Chapter 5

Integrated Service Discovery and Routing in Pervasive Environment

5.1 Introduction

MANET's composition in pervasive environment, its modeling and the routing scheme needed in such environments are discussed in the previous chapter. The various challenges and modeling of MANET are discussed using the results specified in [2]. Pervasive environment characteristics and the features that are to be included in the modeling are referred from [12]. A framework for wireless mobile heterogeneous computing is considered to augment capabilities and to save scanty resources of mobile host by making full use of available resources in the surrounding is discussed in [4].

Service discovery procedures [22][37] and its challenges are taken into consideration for the modeling of pervasive environment. Selecting the proper service possessing device [24] is also considered for the modeling. Clustering and its challenges [28][34][51] are also taken into account in the modeling of pervasive environment. The performance of routing protocols in MANET is always dependent on the availability and stability of wireless links [41] and the topology control in MANETs is needed for reducing interference collisions and in consequently retransmission [42]. This helps to model the MANETs composition in pervasive environment in an efficient way. Various multicast routing protocols [45] with distinguishing feature have been already proposed are used for modeling a better broadcasting scheme. Location-based routing is difficult when there are gaps in the network topology and nodes are mobile or frequently disconnected [29] and so
there may be a possibility of the occurrence of flooding during broadcasting [46] is also taken into the consideration in modeling the environment.

The mobility models and their characteristics [47][75] and their metrics evaluation [37] and the various mobility pattern generations[69] are also considered while modeling the pervasive environment. The various wireless technologies [64][71] from which the characteristics of pervasive devices can be measured depending on the MAC are also included in the modeling. Efficient routing characteristics [27][35][52][61] for both service discovery and routing are modeled with the help of other existing integrated service discovery [60] [62] systems. A distributed service discovery protocol for pervasive environments called Group Service Discovery [63] from which the characters such as caching and group based servicing are modeled.

But all the above said architectures and procedures are meant for obtaining the result of any of the single functions such as broadcasting, service discovery, clustering or routing. The proposed routing scheme works on all these functions at the same time of execution so that it is a complete service discovery and routing scheme for the pervasive environment. From this chapter onwards a ‘node’ can also be termed as a ‘device’ in a pervasive network.

5.2 Background Works

MANET is created temporarily for the purpose of either communication, sharing services and resources are discussed in [30]. By using an integrated scheme which integrates service discovery and routing one can achieve the task in a cost-effective manner. Service discovery is indispensable to Adhoc networking where establishing a stand-alone and self-configurable communication environment is the main objective. A distributed service discovery model for service sharing among diverse mobile devices in MANET using up-to-date routing information enabled by
proactive routing protocols is discussed in [22]. The other base architectures referred for integrating the service discovery and routing are discussed in [60][61][62][63] and are elaborately discussed in chapter 3. The next topics discuss about the various components and working of the proposed service discovery and routing scheme.

5.3 SHMSDP – The Service Discovery Scheme

SHMSDP is the acronym of Simple Hi message Service Discovery Protocol. This is a service discovery scheme that contains various components which are meant for different functions and are service assignment, service request through broadcasting, loop avoidance, service searching, service discovery and service reply. Now we will see each components of this proposed scheme one by one.

Service Assignment

In pervasive environment each device must have at least a type of service or resource in it. A service contains two fields, a Service name ‘S’ and a life time ‘Ts’. The service may differ in nature and that can be shared through broadcasting or it can be requested by a requester. In this research work, it is assumed that each device possesses a single service and is assigned arbitrarily by using the procedure ‘Service_Entry’. The service holder device can share its service at any time. The procedure used for Service Assignment is referred from [99] and is:

Service_Entry (nodeid, Service, description), this procedure assign a service and its description as context to a device which is a part of the pervasive environment. The procedure is the first one to be executed in order to assign a service and id for a node.

Service Request through Broadcasting

Any device in the network can access the service from any of the other nodes available in the network. In Figure 5.1, node Service Request Node (SRN) needs to
access a service ‘S’ from the network and assume the service is available in node Service Provider Node (SPN). In our proposed scheme, broad casting technique is used for service request from a requester.

![Diagram of SRN, SPN and Representative Nodes in a Pervasive Environment](image)

**Figure 5.1 SRN, SPN and Representative Nodes in a Pervasive Environment**

The service may be available in a node of single hop or node that can be reached in multi hop. In a multi hop broadcast the next hop node will act as an intermediate or representative node. Any node in the network can act as an intermediate node. Network agents are used for sending and receiving the messages. All the devices are set to enable by using the network manager. The services in devices are listed in a temporary base table which must contain the device identification, location and service maintained by each device with the help of the service entry mechanism. The service request message contains the parameters specified in the packet structure discussed in Chapter 3. The request message has its own length and context.

**Service Discovery**

A simple service requesting and responding architecture for a single hop network is shown in Figure 5.2 (a, b, c & d). During the service provisioning
problems may arise such as, service holding node will be in sleep mode or in a unreachable location or the service discovery may be slowed down due to environmental factors.

Figure 5.2 (a) Simple Architecture of Service Request and Response (b) Service offering node in one hop (c) No nearby nodes offering services (d) Representative nodes and (e) Service path is established
Service discovery process creates a direct contact with the nearby service holding devices. When the node is in an unreachable position, the next immediate hop devices can act as the representatives for the service requesting node and they can do the search on behalf of the service requested node. During this time, an efficient routing protocol should be needed to create a reliable and trustful path to keep the link to the best service holder.

In the above situation, the SRN Node1 cannot find any nearby services. The service providing node Node4 is not in reachable location, it needs multiple hops to reach the destination. To solve this problem, Node2 and Node3 become representative nodes of Node1. So, Node2 and Node3 will do the search on behalf of Node1 and they can find a service which is requested by the Node1. To choose a best path, both the nodes have to execute an efficient service routing algorithm. That Routing Protocol can find a reliable and trustful path as shown in Figure 5.2(e).

The procedures used for the service discovery [99] in the pervasive environment are:

1. Service Discovery(Service Req Addr, Context, Location) is the procedure shown in Algorithm 5.1, where the ‘Hi’ message send to find out the desired service named ‘Internet’ in the available devices in the environment. This algorithm is discussed in detailed in the next topics.

2. Service Search(Nodeid, Hop Cnt, Service Type) initiates the searching of the specified service in the network. If the search goes beyond one hop then the intermediate nodes are act as representative device and the neighbour devices detail will be maintained by the other devices in connectivity.
3. *Service_Equals(Node_id, Context)* checks for the desired service in the device available in the network. Context returns the exact service required by the service requester.

4. *Service_Reply(Nod_id,Hop_Cnt)* reply from the service possessing device. It returns the identification number and the hop count to the requester. The list of devices send the acknowledgement will be stored by the requester device.

\[
\text{Algorithm SERVICE\_DISCOVERY()}
\]

**Input:** Service Req Addr, Context, Location

**Output:** List of Nodes with desired service,

Broadcast Message “Hi”

\[
\text{Memory_alloted} \leftarrow \text{Space in the device that can hold the list of nodes with service}
\]

\[
\text{List_of_nodes[memory_alloted]} \leftarrow \text{List of nodes surrounded by}
\]

Broadcast interval_delay \leftarrow \text{time interval to broadcast the ‘Hi’ message}

Counter \leftarrow 0;

For each broadcast_interval_delay do

For each list_of_nodes[memory_alloted] do

Counter \leftarrow \text{counter + 1};

Broadcast Message “Hi” to list_of_nodes[counter];

If (receive_message = “Hi”) then

List_of_replied_nodes[counter] \leftarrow \text{service node IP};

End if

End For

End For

For each list_of_replied_nodes [counter] do

Counter \leftarrow \text{counter + 1}; // Whether the replied node exists or not

Broadcast Message “Hi” to list_of_replied_nodes [counter];

If (receive_message = “EMPTY”) then

List_of_replied_nodes[counter] \leftarrow \text{FREE};

End if

End For

End For

Counter \leftarrow 0;

Call ROUTING_PROTOCOL_SFUSP(list_of_replied_nodes);

End

Algorithm 5.1 Algorithm for Service Discovery
The protocol SHMSDP is developed for finding an absolute resource provider or device by the requester according to the context needed. The requester will broadcast a message, based on a context to find the nearby services. Any node offering the same requested service can send acknowledgement to the requester. A connection will be established between the nodes, and at once the reply message will be received by the service requesting node. If the same service is received from more than one node, according to a comparison criterion, the node with best service will be selected and is discussed in the algorithm for self-elimination. In broadcasting of messages, repetition of messages due to looping occurs. This can be avoided by using a sequence number as like in a proactive routing scheme. Next topic discusses about the loop avoidance.

**Loop Avoidance**

The loop avoidance mechanism is based on the use of sequence numbers, similar to the method proposed in a proactive distance vector routing. Since distance vector routing protocols follow a single path Bellman-Ford algorithm, it is needed to use the sequence numbers in a different way. The main idea is to have all destination nodes issue sequence numbers and spread these together with the routing information, and to let data packets follow routes of increasing sequence numbers or decreasing costs.

The general working of the mechanism as given in [55] is as follows:

1. Each node $i$ maintains a local sequence number $S_i$. It also maintains in each routing table entry $TD_{ij}$ a sequence number $SD_{ij}$, which is the last sequence number received for destination D over next hop j. Finally, it maintains a sequence number $SD_i$ for each destination D, which is the highest sequence number i has forwarded for D in its update messages, and a cost
value \(CD_i\), which is the lowest cost forwarded for \(D\) related to this sequence number.

2. For each periodic routing update message, \(i\) increments the local sequence number \(S_i\) by 1 and includes it in the message. Then, it adds the best routing information for each destination \(D\) to the update message, and adds to this the sequence number \(SD_{ij}\) related to the best next hop \(j\) towards \(D\). Finally, it updates \(SD_i\), its local record of the highest sequence number it has forwarded for destination \(D\), and the associated lowest cost \(CD_i\) as follows: if \(SD_i < SD_{ij}\), \(SD_i\) is set to \(SD_{ij}\) and \(CD_i\) is set to \(CD_{ij}\).

3. A node \(j\) receiving an update message from \(i\) stores the sequence number \(S_i\) in its routing table entry \(T_{ji}\) as \(S_j\), the last sequence number received for destination \(i\) over next hop \(i\). Then, for each destination \(d\) mentioned in the update message, it sets \(SD_{ji}\) to the received sequence number.

4. Data packets arriving in a node \(i\) for a destination \(D\) are only forwarded to a next hop \(j\) if \(SD_{ij} > SD_i\), or if \(SD_{ij} = SD_i\) and \(CD_{ij} \cdot CD_i\).

### 5.4 Clustering Scheme

The clustering scheme is based on location based clustering, where the current location of the device is identified and the devices are grouped under some locations in an area. Clustering helps intra communication among devices in selecting the best one and also reduces packet loss due to traffic in network.

**Location Based clustering**

In MANETs, the movement of the network nodes may quickly change the topology resulting in the increase of the overhead message in topology maintenance. Protocols try to keep the number of nodes in a cluster around a pre-defined threshold to facilitate the optimal operation of the medium access control protocol.
A theoretical foundation for the problem of network localization is needed through which some nodes know their locations and other nodes determine their locations by measuring the distances to their neighbours. The data gathered from LBS can be given to the clustering algorithm given in Algorithm 5.2 in partitioning the area where network spreads.

The procedures to get the current location, changed location and servicing location using GPS log are referred from [104] and are:

1. *Get_System_Service*(Context, Location) returns the location of the device which possess the desire service in it.

2. *Show_Current_Location*(LocationManager) the current location of the device will be retrieved from this procedure since for every fraction of seconds the node is under mobility in the environment.

3. *LocationListener*(return ChangedLocation) the change in the location will be notified by this procedure in order to get the current location.

4. *Clustering (GPS Coordinates, device_id)* procedure groups the devices with the desired service in the environment under location. The detailed working is given next.

**Grouping Devices under Locations**

After the eligible nodes are selected, they are grouped by locations by using the clustering algorithm, where no node will be excluded from any of the available clusters. Grouping of nodes in the cluster under an area is done through the algorithm 5.2 which gets input from the GPS system procedures. If the area increases, number of clusters also increases accordingly.
Algorithm CLUSTERING()

Input: - X and Y co-ordinates received from GPS
Output: - Area for each cluster and device

Ci ← 1 (Initialize);
Node[i] ← list of nodes in the area;
Area[i] ← Area ID // Get the area list from the GPS data;

For each Area[i] do
    For each node[i] do
        NodeClusterID ← Ci;
        NodeAreaID ← Area i
    End For
    Ci ← Ci + 1 ; // when each area increases the cluster also increased by one
End For

Algorithm 5.2 Algorithm for Clustering

Clustering enables the comparison of better strengthened node (with respect to service, bandwidth and energy) in that area. In MANETs, nodes often change their native cluster due to the mobility and so get reduced in the efficiency of communication among source and destination. The comparison of the nodes may be done among themselves by using a technique called self-elimination which is given next.

Now the time taken for individual node and all clusters are calculated by their data length, transmission rate and their delay values. Time taken for single node simulation for any nodes in the network can be calculated as follows:

\[
\left( \frac{Data\ Length}{Transmission\ Rate} \right) \times Transmission\ Delay \\
\cdots \cdots (5.1)
\]

where DataLength is the size of the service context.
Cluster simulation time can be calculated as the total number of clusters available in the network with the data length to the transmission rate of each node in the cluster multiplied with the transmission rate. It is calculated as follows:

\[
\left( \frac{\text{Total Number of Nodes in a cluster} \times \text{Data Length}}{\sum_{i=1}^{n} \text{Transmission Rate}_i} \right) \times \text{Transmission Delay} \quad \ldots \quad (5.2)
\]

The node that requests service and the node that provides service are denoted as SRN and SPN respectively. Other nodes may become representative nodes in the routing path or nodes in the network. Figure 5.3 shows clustering of devices.

Cluster optimization is a vital one during clustering of devices under a location since it helps in selecting optimal number of nodes in the routing path. The
optimization technique purely depends on the size of area and the number of
clusters formed by the area separation based on the data logged by the GPS
trajectory. It is found that for k-hop networks, the cluster size can be quasi-circular
in size to manage the cluster efficiently by not leaving any nodes under a cluster
area. Next topics describes about the novel routing scheme which is proposed in this
research work named SFUSP, which consists of two parts, one is self-elimination and
the other is path fixing after self-elimination.

5.5 SFUSP – The Routing Scheme

SFUSP is the acronym of Self-eliminating Fault-tolerant Uninterrupted Service
switching Protocol. This is a routing scheme which is developed by analyzing the
various issues and challenges in routing [48] like medium control access, best
service providers in the route, quality of routes and route maintenance. The route
maintenance is a process which needs route recovery after a fault or delink
occurrence is discussed in Chapter 6. The next topics describes the functioning of
the proposed routing scheme which is specially developed for routing under
pervasive environment.

Proactive Routing

The routing schemes is based on of proactive distance based routing scheme
added with functionalities that enhance the efficiency of the routing during the
device communication. The general working of Proactive Algorithm [56] is:

1. At periodic intervals, node i broadcasts an update message containing for
maximum m of the destinations in its routing table the best routing
estimate $CD_i = \min_j 2N_i (CD_{ij})$ taken over all the possible next hops j in its
set of neighbours $N_i$. m is a constant that limits the size of the update
messages. If there are less than M destinations in the network, all of them
are included in each update message. Otherwise, update messages include subsets of M destinations in a round-robin fashion.

2. Each of the neighbours K of node i receives the update message, and uses it to calculate for each of the destinations D its own new estimate CDK for the cost of the route over next hop i to D. It does so by adding the estimate CDi received from i to the estimated cost of going from k to i. If i indicated in the update message does not have any valid route to D, D would set CDK to infinity.

The route table entry is discussed in the later topic where a temporary route table is maintained by each device at the time of broadcasting message.

**Self-elimination**

SFUSP is Self-eliminating Fault-tolerant based Uninterrupted reliable Service switching mobile Protocol, is a protocol specially designed for pervasive computing environment for MANETs. It is basically a proactive protocol with additional functionalities added such as clustering and self-elimination. It is a well supported context-aware and fault-tolerant service discovery routing protocol. SFUSP is reliable while searching the exact service offering node and thus, it reduces the searching time and balance load among the nodes which are all involved in the process of discovery in pervasive networks. SFUSP follows a new technique to eliminate less strengthened nodes (with respect to Service, Bandwidth and Energy) which get connected during the path discovery process. The additional task of refreshing the existing list of nodes will be reduced, if the unwanted nodes get eliminated in the time of path discovery itself. SFUSP will keep only the best selected node details in its database.

SFUSP is as intelligent as a reliable MANET routing scheme for any pervasive environment which can along with any service discovery protocol, can help in
providing good service provisioning. LBSs can be used along with this scheme for the unfavourable situations to find the previous servicing location details and a new link can be established, if the service gets interrupted due to non-availability of nodes nearby the link. The routing scheme contains three main functionalities such as clustering, self-elimination and routing path establishment and the diagrammatic representation is in Figure 5.4.

The steps involved under SFUSP routing scheme which includes the processes initialized from the service assignment are as follows:

Step 1: Broadcast the message to discover the node which is requested by the requester node.

Step 2: If unfound then representative nodes will be activated to generate a new search in their nearby locations.

Step 3: If requester node is found then

Step 4: Group all the representatives by location.

**Algorithm SELF_ELIMINATION ( )**

**Input**: IP of requesting and providing nodes  
**Output**: Check the enabling status of the device

```java
ServiceRequestNode; // Get IP of the service requesting Node  
ServiceProvidingNode; // Get IP of the service providing Node  
Cluster_id[i]; // list of all clusters in the area

For each cluster_id do
    Cluster_nodes[i]; // list of all nodes in the cluster
    For each cluster_nodes do
        If (me.ServiceRequestNode=NULL) then // if down node is null then the node can be deleted
            Me.ServiceProvidingNode=NULL or DISABLED;
        End If
    End For
    Call PATH_FIXING() 
End For
END
```

**Algorithm 5.3 Algorithm for Self-elimination**
Step 5: Find the most reliable representative.

Step 6: Do the self-elimination technique to reduce to best suited nodes in order to find the most reliable path.

Step 7: Establish the reliable path with the best strengthened nodes in the path.

The broadcasting and clustering procedures are seen in the earlier topics and the procedures related to self-elimination are given in Algorithm 5.3 and 5.4. Now the procedure for path fixing for selecting efficient devices to form efficient devices in the routing is described.

**Path Fixing with Efficient Devices**

The highlight of the SFUSP routing scheme is the self-elimination procedure that helps in finding the most strengthened node in the cluster. Along with the proactive nature of the scheme, the self-elimination process finds effective node in the cluster for establishing reliable links on the basis of the strength of the nodes. The strength may be calculated with the context specified by the requester and the packet structure is given in the next section.

In the self-elimination process, each node will collect the information about their service providing and service requesting nodes ie. Up and Down nodes. Then, the nodes which have not participated in the routing process in the list will be deleted first and this will be repeated until a reliable path is established. The nodes which are discovered with the requested services are selected and some of nodes may act as intermediate or representative nodes. The intermediate nodes are the nearby nodes or nodes in the coverage area. In Figure 5.3, it is shown that all such nodes are clustered on the basis of location. Each node is represented by a value which indicates the strength of the node on the basis of bandwidth, energy and
Figure 5.4: (a) SFUSP Cluster Linking (b) SFUSP Elimination Session starts (c, d, e & f) SFUSP Elimination Sessions and (g) Path for Service Provisioning is fixed
service. All the location wise identified clusters are linked together and so, a location may contain ‘n’ clusters. From which the best suited node in the cluster will be selected by the self-elimination process.

In the first session of elimination, all the grouped clusters, unwanted or low strength nodes will be eliminated. In each group, a comparison will be done to find the most strengthened node to establish a reliable path. Figure 5.4(a) illustrates the cluster linking and the start of elimination session. In the elimination session, in Figure 5.4 (b, c, d, e & f) all the weak and unreliable nodes get eliminated from the reliable link. By this, one can get an exact and reliable path to continue with the communication. The complete routing path fixing using self-elimination technique is coded in the Algorithm 5.4.

```
Algorithm PATH_FIXING()

Input: - Status of the devices in the cluster
Output: - Neighbour node of each device based on strength

Cluster_nodes[i] ← all node ids in the cluster;
Me_node_id ← service providing node id;
THRESHOLD ← n;
DownNode(x) ← Service request node ids;
Up_Node← original first service request node id;
For each cluster_nodes do
  If (cluster_nodes[x].DownNode(x) ≠ EMPTY) then
    { 
      If (cluster_nodes[x].STRENGTH ≥ THRESHOLD) then
        { 
          Cluster_nodes[x]← "ALIVE";   //CONTEXT SERVICE FOUND
          If (cluster_nodes[x].CONTEXT= me_node_idCONTEXT) then
            //check the context
            { 
              If (SERVICE REQUEST NODE ≠ NULL) then
                { 
                  SEND MESSAGE SERVICE REQUEST NODE (Up Node) “CHOSEN” ;
                }
            } 
          End If
        } 
      } 
    } 
  If (Up_Node ≠ NULL) then  //to check the node reached the original first location
```

{  
  SEND MESSAGE UP NODE "PATH FIXED";
  }
  
End If
}

Else // if context not matched and threshold not satisfied
{
  Cluster_nodes[x] ← REPRESENTATIVE NODE;
}

End If
}

Else //if the node has no strength means it can be in a disabled stage
{
  Cluster_nodes[x]← "DISABLED";
}

End If
}
else
{
  Cluster_nodes[x]← "DISABLED";
}

End If //if the path is fixed and the node is not selected it will disable by itself
If (cluster_nodes[x].MESSAGE = "PATH FIXED") then
{
  If (cluster_nodes[x] ≠ "CHOSEN") then
  {
    Cluster_nodes[x] ← "DISABLED";
  }
  End If
}
End If
End For
END

Algorithm 5.4 Algorithm for Routing Path Fixing by SFUSP

This is an intelligent elimination technique, where most reliable path can be set by removing the less reliable nodes which still get connected after the session of elimination. Here each node has to generate a list of Upper Node (UpN) and Down Node (DnN) detail from their connection establishment. After generating this list, the nodes which have no Down Node (DnN) can be voluntarily get off from the connection establishment. Both the elimination and clustering processes are done
without the knowledge of the service requesting node. Finally, the most reliable path for routing in the network will be established as in the Figure 5.4(g). The communication will be done through this path. The path will get often changed in the mobile environment and so the process is repeated at once with any node that senses the nearby node that is left or a new node that gets entered in the cluster or un-necessary nodes in the link. Simulation time of non-eliminated nodes in a cluster can be calculated as the total number of current nodes present in the cluster with the sum of the node data length to the sum of the transmission rate of all nodes multiplied with transmission delay. The following formulae give the time calculation for self elimination with respect to clusters.

The non-eliminated nodes in the cluster can be calculated as follows:

\[
\text{Non-Eliminated Nodes in Cluster} = \frac{\sum_{i=0}^{\text{Total Number of Current Nodes in a Cluster}} \text{Data Length}}{\sum_{i=0}^{\text{Transmission Rate}} \text{Transmission Delay}} 
\]

\[ \ldots \ldots (5.3) \]

Now, one can calculate the self-elimination time of a node in the cluster by the following formula:

\[ \text{Cluster Simulation Time} - \text{Non-Eliminated Nodes Simulation Time of a Cluster} \]

\[ \ldots \ldots (5.4) \]

The following Algorithm 5.5 illustrates the whole routing process which involves clustering, self-elimination and path fixing of the SFUSP scheme. Since the self-elimination takes place cluster wise and since the cluster size is selected by an localization technique [31], an optimal number of best service providing or
representative nodes fall in the routing path. The routing process comprises of the clustering followed by the self-elimination technique in each cluster.

The aim is to find a better path establishment time with minimum number of effective nodes in the service path which is established in a less number of times.

\[
\sum_{i=0}^{n} \text{Cluster Simulation Time} \quad \ldots \quad (5.5)
\]

**Route Table Maintenance**

The elements in the routing table of each mobile node change dynamically to keep consistency with dynamically changing topology of an Adhoc network. To reach this consistency, the routing information advertisement must be frequent or quick enough to ensure that each mobile node can almost locate all the other mobile
nodes in the dynamic Adhoc network. Upon the updated routing information, each node has to relay data packet to other nodes upon request in the dynamically created Adhoc network. The procedure in route table maintenance [55] is:

1. Each node \( i \) in the network maintains a routing table \( T_{ij} \), which has an entry \( TD_{ij} \) for each known destination \( D \) and each available next hop \( j \). \( TD_{ij} \) contains an estimate \( CD_{ij} \) of the cost of the route towards destination \( D \) going over next hop \( j \).

In the routing information updating process, the original node tags each update packet with a sequence number to distinguish stale updates from the new one. The sequence number is a monotonically increasing number that uniquely identifies each update from a given node. As a result, if a node receives an update from another node, the sequence number must be equal or greater than the sequence number of the corresponding node already in the routing table, or else the newly received routing information in the update packet will be old and should be discarded. If the sequence number of one node in the newly received routing information update packet is same as the corresponding sequence number in the routing table, then the metric will be compared and the route with the smallest metric will be used.

5.6 Integrated Service Discovery and Routing Scheme

The integrated service discovery and routing procedure is shown in Algorithm 5.6, where all the processes are integrated to perform an effective service discovery and routing mechanisms in a single instant.

Integrated service discovery and routing is the main objective of this thesis in order to attain an efficient routing in pervasive environment with the above mentioned processes. Efficient devices with good bandwidth, energy and service with desired context has been selected for communication and the above algorithms
ensure the better processing for the creation of routing path in pervasive environment in a less computing time.

![Algorithm 5.6 Algorithm for Integrated Service Discovery and Routing](image)

Next topic discusses the performance evaluation for integrated service discovery and routing by analyzing the various routing metrics.

5.7 Performance Evaluation

Metrics for Routing

The various routing metrics that are to be evolved to evaluate the performance of the proposed routing scheme are:

1. Average End-to-end Delay includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC, and propagation and transfer time. This metric determines how fast the algorithm delivers the data packets to the destination.

2. Normalized Routing Overhead is the total number of routing packets transmitted during the simulation. For packets sent over multiple hops, each transmission of the packet (each hop) counts as one transmission. This metric is also important as it measures the scalability of the protocol, the degree to which the protocol can function in congested or low-bandwidth environments and also shows its efficiency in terms of consuming node battery power.
3. Packet Delivery Ratio can be defined as the ratio of number of data packets delivered to the destination with respect to the number of data packets generated by the traffic (CBR with TCP/UDP) source. It is also known as ‘throughput’ of the routing algorithm. This metric is important as it describes the loss rate as seen by the protocol, which in turn affects the maximum throughput that a network can support. This metric characterizes both the correctness and completeness of the routing protocol.

4. Routing Path Establishment Time is the time taken by the routing scheme to perform all the routing operations to establish a service provisioning path with best devices in the routing path.

**Metrics for Integrated Service Discovery and Routing**

The various metrics to find the efficiency of the routing scheme for the integration of service discovery and routing are,

1. Integrated Service Discovery and Routing Time is the time taken by the routing scheme to perform all the operations including broadcasting, clustering, service discovery and routing for establishing a reliable routing path for service provisioning.

2. Optimal Devices in the Routing Path gives the number of efficient devices in the routing path to perform service provisioning after the path establishment. The optimal number of devices in the routing path depends upon the clusters and the efficient nodes eligible for service provisioning.

**5.8 Network Parameters**

Table 5.1 lists the possible network parameters used for simulation under various conditions in order to get a better result. The simulation is done for number of times in order to get the average value.
Table 5.1 Network Parameters for Integrated Service Discovery and Routing

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Area</td>
<td>1500m x 1500 m.</td>
</tr>
<tr>
<td>Channel Type</td>
<td>Wired and Wireless.</td>
</tr>
<tr>
<td>Propagation Model</td>
<td>Two Way Ground.</td>
</tr>
<tr>
<td>Radio Range</td>
<td>200 m.</td>
</tr>
<tr>
<td>Radio Delay</td>
<td>10 milli secs.</td>
</tr>
<tr>
<td>Traffic Type</td>
<td>CBR with TCP and UDP.</td>
</tr>
<tr>
<td>Duration</td>
<td>100 Seconds.</td>
</tr>
<tr>
<td>MAC Layer</td>
<td>IEEE 802.3, 802.11, 802.15 and 802.16.</td>
</tr>
<tr>
<td>Routing Protocols</td>
<td>SFUSP, AODV, DSR and OLSR.</td>
</tr>
<tr>
<td>Mobility</td>
<td>Random Waypoint, Brownian Motion and Manhattan Model.</td>
</tr>
<tr>
<td>Node Strength</td>
<td>Energy, Bandwidth and Service Context.</td>
</tr>
<tr>
<td>Context</td>
<td>Contexts such as internet, disk, printer, games etc.</td>
</tr>
<tr>
<td></td>
<td>Arbitrarily assigned to all nodes.</td>
</tr>
<tr>
<td>Devices</td>
<td>Mobile devices, stationary devices etc.</td>
</tr>
<tr>
<td>No. of Nodes</td>
<td>50, 100, 200 and 400 with Fixed and Mobility nature. Nodes assigned with different MAC layers.</td>
</tr>
<tr>
<td>Speed</td>
<td>25 m/s with a pause time of 10 milli secs. For Manhattan Model minimum speed of 25 m/s to a maximum speed of 100 m/s.</td>
</tr>
<tr>
<td>Transmission rate</td>
<td>9.6 Kbps.</td>
</tr>
<tr>
<td>Data Payload</td>
<td>512 bytes.</td>
</tr>
<tr>
<td>Traffic Load</td>
<td>Packet Sent in Every 1 second.</td>
</tr>
</tbody>
</table>

The network parameters specified in the above table are used for integrated service discovery and routing. The tools used for the simulation contains libraries from NS2, NS-Miracle, PDP, GPS and Java in Linux Fedora 12 operating systems.
5.9 Result Analysis

Simulation Criteria

The simulation is done with the device density of 50, 100, 200 and 400 nodes in different times, assigning different mobility patterns such as Fixed, Random, Brownian and Manhattan. It is assumed that there are N nodes for the simulation with 20 percentage of the nodes are set to fixed network in that 50 percentage of the nodes are set to WLAN, 30 percentage of nodes are set to WPAN and 20 percentage of nodes are set to WWAN. Among the 40 percentage of nodes set to random mobility the above said proportions of Wireless MAC connectivity is set to the nodes. In the other 20 percentage of nodes that are set as Brownian, the same ratio of MAC is assigned and in the remaining 20 percentage of nodes which are assigned to Manhattan Mobility, the same ratio of MAC layer is used. The simulation is performed for n number of simulations for the same network parameter values and the trace values are used for the calculation of the metrics.

Pervasive Discovery Protocol (PDP) is used to create a pervasive environment for the simulation purpose. The devices are assigned with service, bandwidth and energy. The strength of the device is estimated by calculating the weight of all these parameters. Some of the procedures of PDP are used for the service discovery process and are described in this chapter. SHMSDP is used to broadcast the request in the network and the list of devices with desired service will be stored in the requester. Clustering based on a location is done using GPS to locate the device and group the devices under a location. SFUSP helps in finding out the best strength device in each location for establishing a reliable and efficient path using a proactive nature of routing protocols. Integration of service discovery and routing helps to avoid flooding and traffic in network by handling both the processes involved at the same time in finding the service and establishing a routing path.
## Simulation Result

Table 5.2 Average Simulation Values of Integrated Service Discovery and Routing

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Number of Nodes</th>
<th>SFUSP</th>
<th>Reactive Distance Vector Protocols</th>
<th>Proactive Link State Protocols</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>AODV</td>
<td>DSR</td>
</tr>
<tr>
<td>Average End-to-end Delay</td>
<td>50</td>
<td>1.2</td>
<td>1.6</td>
<td>1.5</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1.7</td>
<td>1.8</td>
<td>1.7</td>
</tr>
<tr>
<td>(in Milli Seconds)</td>
<td>200</td>
<td>2.7</td>
<td>3.6</td>
<td>3.6</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>2.8</td>
<td>4.2</td>
<td>4.3</td>
</tr>
<tr>
<td>Normalized Routing Overhead</td>
<td>50</td>
<td>356</td>
<td>278</td>
<td>611</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>609</td>
<td>1488</td>
<td>851</td>
</tr>
<tr>
<td>(in Packets)</td>
<td>200</td>
<td>1172</td>
<td>3976</td>
<td>2261</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>1880</td>
<td>4372</td>
<td>4223</td>
</tr>
<tr>
<td>Packet Delivery Ratio</td>
<td>50</td>
<td>.99</td>
<td>.97</td>
<td>.94</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>.9</td>
<td>.93</td>
<td>.89</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>.93</td>
<td>.92</td>
<td>.87</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>.93</td>
<td>.89</td>
<td>.86</td>
</tr>
<tr>
<td>Routing Path Establishment Time</td>
<td>50</td>
<td>10.96</td>
<td>11.43</td>
<td>10.99</td>
</tr>
<tr>
<td>(in Seconds)</td>
<td>100</td>
<td>11.27</td>
<td>13.23</td>
<td>13.75</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>11.72</td>
<td>15.32</td>
<td>15.03</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>12.65</td>
<td>17.77</td>
<td>19.75</td>
</tr>
<tr>
<td>Integrated Service Discovery and Routing Path Establishment Time (in Seconds)</td>
<td>50</td>
<td>12.25</td>
<td>13.76</td>
<td>13.02</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>12.66</td>
<td>17.98</td>
<td>18.81</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>13.07</td>
<td>21.37</td>
<td>23.94</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>14.46</td>
<td>23.25</td>
<td>25.08</td>
</tr>
<tr>
<td>Optimal Nodes in Routing Path (in Nodes)</td>
<td>50</td>
<td>4.25</td>
<td>8.03</td>
<td>7.8</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>7.23</td>
<td>14.83</td>
<td>10.25</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>8.1</td>
<td>17.83</td>
<td>15.9</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>10.55</td>
<td>21.6</td>
<td>19.55</td>
</tr>
</tbody>
</table>
The trace files obtained during simulation are processed by using Linux commands, C and Java Programs to find out the average result for the various metrics and are shown in Table 5.2.

The analysis shown in the graph is done by comparing SFUSP Scheme with reactive Distance Vector Routing Protocols and Proactive Link State Protocol.

**Average End-to-end Delay**

![Graph 5.1 Average End-to-end Delay - SFUSP Vs Reactive Distance Vector Routing Protocols](image1)

![Graph 5.2 Average End-to-end Delay - SFUSP Vs Proactive Link State Routing Protocol](image2)
Graph 5.1 shows the performance of SFUSP on Average End-to-end Delay with the reactive distance vector protocols such as AODV and DSR. Graph 5.2 shows the performance of SFUSP against the proactive link state protocol such as OLSR.

Normalized Routing Overhead

Graph 5.3 Normalized Routing Overhead - SFUSP Vs Reactive Distance Vector Routing Protocols.

Graph 5.4 Normalized Routing Overhead - SFUSP Vs Proactive Link State Routing Protocol

Graph 5.3 shows the performance of SFUSP on Normalized Routing Overhead with the reactive distance vector protocols such as AODV and DSR. Graph 5.4 shows the performance of SFUSP against the proactive link state protocol such as OLSR.
Packet Delivery Ratio

Graph 5.5 shows the performance of SFUSP on Packet Delivery Ratio with the reactive distance vector protocols such as AODV and DSR. Graph 5.6 shows the performance of SFUSP with the proactive link state protocols such as OLSR.
Routing Path Establishment Time

Graph 5.7 shows the performance of SFUSP on average routing path establishment time with the reactive distance vector protocols such as AODV and DSR. Graph 5.8 shows the performance of SFUSP with the proactive link state protocols such as OLSR.
Integrated Service Discovery and Routing Time

Graph 5.9 shows the performance of SFUSP on average Integrated Service Discovery and Routing Time with the reactive distance vector protocols such as AODV and DSR. Graph 5.10 shows the performance of SFUSP with the proactive link state protocols such as OLSR.
Optimal Nodes in Routing Path

Graph 5.11 shows the performance of SFUSP on Optimal Nodes in the routing path with the reactive distance vector protocols such as AODV and DSR. Graph 5.12 shows the performance of SFUSP with the proactive link state protocols such as OLSR.
5.10 Observations

The analysis of Average End-to-end Delay of SFUSP with other protocols is discussed in this section. The proposed routing scheme is performing better than the other existing protocols because of the following reasons:

1. In SFUSP, request by a sender is done by broadcasting and also it is done in a proactive way where, a temporary table is maintained with the neighbor node identity. Normally the reactive protocols use unicasting in nature and it may lead to looping of message and not sending acknowledgement makes the bandwidth busy and the message cannot be propagated in a speedy way. Other hand, on demand protocols like AODV and DSR need a time lapse to search for the neighbour nodes. This creates a cumulative delay in delivering the data among all nodes in the network. Thus in scalable networks, average end-to-end delay becomes higher. The link state protocols such as OLSR are table driven protocols and since they maintain the permanent table for maintaining node details it takes much time for the table updations. These are the reasons why SFUSP is performing better than the other protocols without lesser delay in message passing.

2. Clustering in SFUSP is based on area wise and is done for the total area which falls on the whole network. The message passing is controlled by the cluster boundary nodes since they know about which cluster the node belongs. Since both reactive and proactive routing protocols available widely for MANET do not use the clustering concept, there may be a possibility of occurrence of a huge delay in passing the packets.

3. SFUSP maintains a loop avoidance scheme by using a sequence number (since it follows the proactive routing approach) while broadcasting, a
single node gets the message once in a cycle. This also helps in reducing the average end-to-end delay while passing the packets.

In SFUSP, the normalized routing overhead with respect to the scalability of network is lesser when compared to other protocols because:

1. Most probably the normalized routing overhead is depending on the amount of control packets sent at the initial time of routing process. If the size of the control packet is less, the normalized routing overhead becomes low. Since the average end-to-end delay of the SFUSP is less in compare with other protocols, by default the normalized routing overhead seems low for the proposed scheme.

2. The bandwidth utilization is good in SFUSP because of the loop avoidance mechanism used. Once the loop avoidance takes place among the nodes while sending the messages, the bandwidth utilization will not exceed causing lesser routing overhead.

3. SFUSP uses a partial routing table which is not needed to be updated often as it happens in distance vector routing protocols. This helps in minimizing the generation of control packets.

4. Since the node communication is done locally in the cluster in SFUSP scheme, redundant generation of control packets can be minimized.

Packet Delivery Ratio is higher for SFUSP in comparing with other existing routing schemes because of the following reasons:

1. The Packet Delivery ratio depends on the type of message passing scheme in the network. Normally the routing protocols use Unicast or Multicast for message passing. SFUSP uses the broadcasting scheme where the request is passed from the requester to the one hop neighbour. Then the neighbour node acts as a relay to propagate the same message to its
neighbours. Also the list of the neighbours is available in the corresponding requesting or representative nodes table list. This is not the case with the other reactive or proactive protocols. For the high node density SFUSP maintains a constant performance unlike others which fails in maintaining a constant delivery ratio.

2. Since the average end-to-end delay lesser than the other routing protocols by default the proposed routing scheme is better than other protocols.

3. Since the control packet generation is less in comparing with other existing protocols as already mentioned, with lesser delay and better bandwidth utilization, the packet delivery ratio can reach nearer to one for the proposed routing scheme.

For routing path establishment time, SFUSP shows a lesser variation as increase in time in comparing with other protocols. The reasons are:

1. SFUSP scheme contains procedure for efficient broadcasting of messages and location based clustering. Not only of its proactive routing characteristics, had it also based on reactive type of routing. It maintains a temporary routing table with less updating time interval. It selects efficient nodes in the routing path with minimal number of nodes based on the clustering. Instead of unpredictable number of nodes found in the routing path, SFUSP selects a minimal number of nodes with efficient service provisioning capability. That is why it takes lesser time in establishing a routing path.

2. Although the node density increases, proportionally SFUSP maintains the information about the neighbour nodes using their intermediate nodes. For other existing protocols, the change in number of nodes in the network needs to be considered as a new topology as a result of it, they
have to initialize their operations from the scratch. This will take much
time to create a routing path from the requester to the service provider.
These are the reasons why SFUSP is performing better for path
establishment than other existing protocols.

Time taken for Integrated Service Discovery and Routing is less in SFUSP routing
scheme in comparing with other existing protocols because:

1. Service assignment and request schemes pertaining to SFUSP is efficient
   in comparing with other protocols since it is meant for a pervasive
   environment. The support of MAC layer interpretation of devices and their
calculations on the strength of the devices helps in finding a better routing
   path integrated with service discovery in a minimal duration.

2. The efficient broadcasting scheme used by SFUSP helps in lessening the
   routing overhead as it is mentioned earlier, and thus it takes less time for
   increasing number of nodes comparing to other existing protocols.

3. The clustering is not supported by other existing protocols for message
   passing or grouping of devices. The selection of best devices is done using
   clustering and it is done in an efficient way by the proposed routing
   scheme. Also the communication will be entertained always among the
   nodes only through the clusters. Because of these reasons an efficient
   service path among best serving nodes is created in a lesser time.

4. The proactive routing characteristics added with self-elimination
   technique in SFUSP help to find the service as earliest while broadcasting
   itself since the service discovery and routing are integrated as one. This
   helps in reduce the time of routing path creation.

For Optimal number of nodes in the routing path, SFUSP shows a better result in
comparing to other protocols:
1. SFUSP uses cluster based node grouping where the efficient nodes will be selected from the cluster itself and so the number of nodes selected will be proportional to the number of clusters formed. The area partitioning will be always optimal if the GPS data is used for clustering. Unlike SFUSP, other protocols do not rely on clustering and so there may be a chance of many number of nodes in nearby area may be selected for the participation in communication. This leads to a uncountable number of nodes in the routing path if the density of the network increases in the case of other existing protocols.

2. The concept of selecting efficient node in the routing path by SFUSP betters the communication with known MAC and Mobility of the nodes. Other protocols do not support this type of node selection since they are meant only for homogenous networks.

3. For increasing in the density of nodes in the network, SFUSP applies the same procedure for the selection of eligible nodes in the service path. This is not the case with other existing protocols, where they have to repeat the process for each and every node that enters in the network. This leads to increase the number of nodes in the routing table higher than the actual participating nodes in the routing process.

5.11 Conclusion

Thus, SFUSP shows better performance for all routing and service discovery metrics in compared with other reactive and proactive protocols in all normal situations. From the above discussion it is proved that SFUSP can be a suitable routing scheme for resource discovery and dissemination in pervasive environment and can be applied for supporting applications in that environment.