CHAPTER 3
MOBILE DATABASE TRANSACTIONS

A transaction is a set of operations that translate a database from a consistent state into another consistent state. In the context of mobile computing, there exists several interpretations of mobile transactions, but here we use a general definition [37]: “A mobile transaction is a transaction where at least one mobile host is involved”.

3.1 Architecture Issues

This section focuses on architecture issues related to the context of mobile transactions. In classic client server architectures, functions of each actor are statically defined [38]. In the absence of failures, it is assumed that neither the client and server location nor the connection between them changes. In mobile environment, however, the distinction between clients and servers can be temporarily blurred resulting in an extended client-server model [39]. Architectural choices impact application design and data management. In particular, transaction execution on a MH is only possible if the MH provides some minimal capabilities. Data stored on MHs generally come from Data Base Servers (FHs). Therefore, MH’s work must remain consistent with the Data Base Server. Mobile clients may vary from thin to full clients depending on their characteristics as given below [37].

• Thin client architecture, where clients require running operations on servers in which all the computations are done while clients only display text and graphics, play audio, etc.
• Full client architecture, in which clients will do server functions so as to execute applications without being strongly connected to remote servers. In mobile environment clients may be forced to work in disconnected mode or with weak connections.

• Flexible client-server architecture which generalizes both thin and full client architectures. The differences between client and server may be temporarily blurred for the purpose of performance and availability.

• Client-agent-server architecture which is a three-tier model that introduces an agent or proxy located in the fixed network. Agents can act as surrogate to MHs or it can be attached to a specific service or application such as Database Server access.

3.2 Mobile Database System

In contrast to distributed databases operating on traditional hardware, the devices available for mobile databases may exhibit severe resource limitations which can affect the manner in which data is managed. However, in many respects both conventional distributed databases and mobile databases may be considered as special cases of a common architecture and both can borrow ideas from the other [109]. The mobile components can also be added within a distributed database system [110, 111]. In short, many distributed data management issues are similar to those that affect the mobile database systems, and many innovations considered for mobile databases are also applicable to some distributed data management problems.
3.2.1 Mobile Databases

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 servers and mainframes. A mobile database is managed by a Database Management System (DBMS). Due to resource constraints, such a system often has limited functionality compared to a database management system in a fixed wired network. Other DBMS components such as query processing and recovery may also be limited. Queries to the mobile database are usually posed by the user of the mobile device. Updates of the database may originate from the user, or from a central server, or directly from other mobile devices. Updates from the server are communicated wirelessly. Such communication takes place either via a point-to-point connection between the mobile device and the server, or via broadcasting by the server.

The main question that needs to be answered with respect to data management in mobile applications is what data to store on the mobile device, how to get it there and how to keep it synchronized with data on a central server. The underlying assumption here is that mobile users will not constantly be connected to a central server but will need access to data nevertheless. Data will thus need to be transferred to the mobile device and to be managed locally before being reintegrated with the original data source.

3.2.2 Reference Architecture of a Mobile Database System

For the purpose of building the foundation for transaction management, a reference architecture of Mobile Database System (MDS) [2] which is built on PCS
(Personal Communication System) and GSM (Global System for Mobile communication) is given in Figure 3.1. This architecture supports mobile users to initiate transactions from anywhere at any time with a guarantee of consistency preserving execution. Recovery of data from the transaction, system and media failures is taken care of by the MDS.

An MDS is a distributed multi database client-server system based on PCS or GSM. An MDS consists of Fixed Hosts (FHs), Mobile Hosts (MHs), Mobile Support
Stations (MSSs), Base Station Controller (BSC), and Mobile Switching Centre (MSC). BSC [41, 42] coordinates the operations of MSSs using its programs when commanded by MSC. MSS has some additional data processing capability in order to coordinate with Data Base Servers (DBSs). MSS communicates with MHs via wireless channels using transceivers. The geographical area covered by a MSS is called a cell. Each MH communicates with the MSS of the cell it belongs to. MHs can access data from the Data Base Servers installed at MSS or FH. Data can be accessed from each DBS by any MSS or FH. The set of MSCs and PSTN connects the MDS to the outside world.

3.3 Effect of Mobility on the ACID Properties of a Transaction

The presence of mobility gives rise to a number of problems related to the maintenance of ACID properties of a transaction. The ACID properties of a transaction must be maintained in all data management activities. Concurrency control mechanisms and database recovery schemes make sure that ACID is maintained.

The data distribution in conventional distributed database systems can be done using database partition, partial replication and full replication. The presence of processor mobility introduces the concept of Location-Dependent Data (LDD). Since the distribution of LDD depends on geographical locations, its distribution is defined as spatial distribution while the conventional distribution is called temporal distribution. Spatial replication and temporal replication is used in spatial distribution and temporal distribution respectively. In [2], the definitions for spatial and temporal replication are given as:
“Spatial replication refers to copies of data objects which may have different correct data values at any point in time. Each value is correct within a given location area. One of these copies is called a Spatial Replica.”

“Temporal replication refers to copy of data objects all of which have only one consistent value at any point in time. One of these copies is called a Temporal Replica.”

The temporal distribution mainly considers local availability of data and the cost of communication. But in spatial distribution, the geographical location must also be included.

- **Effect of mobility on Atomicity**: The property of atomicity guarantees that partial results of a transaction do not exist in the database. If a transaction fails to commit, then all its effects are removed from the database. Mobility does not alter the definition of atomicity but makes its enforcement quite difficult. Transaction execution log is required for implementing atomicity. The log is stored in the server in a conventional system and it is easily available. Since a mobile unit gets connected and disconnected to several servers when it is mobile, a different logging scheme is required in a mobile environment.

- **Effect of mobility on Consistency**: In a centralized or distributed environment, there is only one correct value for each data object (mutual consistency). A replicated database is said to be in a mutually consistent state if all copies have the exact same value [40]. In addition a database is in a consistent state if all the integrity constraints identified for the database are followed. In mobile system, the presence of location-dependent data defines two types of consistency, namely Spatial consistency and Temporal consistency.
• **Effect of mobility on Isolation:** This isolation property ensures that a transaction does not interfere with the execution of another transaction. Isolation is enforced using a concurrency control mechanism. In a mobile environment, since a mobile unit may visit many data regions and process location-dependent data, the concurrency control mechanism must enforce isolation in each region separately but achieve isolation for the entire transaction.

• **Effect of mobility on Durability:** Durability guarantees the persistence of committed data items in the database. In a mobile system, the durability is regional as well as global. For spatial replica, regional durability is enforced while for temporal replica, global durability is enforced.

### 3.4 Mobile Transaction Execution

In a mobile database system, a global transaction is divided into a number of subtransactions which are then executed at multiple nodes. The entire execution requires a software module called a coordinator which manages a set of activities leading to the termination of the transaction. In mobile database system, MSS serves as a coordinator.

Mobile transaction can be initiated from a FH or from a MH or from both. An MH can be a client with some server capabilities for transaction processing. There can be five execution models [37] based on client capabilities: (a) the mobile transaction originates at MH but entirely processed at FHs and the final result is sent to the MH, (b) the mobile transaction is initiated and executed on the MH. This model supports disconnected transaction execution on the MH. It requires synchronization of the MH’s work on the server located in the wired network, (c) Transaction execution can
be distributed between a MH and a set of data base servers and the final result is sent to the MH. This model is motivated by the availability of resources such as data and power on the MH (d) Distributed execution of transaction takes place among many MHs. It is a peer-to-peer approach. An MH can act as a client as well as server. In Mobile Ad-hoc Networks (MANET) [43, 44] MHs communicate with each other using a point to point connection and (e) Transaction execution is fully distributed among MHs and FHs as in electronic commerce [45]. The problems that arise due to the inclusion of transaction services and distributed data management in the mobile computing environment have been identified in [46, 47, 48].

When a transaction is initiated at an MH, it verifies to find out if the transaction can be entirely executed locally at the MH. If it is so, the MH executes and commits the transaction using its cached data. Then the MH sends transaction updates to the server for installing them to the database and sends the results to the user. If it cannot be entirely executed at the MH, then there are two options.

(i) **Transfer required data items from the server to MH:** After identifying the set of data it requires to process the transaction, the MH communicates with its MSS, which reaches the server and moves the data items to the cache of the MH. Now the MH executes and commits the transaction and sends final results to the user. The updates are also sent to the server for installing them to the database. Though this scheme does not require the services of the coordinator, it generates a significant amount of communication overhead.

(ii) **Distribute transaction to a set of nodes:** The MH divides the transaction into a number of subtransactions. It keeps the subtransactions it can execute and with the help of the coordinator, it distributes the rest of them
to a subset of servers for execution. The current MSS of the MH becomes the coordinator of this transaction and manages the entire execution. This scheme generates less communication overhead.

3.5 Examples of Mobile Database Transactions

Examples of mobile applications which explain the issues in data management and the need for transaction support are given below.

- Coordinating mobile databases is an important issue in mobile applications. Consider an application where real estate agents coordinate their databases in exchanging real estate information with the goal of pushing sales. Since they travel to their customers (who may want to sell or, instead, to buy), they always carry relevant data with them. When one is on the site of a customer, who wants to sell a house, the agent updates his database and makes this data available for other agents. Or, when an agent talks with a potential buyer, and nothing from the agent’s database satisfies the client, the agent may want to query other agents’ databases to look for additional sale options.

- Effective data management has been identified as a significant issue in mobile application development. Consider an application of mobile databases in the medical profession, where within a hospital or medical practice scenario, practitioners often find themselves moving between patients and locations. Using mobile systems, it is possible to provide, in an effective manner, updated data to the medical practitioner without the need for physical connection. The mobile system then enables the practitioner to access current data on the patient for diagnostic purposes, and to update details about the condition of, and recommended actions for, the patients that they are
monitoring. Additionally, outside the hospital or the medical practice, practitioners visiting patients may also benefit from the use of mobile databases. Before leaving, the practitioner’s database may be summarised to maximally contain the data that is required, based on a variety of parameters that will change dynamically. Once the practitioner leaves, connection to the main database will be expected to be weak or non-existent. The practitioner will then need to rely on the data available locally. Any added patient information would then be stored on the mobile computer and merged back with the main database when the connection is reliable. To do all this, a sophisticated and robust mobile data management system must be in place.

- Client mobility also allows new applications that are location-based. For example, consider a mobile application that can tell a user the location of the nearest restaurant. Clearly, “nearest” is relative to the client’s current position and movement can invalidate any data that are currently cached. Upon movement, the client must efficiently invalidate parts of its cache and request updated data from the database.

- M-Commerce scenarios like mobile auction applications assist sellers and buyers, for example, in a flea market, sellers may use their mobile devices to describe their offered goods while buyers search for and locate desired items. In addition, the amount of mobile devices allows multi-hop ad-hoc communication. Contracts between several buyers can be signed to gain volume discounts. In this context, transaction support is necessary in order to achieve consistency for contracts and buying/selling actions across all involved databases.
• Mobile commerce makes business mobility a reality. For example, enabling a stock investor to access the latest stock market information and undertake associated transactions using mobile devices connected to wireless network at any time and from any place.

• Homecare Applications assist nurses with mobile devices like PDAs to get information about patients. Transaction support is required whenever patient data, medical data, or subscriptions get updates. However, the amount of transactions is usually very low. Therefore, the concurrency control mechanisms used do not need to be efficient regarding the transaction throughput, but they must be efficient regarding data blocking.

### 3.6 Characteristics of Mobile Transactions

The presence of mobile environment influences mobile transactions to possess unique characteristics which are different from that of the transactions used in centralized environment. The characteristics of mobile transactions are given below.

• When a MH moves from one cell to another, the currently executing transaction will also move to continue its execution in the new cell. This results in activities like handling hand over processes, establishment of new communication channels or updating the routing tables.

• Movement of MH to another cell causes the movement of states of transaction, related transaction resources and location information.

• Mobile transactions are long-lived transactions compared to conventional ACID transactions because of the mobility of data and users, frequent disconnections which may result in the blocking of execution of transactions and limited storage capacity and power constraints in the MH.
• The execution of transactions in the MH may be interrupted due to either the surrounding environment conditions or the limitation of computing capacity of the MH. For example, wireless network disconnection occurs suddenly during transaction execution or the performance of the mobile host is slowing down due to heavy computing load. These interruptions can result in suspension or abortion of transactions.

• Mobile environment requires distributed transaction processing. Due to the limitations of processing capacity and resources, mobile hosts require additional support from other hosts to carry out transactions. For example, a transaction, which is initiated by a mobile host and accesses a large data set that is not cached at the mobile host, could be moved to stationary hosts for executing. This could reduce transaction processing time and avoid transferring a large amount of data through a slow wireless network, i.e., achieving higher throughput for the transaction processing system. Furthermore, the portable computing devices are easily damaged; therefore, the results of committed transactions must be saved at stationary database servers.

• Mobile environment supports disconnected transaction processing. A mobile host can be disconnected from the database servers for long periods. Therefore, transactions that are executed at the mobile host may suffer from long blocking if the necessary data is not available at the mobile host. To deal with this problem, the mobile transaction processing system should have the capacity to cache enough data so that it can carry out the transactions while being disconnected from the database servers.
• Temporary data inconsistency may result due to long disconnection periods which will cause shared data among different mobile hosts. For example, a transaction at a disconnected mobile host can modify a shared data item that is currently being read-only cached in a local storage of another disconnected mobile host. Data synchronization processes will be carried out when the disconnected mobile hosts reconnect to the database systems so that the data consistency of the database systems will be achieved.

• Mobile transactions involve heterogeneous processing. This is due to the requirement of support for different types of wireless network technologies, protocols and multiple database systems during the execution of mobile transactions.

• Mobile transactions should have concurrency control mechanisms that are efficient in mobile environment to avoid inconsistency of data. Mobile transactions should also support recovery.

### 3.7 Issues in Mobile Database Transactions

There are many issues in mobile database transactions due to the characteristics of mobile computing environment. These issues are described as given below.

• The conventional ACID transaction model was unable to satisfactorily manage mobile data processing tasks due to the following reasons:

  i. Mobility of mobile hosts

  ii. The presence of unpredictable hand off
iii. The presence of doze mode, disconnected mode and forced disconnection

iv. Lack of necessary resources such as memory and wireless channels

v. The presence of location dependent data

vi. Limited battery life of mobile devices

vii. Unreliable communication.

In order to manage these issues, different transaction models (relaxing ACID properties) that can handle mobility during data processing and fulfil wireless environment constraints have been proposed.

- Minimizing the frequency of aborts in mobile transactions due to disconnections.
- Ensuring the correctness of transactions executed on both fixed and mobile hosts by the operations on shared data: Blocking of mobile transactions due to long disconnection periods should be minimized to reduce communication cost and to increase concurrency [51]. After disconnection, mobile host should be able to process transactions and commit locally.
- In Mobile computing, there is always a competition for shared data since it provides users with the ability to access information through wireless connections that can be retained even while the user is moving. Further, mobile users are required to share their data with others. This provides the possibility of concurrent access of data by mobile hosts which may result in data inconsistency. Concurrency control methods have been used to control concurrency. Due to limitations and restrictions of wireless communication channels, it is difficult to ensure consistency of data while sharing takes place.
• The increase in frequency of network and transactions failures and mobility complicates DBMS recovery process. Also, because of mobility, more logging may be required to recover from failures. Caching at MH increases data availability. However, it is essential to maintain cache consistency during frequent updates using new update protocols.

• Replication in mobile environment certainly increases availability but may need certain weaker consistency criteria [64]. Due to the possibility of problems during hand off when MH moves to another cell, transaction failures may occur.

• An MH failure creates a partitioning of the network which will complicate updating and routing algorithms.

In order to enhance the suitability of transactions for mobile environment, the transaction processing systems should be provided with adaptability [50] to deal with different environment conditions and to cope with the constraints of mobile computing resources. However, there are still several major limitations. The architecture of mobile transaction environments [51] relies too much on the mobile support stations. Only few research works focus on caching frequently accessed data in the fixed agents in the wired network [52]. The ability to support both the disconnection and mobility is still a major challenge for mobile transaction models [53].

3.8 Chapter Conclusions

In this chapter, after a discussion on mobile database system, the focus has moved to transaction execution. It is clear from the discussion that transaction execution in mobile database system is more complicated than transaction execution
in distributed database system. This is due to the characteristics of mobile environment such as mobility of mobile nodes and disconnection in communications. It is also found how the presence of mobile environment can influence mobile transactions to possess unique characteristics which are different from that of the transactions used in centralized environment. A detailed review of literature related to traditional transaction models and mobile transaction models are discussed in the following chapter.