ABSTRACT

The recent advancements in portable devices, wireless network technology and satellite services have led to the development of mobile computing environment. This environment gives access to information regardless of the location of the user. With the evolution of PCS (Personal Communication System) and GSM (Global System for Mobile communications) and other technologies, advanced wireless communication services are being offered to the mobile users. Mobile Database System is a distributed client/server system based on mobile communication technology in which clients can move around freely while performing their data processing activities in connected or disconnected mode.

A mobile database is a database that resides on a mobile device such as a PDA, a smart phone, or a laptop. Such devices are often limited in resources such as memory, computing power, and battery power. Due to device limitations, a mobile database is often much smaller than its counterpart residing on traditional servers and mainframes. Mobile users are not constantly connected to a central server but they need access to data nevertheless. Data thus need to be transferred to the mobile device and to be managed locally before being reintegrated with the original data source.

The issues concerned with mobile database transactions and mobile applications are identified through a detailed literature review. Traditional transaction models are investigated to find out their suitability for applying them to a mobile computing environment. Existing mobile transaction models are also investigated and their strengths and weaknesses are observed.
In a mobile computing environment, concurrent access to data at the database server may result in data inconsistency and increase in network overhead. Hence, it is decided to develop a mobile transaction framework which will minimize the network overhead and response time substantially. In this transaction management framework, mobile users are allowed to access data stored in the cache of a Fixed / Mobile Agent. Data Access Manager (DAM) module at the Fixed / Mobile Agent enforces concurrency control using cache invalidation technique. This framework supports priority based transaction scheduling which allows four levels of priority to be assigned to the requesting mobile nodes based on available energy and connectivity. The four schemes which are proposed in this thesis are discussed below.

In scheme I which uses Fixed Agent based Direct Transaction Architecture (FADTA), frequently accessed data are cached in the Fixed Agent situated in the fixed wired network. Whenever an MH enters a Fixed Agent area, it can connect and access data in the cache. But upon update request by an MH updation is done at the local cache and invalidation report is sent to all the MHs which have already accessed the same data. This will force the MHs to refresh their data values. This framework also takes into account transaction update during disconnection. The results of the proposed scheme are compared with the existing scheme for non-disconnected case. The proposed scheme suffers from slight increase of response time as compared to the existing scheme because of agent delay. For a disconnected case, transaction support is provided by the FADTA for small number of transactions.
Scheme II uses Mobile Agent based Direct Transaction Architecture (MADTA) in which, a Mobile Agent is used as a special mobile node which connects to the MSS to cache the frequently accessed data. Disconnected Mobile Hosts can connect to the Mobile Agent using short range wireless communication technologies to form a mobile affiliation work group. Data request and data update operations are carried out as in scheme I. After being disconnected from the server, a Mobile Agent can move along with the connected MHs and MHs can continue their transaction execution with support from the Mobile Agent. If data update at the server is requested, the Mobile Agent will wait for reconnection with the server before updation is made. The response time for the executed transactions is evaluated for both non-disconnected and disconnected cases for the E-I repository model and the proposed scheme. It is found that compared to the E-I model the MADTA scheme gives better results in terms of response time. This is due to the extra time involved in communication overhead for the locking mechanism of the E-I repository model.

Scheme III and Scheme IV use Agent based Multi-hop Transaction Architecture (AMTA) for transaction processing. Scheme III (FAMTA) uses Fixed Agents, while scheme IV (MAMTA) uses Mobile Agents. In both schemes, an MH can directly connect and communicate with other MHs which are within its communication range. Any MH can communicate with another MH in multiple hops using intermediate MHs in case they are not within each other’s communication range. In the FAMTA scheme, Fixed Agents located at different places are connected to the Data Base Server through wired network. In Fixed Agents, cache is used to store the frequently accessed data. An MH can find out the nearest Fixed Agent and connect to it to access data from the cache. In MAMTA
Scheme, the Mobile Agent is a special mobile node which connects to the MSS to cache the frequently accessed data from the server. Disconnected Mobile Hosts can connect to the Mobile Agent using short range wireless communication technologies. In this environment both the user and the Mobile Agent will be moving. So it is necessary to find a route from the MH to the Mobile Agent before submitting a transaction. Since many applications in this environment are time-critical, the transactions are required to be executed not only correctly but also within their deadlines. The transaction scheduler has to consider firm and soft transactions with their deadlines in order to minimize abortion of transactions due to deadline violations.

When FAMTA and MAMTA schemes are compared with the MANET based scheme individually, it is found that both FAMTA and MAMTA schemes provide better results in terms of response time, number of transactions missing deadlines and number of completed firm transactions. The low response time results from the presence of cache in the Fixed / Mobile Agents which allows transactions to finish quickly. The decline in the number transactions of missing deadlines is due to the use of cache invalidation technique for the concurrency control instead of locking based scheme which is used in the MANET based scheme. The increase in the number of completed firm transactions is due to priority based scheduling of transactions based on energy availability and connectivity which allows more number of firm transactions to get completed before deadlines.

From the performance analysis, it is found that these four schemes provide better results when compared with the existing schemes individually. It is also observed that the schemes devised for Agent based Direct Transaction Architecture (ADTA) take less
response time compared to the schemes devised for Agent based Multi-hop Transaction Architecture (AMTA).