CHAPTER 7

PEER TO PEER CLUSTERING AND SECURITY

The major purpose of using a peer to peer network is information sharing. In the absence of fixed networks, peer acts as router and routes information. The problems found in implementing peer to peer network in adhoc environment is configuration, message transportation, reliability and co-ordination. The platform of JXTA (Peer to Peer system) is interoperable. JXTA has three layers – JXTA application, JXTA services and JXTA core [14]. Orion employ semantic overlay to peer to peer architecture to provide connectivity between smart spaces. In the proposed work we have built interoperability among the P2P nodes. Freenet is P2P file sharing system designed to make use of system anonymously. P2P is weakest solution in sense of trust and reputation. Security is brought about in P2P system through multiple key encryption, sandboxing and creation of firewalls. In the designed architecture, authentication check is provided to the users to access the peer to peer environment. Napster and Gnutella are two file sharing network in peer to peer environment. Since peer to peer systems involve distribution across independent systems, care has to be taken for security, reliability and availability [45]. Multi-agent system in the Anthil framework is P2P system that supports swarm intelligence. Multi-agent systems solves the problems in dynamic heterogeneous environments. Multi agent system takes up the ant colony metaphor where ants search for food. In Anthil system, Anthil is being used and the structure of P2P is defined through API. Routing in Anthil [46] is done through query method, and
Routing tables are involved for searching. Gnutant is a document sharing system that provides free search capabilities. Anthill [46] model consists of two phenomena: nest and ant. Once the service offering node is found out, it returns to the source; else, the algorithm is terminated. Gnutant is used for searching documents at the local or remote sites. Gnutant uses the ant colony algorithm with the primitives: insert ant, search ant, and reply ant. Routing is done through a hash algorithm embedded in the nodes [46].

In Gnutella peer-to-peer systems, data is routed through broadcasting among the nodes [47]. In Freenet, each document has a hash value, which is obtained by hashing the document name. Routing is done with the matching of the key. If the key obtained through hashing of the searching node matches the searched node, then the document is retrieved. Gnutant maintains a distributed document index and routes information based on lookup's [47].

Java Message Service supports both message queues and publish-subscribe models. Message queue supports one way of interaction, and publish-subscribe supports many ways of communication through topics. Java Message Service is a message-oriented middleware. Epidemic routing protocol is used to transmit messages to all nodes in peer-to-peer networks. Security aspects of peer-to-peer networks are not taken care of in Java Message Service. The main goal of JXTA is to provide interoperability. JXTA provides a set of protocols to carry out functionality in peer-to-peer environments. The various protocols of JXTA are Peer Discovery Protocol, Peer Revolver Protocol, Peer Information Protocol, Rendezvous Protocol, Pipe Binding Protocol.

MIGRA is a modification of the JXTA environment that provides singleton
services and group services. In MIGRA the user context is not tied to a peer but exist as a separate entity. It provides better establishment of protocol stack for security and trust rather than JXTA. Whereas in JXTA the user context is attached to peer entity. Multiple interfaces are supported by JXTA. The framework could be extended to adhoc networks and code mobility. JMS architecture is not flexible due to the protocol stack. One of the problems of P2P is that querying is very slow and this is due to the inefficiency of the protocol stack [48]. To overcome this problem the best method adapted was the cross layer design.

P2P can be either pure or hybrid model. In pure P2P model there is no central server existing where as in the case of hybrid model there is a central server existing. Gnutella and Freenet are examples of pure peer to peer networks. Napster, Groove and Magi are examples of hybrid peer to peer systems [45].

Avaki is an object oriented middleware designed for peer to peer systems. The middleware is designed for layered virtual machines. The goal of seti@home peer to peer system is to search for extraterrestrial radio emissions from nearby developed intelligent populations based on data collected from huge radio telescope. The capability of Groove is direct communication without central server. Groove is intended for collaboration, communication and content sharing [45] [49].

In the proposed work, the peer to peer architecture have three modular components—peering module, interaction of peering module with the smart space and the peering interface with service discovery.
7.1 Peering Module

OSDA can operate on multiple domains with diverse discovery mechanism and can support large number of users participating over wide area networks. OSDA are designed to be platform independent, extensible and fault tolerant. It provides straightforward ways to access control policies [20]. In OSDA, querying method is adapted to establish communication and uses the concept of overlay networks which incurs heavy overhead.

In the proposed peering architecture seamless connectivity between different domains spaces are established. The connectivity could be one to one, one to many and many to one.

The proposed middleware developed at the middleware layer acts as a broker to connect different peers located across heterogeneous smart spaces. The adapter deployed in the smart space is responsible for converting all the services obtained from various service discovery protocols to a single user understandable format. The functional components of the module are as in Fig 7.1 which includes –

**Peer to Peer Indexing Node**

Advertisement is broadly classified in to local advertisement and global advertisement. Local advertisement is within the cluster and global advertisement moves across smart spaces. The order of mapping required for interoperability from one form to other is in the O(N), which is much lesser than the OSDA architecture [21] of interoperability. Interoperability in the protocols
requires some form of conversion from one format to another. Each service has a service id and service description that describes the service. The same service offered by different service discovery protocol has different ID’s. Adapters are used for such mappings.

The discovery process in the peer to peer network across heterogeneous smart spaces does not make use of handlers. The seamless transmission from the user’s personalized profile is routed to the local peer of peer cluster in the adjacent smart space from where it is routed to the global clusters and if the matching service is located by the cluster head the search propagates downwards inside the cluster and the best matching service is retrieved to the user.

**Bootstrap Module:** The bootstrap module is responsible for joining seamlessly the peer to peer networks to smart spaces.

Communication between smart spaces is asynchronous and adapters in the smart space bring uniformity among different communicating languages and protocols. Adapter can communicate with any index of the clusters. OSDA [20] permits local service discovery seamlessly but requires various brokers and handlers which demands heavy overhead. Bridges are used to bridge the service discovery interoperability across heterogeneous smart spaces. OSDA is designed for fixed networks. But proposed work proved better in clustering across heterogeneous smart spaces without involving overheads.
Each node in the peering module has equal power to act both as the service provider and consumer of service. It is completely a decentralized architecture and does not depend on any central server for service discovery or routing. The peering node change state very often. Change of state is addressed to conserve power which is an important property of adhoc networks. In adhoc networks as there is no infrastructure the nodes are most trusted entity and should remain charged for a longer period of time. If all the nodes remain active, then the power of the nodes will be reduced and the system will fail abruptly. In order to prevent this state of the peer nodes, the smart and dummy nodes change state very often. Smart nodes are the one’s which offer service and dummy nodes are the ones which consume the services of smart nodes. The dummy nodes are put to sleep state for certain lengths of time. If they fail to receive service at that time, buffering is carried out and the services are stored in the buffer and at
the later instant of time when the dummy node changes state to awake, then the services are provided to the dummy nodes from the buffer.

Smart nodes continue to offer service until they remain active in the network. Once the node becomes passive, then the dummy node converts themselves into smart ones and starts offering services. Only the header node is permanent and stable. While all other nodes move in and out of the cluster across smart spaces. The interoperability feature binds all the heterogeneous smart spaces together. Thereby the service offering nodes are discovered even when they move out of the home cluster. Peers across different smart spaces communicate through a common protocol language with the help of adapters. In the clustering hierarchy, the cluster in which the nodes get first attached becomes the home peer and as the node move it enters the foreign peer. There is no registry maintained in the home peer to trace the new location of the node. But the location sensor deployed in the peer to peer networks helps to track the new location of the node. This is done seamlessly and the service is provided back based on the profile. When the power level at the header node gets reduced then the new node is selected as the header. The peer discovery could include various services related to context and thus gets seamless information from the context module. If the user is a new user then the context of the user is sensed by means of the sensors. The context information is available from the context module which is interfaced seamlessly in to the peer to peer cluster to retrieve the best matching service. User context relates to the information pertaining to the latest user activity in the smart space.
environment. An illustration of context information is illustrated below.

</Type>
<smart space>
</Type>
<peerID>
</Activity>
<reading book>
</location>
</mma>
</location>
<presence>
</Time>
</Time>

7.2 Architecture of Smart Space with Peering Nodes

The context service is provided by the context provider. Peers who are active members of the group will receive the context service within the group as in Fig 7.2. Multiple services are accessed in the smart space simultaneously at the same time.
Each peer rest on the smart space as in [20]. With the help of routing protocol it routes the content from one peer to another in a group. As the peer moves it gets accustomed with the new service discovery protocol of that space by the concept of interoperability and starts offering services at the new location. The routing protocol also gets changed as it enters the new smart space but the peers still continue to offer service. This feature of offering services seamlessly is brought about by interoperability. The interoperability of routing is attained by code migration. As the protocol gets downloaded, the code provides the functionality of seamlessness and interoperability.
7.3 Service Discovery Peer Architecture

The service discovery peer architecture connects the service discovery architecture to the peer and embeds it on to the peer architecture. The service discovery architecture has a service discovery protocol embedded in to it. Thereby the service discovery architecture is interfaced in the smart space. The service discovery module has the following components:

1) **Service Cache**: are registries were services are stored temporarily.

2) **Service Discovery Components**: maps the service information with the context information obtained from the environment.

3) **Service Registry**: Acts as a buffer for temporary storage of data.

4) **P2P Interface**: This provides the peer to peer interface for service discovery. If one group of peers are not able to satisfy the request, they are forwarded to the other group which have the service registered at the cluster head and provides the service to the users.

When the request from the user comes to the service provider, the provider gets information from both the context module and the personalized profile and the best matching service is provided to the user depending on the coverage domain. The module is very small with less functional component when compared to the work of [50].
7.4 Service Matching Procedure

The service matching procedure is generated based on the location, distance between the source and destination, bandwidth and other utility factors.

1. If the service belongs to the category of service requested for, then the service is accepted.
2. If the service name is in any of the header cluster node then the service is accepted.
3. If the service is within the limited location of homogenous network then the service is retrieved by the seamless protocol transmission based on the utility factor.
4. If the search is done across heterogeneous platform then the interoperability of protocols occurs across the platforms and services are retrieved based on the best utility value.
5. If success in searching, the services are retrieved seamlessly in the device of the user else returned failure.

7.5 Advertising Service Group

Group based service hierarchy is followed in our proposed work. The nodes arrange themselves in clusters and each cluster forms a group. The service advertisement is done by the header node of the group. Each node stores the advertisement in the service cache. As the node moves and joins another cluster it sends its query to the cluster head to advertise service. Group based service discovery is used to get
matching service from all the service offering node within the cluster and also from the vicinity cluster.

Function Get Vicinity ()
Other groups {}
For each service s do:
{
    If s is not in the local cluster group {
For (each group G belonging to service offering cluster do or search in the other cluster)
{
    {if G1 is not in other cluster) then
        Add G1 to other cluster groups
    }
}
}

A node apart from advertising its own service also advertises the service of local vicinity. The service group information moves from one node to another through the whole network. Functional related information’s are grouped together. The service request from the user is propagated to the group, and if the service is not found in the group it is broadcasted to the neighboring group. Due to the intelligent routing, the request is transmitted to the selective header node which has the matching service. Here intelligent routing refers to sensor based routing. The selection of protocol is done by the sensors. Therefore the time taken to search at the vicinity is reduced. Broadcasting the request to all cluster head will also result in the failure message. The service is routed from the user to the peer node using the routing protocol. Inside the peer cluster, interoperability is among the service discovery protocol and is routed
back using the routing algorithm. Integrating routing with service discovery improves the system efficiency to a larger extent.

P2P approach holds good for smart space applications. Dynamic information transmission, redundancy removal, fault tolerance and content reflective addressing are benefits of P2P. P2P is a suitable architecture for smart space environment.

Fig. 7.3 Peer Communication across Smart Spaces.

From the Fig 7.3, it could be inferred that the peers could move across smart spaces in a heterogeneous environment. The peer nodes have equal priority to offer service and also to consume the services offered by other peers. All the three interoperable architectures
designed for peer to peer clustering are flexible and efficient in offering service to the users. The mapping and advertising of services are also discussed in the above section.

7.6 Clustering Algorithms

Clustering algorithms are generally used to traverse through the peer to peer architecture. There are several clustering algorithms designed with the header node selected based on the lowest id, highest degree or highest weight. In our proposed work, the cluster head is selected as that node, which traverses all other nodes in the network with a minimum distance. The node must have the capacity to link and get information from all other nodes. In clustering, only the cluster heads remains constant while all other nodes move in and out of cluster.

In our proposed work no route tables are maintained in the cluster. There by the overhead in the cluster head is minimized. Routes are detected intelligently with the help of sensors. Honey bee optimization algorithm is used to find the optimal path of routing between the source and destination. Generally honey bee algorithms are used for flat structure and SOAC coloring algorithm is used for hierarchical structure. As we have considered peer to peer architecture in the proposed work, we have extended the honey bee optimization algorithm to hierarchical structure.
7.7 Existing Algorithms

Ant Colony Algorithm

Ants are created by service manager and travels through server manager to updates pheromone table. Pheromone is the path set up by ant after routing once.

Random Round Robin/Load Distribution Algorithm

Pushes load on overhead server (or) round robin server which would observe if it would have capacity else it is transmitted to the next neighboring server.

Simple Greedy Algorithm

A simple greedy algorithm pushes load to the least loaded neighbor. R3 and greedy chooses the closest server that is least loaded. One of the hardest problems is resource allocation problem and server placement problem.

Broadcast of local Eligibility Based Control Algorithms

A cluster of servers are formed with the cluster head. The server broadcast a list of services and each server will host a particular service. The algorithm supports fault tolerance and scalability [51].
Anthil Model

Anthil model provides communication between Ant and nest. The service is provided by ant algorithm. Each nest has unique identifier. Ant moves across nest to satisfy request. If it cannot satisfy the request it is terminated [47].

Ant Colony Metaheuristic

Clustering Algorithm can be arranged for routing. Swarm optimized algorithm is used for optimization. SOAC algorithm is used for coloring based on probabilistic rule [41].

7.8 Proposed Clustering Algorithm

1. The Cluster groups are formed with header node. The header nodes are selected based on the condition that they are smart nodes and secondly depending on the longevity life of the node in the network. The header nodes connect all other nodes in the network with minimum distance.
2. Smart spaces are made interoperable by the service discovery protocol.
3. The nodes are free to move within the cluster in the same smart space and also across spaces.
4. Each node is to work on the principle of the ant searching for its food. The ant generates a pheromone which traces the location of food grains directly and easily and thus the entire path of search is optimized. The service discovery path is
optimized once the services are searched for and the optimal path remains in the network until it is not being used.

5. Here in the proposed algorithm, optimization is done on the header nodes, which are smart ones and through the swarm intelligence of sensor nodes the request of the personalized profile and context is routed to the matching node that offers services and the required service is offered to the user.

6. By this process, the nodes that offer similar service could be traced by the pheromone and could be grouped together irrespective of position of the nodes in the network. Thereby optimization in transmission, clustering and header node tracking could be obtained. Peer to peer clusters forms a honey bee hive, and the nectar is the exact matching node that offers the service to the user.

7.8.1 Swarm Intelligence based Honey bee Optimization Algorithm

Swarm intelligence based optimization of optimal path is followed in the honey bee optimization algorithm. The ant generates a pheromone which traces the exact path in searching for food. Once the pheromone is generated, the search during the second time in the same path becomes easier. During forward movement the ant searches for the food and when not found it returns back to the nest without generating pheromone else when the path is found it generates pheromone. In the proposed work we have deployed sensors along with the swarm intelligence technique.

Two sets of ants are used in the existing honey bee optimization-the forward ant and the reverse ant. The forward ant pushes the information in to the stack on reaching the destination and
then the reverse ant traces back and moves back generating pheromone until it reaches the source node [52] [45]. In the proposed work, using sensors, the personalized profile is forwarded to the cluster head, where the matching service is found. From the cluster head the honey bee optimization of clustering is done and the node which offers the best utility value function is returned back. If the selected node fails then automatically the next node that offer the best utility service at the optimal distance from the source is returned. Honey bee optimization principle with ant and nest methodology is followed as a principle for optimization. Optimization is done both in the case of Adhoc On-demand Distance Vector Routing and Destination Sequence Distance Vector routing. As intelligence is applied to the route the best optimal and the shortest path that provides the service through the utility value is selected.

7.9 Security

Security is an important feature of all networks and especially demands high recognition in wireless adhoc networks, as there are no fixed infrastructures in this set up. As the devices move across the networks, there are heavy chances of attacks which demands security. Security in the proposed architecture is obtained by digital signature scheme thereby checking the validity of the user. Only when the user is proved to be a valid user, he/she is given rights to access the services of the smart space. Thus security plays a vital role in designing the proposed architecture of mobile adhoc networks.

This chapter deals with the efficient clustering algorithm. Service discovery and matching algorithms are also discussed. The chapter
concludes that clustering of peer nodes increases the speed of delivery of services. The next chapter provides qualitative performance analysis of the protocols.