. Each member node forwarding the SREP packet and inspect the service description and may choose to store one of the service descriptions according to the following condition:

- If the service is matched with the CHN service list check the EXP_T, is less than the threshold.
- The node must have the sufficient space to store the SREP.
- The service is offered by the SPN is not more than the H hops away from the Service Request node (SRN).

When the SPN registering their services SER1 at a CHN with an EXP_T time. If the service SER1 EXP_T comes to an end at an SPN then the SPN removes the service SER1 from the SPN and forwards the service remove packet (SRMP) to all its neighbor and CHN. Then CHN also removes the SER1 from the service list. The SRMP will help in avoiding the unnecessary traffic to send the SERP from SPN to the SRN in the cluster.

Suppose the service is not in the particular cluster the SREQ query is passed to the next

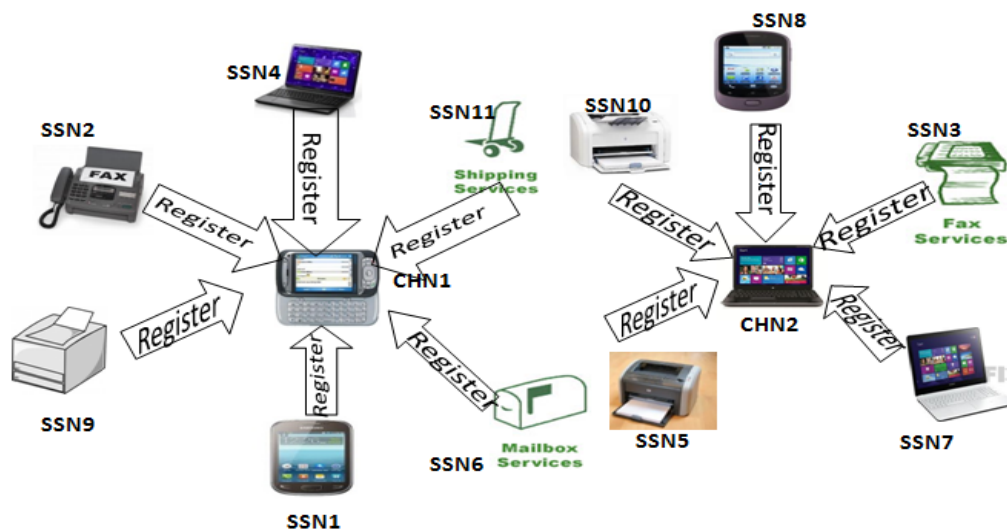


Fig. 1. Service registration in cluster head (service repository)

Nearest cluster head (CHN) with the help of the gateway node (GWN). This gateway node should be nearer to the CHN node or number of hops between the CHN must be less. When all the SPN register their services with the CHN in the cluster, if more than one similar services provided by different SPN means CHN selects the best service according to EXP_T , T_r . T_r denotes the response time, that is time taken to send the request and receiver receives the response. For a long time SRN doesn't receive any response from SPN, SRN sends the same request with change service provider (CSP) packet to CHN to get the service. Hence the CHN maintains the list of service and provide the service that is needed.

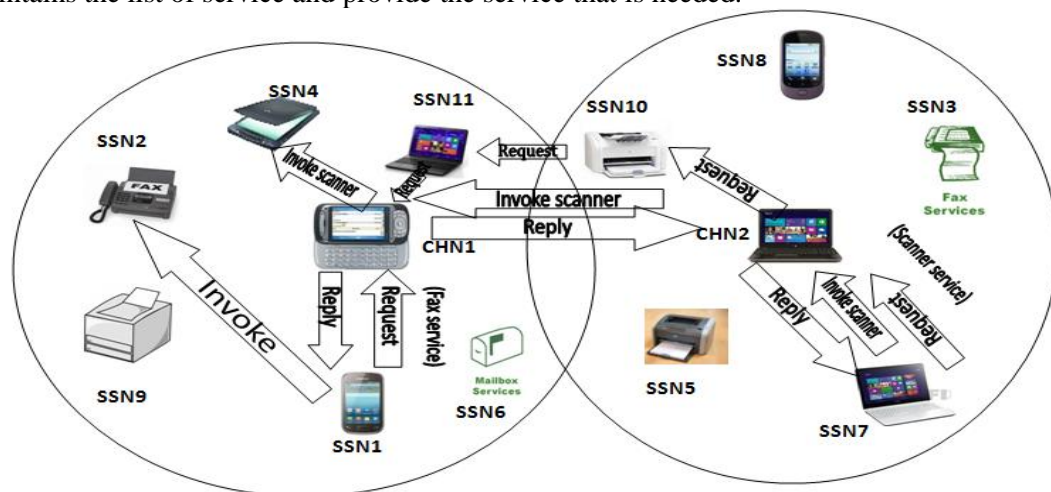


Fig. 2. Example scenario to search a service in two different clusters

Fig. 2 shows the simplified examples of service discovery. The first case SSN1 as a SRN sends the service request (fax service) to CHN1, if it is available it sends the SREP with the

address of SPN (SSN2) back to the SSN1 and SSN1 invoke the fax service within the cluster by SEINVK message and SSN2 as a SPN sends the service response packet (SRSP) to the SSN1. Second case SSN7 (SRN) wants to get the scanner service but it is not available in the request node cluster (right side), then it forwards the service request to the gateway node (GWN) according to nearest and select the less number of hops from CHN (CHN2, SSN10, SSN11, CHN1). The scanner service is available in the CHN1 (left side cluster). Now CHN1 sends the SREP it includes address or location of the service to SSN7 and SSN7 invoke and get the SERSP from N4.

4. Cross Layer Integration with MAC layer for energy efficient

While constructing cluster the *setHead* message is broadcasted, and the CHN identity is disseminated to the entire cluster and to GWN belonging to the neighbor cluster. The integration of control packets with the *setHead* message is processed based on the TDMA based MAC protocol. This will reduce the communication cost and increases the network lifetime. In our CSDM network the nodes are grouped as a cluster, the gateway nodes are identified to communicate with one cluster to another cluster through the wireless links. The gateway nodes such as PDAs, Smartphone which is connected to the GSM network or through the internet. By this method the multiple mobile gateway nodes are integrated. With the help of the GWN, the following communication is possible. (1) Node can communicate with the network by discovering available services and the (2) the network can discover the GWN nodes to share the information.

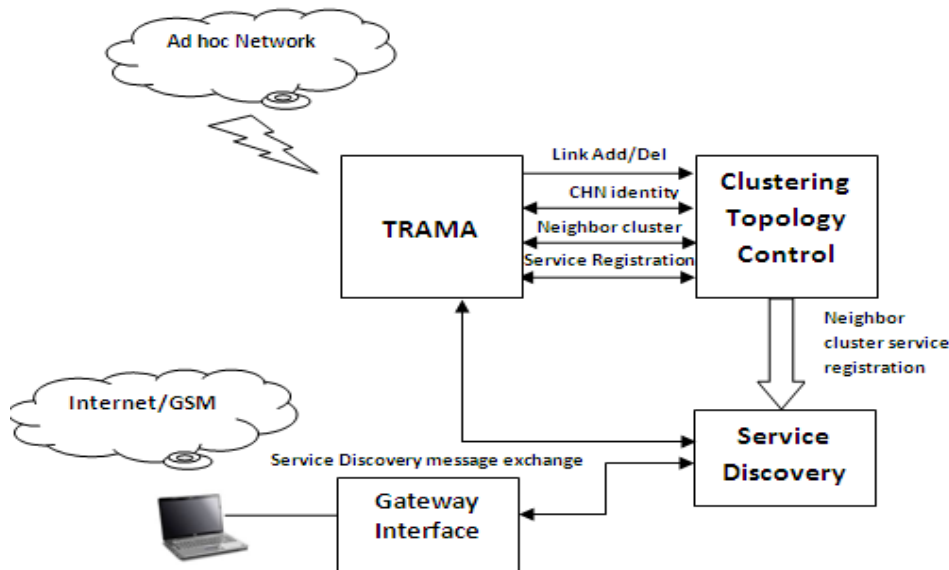


Fig. 3. Service Discovery Architecture

For standardizing the access to the nodes to the wireless, we need to use TDMA based MAC protocol Traffic Adaptive multiple access protocol (TRAMA) [16]. This is used to improve the energy efficiency in collision free channel access in ad hoc network. Each node is familiar with its two-hop vicinity and their present schedules. Each node can use the time slot as per the node identifier 'x' and the globally known hash function 'h'. When the traffic is found in the

network it allocates the timeslots for each node, otherwise it won't allow assigning the timeslots in each node. For time slot t , compute priority $p = h(x, t)$. Every node and its two hop neighbors compute this priority for next 'k' time slots. It provides the high packet delivery guarantee and energy efficiency, without any delay. During its $(k + t)$ timeslots, a node transmits a control packet (any requested service) followed by the data. Upper layer protocols can use the control packet and associated some information for energy efficient cross layer integration. TRAMA maintains a neighbor table by analyzing the control messages received from the adjacent nodes. TRAMA maintains a neighbor table properly. By using this table TRAMA can inform the upper layers about the communication link either is added or deleted in the topology. **Fig. 3** shows the implementation overview of the CSDM model. The clustering topology control is developed and maintained with the use of TRAMA information. Using this cross layer optimization, the clustering topology control module identifies the CHN that is exchanged periodically by TRAMA. By using this TRAMA, control module receives the information from the member node and it is transmitted to the root node. Service Discovery and module receives the information from the TRAMA or from the internet/Bluetooth.

5. Implementation

We implement CSDM model as a proof of concept of our model. **Fig. 4** shows a set of ad hoc network nodes arranged into two clusters and connected to the network via GSM/internet/Bluetooth interfaces. The NWV is evaluated for the nodes in the cluster are determined during the formation of clusters. The network have the two highest capability grades (4 and 10), as they are connected to the wireless link. As a result, they are chosen as cluster heads of clusters 1 and 2. The clustering structure dynamically reorganizes in case of mobility or node addition/removal.

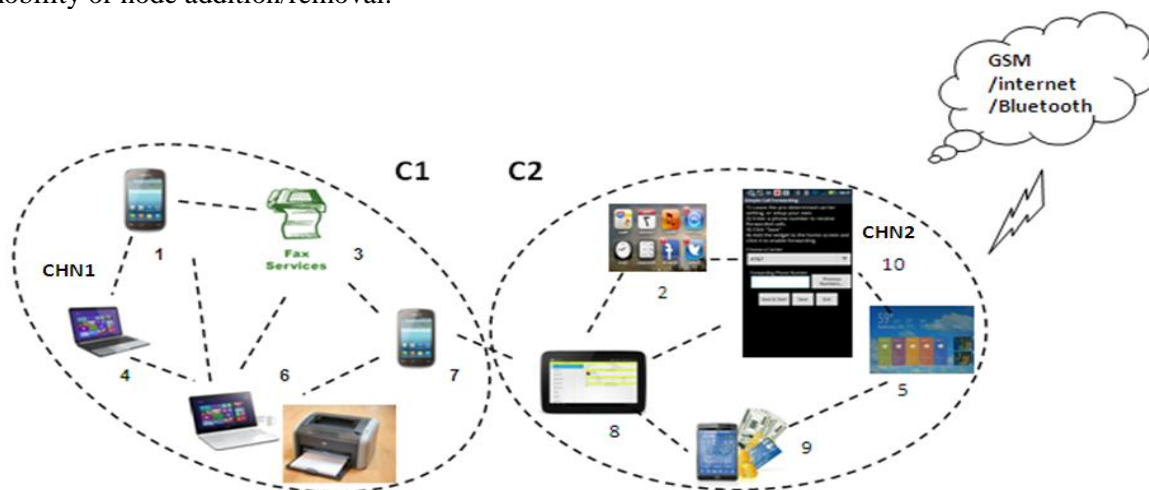


Fig. 4. Implementation Setup

In our CSDM nodes are arranged in the network area. Each node has appropriate services that are to be implemented and by using multicast, neighbor nodes can be detected and updated in



Fig. 5. (a)Node information in neighbor table (b) Service Registration (c) SPN service details list (d) Searched service “currency converter” from node 9.

neighbour table. Each node has weight which is described in section 3.2. Nodes are equipped with the weather report, email, social networking, currency converter services, etc. Initially the CHN election algorithm is implemented by sending the CHN weight packet (CHNWP). CHNWP contains list of node addresses, weight of each nodes. CHNWP values of all the nodes in the network are updated if the node is moved from one cluster to other cluster. Node which contains highest NWV value is selected as cluster head. The **Fig 5 (a)** shows the list of nodes connected to the network. The connected nodes in the cluster is displayed in the neighbor table. **Fig 5(b)** shows the service registration, the list of services in the SPN is registered to the CHN. Available services in the CHN is displayed in the **fig 5(c)**. The SRN can select the desired service (currency converter) from a list and launch the search command which initiate the service discovery. **Fig 5(d)** shows the currency converter service which is retrieved from the node 9. The location of SPN (node 9) is sent by the CHN then the SRN can use that service.

6. Analysis

In this section, some mathematical analysis of our work has been carried out to analyse how some of the system parameters affects the performance. As said before, when SPN register its service SER1 at its nearest CHN, for example CHN1, service details of SER1 is sent to CHN1, that may hold or it may not because of the EXP_T of that service or similar type of services already exist, it takes any one of certain service among all the similar services. Hence, when SRN sends the SREQ to CHN, it gets the SREP from CHN. It will be explained later that as the number head nodes (CHNs) increases, responses to receive for the request is reduced that means delay is reduced and the unwanted responses reduced by the RPM so traffic is also reduced. Furthermore, the entire network is divided into clusters so the load among the network is also reduced during service discovery.

We now take some parameters to the computation of service discovery. The proofs have been carried out to prove that our CSDM model gives the stable cluster to discover services easily and also analysis the cluster overhead during formation.

6.1 Stability Analysis

Our model is stable when it adapts itself during topology changes. The frequent changes of cluster head leads to route invalidation and need re clustering. Our CSDM model reduces the frequent changes because of finding weights. Life time of the cluster head increases the stability nature of the cluster head is also increased. The stable clustered structure is proven by the following properties.

- Expected number of hops between two nodes*
- Uniqueness of two cluster head nodes (CHNs) can be closest.*
- Expected number of distance to the nearest CHN.*

6.1.1 Expected number of hops between two nodes

Number of hops can be defined as the number of intermediate nodes in the route (n_1 to n_2). The main assumption is that each hop results in the same progress towards the destination, equal to the average distance covered by a node in one hop. Number of hops should be as low as possible. It will reduce the probability of link breakage and improve the path duration between nodes [17]. In our CSDM, Nodes are deployed in the coverage area $m \times n$. A link between the two nodes can be formed while the distance d between the n_1 and the next hop in the randomly deployed nodes in the area is $\leq r_0$ (transmission range). If n_2 node present in the n_1 's transmission range then the probability of finding destination node is same as the probability of finding next-hop node. We derive the equation to find the probability density function of distance d between two nodes in the coverage C is $P_D(d)$ which is used to compute the expected hop count [18].

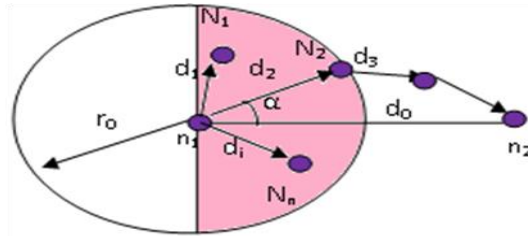
$$f(d_1) = 2\pi m n d_1 \cdot e^{-\pi m n d_1^2} \quad (8)$$

The probability of 1 -hop can be computed as follows:

$$P_1(d) = \int_0^{r_0} f(d_1) dd = 1 - e^{-\pi m n r_0^2} \quad (9)$$

Suppose that there are n forwarding neighbour nodes ($N_1 \dots N_n$) distributed over the shadowed circle in the direction of the destination n_2 . The angles from the n_1 to n_2 is α and distances from one node to other node is d_i where $i=1 \dots n$. We concluded that, if two nodes (n_1 to n_2) are at distant d_0 from each other, the number of hops between them is a sufficient number of nodes that would tend toward d_0/r_0 . We also say that $P_D^*(d)$ substituted by the coverage C or any two

nodes n_1, n_2 .



However, the n_2 node can be far away from the n_1 source node that may be two, three or more hop count. If the n_2 is out of transmission range, r_0 and $\geq 2r_0$, then as a minimum one intermediate node is required between n_1 and n_2 to transmit the packet further in the network. The probability of a two-hop count can be calculated as follows:

$$P_2(d) = \int_0^{2r_0} 2\pi mnd_1 \cdot e^{-\pi mnd_1^2} dd \times \left[1 - e^{-\frac{mn}{2} \cdot C}\right]$$

$$= [e^{-\pi mn r_0^2} - e^{-4\pi mn r_0^2}] \times \left[1 - e^{-\frac{mn}{2} \cdot C}\right] \quad (10)$$

$$\text{Where, } C = 2r_0^2 \left[\frac{\pi-2}{4}\right] \left[\alpha - \frac{\sin(2\alpha)}{2}\right]$$

Likewise we can compute the h hop counts,

$$P_{Hn}(h) = [e^{-(h-1)^2 \pi mn r_0^2} - e^{-h^2 \pi mn r_0^2}] \times \left[1 - e^{-\frac{mn}{2} \cdot C}\right]^{h-1} \quad (11)$$

Hence the expected number of hops between two nodes in the network is $P_H^*(d)$ is derived by using the above equations (8) (9) (10) (11), which is the probability density function of the distance d when the hop h is said to be H^* . This is derived by

$$E[H] = P_H^*(d) = \sum_{H=1}^h H [e^{-(H-1)^2 \pi mn r_0^2} - e^{-H^2 \pi mn r_0^2}] \times \left[1 - e^{-\frac{mn}{2} \cdot C}\right]^{H-1} \quad (12)$$

$$P_D(d) = \frac{1}{\pi m^2 n^2} \int_0^d \left[4m^2 n^2 \left(\frac{1}{2mn}\right) - d\sqrt{4m^2 n^2 - l^2}\right] l dl \quad (13)$$

$$\text{Hence, } f_D(d) = \frac{d}{\pi m^2 n^2} \left(4mn \frac{\pi d}{2} mn - d\sqrt{4mn - d^2}\right) \quad (14)$$

Following equation 15 represents that dividing $E(n_1)$, the expected distance of n_1 by r_0 . That is number of hops inevitably increase due to having to route through longer distance to reach a particular node. The expected number of hops is

$$E[H] = 0.752n_1/r_0 \quad (15)$$

Let we take a 500 m \times 500 m area and a value of 100 m for transmission range r_0 . The average number of hops between two randomly chosen nodes is 3.75 which are yielded from Eq.13.

6.1.2 Uniqueness of two cluster head nodes (CHNs) can be closest.

In this section we examine two cluster head nodes (CHNs) cannot be neighboring. Consider the cluster formation of clusters as shown in Fig. 6, where CHN1 receives the service request from CHN2 that resides with its r_0 . Source CHN2 forwards the packet using directional flooding with an angle of α via its gateway node. Now the packet hops from one cluster to another cluster by keeping closer to the axis of imaginary line between node CHN1 and CHN2. Now set the timer to check whether the node CHN2 resides after the time expired. Using stochastic, then compute the position of the CHN1 and CHN2.

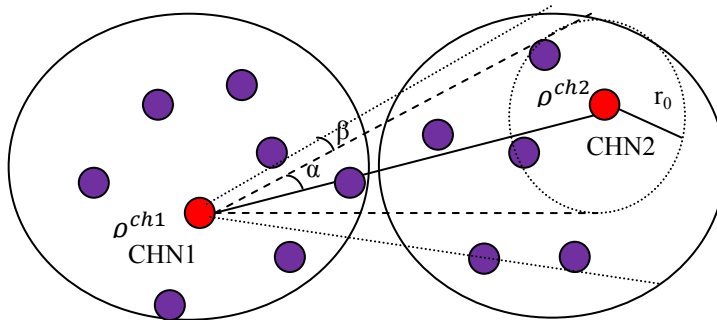


Fig. 6. Relationship between two cluster heads

$$CHN2(CHN1, \rho^{ch1}) = \begin{cases} y \in \rho^{ch1} \{ \|\rho^{ch1} - \rho^{ch2}\| \}, & \text{if exist and unique} \\ \infty, & \text{otherwise} \end{cases} \quad (16)$$

Where, ρ^{ch1}, ρ^{ch2} is the position of the CHN1 and CHN2 respectively, the receiver CHN2

of the CHN1 at ρ^{ch1} represents the $CHN2(CHN1, \rho^{ch1})$.

6.1.3 Expected number of distance to the closest CHN.

In this section we analysis the average distance needed to reach the nearest CHN. Already we found the number of hops between two nodes E [H]. The distance d is higher than the transmission range by using equation 13, then the probability of the distance r is, $P_{d>r}(r) = \int_r^\infty P_d(d)dd$. Let we take the random variables $Y = \{y_1, y_2, \dots, y_{T_{CHN}}\}$ to represent the T_{CHN} . The probability density function of selecting the minimum of Y is,

$$\begin{aligned} P_{\min(Y)}(r) &= \sum_{i=1}^{T_{CHN}} P_Y(y_i = r | y_j > r, \forall j \neq i) \\ &= \sum_{i=1}^{T_{CHN}} \left(P_{y_i}(y_i = r) \prod_{j=1, j \neq i}^{T_{CHN}} P_{y_j} > r(r) \right) \\ &= \sum_{i=1}^{T_{CHN}} \left(P_{y_i}(y_i = r) (P_{y>r}(r)) \right)^{T_{CHN}-1} \\ &= T_{CHN} P_{y_i}(y_i = r) (P_{y>r}(r))^{T_{CHN}-1} \end{aligned} \quad (17)$$

Hence the expected number of distances to closest cluster head CHN is divided by transmission range r_0

$$E[D_{CHNclose}] = \frac{T_{CHN}}{r_0} \int_0^\infty r P_Y(r) \left(\int_0^\infty P_Y(r) dr \right)^{T_{CHN}-1} dr \quad (18)$$

6.2 Control Over head

The control overhead of our model has been analysed mathematically for Beacon messages transmitted by the nodes and cluster formation and maintenance phase.

6.2.1 Cluster formation overhead

Let the SSN sends request messages to neighbouring nodes (n-1) to provide the service information in the network. On receiving the request message from cluster head, the n-1 SSN nodes respond back with their service information. Therefore, 2(n-1) messages are required to obtain node's parameters of all the nodes in the network. The SSN selects N number of cluster heads and sends *head-selection* messages to all other CHNs in the network. Each cluster head broadcast *head-selection* messages to its neighbours. Let m_i be the neighbours of i cluster head in the network. Thus the cluster formation overhead of our CSDM, C_f has been derived mathematically as

$$C_f = \sum_{i=1}^N m_i + [N + 2(n - 1)] \quad (19)$$

6.3 Network traffic and bandwidth utilization

The total network traffic for service discovery is created by control packets and data packets. The control packets are shown in [Table 2](#). The total network load is calculated by the sum of all packets which are used in the network.

Table 2. Types of packets used in CSDM

Packet	Meaning	Operation
SREQ	Service Request Packet	SRN request a service
SREP	Service Reply Packet	CHN sends the reply to SRN
SERP	Service Response Packet	SPN sends the requested service to SRN

RMP	Remove the service Packet	SPN removes the service when it is expired
CSP	Change service provider packet	SRN request to change the SPN.
SREG	Service Registration packet	SRN registering their services
SEINVK	Service invocation packet	After getting reply from CHN, SRN invoke the service.
NAK	Negative Acknowledgement packet	SRN sends the response as NAK to SRN
WP	Weight packet	Advertising their weights to its entire neighbor in a cluster.
CHAP	Cluster head Assignment Packet	When new node w is high then advertising the change of cluster head.
MEMB	Member Beacon message packet	Member nodes sends their alive message to its neighbor
CHNWP	Cluster Head node weight packet value	Contains list of addresses and weight of each node

The control packets load generated by a node is

$$M = \sum_{i=1}^n S_i \quad \forall S = \{SRQ, SRP, SER, CSP, SINVK\} \quad (20)$$

The total number of service messages generated in a single bounded advertisement from a single node is n . Thus, the total number of messages generated by advertisements in time T by all nodes is $D \times n \times T$. Control Packet by n nodes (M) $CP = M \times n \times T$. Data packets n by nodes (D) $DP = D \times n \times T$.

Total traffic (TT) = $CP + DP$

$$\text{Control overhead } COH = \frac{CP}{DP} \quad (21)$$

If control packets are increased, the total network traffic increases.

6.4 Success hit ratio

Since each cluster head has together about nearly 10 nodes information, there is a possibility to find the service in the local CHN. The SRN can measure the success hit ratio to obtain the service is $SHR = \frac{N_{RS}}{N_{SS}}$ (22)

Where N_{RS} is the number of requested services and N_{SS} represents the number of services successfully found from the service list of CHN.

6.5 Mobility vs. Packet delivery ratio(PDR)

When the nodes are moving within the cluster, the topology does not affect route discovery and service discovery. When the nodes move across the clusters, route rediscovery is done automatically. The throughput of the system is not affected by low mobility. The average number of data packet received (DPR) by destination nodes divided by the number of data packet that received (DPSR) is called data PDR. The PDR for any session is calculated as follows: $PDR = \frac{DPR}{DPSR}$ (23)

7. Experimental Results

We implement CSDM and we extent the performance by deploying a sequence of experiment of up to 50 nodes. We analyse the number of messages exchanged until network convergence,

and the number of resulting clusters, the cost of service discovery and the discovery time. In order to estimate, we generate a sequence of random topologies in an ad hoc network. The nodes are considered to be placed on a dynamic environment. The number of nodes from 5 to 50 and thus the network density (e.g. average node degree) increases from 1.4 to 7. For each network density in turn we run 20 experiments, each consisting of the following steps:

- 1) Deploy the nodes in the network.
- 2) Calculate the number of SREQ messages exchanged by the SRN to cluster in the network topology.
- 3) Identify the number of clusters finalized.
- 4) Issue the SREP from the CHN in the network.
- 5) Compute the number of SREP messages exchanged and the time until the service found.
- 6) Count the number of SERINV messages to obtain the service.

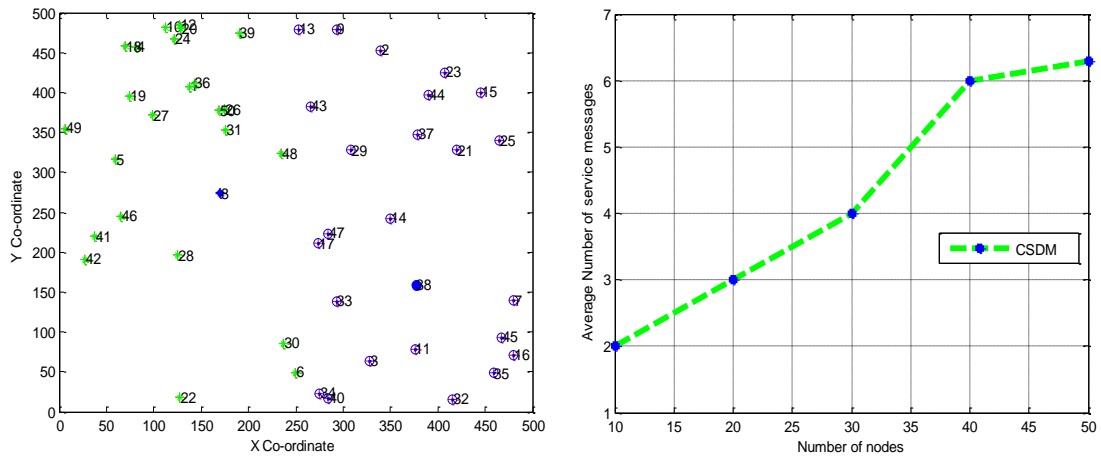


Fig. 7. (a)number of clusters (b) Average number of service messages

Fig. 7 (a) shows the number of clusters in the network with 50 nodes. The clustering algorithm performance is identified by the number clusters available in the network. A number of clusters lead to a large number of repetitions that arise during the discovery process. The nodes are distributed in the area (500m x 500m) and form clusters with only one CHN. In our CSDM only 2 clusters are obtained with 2 CHNs (node 8 and 38). When the network size increases, the new node enter into the already existing cluster or they form their own cluster and compel the CHNs in the neighborhood to join. We see the general trend that adding new nodes to the network does not significantly change the number of clusters, and accordingly, the number of cluster head nodes. The proposed work results confirm the network is efficient one with minimum number of clusters that can decrease the routing cost of the network.

Fig. 7 (b) shows the average number of service messages sent and received per node for updating the information on neighboring clusters and the service registrations. The messages are calculated starting from a newly initialized network until the network convergence. In this figure initailly we see that, the number of messages per node raises linearly with the network size. The reason is that when increasing the number of nodes within the cluster, the network becomes more linked and thus, the CHNs receive more information about the clusters in vicinity. At last of the graph, the number of messages per node in the cluster remains comparatively stable. For dense networks, it is likely that the information received from a new neighbour does not change the already obtained information of neighboring clusters, so it is

not propagated further in the cluster.

Packet Delivery ratio is the ratio between amounts of packets transmitted and amount of packets received. This factor finds the fraction of successful packet reception. The packet delivery ratio in CSDM is depicted in Fig. 8(a), it becomes greater than in DSDM [12] in dynamic environment. Increasing the number of nodes decreases the packet delivery ratio due to stable clustered structure which is proven in section 6.1. In our case, the packet delivery ratio is high by sustaining a stable route by thinking residual energy and transmission power taken from PHY and MAC layers. It decreases the rate of packet loss owing to link break by choosing a different path according to residual energy and transmission power.

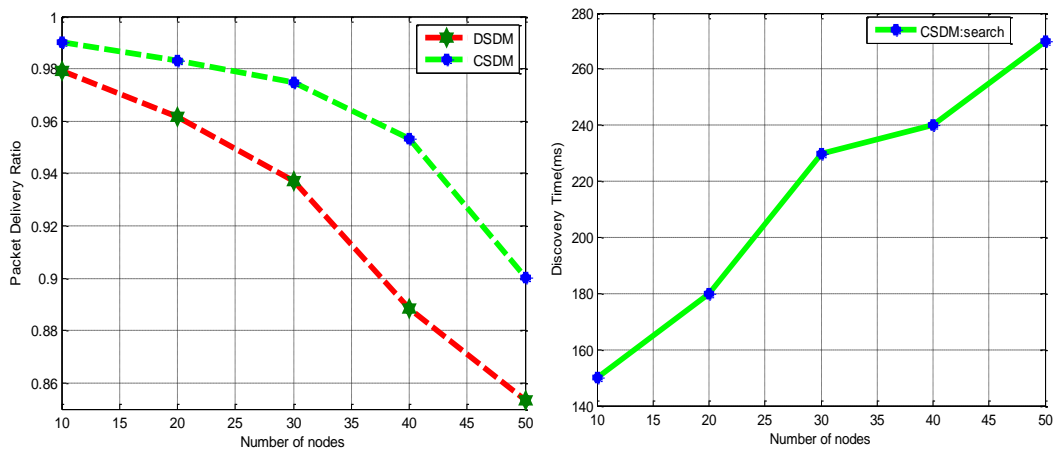


Fig. 8. (a)Packet Delivery Ratio (b) Average number of SREQ message

In the first run, the discovery time is measured, and the effect of increasing the number of nodes on discovery time is examined. The discovery time is the total time needed for finding all the services offered by all the nodes by sending CSDM: search messages to all SPN nodes and receiving the replies from all of them. Fig. 8 (b) depicts the results taken from this test.

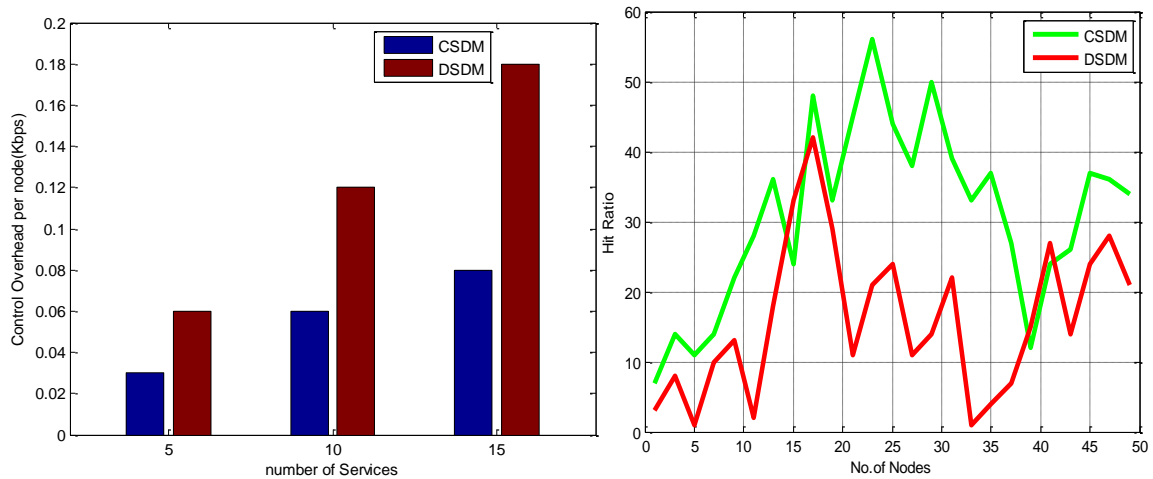


Fig. 9. (a)Control overhead(b) Success hit ratio vs number of nodes

Fig. 9(a) shows that the average sending rate of control packets and where the total traffic

load per node was reduced. We also measured the average number of control packets sent per second at each node. **Fig. 9(b)** shows the success hit ratio comparison of our proposed CSDM with DSDM. To estimate the success hit ratio i.e. the number of service discovery requests locally fulfilled in comparison with the requests offered by a node. In our experiment, 50 nodes with 10 services with the simulation time of 5000 seconds. The results show that the CSDM performs very well. The reason for this improvement is that the nodes are predicting future requests and attaching the responses of these requests with current response. Some of these piggy-backed are then actually used by requesting nodes in future. This gives rise to efficient in success hit ratio. In some of the cases, the result however degrades. This might be due to the fact that the future prediction of service requests is wrong.

We summarize the most significant comments that validate from a practical point of view:

- When network size is high, increasing the number of nodes in the network does not further increase the number of clusters.
- The average message overhead per node for cluster setup and maintenance is limited, since there is no need to transmit the information on neighbour clusters that is previously identified in the higher levels of the cluster.
- The discovery cost per node decreases with the network density, as the discovery messages travel only among cluster head nodes.

8. Conclusion

In this paper, we have proposed a cluster based service discovery model for discovering services, and sharing the services in network of cluster head nodes and using them as a service repository and finally some results taken from evaluations made on the performance of the system. The main feature is to avoid the unnecessary packet forwarding to avoid the traffic during service discovery. Further, the results taken from simulations showed that to minimize the computation cost during the discovery phase and it is very flexible, efficient and can be adopted to work in MANET environment. It can save the computing power and success hit ratio is increased while discovering the services. Both analytical and simulation results showed the performance in terms of stability, control overhead, traffic and bandwidth utilization. An overhead to the service discovery process is reduced, it directly used by the node for finding the most suitable services in the cluster. Then again, the results taken from CSDM proved that the discovery time for all the existing services increase when the number of nodes exceeds. In addition to that our model has been analysed with some system parameters in terms of delay and packet delivery ratio.

9. References

- [1] Katsigiannis CO, KaterosDA, Koutsoloukas EA, Tselikas ND, Venieris IS, "Architecture for reliable service discovery and delivery in manets based on power management employing slp extensions," *IEEE Wireless Communication*, Vol. 13 pp. 90 – 95, 2006. [Article \(CrossRef Link\)](#)
- [2] Mallah RA, Quintero A, "A light weight service discovery protocol for ad hoc networks," *Journal of Computer Science*, Vol 5, pp. 330 – 337, 2009. [Article \(CrossRef Link\)](#).
- [3] Dhurandher SK, Singh GV, "Weight based adaptive clustering in wireless ad hoc networks," in *Proc. of IEEE Personal Wireless Communication Conference*, New Delhi, India, pp.95–100, 23-25 January, 2005.
- [4] Basagni S, "Distributed clustering for ad hoc networks' IEEE International Symposium on Parallel Architectures," in *Proc. of Algorithms and Networks Conference*, Fermentle, Australia,

- pp.310–315, June 1999.
- [5] Dhurandher S K, Singh G V, “Power aware clustering technique in wireless ad hoc networks,” in *Proc. of International Symposium on Ad Hoc and Ubiquitous Computing*, NITK, Surathkal, India, ISAUHC, pp.75–80, December 2006.
 - [6] Bokar A, Bozyigit M, Sener C, “Scalable energy aware dynamic task allocation,” in *Proc. of IEEE 2009 International Conference on Advanced Information Networking and Applications Workshops*, Bradford, United Kingdom, IEEE, pp.371–376, May 2009. [Article \(CrossRef Link\)](#)
 - [7] Jiangang MA, Zhang Y, Jing H., “Efficiently finding web services using a clustering semantic approach,” in *Proc. of International workshop on Context enabled source and service selection, integration and adaptation*, Beijing, China, pp. 5–5, April 2008.
 - [8] Funk A, Bontcheva K, “Ontology based categorization of web services with machine learning,” in *Proc. of Language Resources and Evaluation Conference*, Valletta, Malta. pp. 482–488, 2010.
 - [9] Jamous Mamoun Mohamed, Deris SB, “Web services non-functional classification to enhance discovery speed,” *International Journal of Computer Science Issues*, Vol 8, pp. 150 – 155, 2011.
 - [10] Chakraborty D, Joshi A, Yelena Y, Finin T., “A novel group based service discovery protocol for manets,” in *Proc. of IEEE Mobile and Wireless Communications Networks Conference*, Stockholm, Sweden, pp.140 – 144, 2002.
 - [11] Hassan Artail, Khaleel Wafiq Mershad, and Hicham Hamze, “DSDM: A Distributed Service Discovery Model for Manets,” *IEEE Transactions on Parallel and Distributed Systems*, VOL. 19, NO. 9, pp 1224-1236, 2008.
 - [12] Basu P, Khan P, Little TDC, “A mobility based metric for clustering in mobile ad hoc networks,” in *Proc. of IEEE Workshop on Distributed Computing Systems Workshops*, Mesa, pp. 413–418, 2001.
 - [13] Buvana M, Suganthi M, “Novel architecture for cluster based service discovery and load balancing in mobile ad hoc network,” in *Proc. of IEEE International Conference on Advanced Computing*, pp. 292–297, 2011.
 - [14] Hammouda KM, Kamel MS, “efficient phrase-based document indexing for web document clustering,” *IEEE Transaction Knowledge Data Engineering*, Vol16, pp. 1279–1296, 2004. [Article \(CrossRef Link\)](#).
 - [15] Sajjanhar A, Hou J, Zhang Y, “Algorithm for web services matching,” *Advanced Web Technologies and Applications. Lecture notes in computer science*, Springer press, pp. 665– 670, 2004. [Article \(CrossRef Link\)](#).
 - [16] V. Rajendran, K. Obraczka, J. J. Garcia Luna Aceves, “Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks,” in *Proc. of the ACM SenSys’03*. Los Angeles, USA, pp. 126-137, 2003.
 - [17] Y. Han, R. J. La, and A. M. Makowski, “Distribution of path durations in mobile ad hoc networks – Palm’s Theorem at work,” *ITC Specialist Seminar on performance Evaluation of Wireless and Mobile Systems (ITCSS)*, 2004.
 - [18] S. M. Harb, and J. McNair, “Analytical Study of the Expected Number of Hops in Wireless Ad Hoc Network,” *LNCS 5258*, pp. 63–71, 2008.



M.Buvana has over 10 years of teaching experience and is currently working as a Assistant Prof in PSNA College of Engineering and Technology. She finished her bachelor of Engineering (computer science) at PSNA College of engineering and Technology, Dindigul in the year 2004. And She completed her M.E in PSNA College of engineering and Technology in 2008. She published two papers in International journal and she has published papers in the 7 national and 5 International and IEEE conferences such as ICoAC 2011, ICICCA 2013, etc.. She is pursuing PhD degree in the field of Mobile communication at Anna University Chennai, India.



M.Suganthi is currently working as Professor of Department of Electronics and Communication Engineering in Thiyagaraja College of Engineering, Madurai. She graduated her B.E. in Electronics and Communication Engineering from Thiyagaraja College of Engireeing, Madurai, M.E. in Communication Systems from P.S.G College of Technology, Coimbatore, and Ph.D in Mobile Communication from Madurai Kamaraj University, Madurai. She has been an educator in the technical teaching society for the past 25 years. She has published various papers about the wireless communication in different conferences and journal and she has guided 8 research scholars.