CHAPTER 5

CROSS - LAYER APPROACH

Cross-layer approach integrates the service discovery and selection functionality with adhoc routing algorithms. On one hand cross layering supports routing mechanism and on the other hand it is for service discovery and provides the best matching service. Shakkottai suggest a cross-layer design to improve performance in Manets. Conte et al proposes cross-layer to maintain layering principle. Cartel et al shows the application layer on top to support for group communication. Kodi et al proposes extension of Manet routing protocol to service discovery [33].

In the work [34], Meenakshi Venkatramanan designed a new protocol called CDR - cross layer discovery and routing protocol. Two assumptions are made in the work, one is that the destination is assumed to be in the network and only one application is sent at a time. If route is found, then the routes are stored in cache or look up server and are initiated using the client-server technology for matching. Additional protocol is added to determine the state information. Cross layering allows non adjacent protocols to directly interact attaining overall optimization. Cross-layer discovery and routing protocols enables routing to be established based on the attributes of the receiving host. Protocol actions are triggered by input preconditions. Actions without preconditions are triggered by different host. The application sends packets and when no satisfactory route exists it sends resource discovery packet. When the resource discovery completes, at least one route will be available in
cache. The applications send packets when it has data to send. It selects a route and then sends a packet and maintains a copy of the send packet in the cache. The application will then be delivered to the destination. The acknowledgement will be received in the reverse direction. Cross-layer discovery and routing allows multipath propagation of data also. If the route error occurs along a path, it is propagated back to the sender and the sender removes the packet that has error and again reassembles and resends the message. Optimization of the route is achieved based on the recovery. During the resource discovery process the application creates route discovery and stores it in the cache. Thus during the discovery, the route cache will contain at least one cache to the destination. Interaction between routing and middleware layer allows two layer to share information by means of system profile. Cross-layer design increases the overhead of protocols. Routing decisions are made in the application layer and the packets are no longer of fixed length. The query from the server is passed on to the lookup and the path to the destination is found from the query. The distance between the sender and the receiver and the length of the data packets affects transmission. Using cross-layer approach the application layer information can be accounted to the routing layer [34].

The main goal of the cross layer architecture according to [35] is to increase overall network throughput, to reduce service discovery cost, optimize service selection and reduce the cost of network transmission. The cross layer architecture for service discovery always maintains a higher throughput than the normal service selection protocols [35].
5.1 Layers of Cross Layer Approach

Cross layering is a concept to skip the intermediate layers while traversing the protocol stack. The intermediate layers are skipped out but the functionality of the layer is taken into consideration. The time taken for service discovery and selection is reduced by this process. Proposed architecture of cross-layer approach consist of five layers – data link, MAC layer, network layer, middleware layer and transport layer. The data link and MAC layer controls power during transmission and also takes care of link management. The network layer takes care of routing between source and destination. The routing includes the selection of protocol for routing - whether it should be a proactive or reactive routing. This is decided by the sensor deployed in the middleware and the reflective context aware feature of the middleware. It is in the middleware layer the actual cross-layer approach is built. Reflectiveness subjects the middleware to the changing environment. Transport layer plays a role in discovering service.

Cross layering in the proposed work is done at the middleware layer, which is a new layer created in between the transport and routing layer. It is in the middleware layer, the interoperability of service discovery protocol and routing protocol occurs. The selection of route and seamless service discovery is automatically taken care of by the interoperability feature of the middleware. The requested service is seamlessly routed back to the device of the user. The communication is completely seamless and asynchronous. Asynchronous communication does not demand the existence of service provider and consumer in the network at a particular instant of time. The communication could also be
synchronous i.e. the service provider and consumer of the service could be present instantly during communication.

Cross layering along with context awareness and seamlessness increases the speed of service selection and delivers the service at the faster rate. Normally in the protocol stack all layers have to be traversed to obtain and provide service to the user and this is not necessary in cross layering.

In the existing work, CrossROAD is a Pastry like middleware for adhoc networks [36]. The disadvantage of Pastry on adhoc network is the lack of knowledge of physical location of other peers. CrossROAD defines the service discovery piggybacked along with the routing protocol. CrossROAD identifies each node by the logical address and assigns address for each node. The service in the CrossROAD registers with the plugin’s. Through the plugin’s discovery of the first peer node is done in the local registry, from where it is forwarded to global service discovery. When the CrossROAD or Pastry wants to detect the correct matching service, it interfaces with the plug-in’s and directly routes the information to the destination. In the CrossROAD approach, the service discovery protocol gets piggybacked along with the routing protocols. Until the CrossROAD is active the plugin’s send messages between the nodes, and once the CrossROAD becomes inactive the plugin’s gets disconnected. Overlay network is randomly selected one after the other. In the distributed computing environment selection of overlay networking is very important and is done spontaneously [37] [38] [39].
In the already existing work, of Rolf Winter [40], Crosstalk is designed for load balancing. Interoperability feature is not dealt with in the architecture. It also involves piggybacking from the local to global view. In the proposed work overlay networks are not considered, but nodes move across heterogeneous networks offering services. In the proposed work cross layer approach is designed to bind the feature of interoperability seamlessly in the nodes and thereby make the peering nodes aware of services offered by all the other peers. In the Existing architecture, local nodes formally receive information as in Fig. 5.1.

From the local nodes it is forwarded to the global node maintaining the information of all services in the route. When the
application is to be sent, first it checks the internal local storage node and then selects the destination among all the P2P nodes.

5.2 Conclusions

This chapter deals with the cross layer approach where in which the routing protocol of the routing layer seamlessly interoperates with the service discovery protocol of the transport layer by cross layering. The middleware layer is skipped while traversing but the functionality is taken in to account and interoperability among the protocols is attained. By this method, the search time is reduced and efficiency is increased to the larger extent in discovering services. The next chapter deals with the design of routing and service discovery protocols.