CHAPTER ONE

INTRODUCTION

1.0 Background of Study

Rapid advancements in wireless technologies and portable devices have given mobile computing considerable attention in the past few years as a new dimension in data communication and processing, and a fertile area of work for researchers in the areas of database and data management (Barbara, 1999; Nishio and Tsukamoto, 2002). Mobile computing can be seen as an integration of portable devices and wireless communication. As a consequence of the commonalty of mobile computing devices (e.g. laptop, PDA, and cell phones), the environment for accessing and processing data is changed from stationary to mobile due to the increased utilization of portable devices to carry the shared data, which are needed to be accessed by the users from anywhere and at anytime in the network. This increased usage has led to popularity of mobile databases (Phatak and Nath, 2004; Waluyo, Srinivasan, and Taniar, 2005). According to Connolly and Begg (2004) and Holliday, Agrawal, and Abbadi (2002) , the mobile database can be defined as a database that is portable and physically separate from a centralized database server but is capable of communicating with server from remote sites allowing the sharing of data. This database is more advanced than the fixed distributed databases as it offers the feature that data are available anywhere independent of the availability of the fixed network connection (Nishio and Tsukamoto, 2002).

From a data management standpoint, mobile databases can be considered as a variation of distributed databases that is distributed among wired and wireless components. Thus, a mobile database is a distributed when it is stored on a set of autonomous mobile hosts that are connected via wireless channels to each other and to the stationary hosts. The software that is used to manage the mobile database can be referred to as a Mobile Database Management System (MDBMS). The combination of mobile database and MDBMS is refereed to as a Mobile Database System (MDS) (Ozsu and Valduriez, 1999). This system is considered as a Large-Scale Mobile Distributed Database System (LMDDBS) when the mobile database is distributed among a set of large number (i.e. thousands) of autonomous mobile hosts that located on different geographic areas. This consideration is based on the two aspects of large scale, which are geographic separation and number of sites (Baboglu, Bartoli, and Dini, 1995). In this thesis, LMDDBS is defined as a distributed database system where a huge part of the processing environment is a mobile environment in which the database is stored and accessed by a large number of highly mobile nodes, which are distributed over wide areas with the aim of accessing desired data to process updates from anywhere and at any time. Typical examples of such systems include mobile health care, mobile data warehousing, news gathering, military inventory control, and traffic control management systems.

1.0.1 Architecture of LMDDBS

The general architecture of LMDDBS is illustrated in Figure 1.0.1.1. In this architecture, a set of general purpose devices (PC, workstations, etc.) are interconnected through a wired network. These devices are categorized into Fixed Hosts (FH) and Mobile Support Stations (MSS). MSS can connect to some of the other nearby MSSs via wireless communication links (radio or satellite links). A large number of portable devices, referred to as Mobile Hosts (MH) are connected to the wired network components only through MSSs via wireless channels. MHs are battery powered portable computers that move around freely in a geographic mobility domain, which is the total area covered by all MSSs (Dunham and Helal, 1995**;** Madria and Bhowdrick, 2001).

To support the mobility of MHs, the entire mobility domain is divided into smaller areas called cells. A particular MSS manages each cell. MSSs are augmented with wireless interfaces to communicate with mobile hosts, which are located within its cell. The mobile discipline requires that a MH must have unrestricted movement within the mobility domain (inter-cell movement) and must be able to access desired data from any cell. The process of crossing a cell boundary by a MH and entering into another cell is referred to as a handoff. This hand-off involves establishing a new communication link with the MSS that is responsible for managing the new cell.

Within this architecture, shared data are stored and controlled by a number of Distributed Database Management Systems (DDBMSs) in both fixed hosts and mobile support stations, and by a large number of MDBMSs in mobile hosts.

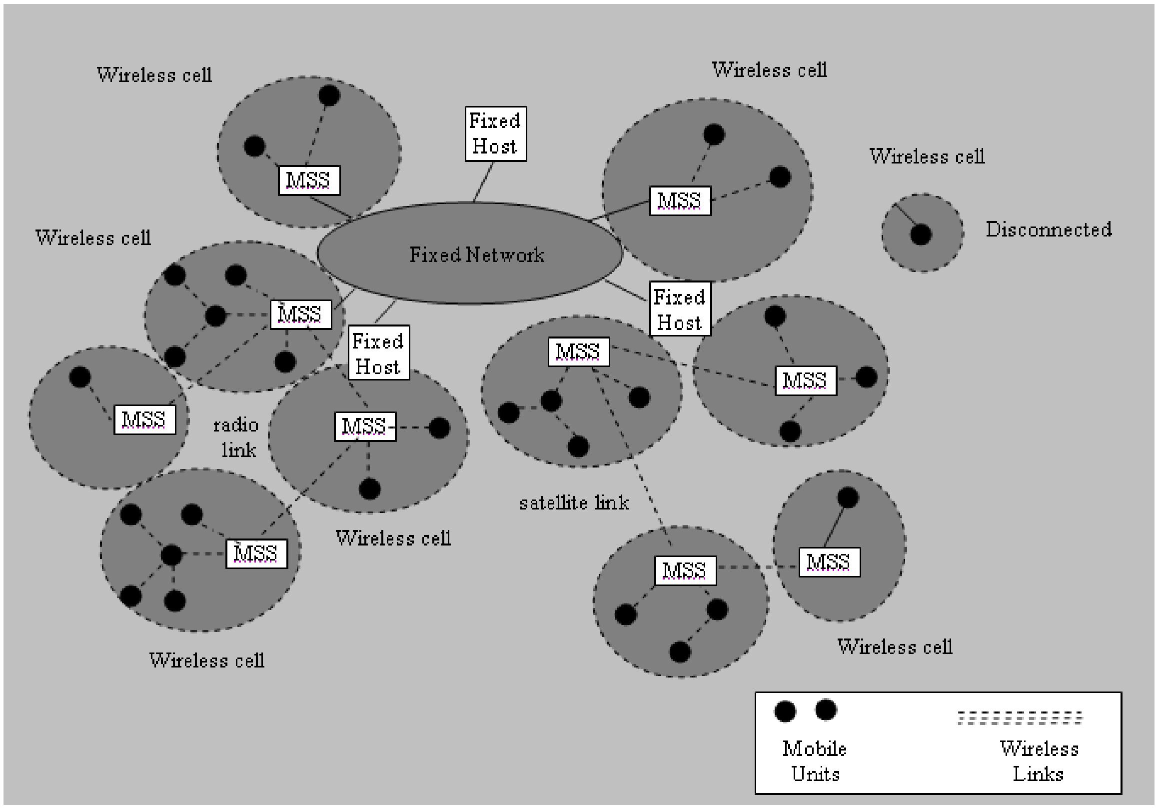

Figure 1.0.1.1 A General Architecture of LMDDBS (Adapted from Dunham and Helal (1995), Lim and Mok (1998) and Madria and Bhowdrick (2001))

1.0.2 Characteristics of LMDDBS

Although the LMDDBS is essentially a distributed database system, there are some characteristics that make the system unique and involve designing of a new data management solutions in such systems (Madria et al., 2002; Schneiderman, 2002). In addition to the characteristics of wide areas that are covered by LMDDBS and a large number of mobile users, there are many characteristics related to wireless communication and mobile units. These characteristics are described as follows.

1.0.2.1 Wireless communication's characteristics

Mobile hosts communicate to the other hosts via wireless networks. Compared to wired networks, wireless networks have fundamentally different characteristics that affect data sharing in LMDDBSs. These characteristics include:

- **High communication latency.** Latency caused by the processes unique to the wireless medium, such as coding data for wireless transfer, and tracking and filtering wireless signals at the receiver.
- **Intermittent wireless connectivity.** Can be intentional or unintentional. Unintentional disconnections happen in areas where wireless signals cannot reach, e.g., elevator shafts or subway tunnels. Intentional disconnections occur by user intent, e.g., during an airplane takeoff, or when the mobile device is powered down.
- **Limited bandwidth.** Wireless networks deliver lower variable bandwidth than the wired networks. The wireless network does not have the capacity as the wired network. For example, a wireless network has bandwidth in the order of 10Kbps or a wireless local area network (WLAN) has bandwidth of 10Mbps; while gigabits (Gbps) are common in wired LAN (Schneiderman, 2002). Therefore, it can take longer time for a mobile host to transfer the same amount of information via the wireless network than the wired network. Consequently, the wireless network introduces more overhead in transaction processing.
- **Unreliability.** A wireless network has high error-rates, and the bandwidth of a wireless network is variable. Due to errors during data transmission, the same data packages are required to re-transmit many times, thus, extra overhead in communication and higher cost. Due to the varying bandwidth, it is hard to estimate the time required to completely transmit a data package from/to a mobile host. These problems will affect the data availability at the mobile hosts and exchanging of recent updates that are performed to the shared data.

1.0.2.2 Mobile units' characteristics

Similarly, mobile hosts behave differently and have fundamentally different limitations than stationary hosts, including:

- **Power limitations.** Mobile units generally operate on limited battery power, before the batteries have to be recharged. This limited power supply is one of the major disadvantages of mobile computing devices. The energy consumption of a mobile device depends on the power of electronic equipments installed on the mobile device, for example types of hard disks or CPU. Moreover, the battery life also depends on the number of applications and the application types that operate on the mobile devices (Flinn and Satyanarayanan, 1999). Data update operations that are being carried out at a mobile host can be interrupted or re-scheduled if the mobile host is exhausting its power supply.
- **Limited functionality.** Small size and weight of mobile devices mean that the functionality of these devices is also limited in terms of the small storage capacity, graphical user interface, the application functionalities, and the processing power. Therefore, a mobile host may be unable to perform some of data update operations, or requires longer processing time to perform these operations.
- **Frequent disconnections.** Mobile units do not stay connected to the network continuously (as fixed hosts do), For example, a mobile host may move and become disconnected from, or change its point of attachment to, the network. While being disconnected, the mobile host will not be able to communicate to other hosts for sharing of data. If the mobile host holds vital shared data, it can block data updates on other hosts. Furthermore, the duration of a disconnected period of a mobile host is not always as planned, i.e., it can be longer than expected. These frequent disconnections affect exchanging of recent data that affected by the update operations during the disconnection time between the mobile host and other hosts.

Figure 1.0.2.1 summarizes the characteristics of both wireless communication and mobile units.

- **High communication latency.**
- **Intermittent wireless connectivity.**
- **Low bandwidth.**
- **Unreliable communication.**
- **Expensive communication.**
- **Limited battery power.**
- **Limited Resources.**
- **Frequent disconnections.**
- **Limited reliability.**

Figure 1.0.2.1 The Characteristics of Wireless Communication and Mobile Units. (Madria et al., 2002)

1.0.3 Replication in LMDDBSs

Replication is the process of maintaining multiple copies of data that are stored on different sites (Son, 1988; Saito and Shapiro, 2005). Replication is an important mechanism for any system that attempts to efficiently support shared data in mobile and weakly-connected environments (Ratner et al., 2001; Cetintemel et al., 2003). The reason is that disconnected or poorly connected hosts must rely primarily on local resources. Moreover, the monetary costs of communication when mobile, combined with the lower bandwidth, higher latency, and reduced availability, effectively require that important data be stored locally on the mobile host. In the case of shared data, between multiple mobile hosts or between mobile and fixed hosts, replication is often the best and sometimes the only viable approach.

In LMDDBSs, the replication of data will be essential to support data sharing over wide-area networks by improving the availability of data, increasing the performance, and supporting mobility (Helal, Heddaya, and Bhargava, 1996; Cetintemel et al., 2003; Goel and Buyya, 2006; Deris, Abawajy, and Mamat, 2008) as follows.

- **Availability.** By replicating the data that are needed in each host, if a host fails, this will not affect the other host and the system can operate using replicated data, thus increasing availability and fault tolerance.

- **Performance.** Performance refers to the response time and the throughput of the system. When data are stored only at specific hosts (i.e. servers), these hosts can be a bottleneck if too many requests need to be served at the same time and the whole system slows down, i.e. slow response time and limited throughput capacity in terms of requests per second. By offering multiple replicas at multiple locations, requests can be served in parallel and one replica provides data access to a smaller community of users, thus increasing the performance of the system.
- **Mobility.** Replication in a LMDDBS supports mobility because the large number of users and their updates move from one cell to another while accessing the database. Unless the data moves with the user in the form of a cache or mobile disk storage, data will need to be re-transmitted back and forth. Retransmission wastes the limited communication bandwidth and slows down update processing because of retransmission delays. Moreover, re-transmission/reception results in faster power consumption of the limited battery resource.

These benefits of replication in LMDDBSs come with the overheads of:

- **Creating and allocating the replicas.** This overhead concerns the allocation of replicas among the large number of sites that are distributed over the wide-area networks (Abawajy, Deris, and Omer, 2006).
- **Maintaining consistency among replicas.** If the application has read-only nature, replication can greatly improve the performance. But, if the application needs to process update operations, replicas at different sites need to be kept consistent for application correctness. Thus, realizing the benefits of data replication is difficult since the correctness of data must be maintained. Accordingly, the main challenge of database replication is to keep the data copies consistent in the presence of updates (Nicola and Jarke, 2000).
- **Supporting large number of updatable replicas.** The distributed applications over wide areas may scale to thousands of replicas located in every part of these areas. The replication system must provide the ability for all replicas to generate updates (even though some may never do so) as well

as the ability to access updates that occurred on other replicas. However, supporting access to updates that occurred on these replicas without affecting the performance represents a challenge in large scale systems (Golding, 1993 ; Ratner et al., 2001)

1.1 Problem Statement

This research addresses the problems of maintaining consistency and improving availability of replicated data in LMDDBSs. This type of systems is characterized by a large number of updates (e.g. tens of updates may be performed on each replica at any period of time) that occur in a large number of replicas (i.e. thousands of replicas), which are distributed over both mobile and fixed environments. The characteristics of LMDDBSs, and the ability of the highly mobile users in such systems to issue a large number of concurrent update operations on the shared replicated data, which occur during the disconnection periods extremely aggravate these problems than in the ordinary replicated database systems, which have a small number of replicas. The concurrent updates during disconnection periods influence availability and consistency of the replicated data by leading to unavailability of recent updates, and a divergence in the database states (i.e. the data that are stored in the database at a particular moment in time) within replicas that are hosted in mobile networks, and between these replicas and the replicas that are hosted in fixed networks.

Long disconnection periods in communication affect data consistency and availability by delaying the process of propagating recent updates that occurred at disconnected mobile hosts to the other hosts (either stationary or mobile hosts) for a long period. Also, disconnection periods hinder the mobile hosts by delaying the process of receiving recent updates that occurred in other hosts that are located in both wired and wireless networks. The duration of disconnected or connected periods is not always as planned, i.e., varying in time, due to many factors, for example, the limited battery life of mobile hosts, and the unstable bandwidth of wireless networks. This might result in longer delay time or interruption periods for sharing recent updates through updates propagation.

To cope with these problems (i.e. maintaining consistency and improving availability), several replication strategies are proposed. These strategies are divided into optimistic and pessimistic strategies (see section 2.3.1).

Pessimistic strategies such as primary copy, ROWA, and voting restrict updates to a single replica or a specific number of replicas. They require that update operations be synchronized in order to ensure that replicas are mutually consistent. This makes these strategies well suited only for environments where the network is stable, hosts have well known locations, and message delivery is guaranteed.

However, strong consistency guarantee provided by these strategies is inadequate for large scale systems, especially wide area systems with mobile devices (Adly, 1995; Saito and Shapiro, 2005; Baliga, 2006). Moreover, such strategies requiring synchronization among a large number of replicas are difficult to be implemented in wide areas and mobile environments for the following reasons:

(i) They suffer from high latency and low throughput, since links tend to be slow and unreliable and a large number of replicas generate considerable traffic over the network.

(ii) They reduce the system availability when one or more nodes fail, or when the network is partitioned.

(iii) They lock or restrain access to replicas during operation execution.

(iv) Mobile clients are mostly disconnected and connect often with low bandwidth connections. Hence these mobile clients are unable to participate in synchronous algorithms.

(v) A pessimistic replication strategy, attempting to synchronize with an unavailable site, would block indefinitely according to the intermittent connectivity.

Accordingly, Pessimistic strategies scale poorly in the wide area systems. It is difficult to build a large, pessimistically replication system with frequent updates because its throughput and availability suffer as the number of sites increases.

Optimistic strategies, in contrast, allow multiple replicas to be concurrently updatable. Then, the replicas perform a process of reconciliation to make their copies consistent. Accordingly, these strategies enable the users to access any replica at any time, which means higher write availability to the various sites and better response time for both read and update operations. Thus, these strategies are suitable for mobile environments. However, they provide weak consistency guarantee based on the fact that replicas can tolerate inconsistencies for at least some time period before the reconciliation process occur. To provide such weak guarantee, these strategies should provide a propagation mechanism which ensures that updates are efficiently propagated to all replicas during the reconciliation processes.

In the reconciliation process, most of the proposed optimistic strategies rely on version vectors (see section 2.7.2) (Kang, 2004) either for determining the set of updates to be exchanged (such as Petersen et al. (1997) and Yu and Vahdat (2001)) or for detecting conflicts (such as Ekenstam et al. (2001) and Ratner, Reiher, and Popek (2004)). Unfortunately, strategies based on version vectors have fundamental limitations with respect to scalability, since the size of version vector grows as the number of replica sites increases and each version vector entails complicated management in site addition/deletion. On the other hand, some of the optimistic strategies (such as Kang, Wilensky, and Kubiatowicz (2003)) rely on keeping new updates that occurred in each replica in a log in order to propagate them to the other replicas at a later time. The update log is, however, the main source of memory overhead in an optimistic replication system. This is especially relevant in mobile environments, where the memory resources of mobile devices are typically scarce (Barreto, 2003).

Some of the optimistic strategies implement updates ordering as another mean instead of one-copy serializability (see section 2.7 for updates ordering and section 2.4 for one-copy serializability), which represents strong consistency guarantee that is provided (achieved) by pessimistic strategies (Jia and Zho, 2005), which are difficult to be implemented in mobile environments according to the abovementioned reasons.

These optimistic strategies implement updates ordering through ensuring causal ordering of updates. However, the existing causal ordering approaches for maintaining consistency (i.e. causal ordering) such as Petersen et al. (1996), Adly and Nagi (1995), and Cetintemel et al. (2003) are not scalable because they require too much information be sent around between sites for maintaining causal ordering . The scalability problem is especially important in mobile computing environment due to limitations of available bandwidth, memory and energy supply, as described earlier. Hence, there is a need for an implementation of causal ordering of updates that has low communication, computation, and memory overheads.

Many optimistic replication strategies have been proposed for mobile environments such as TLRSP (Zhiming, Xiaofeng, and Shan, 2002), Roam (Ratner, Reiher, and Popek, 2004), Cedar (Tolia, Satyanarayanan, and Wolbach, 2007). However, these strategies have not explicitly addressed the following issues:

1. The issues of consistency and availability of data in large scale systems that demand, in its nature, large number of updatable replicas, which are distributed among both stationary and mobile environments.
2. The issue of ensuring rapid propagation of updates between large number of mobile replicas and fixed replicas in order to bound the inconsistency between these replicas. Unfortunately, the proposed optimistic strategies provide no bounds on the inconsistency of the replicated data.
3. The issue of availability from a point of view that the mobile host needs to receive recent updates that have been performed on the other hosts (in both fixed and mobile networks) to the data items that represent an interest to the user. Regarding this issue, our research looks abstractly at the availability as obtaining recent updates that occurred in the other replicas, while it looks abstractly at the consistency as an agreement between a set of replicas on an identical set of data values for the same replicated data items. Accordingly, the data that are stored on a given host may be consistent with other replicas, although it may not receive last updates (i.e. availability). For example, assume that there are 10 replicas, and the consistency agreement between them is that each replica should have identical values as the values that are stored in replica number 5. Based on this agreement, these replicas are assumed that they have consistent data initially. In the case of two replicas generated updates, and these updates did not reach replica number 5 and replica number 3, this means that replica number 3 still having consistent data as replica number 5, although it did not receive recent updates that are generated on those two replicas.

Thus, the previous issue dealing also with bounding the unavailability of recent updates.

4. The issue of providing a unified causal ordering of large number of updates that are performed optimistically in a large number of both fixed and mobile replicas.

Therefore, this research comes to a conclusion that additional research toward a new replication strategy is needed to investigate and address these issues in LMDDBSs that consist of large number of updatable replicas.

1.2 Research Questions

Based on the aforementioned issues, our research questions are:

1. What are the specifications of a suitable replication strategy that will maintain consistency and improve availability of replicated data in LMDDBSs that are characterized by frequent disconnections and frequent updates of large number of replicas?
2. What are the required components of the replication architecture that is needed to support the suitable replication strategy?
3. How recent updates are propagated between the components of the replication architecture in a way that achieves the consistency and availability of interested data to mobile users?
4. How can we ensure that all updates that occurred in both fixed and mobile networks will be stored on all replicas in the same unified total order?

The answer to question 3 should consider the characteristics of LMDDBS, especially the frequent disconnections of mobile hosts. These characteristics require that limiting the bound of inconsistency and unavailability of recent updates by considering disconnection periods of mobile hosts. Accordingly, we need to ensure rapid propagation of updates and providing mobile hosts with the ability of automated propagation of their recent updates once they are connected with the fixed network. Also, we need to enable fixed hosts to provide the connected mobile hosts with recent updates that are generated on the other hosts in an automated manner. Accordingly, the following sub questions arise:

(a) How updates propagation in LMDDBSs is performed in order to ensure fast propagation of updates?

(b) How updates propagation can be automated in a manner that ensures all recent updates are propagated to the other replicas?

1.3 Research Objectives

The aim of this research is to propose a new replication strategy that maintains consistency and improves availability of replicated data in LMDDBSs with large number of updatable replicas. In order to realize this aim, the specific objectives have been specified as follows.

1. Developing a new replication architecture to provide a solid infrastructure for improving availability, maintaining consistency, and supporting large number of replicas (i.e. thousands of replicas) by determining the required components in LMDDBSs that are involved in the replication process.
2. Proposing a scalable updates propagation protocol for exchanging distributed updates that occurred on large number of replicas in a manner that achieves load balance between the components of the replication architecture, and reduces the propagation delay in order to bound inconsistency between these components and unavailability of last updates.
3. Providing a replication method to allocate replicas to mobile hosts and automate propagation of updates between the components of the replication architecture in a manner that ensures propagating only recent updates among the large number of replicas, which leads to better utilization of the connection time.
4. Proposing updates ordering mechanism that ensures a unified causal ordering for all updates that occurred in all replicas in LMDDBS.

Accordingly, this research is concerned with providing a useful replication strategy that performs in a good manner under a large workload in terms of large number of replicas and updates.

1.4 Scope

Among other data management issues (e.g. concurrency control, query processing and optimization, transaction management, and recovery) in mobile database systems, the research concerns on data replication issues. These issues include:

- Appropriate architectures for distributing both mobile and fixed replicas among the areas that are covered by the replication systems.
- Allocation of the replicas to the different hosts that are involved in the replication process.
- The levels of the consistency of replicated data that satisfy the needs of mobile database systems.
- Efficient mechanisms and proper structures for exchanging of recent updates between replicas.
- Ordering mechanisms and constraints that are required to provide sensible order for all updates occurred in the replication system.
- Interaction between replicas and restriction of updates to be performed on specific replicas.
- Automation of the processes replicas allocation, updates propagation, and updates ordering.
- Scalability of the replication systems in mobile environments in terms of the system size (covering new areas) and supporting new replicas.

1.5 Organization of the Thesis

The remainder of the thesis is organized as follows.

Chapter 2 presents a literature review in the related areas of data replication and previously proposed strategies that devoted to mobile environments and large-scale systems.

Chapter 3 presents components of the proposed replication strategy, a formal system model and an abstract replication model for LMDDBSs, and details of replicas allocation.

Chapter 4 contains a detailed description of a structure and mechanisms that establish a Wheel-based Updates Propagation Protocol, and details of the performance evaluation of this protocol.

Chapter 5 discusses an implementation of the Wheel-Based propagation protocol via IIRA-Based Propagation System, data structures and algorithms that are provided by this system, a Stochastic Perti Net model that provides an analysis of the behavior of the replication system.

Chapter 6 presents a detailed description of a hierarchical Multi-Criteria updates ordering mechanism including formal models of both casual and total ordering, algorithms for providing a unified ordering and correctness proof, and performance issues.

Lastly, Chapter 7 concludes the thesis by presenting a summary of the main conclusions, focusing on contributions, limitations and future work.