CHAPTER 4

CLUSTER BASED SERVICE DISCOVERY MODEL USING
LINEAR DISCRIMINANT ANALYSIS

A Mobile Ad Hoc Network (MANET) is a self-designing system that is revealed as infrastructure less, mobile, self-sufficient, multi-hopped, self-organized and self-controlled, by having a dynamic topology and unpredictable traffic designs and more. Several examinations have been carried out to settle different issues of MANET. These issues incorporate routing, medium access layer issues, multicasting, energy management, transport layer convention, Quality of Service, billing, deployment consideration, addressing, service discovery, data management and security and so on (Sesay et al. 2004; Xiao et al. 2000; Royer & Toh 1999). One of the most vital inquiries in MANET is the discovery of services accessible around the region of any node. A service can be any hardware, software or any other substance that a client may be interested to use and service discovery is the procedure for finding the services based on the requester/user desires. An effective and adaptable way to deal with service discovery can prompt in improving countless applications. For example, in a vehicular ad hoc network, vehicles may be keen on knowing the services gives by an adjacent fuel station. Correspondingly, in a war zone, warriors may be occupied with sharing the circumstance about the whole battlefield. Because of the dynamic way of MANE, there are constantly spatial and transient varieties in the accessibility of services (Filip Perich et al. 2001). Subsequently a service discovery methodology ought to be highly robust, effective and dynamic in nature.
Cluster based solutions ease the requesters needs and the discovery process. Thus, the cluster based service discovery has various benefits (Marin et al. 2006).

- The knowledge can be distributed among the members of the clusters depending on the hierarchical level.
- It would allow for low maintenance overhead and low discovery cost even in highly dense ad hoc networks.

The objective of this chapter is cluster based service discovery model to perform service discovery and service selection. The OC-AHPPSO algorithm is extended to enhance the system performance in terms of success hit ratio and improve the response time and reduce the maintenance overhead. Several service discovery approaches are based on the principle component analysis (PCA) (Joliffe 2002, Batra & Seema Bawa (2011)) and semantic based methods proposed in the literature. From the literature the existing method support only the static data which is stored already prior to design and development of the model, and does not know about the dynamic service information. This drawback is handled, by employing the linear discriminant analysis method. Due to the dynamic nature of the mobile node any node can enter or leave the network. At that time, new services can be introduced, LDA avoids in recalculating the entire service classes. The proposed CSDM-LDA method efficiently can be used to update a classifier while new service entered.

The main contribution of this chapter is therefore, CSDM (Cluster based Service Discovery), a clustering algorithm that builds a distributed directory of service registrations. By making decisions based only on the neighbourhood information, the clustering algorithm reacts rapidly to topological changes.
Artail et al. (2008) describes the distributed service discovery architecture. This distributed service discovery model (DSDM), where service directories act as a virtual backbone of locating and registering the services is proposed. The mediator nodes between the provider and requestor cache all the information. DSDM does not consider higher mobility. Due to mobility, the service has a higher chance of moving out the route breaking between two nodes for the service reply to propagate back. Its packet overhead is high due to its redundant packet transmissions. In light-weight service discovery, the protocol for adhoc networks (Malla et al. 2009), selects virtual backbone nodes based on stability constraints such as residual energy, average node speed, the total number of neighbours, bandwidth usage, a period of transmitter range. This protocol does not consider the partitioning method in the network. Only stable nodes maintain the list of services available in the network. Only 80% of the message signalling was observed in the network. Nodes are not balanced while distributing their services. The 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. CSDM-LDA shows the realistic and effective solution by CSDM implementation on the real network.

The two important necessities for a service discovery system, obtained from the application scenarios of mobile ad hoc networks are as follows (Lenders et al. 2005):

a. Optimal Service Selection: If multiple service providers provide the same service in a large system, the performance is significantly enhanced by “efficient” service selection.

b. Robustness faced to mobility: The nodes are open to join or leave any instance, i.e. the network by nature is dynamic. When there are recurrent changes in the network topology, the system performance should continue to be steady.
This chapter describes the Cluster based Service Discovery Model using LDA (CSDM-LDA), by adopting the distributed clustering protocol, service description in the form of WSDL and the LDA and Functional comparison (FC) match making algorithm. In the distributed directory based architecture, the cluster head nodes are considered as service repository nodes which are selected based on the OC-AHPPSO, as discussed in chapter 3. The LDA method is used to classify the services for representing dynamic services in cluster head. Matchmaking algorithm is integrated with LDA method. These combined methods enhance the effectiveness and flexibility of the service discovery process.

4.1 SCOPE AND OBJECTIVE

The experimental techniques adopted in the present work for the OC-AHPPSO of an optimized cluster based service discovery model by using Linear Discriminant Analysis (LDA) to increase the success hit ratio, and minimize the delay and control overhead due to mobility are presented in this chapter. The Linear Discriminant Analysis, the basic theories involved with special reference to the behaviour of various QoS parameters for each service have also been presented. Further, service matching technique is also outlined.

4.2 CSDM-LDA ARCHITECTURE

Service discovery in MANET entails a decentralized design approach that better handles with its active locations (Tyan & Mahmoud 2005). The proposed model is called Cluster based Service Discovery Model by using Linear Discriminant Analysis (CSDM-LDA). Service Discovery Bokar et al. (2009) is a method of allowing nodes to publish their services, enquire about services provided by other nodes, select the exactly matched services and invoke the services. In MANET, node mobility affects service availability, and energy is more consumed during service discovery. To decrease those problems, the clustering method is employed. In the proposed scheme (depicted
in Figure 4.1), various components of the functional models are used such as Service Requestor Node (SRN), Service Provider Node (SPN), Service Repository or Cluster Head Node (CHN). DA has the connection between the service requestor and the service repository to select the appropriate services. Service repository is responsible for the local service registration / deregistration and it collects all the service information such as service name, id, service description and QoS values by service advertisement of SPN to the repository. Repository updates the service information with QoS values whenever a new service comes. Orchestrator is used to correct atomic services into several services. Number of services can be provided to SRN with the use of orchestrator. The SPN is responsible for advertising the service information and operations in the correct format to the DA and vice versa.

Figure 4.1 Service Discovery Architecture
SPN can periodically distribute its information with member beacon (MEMB) to the service repository in the cluster. SRN sends any request (SREQ) to the discovery agent, and then the agent discovers the services which will satisfy the SRNs functional requirement. Discovery Agent deals with a finding technique with QoS range cited by which the SRN could fix QoS constraints. Hence, the searched services would meet the SRN’s QoS constraints. Then the discovery agent does extraction and pre-processing, and it will be passed to LDA-based service selection component to analyze the QoS of every requested service based on the QoS information and the service information on the service. Finally, it will select the optimal service class. By providing the analytic method, it reduces the packet size of the services without any loss of information, if it is repeated. To avoid the packet overhead, the packet transmission can be reduced while sending duplicates services from the same provider. Scalability is the main issue in MANET. The CSDM-LDA handles the scalability. In CSDM-LDA once the services are advertised by the SPNs and the Service Advertisement (SADV) messages are stored into the service repository, the CHN receives the services with a service type, but it is not already present. Later it stores in the service table with its descriptions and it forwards to the adjacent nodes. Suppose if the same types of services are forward by the different SPNs, then the CHN updates it’s already presented services and then forwards as a single SADV message. This single SADV can reduce the overhead. In CSDM-LDA, the services are grouped depending on the classes. So service discovery performance is not degraded, and can optimize the service discovery with high accuracy.

Then the parameter is matched with matchmaker in order to check whether the requested services are related to their repository if it is found to be related to the service requester. QoS monitor evaluates QoS values of service while execution. After getting the result from the agent, the request process is over. Now, the SRN can access the service and it can be evaluated. Reporter
evaluates the feedback and it is sent to the service repository for further processing.

4.3 CSDM-LDA

This section presents a general overview of the CSDM-LDA approach and describes about the service registration, service request propagation in to the cluster and the LDA based service selection during service discovery process.

4.3.1 Overview

The general idea of the proposed service discovery is as follows:

- First, Cluster Head Node (CHN) is elected from OC-AHPPSO. The head node is invoked when the new node joins the cluster or is moved from the cluster or a cluster head retires. Initially, each node broadcasts the message to notify its presence to the neighbour nodes in the cluster. Identify the cluster head node which manages the services on a cluster. setHead message is broadcast, and the CHN identity is disseminated to the entire cluster.

- The Service Provider Node (SPN) is responsible for publishing or advertising the service information and operations in the correct format to the CHN as shown in Figure 4.2. Service Advertisements are explained in section 4.4.2.

- CHN maintains the list of all registered services contained with a description which is offered by SPN and it also maintains the DA which plays a major role in discovering optimal services. The Registered services would be extracted and pre-processed and send to the LDA handler to perform the service discovery process.

- SRN sends any Service REQuest (SREQ) query to the discovery agent in a CHN, and then the agent discovers the services which will satisfy the SRNs functional requirement.
• Then the discovery agent receives SREQ and verifies the availability of services in LDA handler in the cluster to analyze the QoS of every requested service based on QoS information and the service information of service. Finally, it will select the optimal service class. If the requested service is available, then apply match making algorithm to retrieve the optimal service.

• If the requested service is not available in local cluster, SREQ is passed on to the neighbouring cluster to discover the services.

4.3.2 Service Registration

Service provider service instances should be registered with and deregistered from the service registry (CHN). During service registration, each node in the cluster is registering its services with its service description to the CHN. The registration process requires unicast messages to be transmitted from CHN to member nodes. It can be integrated with the transfer of knowledge on adjacent clusters. Thus, the communication cost is improved by using the same message Update Info for both service registrations and is transferred to the knowledge on adjacent clusters.

The distributed directory maintains consistency when the topology changes. In case of a cluster head reselection, an SPN registers the services from its cluster with the new CHN, and notifies the old CHN (if it is still reachable) to remove the worthless service information. The process is transparent to the other nodes in the cluster. If overall service information at old CHN and new CHN change due to cluster head reselection, the modifications are propagated in the clustering hierarchy.

Service provider advertises its services with the service id, service description, service name, documentation within its lifetime. When expiry time (EXP_T) of a service reaches the service information, it is removed from the service registry by the CHN.
/*****Service Registration/Advertisement message******/
SADV
{
Initialize ser_pubmsg;
ser_pubmsg[ser_id]=Get localinformation[ser_tab];
ser_pubmsg[ser_name]=Get localinformation[ser_tab];
ser_pubmsg[serdescpn]=Get localinformation[ser_tab];
ser_pubmsg[serdocum]=Get localinformation[ser_tab];
}
Publish SADV to service repository;

When the SPN registers its 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 neighbour and CHN. Then CHN also removes the SER1 from the service list. The SRMP will help in avoiding the unnecessary traffic to send the SRSP from SPN to the SRN in the cluster.

Figure 4.2 Service registrations in cluster head (service repository)
Figure 4.2 shows the service registration among the service directory nodes (CHN). The SSN is either service provider (SPN) or Service Requester (SRN). The SPN registers its service information with the CHN. Each CHN maintains the service lists and the DA. Each CHN is associated with the DA. The DA extracts the service and the extracted service is send to the LDA handler. Extraction and LDA-matching is described in the next section.

### 4.3.3 Service Request Propagation Scheme

A Service Requestor is a service discoverer. That service requestor node (SRN) is willing to utilize the service which is available in the network to send the SREQ query that includes the service name, type or service code. In propagation scheme, service requesters send the (SREQ) query to the Query Acceptance module, which receives the SREQ from the SRN. The query undergoes various pre-processing steps like stemming, stop word removal and splitting. The pre-processed query is send to the service repository history. If the query is matched with the query in the service repository history then the corresponding services are retrieved. Otherwise the pre-processed query is send to the Discovery Agent to discover the services.

Suppose the service is not in the particular cluster, the SREQ query is passed to the next 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 should be lesser. When all the SPN register their services with the CHN in the cluster, if more than one similar service is provided by different SPN, it means CHN selects the best service according to LDA model. When SRN does not receive any response from SPN for a long time, then 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.
### Table 4.1 Service request (SREQ) message format

<table>
<thead>
<tr>
<th>Service name</th>
<th>Service type</th>
<th>Service Description</th>
<th>SREQ ID</th>
<th>Pkt-Size</th>
</tr>
</thead>
</table>

Where, Service name - Name of the service, Service category - Type of the service, Service description - details of the service or a URL where details can be found, SREQID - service request identifier, Pkt-size - amount of packets used.

#### 4.3.3.1 Query analyzer

The Figure 4.3 shows how the service requester query is processed by the Query Analyzer (Uma Maheswari & Karpagam 2014). The SRN’s SREQ is send to the Query Accepter to accept the SREQ and then ‘stemming’ is carried to removes the stemmed letters. Then remove the stop_words after that analyzed query is send to the service repository to discover the services. The following sections describe the pre-processing steps, preliminary evaluation of LDA service selection for service matching and LDA based service discovery.

![Figure 4.3 Functions of query analyser](image-url)
4.3.3.1.1 Stemming

Query Accepter receives SREQ and does the pre-processing steps to identify the service terms from the stored service repository. The terms are then filtered to remove the irrelevant words, called stop words and stemming (Porter 1980) is done to reduce the number of terms to their stems. If a term is represented two or more times, then it will be considered only once as it is representing the same concept or the word again and again.

Table 4.2 Example of stemming

<table>
<thead>
<tr>
<th>Before stemming</th>
<th>After Stemming</th>
</tr>
</thead>
<tbody>
<tr>
<td>emails</td>
<td>email</td>
</tr>
<tr>
<td>cows</td>
<td>cow</td>
</tr>
<tr>
<td>hopeful</td>
<td>hope</td>
</tr>
<tr>
<td>reports</td>
<td>report</td>
</tr>
</tbody>
</table>

4.3.3.1.2 stop_word removal

All the collected services like XML, SOAP, DSN services get into the desired format namely WSDL description language. They are made separate words using WSDL services. Tokenizer utilizes white space (tabs, space, new lines) as token delimiters for the text. The service descriptions and service uri are also tokenized after replacing all non-alphanumeric characters with space.

The extracted word matched with the initialized stop_word list removes the word from the SREQ Query (UmaMaheswari & Karpagam, 2014).

\[
\text{stop\_word[10 ]}[10] = \{ "i", "a", "in", "an", "was", "are", "as", "at", "be", "by", "com", "when", "where", "the", "for", "from", "about", "how", "is", "with", "it", "of", "on", "or", "that", "this", "to", "what", "who", "will", "ing" \} \\
\forall \text{ word in a SREQ} ;
\]
\[ \forall \text{word}_{\text{out}} \text{ in a stop_word list} \]
\[ \text{If}(\text{word}_{\text{in}} \text{ of a SREQ} == \text{stop word}_{\text{out}}) \]
Remove the word

Extracted services are sending it to the LDA service selection component. The section 4.3.3.4 describes the LDA based service selection in detail.

4.3.3.2 Preliminary evaluation of LDA for service matching

In the preliminary evaluation, there is a search for services within the service repository. The proposed algorithm is applied to the set of services to find matching services for requested service. Measure the similarity between services based on keyword matching (Papakonstantinou & Xu 2005) techniques. The proposed algorithm is denoted as LDA based algorithm. Before applying the LDA based selection match the relevant services and list out the services. After pre-processing extract the terms, the words are extracted from the service descriptions, service name and a unique term set with unique terms is obtained. For services that have no link, names, and documentation, use the topic as the default link.

4.3.3.3 Local service reply generation message

Service Repository consists of different type of services to find the user’s needs. Service Repository maintains the service table to identify the particulars of each web service. Table 4.1 shows the details of service repository. Service registry maintains the service history database and LDA service discover. Service history maintains the frequently entered queries and the corresponding location of the services. If the SRN query is pre-processed and is matched with the service history, then the service reply generation scheme and service Local Reply (LocReply) messages are sent by CHN to SRN. A LocReply message includes all frequently requested service providers list that satisfies the sent by the service requester. The head node will receive
the information about all the service providers inside the cluster that satisfy the requester. At this stage, it generates the reply message that will be propagated to the service requester. If the history does not have the SREQ service, then it sends it to the LDA selector module to discover the services.

4.3.3.4 LDA based service selection

The proposed method uses Linear Discriminant Analysis (LDA) technique for service selection. LDA (McLachlan & Wiley, 1992) is a supervised technique which is used for selecting services in a large set of dataset. However, these techniques are not used for service selection previously. The proposed technique uses different type of services from which the optimal is selected. In this section, the implementation of LDA technique for the selection of services is described.

4.3.3.4.1 Linear Discriminant Analysis

LDA is a good classification method (Shailendra Singh et al. 2009; Zhao et al. 1998). It is mathematically robust and often generates models whose correctness is on par with the more difficult procedure. None of the work has been carried out in LDA on service selection. In this work the, focus is on linear transformation since LDA commonly attains good performances in the service selection, even if the assumptions of common covariance matrix among groups and normality are often violated. LDA is based on the concept of searching for a linear combination of variables (predictors) that best separates among classes (targets) and the categorization is then carried out in the transformed space based on some metrics such as Mahalanobis distance. It finds the optimal transformation matrix to preserve most of the information that can be used to discriminate among the different classes. Therefore, the analysis requires the data to have suitable class labels and it mathematically formulates the optimization. Let $X= \{x_1, x_2, \ldots, x_d\}$ be the dataset given d-dimensional vector of Quality of Web Service (QWS). Each data point belongs to one of the
k object classes. The following Equations 4.1 and 4.2 defines the within a class scatter matrix and between class scatter matrices.

\[ S_b = \sum_{i=1}^{k} M C_i (x_i - m)(x_i - m)^T \]  
\[ S_w = \sum_{i=1}^{k} \sum_{x_p \in X_i} (x_i - m_i)(x - m_i)^T \]

Where \( m \) is mean of the \( k^{th} \) class, \( x_p \) represents the \( p^{th} \) data of class. \( m_i \) represents the input as 0 or 1. The number of vectors in class \( X_i \) is denoted by \( MC_i \).

Next, to compute the optimal linear transformation matrix \( L \). Final matrix

\[ L = \max L \frac{|L^T S_b L|}{|L^T S_w L|} = [l_1, l_2, \ldots, l_d]. \] (4.3)

Where \( \{l_i \mid i=1, 2, 3, \ldots, d\} \) is a eigen vectors of \( S_b \) and \( S_w \). \( S_b = \lambda S_w l_i \) (i.e) there are ‘l’ features of that maximizes the between class separation of data while minimizing the within class scatter.

Table 4.3 shows the notations used in LDA. Its aim is to maximize the between class separation of data \( (S_b) \) while minimizing the within class scatter \( (S_w) \).

### Table 4.3 Notations used in LDA

<table>
<thead>
<tr>
<th>Notations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>Number of dimensions, (i.e) terms</td>
</tr>
<tr>
<td>k</td>
<td>Number of classes</td>
</tr>
<tr>
<td>( S_b )</td>
<td>Between scatter matrix</td>
</tr>
<tr>
<td>( S_w )</td>
<td>Within scatter matrix</td>
</tr>
<tr>
<td>( m_i )</td>
<td>Centroid of the ( i^{th} ) class</td>
</tr>
<tr>
<td>MC</td>
<td>Mean of the class</td>
</tr>
</tbody>
</table>
4.3.3.4.2 LDA based service selection

Assume that ‘n’ number of candidate services $S = s_1, s_2, ... s_n$ are available in the service repository which is advertised by various service providers. These ‘n’ services which meet the service requestor’s needs based on the functional requirements and ‘m’ QoS attributes, $1 \leq i \leq n$, then get a $n \times m$ matrix $S$ shown in Equation (4.4), where each row and column represents the candidate service corresponding to the QoS property, and value of the attributes of all services respectively. The LDA algorithm computes the service selection score based on the service requester’s needs.

$$S = \begin{bmatrix}
S_{11} & S_{12} & ... & S_{1m} \\
S_{21} & S_{22} & ... & S_{2m} \\
S_{31} & ... & ... & S_{3m} \\
. & . & . & . \\
S_{n1} & S_{n2} & ... & S_{nm}
\end{bmatrix} \quad (4.4)$$

Service requestor may have a variety of QoS requirements which are more relevant to the application. Service repository contains number of services with different QoS attributes. The following attributes are considered for response time ‘RT’ (The time lagged between the request sent and the response received by the user), throughput ‘TPT’ (is the time of successful message delivery over a time), reliability ‘R’ (The probability that a request will perform a required function without any error), best practices ‘BP’ (the amount to which a web service follows web service information basic profile), and documentation ‘D’ (measure number of description tags in WSDL). CSDM-LDA algorithm uses these QoS attributes for the case in the experiment.

Lenders et al. (2005) describes the services as a whole and does not consider any categorization into classes. But, in CSDM-LDA, it gives an optimal to discriminate among the different classes of services. In general the number of services with QoS can be computed by the CSDM-LDA method.
which is less than the number of classes in the problem. The following steps are involved in CSDM-LDA algorithm.

**Step1: Normalization**

Let the ‘m’ criteria be considered for evaluating the QoS for each service \( s_{ij} \) as it describes \( j^{th} \) QoS attribute of \( i^{th} \) service. The QoS of each service is different from each attribute. If there are higher throughput and less response time, it would be considered as negative criteria. If the reliability and documentation is higher, then it would be considered as positive criteria. Here, the attribute criteria are different to normalize. The purpose of the normalization is to give uniform distribution of the QoS values. The preferences of the SRN are obtained in the form of binary decisions in an array (about preference[\( j \)]). This array will have a ‘0’ for parameters which need to be low and ‘1’ for parameters which should be high. Then formulate the service sets with the QoS attribute, which are to be classified in the original space. For each matrix element

\[
\text{If}(s_{ij}=="positive") \quad (4.5) \\
Y' = \frac{Y - Y_{\text{min}}(j)}{Y_{\text{max}}(j) - Y_{\text{min}}(j)} \\
\text{Else if } (s_{ij}=="negative") \\
Y' = \frac{Y_{\text{max}}(j) - Y}{Y_{\text{max}}(j) - Y_{\text{min}}(j)}
\]

Where \( Y = s_{ij} \) and \( Y' \) is scaled by \( Y \), which is specified by Equation (4.5) where \( Y \) represents the normalized value of each service and \( Y_{\text{min}} \) denotes the minimum QoS value and \( Y_{\text{max}} \) represents the maximum value of QoS.

A discriminant function is defined as in Equation (4.6)

\[
X_i^C = w_1s_{i1}^C + w_2s_{i2}^C + w_3s_{i3}^C + \cdots + w_ns_{in}^C \\
(4.6)
\]
Where \( w_1, w_2, \ldots, w_n \) are considered as a weighting factor. Then the corresponding discriminant function for each class \( S_1, S_2 \) can be defined as follows.

\[
X_1 = w_1 s_{11} + w_2 s_{12} + w_3 s_{13} + \cdots + w_n s_{1t} \\
X_2 = w_1 s_{21} + w_2 s_{22} + w_3 s_{23} + \cdots + w_n s_{2t} \\
X_3 = w_1 s_{31} + w_2 s_{32} + w_3 s_{33} + \cdots + w_n s_{3t}
\]

The service sets are represented as \( S_1, S_2 \) matrices consist of services as given below:

\[
S_1 = \begin{bmatrix}
    s_{11} & s_{12} & \cdots & s_{1t} \\
    s_{21} & s_{22} & \cdots & s_{2t} \\
    s_{31} & s_{32} & \cdots & s_{3t} \\
    \vdots & \vdots & & \vdots \\
    s_{n1} & s_{n2} & \cdots & s_{nt}
\end{bmatrix}
\]

\[
S_2 = \begin{bmatrix}
    s_{11}^2 & s_{12}^2 & \cdots & s_{1t}^2 \\
    s_{21}^2 & s_{22}^2 & \cdots & s_{2t}^2 \\
    s_{31}^2 & s_{32}^2 & \cdots & s_{3t}^2 \\
    \vdots & \vdots & & \vdots \\
    s_{n1}^2 & s_{n2}^2 & \cdots & s_{nt}^2
\end{bmatrix}
\]

**Step 2: Covariance matrix**

Calculate the mean (average) vectors, covariance matrices of each service set \( S_1 \) and \( S_2 \) as given in Equation (4.8). The mean of \( S_1 \) and \( S_2 \), which is obtained by merging \( S_1 \) and \( S_2 \) by Equation (4.7) covariance matrices, is obtained by (4.9). The mean value of each set can be defined as

\[
m_1 = \frac{1}{n_1} \sum_{i=1}^{c} s_{ij}^1 \\
m_2 = \frac{1}{n_2} \sum_{i=1}^{c} s_{ij}^2
\]
Where \( j = 1, 2, 3, \ldots t \) t = ‘m’ criteria.

where \( n_1, n_2 \) is the total service in the S1, S2.

Now, define the Total mean by using the following Equation 4.10

\[
M = \frac{1}{S_{t_{\text{total}}}} \sum_{i=1}^{C} S_{i_{\text{total}}} m_i = \frac{1}{S_{t_{\text{total}}}} \sum_{i=1}^{C} \sum_{j=1}^{S_{i_{\text{total}}}} m_{ij} \quad (4.10)
\]

Where \( S_{i_{\text{total}}} \) is \( n_1 + n_2 \) . \( MC_i = \sum_{i=1}^{S_{i_{\text{total}}}} m_{1i}, m_{2i} \). Sample Covariance’s

\[
SC_i = \frac{1}{S_{i_{\text{total}}}} (M_i - m_i) (M_i - m_i)^T \quad (4.11)
\]

\[
S_b = \sum_{i=1}^{C} S_{i_{\text{total}}} (M_i - m_i)(M_i - m_i)^T \quad (4.12)
\]

where, \( i = 1 \ldots C \). \( S_b \)-Covariance matrix.

Normalize this into \( aS_b \) by dividing \( S_{i_{\text{total}}} \), but it is not necessary to do so. If all classes were the same size of services, and then \( S_{i_{\text{total}} - 1} \) could be removed.

\[
S_w = \sum_{i=1}^{C} (S_{i_{\text{total}} - 1}) SC_i \quad (4.13)
\]

**Step 3: Pooled Covariance**

The \( S_w \) (Equation 4.13) is computed by pooling the estimates of the covariance matrices of each class. Since each \( SC_i \) has rank \( S_{i_{\text{total}} - 1} \) its rank can be at most \( S_{\text{total}} - C \). The main objective of LDA is to find a projection matrix \( L \) that maximizes the ratio of the determinant of \( S_b \) to the determinant of \( S_w \). This can be written by using Equations (4.12) and (4.13).

**Step 4: Optimum Service Class**

Optimum service class from all services \( S \) is obtained by the following Equation 4.14

\[
S_b = \lambda S_w L. \quad (4.14)
\]
If the $S_b$ equation (4.14) is large and $S_w$ equation (4.13) is small, then it will yield a better discriminant value $L$.

\[
\left( s_{ij} - \left( \frac{m_{ij} - m_{ij}^2}{2} \right) \right) > P(C_i) \quad \forall i = 1..C \quad (4.15)
\]

Where $P(C_i)$ is the probability that a service $s_i$ is belonging to class C. An easy linear correlation among the service scores and predictors can be used to test which predictor contributes extensively to the discriminant function. Correlation varies from -1 to 1, where -1 and 1 mean the highest contribution, but in different directions and 0 means that there is no contribution at all. Highest contribution is considered as an optimal class.

**Step 5: Score Evaluation**

Then service matching is necessary to select the best services from the optimum service class. So it applies the Mahalanobis distance to find the best service. From Equation (4.14), compute the weighting parameter to get the score of each service.

\[
L(w_1, w_2, \ldots w_n) = \frac{(m_1 - m_2)^2}{\sum_{i=1}^{n_1} (m_{1i} - m_{2i})^2 + \sum_{i=1}^{n_2} (m_{1i} - m_{2i})^2} \quad (4.16)
\]

From the optimized function, the weighting parameter $(w_1, w_2, \ldots w_n)$ is computed as follows:

\[
S. \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} m_{11} - m_{12}^2 \\ m_{21} - m_{22}^2 \\ \vdots & \vdots & \vdots \\ m_{n1} - m_{n2}^2 \end{bmatrix} \Rightarrow \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = S^{-1} \begin{bmatrix} m_{11} - m_{12}^2 \\ m_{21} - m_{22}^2 \\ \vdots & \vdots & \vdots \\ m_{n1} - m_{n2}^2 \end{bmatrix} \quad (4.17)
\]

By Equation (4.17), calculate the overall score of the service with their QoS and select the highest score is the best service.
4.3.3.4.3 Service reply propagation message

In service reply propagation scheme, Service Reply Packet (SREP) message is generated by the head node in the cluster. It is propagated towards the service requester. During the service discovery process, the service requester could be moving and the SREP message is propagated towards the service requester. Only mobile nodes that are closer to the expected cluster region of the service requester forward the SREP message. Moreover, nodes that satisfy the requirements specified by the requester can forward the SREP message. Because the service requester can use the alternative path received in a service reply message to communicate with the chosen service provider.

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 optimal service of the SPN is selected based on some criterion by the service requestor (SRN) node. The SPN will receive the service with the service description, EXP_T. If the route exists already in the routing table or the service exists already in the service table, then the routing table or the service table is updated respectively.

In the cluster, CHN provides the reply regarding the service details that will pass on through the number of intermediate nodes. Each member node forwards the SREP packet and inspects the service description and it 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, which is less than the threshold.
- The node should have the sufficient space to store the SREP.
- The service is offered by SPN which is not more than the H hops away from the SRN.
Figure 4.4 Example scenario to search a service in two different clusters

Figure 4.4 shows the simplified examples of service discovery. The first case SSN1 as a SRN sends the (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 SREQ to the gateway node (GWN) according to nearest and selects the lesser number of hops CHN from that 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 gets the SRSP from SSN4.
Consider, SSN7 (SRN) wants to get the scanner service, but it does not have the service in the request node cluster (right side). Then it forwards the SREQ to the adjacent cluster. In the meantime, SSN4 moves to the requester node cluster and it becomes the new member node of the cluster head CHN2. Hence, the service from node SSN4 is not provided anymore by CHN1, so the discovery fails. Though, CHN2 caches the SREQ for a period of time the newly joined node registers its information to the CHN2 and the CHN2 checks its service list. The scanner service is available in the CHN2 (right side cluster). Now CHN2 sends the SREP. It includes address or location of the service to SSN7 and SSN7 invoke and gets the SRSP from SSN4. In this case, the discovery is successful and the service is found.

Consider one of the service advertisements based on the disaster alert information system is advertised by SPN:
SADV
{
  serv_pubmsg(SER1,SMS, http://ws.across communications.com /SMS.asmx?)
}
serv_pubmsg(SER2,DOTSGeoPhone,http://ws2.serviceobjects.net/gp/GeoPhone.asmx?wsdl)
serv_pubmsg(SER3,Phone,http://ws.acrosscommunications.com/Phone.asmx?wsdl)
serv_pubmsg(SER4,Email,http://ws2.serviceobjects.net/ev/EmailValidate.asmx?wsdl)
serv_pubmsg(SER5,Fax,http://ws.acrosscommunications.com/Fax.asmx?wsdl)
serv_pubmsg(SER6,interFax,http://ws.interfax.net/dfs.asmx?wsdl)
serv_pubmsg((SER8,WeatherFetcher:
}

This SADV is sent by a SPN which is stored in to the service repository (CHN) in the cluster. The service names are pre-processed and are stored into the service discovery agent. According to the experience, SADV messages are broadcast for every second. The proposed CSDM-LDA advertisement messages are stored between 0 and 5 seconds depending on the arrival time before forwarding. Adjacent nodes once in a while exchange their possible values every 5 seconds with broadcast packets. Services are collected from QWS. Each service is assigned by a QoS value. Datasets provide 365 real services having real values with 6 attributes were collected using the web service crawler engine (WSCE). Each service was tested over a ten-minute period of three consecutive days. Each row in the dataset corresponds to an existing web service implementation available to the public.

Consider one of the service requests based on the disaster recovery management information system requested by SRN:
SREQ
{
Invoke (Email and SMS Weather Alert Services, emergency alert through sms or phone call, Fax the weather reports)
}
}

Service requestor needs to obtain the ‘sms’ related services. The SREQ query is sent to the Query accepter and it passes the Query to the preprocessor to stem the word within the CHN

**Step 1:** After stemming

Query = (Email and SMS Weather Alert Service, emergency alert through sms or phone call, Fax the weather report)

**Step 2:** Removal of Stop Words

Query=(Email SMS Weather Alert Service, emergency alert sms phone call, Fax weather report)

- Q1=Email SMS Weather Alert Service
- Q2=emergency alert sms phone call
- Q3= Fax weather report

Terms along with the description, identifier, and QoS data are stored in the repository. SREQ is matched with the service repositories’ service history. If it is available, send the Local Reply(SPN1_id,sms) and it forwards the SREQ pre-processed content to the LDA selector.

After pre-processing the words extracted from the service descriptions, obtain a keyword set with unique words. For services that have no description, use the topic as the default description. It discards the documents with no label or with multiple labels. The pre-processed results will go for LDA model to separate in to different classes. Here take the Q1 and Q2 SREQ, keyword is ‘sms’ services. After pre-processing, it returns 8 ’sms’ related
services and these services meet the service requestors QoS constraints. The overall performance is obtained from the proposed CSDM-LDA algorithm. The extracted and pre-processed 'sms' services are listed out in Table 4.4.

<table>
<thead>
<tr>
<th>Service</th>
<th>Service name</th>
<th>Service Description</th>
<th>Response Time (ms)</th>
<th>Throughput (hits/sec)</th>
<th>Reliability (%)</th>
<th>Best Practices (%)</th>
<th>Documentation (%)</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>SMS</td>
<td><a href="http://ws.acrosscommunications.com/SMS.asmx?wsdl">http://ws.acrosscommunications.com/SMS.asmx?wsdl</a></td>
<td>113.8</td>
<td>5.2</td>
<td>81</td>
<td>84</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>s2</td>
<td>SMS</td>
<td><a href="http://www.abctext.com/webservices/SMS.asmx?wsdl">http://www.abctext.com/webservices/SMS.asmx?wsdl</a></td>
<td>179.2</td>
<td>0.7</td>
<td>65</td>
<td>69</td>
<td>36</td>
<td>1</td>
</tr>
<tr>
<td>s3</td>
<td>SendSMS</td>
<td><a href="http://www.webservicex.com/SendSMS.asmx?wsdl">http://www.webservicex.com/SendSMS.asmx?wsdl</a></td>
<td>1308</td>
<td>6.3</td>
<td>67</td>
<td>84</td>
<td>41</td>
<td>1</td>
</tr>
<tr>
<td>s4</td>
<td>SendMessages</td>
<td><a href="http://www.webservicex.com/sendsmsworld.asmx?wsdl">http://www.webservicex.com/sendsmsworld.asmx?wsdl</a></td>
<td>291.07</td>
<td>5.2</td>
<td>53.6</td>
<td>84</td>
<td>95</td>
<td>2</td>
</tr>
<tr>
<td>s5</td>
<td>SMS</td>
<td><a href="http://smsinter.sina.com.cn/ws/smswebservice0101.wsdl">http://smsinter.sina.com.cn/ws/smswebservice0101.wsdl</a></td>
<td>436.5</td>
<td>4.5</td>
<td>43.2</td>
<td>84</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>s6</td>
<td>SendSMSWORLD</td>
<td><a href="http://www.sms.mi.o.it/webservices/sendmessages.asmx?wsdl">http://www.sms.mi.o.it/webservices/sendmessages.asmx?wsdl</a></td>
<td>3103</td>
<td>5.3</td>
<td>64.3</td>
<td>87</td>
<td>33</td>
<td>1</td>
</tr>
<tr>
<td>s7</td>
<td>SMSWS</td>
<td><a href="http://www.barnaland.is/dev/sms.asmx?wsdl">http://www.barnaland.is/dev/sms.asmx?wsdl</a></td>
<td>751</td>
<td>6.8</td>
<td>79.3</td>
<td>91</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>s8</td>
<td>emSoapService</td>
<td><a href="https://www.emsoap.net/smsservice.asmx?wsdl">https://www.emsoap.net/smsservice.asmx?wsdl</a></td>
<td>424.54</td>
<td>4.3</td>
<td>11.9</td>
<td>80</td>
<td>34</td>
<td>2</td>
</tr>
</tbody>
</table>

Group the services from all the services defined. In order to find an optimum class, it is necessary to define a measure of separation between the services. So that formulates the QoS data of services are formulated and
classified in the new space and the corresponding value of QoS after normalization is shown in Table 4.4.

**Table 4.5 QoS data information of SMS services after normalization**

<table>
<thead>
<tr>
<th>Service</th>
<th>Response Time (ms)</th>
<th>Throughput (hits/sec)</th>
<th>Reliability (%)</th>
<th>Best Practices (%)</th>
<th>Documentation (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>S1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s1</td>
<td>0.08651</td>
<td>0.00344</td>
<td>0.06142</td>
<td>0.06372</td>
<td>0.00788</td>
</tr>
<tr>
<td>s2</td>
<td>0.13654</td>
<td>0.00000</td>
<td>0.04919</td>
<td>0.05225</td>
<td>0.02700</td>
</tr>
<tr>
<td>s3</td>
<td>1.00000</td>
<td>0.00428</td>
<td>0.05072</td>
<td>0.06372</td>
<td>0.03083</td>
</tr>
<tr>
<td>s6</td>
<td>0.22211</td>
<td>0.00344</td>
<td>0.04047</td>
<td>0.06372</td>
<td>0.07213</td>
</tr>
<tr>
<td><strong>S2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s4</td>
<td>1.00000</td>
<td>0.00106</td>
<td>0.02009</td>
<td>0.02741</td>
<td>1.00000</td>
</tr>
<tr>
<td>s5</td>
<td>0.24153</td>
<td>0.00155</td>
<td>0.02493</td>
<td>0.02870</td>
<td>0.24153</td>
</tr>
<tr>
<td>s7</td>
<td>0.14012</td>
<td>0.00081</td>
<td>0.01329</td>
<td>0.02644</td>
<td>0.14012</td>
</tr>
<tr>
<td>s8</td>
<td>0.13626</td>
<td>0.00074</td>
<td>0.00319</td>
<td>0.02515</td>
<td>0.13626</td>
</tr>
</tbody>
</table>

From the values show in Table 4.5, find the mean of each service sets and compute the mean for entire sets, which are obtained by Equation (4.8) and (4.9) according to that compute the covariance matrix for each service class. S1 consists of s1, s2, s3, s6 services. S2 consist of s4, s5, s7, s8. In LDA, $S_b$ (equation 4.12) and $S_w$ (equation 4.13) are used to formulate the optimum criteria for class separability which are shown in Table 4.6 and 4.7.

**Table 4.6 Service Between class $S_b$**

<table>
<thead>
<tr>
<th>QoS</th>
<th>RT (ms)</th>
<th>TPT (hits/sec)</th>
<th>R (%)</th>
<th>BP (%)</th>
<th>D (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RT(ms)</td>
<td>0.05373</td>
<td>0.00007</td>
<td>0.00013</td>
<td>0.00010</td>
<td>0.00016</td>
</tr>
<tr>
<td>T</td>
<td>0.00007</td>
<td>0.00000</td>
<td>0.00001</td>
<td>0.00001</td>
<td>0.00001</td>
</tr>
<tr>
<td>R (%)</td>
<td>0.00013</td>
<td>0.00001</td>
<td>0.00015</td>
<td>0.00012</td>
<td>0.00006</td>
</tr>
<tr>
<td>BP</td>
<td>0.00010</td>
<td>0.00001</td>
<td>0.00012</td>
<td>0.00012</td>
<td>0.00010</td>
</tr>
<tr>
<td>D (%)</td>
<td>0.00016</td>
<td>0.00001</td>
<td>0.00006</td>
<td>0.00010</td>
<td>0.00020</td>
</tr>
</tbody>
</table>
Table 4.7 Service Within class $S_w$

<table>
<thead>
<tr>
<th>QoS</th>
<th>RT (ms)</th>
<th>TPT (hits/sec)</th>
<th>R (%)</th>
<th>BP (%)</th>
<th>D (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RT (ms)</td>
<td>0.0269</td>
<td>0.0000</td>
<td>0.0001</td>
<td>0.0000</td>
<td>0.0001</td>
</tr>
<tr>
<td>T</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>R (%)</td>
<td>0.0001</td>
<td>0.0000</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td>BP</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>D (%)</td>
<td>0.0001</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

Table 4.8 Optimum class selection

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2</td>
<td>0.93197</td>
<td>A better</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

An easy linear correlation among the model scores and the predictors can be used to test the predictors which contribute extensively to the discriminant function. Correlation differs from -1 to 1, if the value is 1 then it finds the best predictor and 0 means that there is no contribution at all.

Table 4.9 Score values for services $S_2$

<table>
<thead>
<tr>
<th>SER</th>
<th>Service Name</th>
<th>RT (ms)</th>
<th>TPT (hits/sec)</th>
<th>R (%)</th>
<th>BP (%)</th>
<th>D (%)</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>s4</td>
<td>Send Messages</td>
<td>0.3000</td>
<td>0.0002</td>
<td>0.0020</td>
<td>0.0055</td>
<td>0.2000</td>
<td>0.5077</td>
</tr>
<tr>
<td>s5</td>
<td>SMS</td>
<td>0.0725</td>
<td>0.0003</td>
<td>0.0025</td>
<td>0.0057</td>
<td>0.0483</td>
<td>0.1293</td>
</tr>
<tr>
<td>s7</td>
<td>SMSWS</td>
<td>0.0420</td>
<td>0.0002</td>
<td>0.0013</td>
<td>0.0053</td>
<td>0.0280</td>
<td>0.0768</td>
</tr>
<tr>
<td>s8</td>
<td>emSoap Service</td>
<td>0.0409</td>
<td>0.0001</td>
<td>0.0003</td>
<td>0.0050</td>
<td>0.0273</td>
<td>0.0736</td>
</tr>
</tbody>
</table>

According to Table 4.8, $S_2$ is the better service class from which one can find the optimal service. In favour of that, CSDM-LDA can calculate the score of each service. The score calculation for each service is done from the combination of quality value of services in terms of five parameters including response time, throughput, reliability, best practices and documentation by the use of Equations (4.14) and (4.16). According to the table 4.9, s4 is the best SMS service.
4.5 RESULTS AND DISCUSSION

In order to identify the performance of the service discovery during service selection, experiments are conducted to prove the performance and convergence with respect to mobility and the control traffic overhead caused by the discovery protocol. Simulation time for the test is set to 1000 seconds. Thirty runs are carried out in the graph and the results of all runs are averaged to produce the resulting graphs. The network size is limited to 400 nodes on a network area (300m x 300m) topology.

The performance metrics considered for evaluation are the service success hit ratio, control overhead and the average response time. The Success hit ratio is the ratio of number of SREQ packets entered at any service instance to the total number of packets sent by the SRNs. It is also used to measure the number of services found in the MANET. By adding the CSDM-LDA with OC-AHPPSO, the control overhead is also determined by varying the nodes under mobility. The effectiveness of the LDA is also analyzed in terms of false positive. The average response time is the time interval between the arrivals of the SREP for the corresponding SREQ which is averaged over all the services that are discovered successfully. The next metric is called the matching time for the SREQ measured. The scalability is also measured while adding the new services in the network.

Two metrics success hit ratio and the control overhead are evaluated first in various service instances for the performance of the service discovery model during mobility of the node. Figure 4.6 shows the success hit ratio by varying pause time (sec). For simulation, it takes 5, 10, 15 similar type of instances on different nodes. Nearly 10 SRN sends SREQ messages at a rate of two (packets/sec). For 1000 pause time (High pause time, low mobility), the success hit ratio is 99(%). At the end of the pause time (low pause time, high mobility), success of discovery is 87(%), and the performances quite stable.
Finally conclude that the success hit ratio is increased when number of service instances increase because SRN and services tend to get closer on average.

Figure 4.6 Success hit ratio under mobility

The Figure 4.7 shows the performance of the control overhead, which is specified as the number of SADV, SREQ and Service Reply(SREP) messages exchanged among the adjacent nodes in the network. Control overhead is autonomous of the node’s mobility because the packets are periodically distributed. Overhead linearly depends on the number of service instances used. CSDM-LDA discovery overhead is reduced. CSDM-LDA uses a fewer directories to handle the service descriptions. When a requester cannot find the requested service information in its service table, it should issue a packet to discover the service to the service repository. The CSDM-LDA utilizes the fewer head nodes at the higher density network. So it leads to lesser overhead while sending and receiving the request and response packets. CSDM-LDA does not account the service query and reply messages in the control overhead which depends on the search activity of requesters. These messages are not critical to the scalability of the system since they are unicast and not flooded.
The result of adding CSDM-LDA together with the clustering protocol OC-AHPPSO on the control overhead is determined, by varying the number of nodes services.

The Figure 4.8 shows that the average sending rate of control packets and where the total traffic load per node was reduced after adding CSDM-LDA. The Figure 4.8 indicates that there is no significant overhead on the network while adding control packets in LDA to the OC-AHPPSO, which also measured the average number of control packets sent per second at each node.
The simulation runs for 150 seconds by varying the Average Energy of SPN nodes. The initial energy of all the nodes is 100 joules. The service provider nodes continuously service the requesters. As the load is distributed by using stochastic process model (chapter 5), all the SPNs get a chance to service the clients. In each scenario, some service providers are randomly chosen to check the residual energy. Figure 4.9 analyze the residual energy for 0 to 150 seconds.

The residual energy of all the SPNs is approximately same for 50 seconds when compared to the existing system. LWSD, DSDM and the nodes drain energy at constant rate. So the average energy dissipation in each second is less compared with existing as shown in Figure 4.9. Thus all the SPN nodes in a cluster started with same energy dissipate uniformly and die almost at the same time in CSDM-LDA.
The response time acquired by LWSD and DSDM with CSDM-LDA over maximum speed of the node is evaluated as shown in Figure 4.10. During simulation period, for all the speeds of 2m/s to 15m/s, the nodes reach the steady-state average speed. In the DSDM, if the node does not have the services within their cluster head node (CHN) then DSDM Information Packet (DIP) is broadcasted to all other nodes for the SREQ, sends the queued packets, and caches the registrations received for these packets from the nearest CHN until they are matched the SREQ. In LWSD, it does not consider the partitioning the network and it broadcast the SREQ to the entire nodes. In contrast, in the CSDM-LDA, the SRN’s service is not found within the local cluster CHN, and then it forwards the SREQ to the other neighbour CHN. Hence, CSDM-LDA response time is improved. This includes the time taken by the SREQ to travel to the gateway node to the other CHN and LDA. It avoids recalculating the entire service classes again when new service or SREQ enters to the CHN, to search the service for the SREQ, and identify the exact services and retrieved requested services to the SRNs.
The Figure 4.11 shows the service discovery success hit ratio for varying number of nodes acquired by DSDM and LWSD with CSDM-LDA which is evaluated. When the number of nodes increased, the success hit ratio gets reduced slightly in CSDM-LDA. The node maintains the steady-state average speed. In DSDM and LWSD the nodes broadcast the SREQ to the network. Hence it takes more time when the number of nodes increased. When a node needs a service due to increasing number of nodes, the services also increased. It forwards the SREQ to node, until SREQ matched with the SRN request. But, CSDM-LDA, first the SREQ of a node is searched with the CHN’s DA. If it is available directly, send the SREP to SRN. Or else SREQ is forwarded to the next nearest CHN to find the desired service as discussed in section 4.3. The LDA performs the service discovery quickly and efficient search has been done by the CSDM-LDA.
The scalability of the MANET by adding new services or by increasing the SREQ is evaluated. Let, the number of SREQ is set to 200. The total number of SREQs generated by the CSDM-LDA is increased, and the average discovery of services and intra and inter-cluster discovery of services are measured.

Figure 4.11 Success hit ratio for varying number of nodes

Figure 4.12 Scalability when adding new SREQ and services
Figure 4.12 shows the intra-cluster and inter-cluster discovery which is 93% and 77% respectively. Hence, the average discovery of services is around 85%. Thus, the CSDM-LDA maintains high success ratio, even when SREQ is increased. Hence, the scalability of the CSDM-LDA is improved.

The precision and recall (Retna Raj et al. 2014) values are computed from the following equations to measure the accuracy of the services discovered.

\[
\text{precision} = \frac{|\text{relevant services} \cap \text{retrieved services}|}{|\text{retrieved services}|} \quad (4.18)
\]

\[
\text{recall} = \frac{|\text{relevant services} \cap \text{retrieved services}|}{|\text{relevant services}|} \quad (4.19)
\]

Let 15 services be taken for consideration. The precision and recall values are evaluated and it shown in Figure 4.13 and 4.14. The proposed approach CSDM-LDA provides higher precision and recall values due to the LDA which removes the duplication of services and eliminates the discriminate values from the service class and selects the most appropriate optimal service. The test results show that the CSDM-LDA provides better precision and recall values than DSDM and LWSD.

![Figure 4.13 Precision](image-url)
Recall

The Figure 4.15 shows that the minimum frequency of SREQ initiated then the CSDM-LDA is more efficient than the DSDM and LWSD depending on the number nodes. The Figure 4.15 shows that the minimum frequency of SREQ initiated then the CDSD-LDA is more efficient than the existing protocols depending on the number of nodes.
4.6 SUMMARY

Cluster based service discovery for OC-AHPPSO by using LDA method is implemented. The service information is acquired by designing a dynamic topology for disaster recovery information system. LDA is used for separating the classes of service and by ranking the service. Because of the limited memory capacity of mobile nodes, the LDA is well-suited algorithm for representing the services and for selecting the services dynamically with lesser space. Thus, it provides the optimized services and service discovery is improved by additionally considering service matching and it avoids the duplication of services. Finally, some results are taken from evaluations made on the performance of the system in terms of service discovery success hit ratio and control overhead under varying mobile speed, average response time, scalability of the MANET by adding new services or by increasing the SREQ which is evaluated.

The result of adding CSDM-LDA together with the clustering protocol OC-AHPPSO does not cause any control overhead by varying the number of nodes services. The scalability of the CSDM-LDA is better than the LWSD and DSDM. CSDM-LDA maintains high success ratio, even when SREQ is increased. CSDM-LDA matches the SRNs SREQ with the available service description in the service registry which provides better precision and recall values than the other method such as DSDM and LWSD. Irrespective of the varying number of nodes the CSDM-LDA takes minimum amount of time in the discovery process. When the SREQ is increased inter and intra-cluster service discovery is about 93% and 77% respectively. Thereby the success hit ratio also increased by 20% and 34% compared to LWSD and GSD respectively. When the number of service requests increased the success hit ratio is increased, thereby the delay gets reduced by 12%.