The success and improvement of the P2P system is dependent on the ability to satisfy the QoS measures. This research work has recognized a way to improve the QoS metrics in resource management and file sharing process thereby improving the fault tolerance of the entire system in a distributed environment.

In this research work, three new techniques QASAC, PSONPCT and FT-TREE have been proposed for the fulfillment of QoS needs. The QASAC (QoS Aware Shared Application Cluster) architecture is used for effective allocation of resources by selecting the appropriate middleware. The QASAC technique resulted in improvement of QoS metrics thereby improving the load balancing capability and guaranteeing efficient resource management. A new caching technique for a P2P network called the PSONPCT (P2P Semantic Overlay Network Partial Caching Technique) has been proposed for handling intricate queries in the network. Based on the popularity of the object, it is placed proactively or reactively using the caching technique. This technique has yielded a good improvement in throughput and reduced the delay. In FT-Tree, the nodes are arranged in tree fashion. When a node fails, the adjacent node takes in-charge of the failed node and handles fault tolerance effectively.

The architecture proposed in Chapter 3 gives an efficient clustering and load balancing capacity for satisfying the QoS guarantees. The QASAC is
used along with SLA for providing good response time with improved service availability. The average response time of the QASAC architecture shows 4\% improvement when compared to the existing architecture. The Pseudo Domain Name System used in this architecture is used for keeping track of any failure and elects the alternate resource manager. The average waiting time for a client to get the service from the server is 0.071ms, which is a better result that reduces the waiting time of the client. Altogether, the entire system has exhibited its contribution towards the enhancement of QoS metrics like resource availability, timely response and expendability.

The PSONPCT proposed in Chapter 4 showed a good improvement in throughput and bandwidth and reduced the delay. A metric called popularity value is calculated based on familiarity of an object considering the QoS metrics: hit ratio and access time. Based on the popularity value, the objects are categorized and stored in the partial cache proxy for easy access of the data. The search operation adopted here follows a replication strategy for increasing the probability of the success of search. While compared with the existing PSON protocol, the PSONPCT showed 0.1\% increase in bandwidth, 1.39\% increase in throughput and the delay is reduced by 0.5\%.

The Chapter 5 concentrates in achieving Fault tolerance in a tree based Peer to Peer Overlay Network. Selecting the appropriate middleware plays a main role as it assists in providing good fault tolerance. ALIVE packet is used to detect the fault. It indicates whether a node and its link are reliable. The fault tolerant technique provides a way for fault detection and fault recovery in order to satisfy the QoS of the system. The middleware takes care of the server side replication thereby paving way for high availability and reliability. The results obtained proved that the proposed approach (FT-Tree) reduces the packet drop and delay by 50\% and improves the throughput by 8.39\% compared to the PSON-fault technique.
There are possibilities of variety of researches in Peer-to-Peer Network keeping this research work as a basis. Future research to extend the current work may take various directions. Providing security is one challenging task in a peer-to-peer network which could be implemented in future. As the nodes join and leave the network dynamically and the nodes are not trustworthy, security becomes a difficult task. Research is also needed to widen the database management of the network by assigning unique naming details to peers before they join the network. As a further extension, peer virtualization can be carried out to achieve elasticity of the network in handling queries flooding.