Service Discovery Mechanism

Service discovery is a process of mapping a service description to a service location. It normally occurs in a client-server environment where a client requests a service and waits at a port and the server responds to the service. Because there are multiple instances of service match there will be multiple matching. If multiple mappings are found then the client can choose the best server based on the shortest path between the sender and the receiver. A common language is used for different service provider to provide service. XML is the standard language used for service discovery across heterogeneous protocols and platforms [35].

Service discovery is either a centralized one or a distributed one. The centralized service discovery maintains a server that register all the services and the client request the service from the server. In the distributed service discovery mechanism the services are distributed among all the nodes in the network and are broadcasted or multicasted among all the nodes. In the distributed approach the discovery of service is done by push and pull mode. In case of the push mode the server pushes the information to all the clients and in the case of pull mode the client send request to the server. Central service discovery totally depends on the server and if the server fails, the entire system fails. It mostly follows a unicast method of service discovery. When the network is partitioned it creates a heavy problem in the central
service discovery. Moreover it requires manual configuration which is highly time consuming whereas the distributed approach depends on multicasting and broadcasting which incurs heavy overhead. Rediscovery in this type of method could be either server driven or client driven. One method highlighted in the existing work for service discovery is pinging [35]. Pinging traverses through the entire server and generates the shortest path among all the servers. The cost of service discovery highly depends on the routing protocol used. As the pinging involves traversing through all the server, the cost and time incurred in search is definitely very high. Message based middleware with context aware routing is taken for comparison as the routing is completely context based and need not traverse the entire server. And the time and cost factor is definitely lesser than the pinging technique thereby message oriented middleware with CAR protocol is taken for comparison with the proposed work. Thus in our proposed work we have interoperated the service discovery protocol along with the routing protocol to reduce cost to a minimum extent by cross-layer approach which excels the centralized and distributed approaches [35].

A good service discovery mechanism for pervasive environment must consider highly changing nature. It must consider the power constraints of the devices and exploit cooperation between devices to achieve best performance. Service description mechanism should provide whatever service available in the environment. Moreover it should be integrated with service access for compatibility in forward and backward directions. Service discovery in dynamic environment is highly challenging. There are various service discoveries in fixed environment like SLP, JINI, and UPnP but in dynamic environment it
is difficult. Accepted method of service discovery in pervasive environment is the push-pull method [26].

6.1 Cross Layer Support to Service Discovery

In already existing cross-layer approach of service discovery, library is maintained to record all the service related information. Routing layer driver is to maintain all the routing information. SDL’s main data structure is to store all service related information in the service table. RLD stores all the available routes to the destination. Discovery in the already existing work is either client driven or server driven. Active service discovery is client driven and passive service discovery is server driven. In case of active discovery the server waits for the client to give back the reply whereas in case of passive service discovery the server request the client to advertise the service continuously. In the already existing work, cross layering is established using client server architecture whereas in the proposed work cross layering is established in the peer to peer architecture [35]. Peer to Peer paradigm provides an environment for adhoc networks. Service discovery protocols are used to discover services. Services could be in any of the space. These spaces are brought together in to a homogenous network by interoperability.

In the proposed work discovery tables are not maintained for service discovery. The routing and service discovery protocols are selected seamlessly by the sensors depending on the profile and the proximity of the user from the service offering node. The shortest path between the sender and receiver is traced by the utility function. Based on the utility function the sensor routes the personalized profile to the service matching peer in the cluster. The reselection and rediscovery of path...
in case of any route break is done seamlessly within the underlying protocol by means of intelligent routing.

**Service Location Protocol:** Existing SLP provides three Agents: user agent that works on behalf of the client, service agent that advertises the service based on location and attribute. Directory agent that stores information about the services announced by the service agent. SLP provides two type of operation – one when DA is present, the request reply unicast communication exist between DA’s and UA’s. When there is no DA, the UA’s repeatedly multicast their request. SA’s listen to the multicast of UA’s and unicast back the response to UA [41].

The Service Location Protocol in our proposed work involves the discovery of peer node in a cluster without any centralized server. The nodes are free to move across in the smart space. Each node offers particular service to the user. Service Location Protocol is used to find the location of the service offering node in the cluster. If the service offering node has moved to a different cluster, the interoperability feature helps to deliver the best matching service to the user. In our proposed work there are no directories maintained either by the user or the service provider. The search is completely seamless with the help of sensors.

**Pervasive Discovery Protocol:** Existing PDP is used for low power, short range wireless networks and is composed of devices with limited transmission power. The service discovery protocol use centralized server that listens to broadcast or multicast of available services. One of the objectives of PDP is to minimize power. The amount of transmission necessary to discover service is reduced, service announcement are broadcast to all devices in the network, which will know the service simultaneously without having to query
for it. In pervasive discovery environment there are two means of communication of services - pervasive request and pervasive reply. PDP request is send to the service discovery agent. The request could be for a particular service or a group of services. When PDP request is received the SDA is checked for the service. Time span is being sent out and the reply is sent back within the speculated time. When multiple services match, the most appropriate service is offered to the user. The remaining services are aborted [26].

In proposed work we have designed PDP in adhoc environments. Pervasive discovery is used to solve the problem of power usage, reduction of memory footprint and processing power. Pervasive Discovery Protocol does not depend on a centralized server. In Pervasive Discovery Protocol the service is announced only when it is being requested. The Pervasive Discovery Protocol is mainly used in the proposed work to offer seamless service to the user. It is used to bring interoperability among heterogeneous smart spaces i.e. between adhoc, wired and sensor enabled spaces. Pervasive Discovery Protocol in our work does not depend on the specific characteristic of the space. The profile of the user is seamlessly routed in to the smart space. Based on the interoperability factor, the exact matching services are retrieved by the protocol and given to the user.

**Universal Plug and Play Protocol:** It is independent of discovery architecture to connect services in an unmanaged (or) adhoc networks. Internet protocol plays an important role in the architecture. Devices advertise the service at the control point. The control Point listens to the service that matches the requirement. After discovering the service, the control Point downloads the description of device and
finally SOAP protocol is used to control services. The discovery architecture of UPnP is multicasting and requires no central component. UPnP is well suited for mobile computing environment as the information is highly ad hoc. The members of UPnP forum can create a means to easily connect device and simplify the implementation of networks. UPnP architecture consists of TCP/IP communication channel and web services that are used for transferring data among multiple devices. These devices describe their capabilities and features when they join the network using the UPnP protocols. When other devices join the network they can receive the description of existing devices and they can use them without a complex configuration [40].

The Universal Plug and Play Protocol is used in our proposed work for the publisher to publish the service at the header node and the subscriber to subscribe for the service by passing a request seamlessly to the header node. Any node that offers service within the cluster publishes the service at the header and the node that consumes the service listens to the publishing node during its motion. The moving node seamlessly interacts with the service provider. Here there is no control point as in the already existing work. The peer nodes themselves act as a control point in offering services. These nodes have links to the cluster head which acts as an overall controller of all the nodes under it. The cluster head maintains the details of the nodes within the cluster and of the nodes that have migrated across the clusters. The details of all the nodes within the cluster are stored in the cluster head cache. Thus the above mentioned protocols support interoperability across ad hoc and pervasive environment.
6.2 Routing Protocols

An adhoc network is a collection of mobile host without a centralized access point. Each mobile host acts as a router to advertise services. Wireless media are of limited and variable range in differentiation to wired media. The routing protocol is Adhoc On-demand Distance Vector Routing and Destination Sequence Distance Vector Routing which is extended to wireless media in the proposed work. Network establishment includes node discovery, network interoperability, mobility and service discovery. The routing protocol depends on the network condition and also scalability.

The routing protocols are classified in to proactive and reactive routing protocol. The reactive protocols are on demand protocols. The protocol initiates a route discovery mechanism. This is simple with low computational requirements but it requires a big amount of broadcast message when requesting new routes. The proactive protocol is based on link state information where each node has knowledge about the network topology. This means that the node requires high computational power and continuous update of link state information. In highly dynamic environment the routes are obsolete after a short period of time [40].

**Adhoc On-demand Distance Vector Routing**

On demand routing protocol minimizes the number of broadcasts by creating routes on an on demand basis. In simulations AODV has shown very good performance in highly mobile network
provided that link brokerage is performed using lower layers such as MAC to detect transmission errors [41] [42][43].

Adhoc On-demand Distance Vector routing is a reactive routing algorithm that is chosen instantly by the nodes. There are no routing tables maintained in the route or at the nodes. The architecture of the proposed system is the peer to peer architecture where the nodes move within and across clusters. The cluster head of each cluster group remains stationary and all other nodes are movable within and across clusters does offering services to the users. As the adhoc environment does not depend on any infrastructure, peer to peer architecture is best chosen for adhoc environments. The routes are discovered instantly in this case. If a particular route is chosen for more number of times for the discovery of service, the route is optimized. In the proposed work honey bee optimization is used for hierarchical clustering which is a novel work. The routes are chosen instantly and the personalized request is transferred to the header node of the cluster. Cluster head selects the best matching service from available service offering node. This is done with the sensor nodes deployed in the space. Due to the absence of storage in the proposed work, the overhead is reduced considerable.

**Destination Sequence Distance Vector Routing**

Existing DSDV is a table driven proactive routing protocol, where every node has a routing table entry for each destination node in the network. Each routing entry includes a destination address, the next hop to the destination and the metrics. Nodes exchange route
information periodically, thus propagating network topology change through the network [35] [44].

The routing protocols are generally used to route the information from source to the destination. The routing protocols are selected at the routing layer. In our work we have considered two routing protocols - the proactive routing protocol which includes Destination Sequence Distance Vector routing and the reactive routing protocol which includes Adhoc On-demand Distance Vector routing.

In the proposed work, the proactive protocol is chosen for routing instantly. The link state information of the route is provided spontaneously by the sensors. The routing protocols are selected based on the distance between the sender and the receiver. The path which would traverse the receiver at the minimum distance is chosen. In our proposed architecture no tables are maintained even for proactive type. Seamless selection of the protocols spontaneously depending on the environment and the user's proximity is done so as to minimize the overhead incurred due to the maintenance of the table. This algorithm is extended to asynchronous mode for mobile adhoc communications. Protocols can work in both synchronous and asynchronous mode. Synchronous mode expects the sender and the receiver to be in the network during communication. The service discovery protocols are used to discovery services. The service discovery protocol detects the service and the service is seamlessly routed back to the device of the user by the routing protocols.
The routing algorithm is chosen generally based on the overhead of the network, the distance between the source and destination and the number of best matching service found in the proximity of the particular routed cluster head. Depending on the network parametric conditions AODV and DSDV are chosen.

<table>
<thead>
<tr>
<th>Packet Delivery Ratio</th>
<th>Roaming Node</th>
<th>End Node swap</th>
<th>Relay swap</th>
</tr>
</thead>
<tbody>
<tr>
<td>AODV</td>
<td>0.86</td>
<td>0.66</td>
<td>0.62</td>
</tr>
<tr>
<td>DSDV</td>
<td>0.83</td>
<td>0.76</td>
<td>0.68</td>
</tr>
<tr>
<td>AODV+DSDV</td>
<td>0.78</td>
<td>0.72</td>
<td>0.66</td>
</tr>
</tbody>
</table>

Table 6.1 Overall Packet Delivery Ratio of Routing Protocols

Table 6.1 tabulates the performance of the routing protocol, which was compared to the work of [37] with respect to AODV protocol and was found almost equivalent to existing work in efficiency. The roaming node denotes the intermediate node (within a selected network) that does the transmission, end node denote the sender and receiver and the relay swap is change in the direction of routing, so the different set of nodes are taken in to account during transmission. This generally happens when routing happens across heterogeneous networks. This chapter discusses on the service discovery and routing protocol.

Cross-layer functionality is obtained in the proposed work by the interoperability of service discovery and routing protocol at the middleware. The qualitative performances of the protocols are given under results in chapter 8. The next chapter deals with designing the peer to peer architecture, clustering of peer nodes in the smart spaces and security of spaces.