Journal of Engineering Science and Technology
Vol. 6, No. 3 (2011) 268 - 299

A NOVEL REPLICATION STRATEGY FOR LARGE-SCALE MOBILE DISTRIBUTED DATABASE SYSTEMS

ASHRAF AHMED, DANAPAL DURAI DOMINIC*, AZWEEN ABDULLAH

Department of Computer & Information Sciences, Universiti Teknologi PETRONAS
Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia
*Corresponding Author: dhanapal_d@petronas.com.my

Abstract

An important challenge to database researchers in mobile computing environments is to provide a data replication solution that maintains the consistency of replicated data. The paper addresses this problem for large scale mobile distributed database systems. Our solution represents a new binary hybrid replication strategy in terms of its components and approach. The new strategy encompasses two components: replication architecture to provide a solid infrastructure for distributing replicas and updates propagation protocol to propagate recent updates between the components of the replication architecture in a manner that achieves the consistency of data. The new strategy is a hybrid of both pessimistic and optimistic replication approaches in order to exploit the features of each. These features are supporting large number of replicas and lower rate of inconsistencies between them as well as supporting the mobility of users. The proposed replication strategy is compared with a baseline replication strategy and shown that it achieves updates propagation delay reduction, less communication cost, and load balance as important requirements for maintaining consistency in large scale environments with large number of replicas and highly mobile users.

Keywords: Pessimistic Replication, Optimistic Replication, Data Consistency, Updates Propagation, Propagation Mechanism.

1. Introduction

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 [1, 2]. As mobile computing devices (e.g., laptop, PDA, and cell phones) become more and more common, mobile databases are becoming popular [3]. Mobile database has been 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 corporate data [4, 5].

Mobility of users and portability of devices pose new problems in the management of data [6, 7], including transaction management, query processing, and data replication. Therefore, mobile computing environments require data management approaches that are able to provide complete and highly available access to shared data at any time from any where. One way to achieve such goal is through data replication techniques. The importance of such techniques is increasing as collaboration through wide-area and mobile networks becomes prevalent [8]. However, maintaining the consistency of replicated data among all replicas represents a challenge in mobile computing environments when updates are allowed at any replica.

This paper addresses the problem of maintaining consistency of replicated data for large scale distributed database systems that operate in mobile environments. This type of systems is characterized by a large number of replicas (i.e., hundreds of replicas) and a large number of updates (i.e., tens of updates per data items are expected at any period of time) are performed on these replicas. Examples of such systems include mobile health care, mobile data warehousing, news gathering, and traffic control management systems. In such type of mobile environments, the concurrent updates of large number of replicas during the disconnection time influences consistency of the replicated data by leading to divergence in the database states (i.e., the data that are stored in the database at a particular moment in time).

To cope with this problem (i.e., maintaining consistency), several replication strategies are proposed. These strategies are divided into optimistic and pessimistic approaches [9-11]. Pessimistic replication avoids update conflicts by restricting updates to a single replica based on a pessimistic presumption that update conflicts are likely to occur. This ensures data consistency because only one copy of the data can be changed. Primary-copy algorithms [12] are an example of pessimistic approaches. However, pessimistic approaches cannot be used directly in large-scale mobile environments, because they are built for environments in which the communication is stable and hosts have well known locations. An optimistic replication, in contrast, allows multiple replicas to be concurrently updatable based on an optimistic presumption that update conflicts are rare. Conflicting updates are detected and resolved after they have occurred. Therefore, this schema allows the users to access any replica at any time, which means higher write availability to the various sites. However, optimistic approaches, which include [13-18] can lead to update conflicts and inconsistencies in the replicated data. Moreover, these strategies have not explicitly addressed the issues of consistency and availability of data in large scale distributed information systems that operate in mobile environments.

Therefore, this paper comes to a conclusion that additional research toward a new replication strategy is needed to investigate and address data consistency issue in large-scale mobile environments. Accordingly, the paper proposes a new replication strategy that acts in accordance with the characteristics of large scale mobile environments (i.e., large number of updateable replicas).

This paper is organized as follows. The next section provides the related work. Section 3 describes the proposed replication strategy. Section 4 gives the details of the performance evaluation. Section 5 concludes the paper.

2. Related Work

Using optimistic replication in mobile environments has been studied in several research efforts. ROAM [13, 14] is an optimistic replication system that provides a scalable replication solution for the mobile user. ROAM is based on the Ward Model [15]. Replicas are grouped into wards (wide area replication domains). All ward members are peers, allowing any pair of ward members to directly synchronize and communicate. Each ward has a ward master that maintains consistency with the other wards. Updates are exchanged within each ward (i.e., between ward members) and among wards (i.e., between ward masters) using ring topology. Accordingly, Roam employs optimistic replica control mechanism that ensures an eventual convergence for replica updates to maintain the consistency within each ward and among wards. ROAM tries to provide high scalability without discussing a mechanism of ensuring fast propagation of large numbers of updates that can be performed in replicas that are distributed over wide geographic areas.

A multi-master scheme is used in [16], that is, read-any/write-any. To reach an eventual consistency in which the servers converge to an identical copy, an adaptation in the primary commit scheme is used. In this adaptation, a server chosen as the primary has the responsibility to synchronize and commit the updates. The committed updates are propagated to the other servers. This schema inherits the drawbacks of primary-copy algorithm since it relies on a selected server that is responsible for synchronizing all updates between the different replicas.

A hybrid replication strategy is presented in [17] that have different ways of replicating and managing data on fixed and mobile networks. In the fixed network, the data object is replicated synchronously to all sites in a manner of logical three dimensional grid structure, while in the mobile network, the data object is replicated asynchronously at only one site based on the most frequently visited site. The synchronous replication hinders the fixed network to be scalable to wide areas.

Cedar [18] uses a simple client-server design in which a central server holds the master copy of the database. At infrequent intervals when a client has excellent connectivity to the server (which may occur hours or days apart), its replica is refreshed from the master copy.

A mobile database replication scheme called Transaction-Level Result-Set Propagation (TLRSP) is proposed in [19]. Each fixed and mobile units store a replica of the data. When the data in both mobile and fixed nodes are consistent, a mobile host is said to be operating in consistent state. When the mobile host is connected to a host in fixed network, it sends the locally committed transactions to the fixed host for conflict detection. The fixed host updates those transactions that passed the validation test and the recently updated copies of the objects are forwarded to the mobile host to refresh its local copies.

In summary, we argue that existing replication strategies are not coping well with characteristics of large-scale mobile systems containing large number of geographically distant replicas. Accordingly, such systems demand new solutions for addressing data consistency through ensuring fast propagation of recent updates as well as supporting scalability for encompassing new replicas when the replicated system covers new geographic areas.

3. Replication Strategy

The proposed replication strategy encompasses two components: replication architecture and updates propagation protocol. The purpose of the replication architecture is to provide a comprehensive infrastructure for improving data availability and supporting large number of replicas in mobile environments by determining the required components that are involved in the replication process. The purpose of the propagation protocol is to transfer data updates between the components of the replication architecture in a manner that achieves the consistency of data and improves availability of recent updates to interested hosts.

The new strategy is a hybrid of both pessimistic and optimistic replication approaches. The pessimistic approach is used for restricting updates of infrequently changed data to a single replica. The reason behind this restriction is that if the modifications of these data are allowed on several sites, it will influence data consistency by having multiple values for the same data item (such as multiple codes for the same disease or multiple codes for the same drug). On the other hand, the optimistic replication is used for allowing updates of frequently changed data to be performed in multiple replicas. The classification into frequently and infrequently changed data is specified according to the semantic and usage of the data items during the design phase of the database.

3.1. System model

This research considers a large-scale environment that consists of Fixed Hosts (FH), Mobile Hosts (MH), a replica manager on each host, and a replicated database on each host. A replicated database is called mobile database when it is stored on a mobile host. A part of fixed hosts represent servers with more storage and processing capabilities than the rest. The replicated database contains a set of objects stored on the set of hosts. The database is fully replicated on the servers, while it is partially replicated on both fixed and mobile hosts. Update can take place at any host. Update information is sent to other hosts in a form of message. The information of hosts and their replicated data is stored on an object called hosts-Obj, which is replicated in each server. In this paper, the terms replica and host will be used interchangeably because each has a replica.

Definition 3.1.1 An object O is the smallest unit of replication and it represents a tuple $O = <D, R, S>$, where $D = \{d_1, d_2, \ldots, d_n\}$ is a set of data items of the object O, $R = \{r_1, r_2, \ldots, r_m\}$ is a set of replicas of O, and S is the state of the object O.

Definition 3.1.2 The state S of an object O is a set consisting of states that identifies current values for each data item $d_i \in D$, i.e., $S = \{s_1, s_2, \ldots, s_n\}$.

Definition 3.1.3 A replica R is a copy of an object stored in a different host and is defined as a function as follows. For a set of updates U that is performed on a set of objects $\bar{O}$, the function $R : U \times \bar{O} \rightarrow S$ identifies a new separate state $s_i \in S$ for an object $O \in \bar{O}$ as a result of performing update $u \in U$ on an object O in a different host.

Definition 3.1.4 A replicated data item $d_i \in R$ is consistent if and only if all updates that are performed on d_i in other replicas (either in fixed hosts or mobile hosts) are merged with the updates that are performed on d_i in R.

Definition 3.1.5 A replica R is consistent if and only if each data item $d_i \in R$ is consistent.

3.2. Replication architecture

The proposed replication architecture (Fig. 1) considers a total geographic area called the master area that has a server called Master Server (MS) and a set of fixed hosts. The master area is divided into a set $Z = \{z_1, \ldots, z_n\}$ of zones. Each zone has a server called Zone Server (ZS) and a set of fixed hosts and it consists of a set $C = \{c_1, \ldots, c_m\}$ of smaller areas called cells. Each cell represents an area, where the mobile users can perform their duties at a particular period of time before moving to another cell. Each cell has a server called Cell Server (CS). In this architecture, the network is divided into fixed network and mobile network. The fixed network consists of fixed hosts and wired local area network to connect them in the master area. Also, it includes wide area network to connect the master server with zone servers, and to connect zone server with underlying cell servers. The cell server is augmented with a wireless interface and acts as a mobile support station for connecting mobile hosts to the fixed network. On the other hand, the mobile network consists of wireless network and mobile hosts in the cell area. To provide more flexibility and application areas for this architecture, replicas are divided into three levels:

Master Level: In this level, the replica that is stored on the master server must be synchronized with replicas from the zone level. The master server is responsible for synchronizing all changes that have been performed on both infrequently changed data and frequently changed data with the lower level.

Zone Level: In this level, each replica must be synchronized with replicas from the lower level. The zone server is responsible for synchronizing all intra-level changes with the master server.

Cell Level: Each replica in this level is updated frequently, and then synchronized with the cell server's replica and in turn the cell server synchronizes all intra-level data with the zone server.

In this architecture, initially, the database is stored on the master server. When dividing the master area into multiple zones, a replica of that database is distributed to zone servers. Similarly, new replicas are created when dividing the zone area into multiple cells to represent the cell servers and when registering new mobile hosts and fixed hosts in the replicated system. The information of each new replica is stored on the host's object in the server of the area where the replica is created and then it is replicated to other servers. This information includes the Host ID, Host Type (FH, MH, CS, ZS, and MS), Region where it is registered, and the replicated objects on that host.

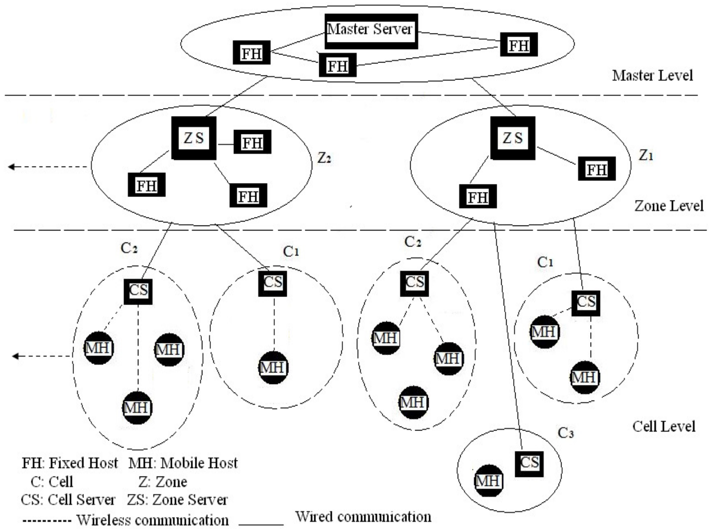


Fig. 1. The Replication Architecture for Large-Scale Mobile Environments.

3.3. Wheel-Based updates propagation protocol

This section provides the details of the proposed protocol for updates propagation through the components of replication architecture. The protocol consists of logical structure for arranging replicas and propagation mechanisms for exchanging updates among these replicas. The logical structure is a wheel-like structure that organizes replicas according to their types, areas where they inhabit (cell, zone, and master areas), and responsibility with regard to updates propagation. The propagation mechanisms act as interaction mechanisms between replicas for propagating recent updates from their sources to other replicas that are distributed over the wheel. Accordingly, the resulted protocol is called Wheel-Based updates propagation protocol.

3.3.1. Updates propagation wheel

The logical structure that is involved in the updates propagation is a water wheel inspired structure called updates propagation wheel, which represents a logical structure for exchanging recent updates between the hosts that are distributed over the replication architecture.

The applying of the water wheel structure here is arising from its general design (Fig. 2) and functionality. The water wheel structure [20] links an axle (i.e., acts as a central point) with multiple buckets (act as points) that are located in different directions on a circular rim through spokes. The functionality of the water wheel depends on the rotation of the buckets that are located on the rim after they are filled by the water. This rotation leads to the revolution of the whole wheel including the centre point. To apply this idea, updates propagation wheel is structured in a manner that includes the basic components of the water wheel with different explanations and functionalities. Table 1 depicts water wheel features that applied and mapped to the proposed architecture.

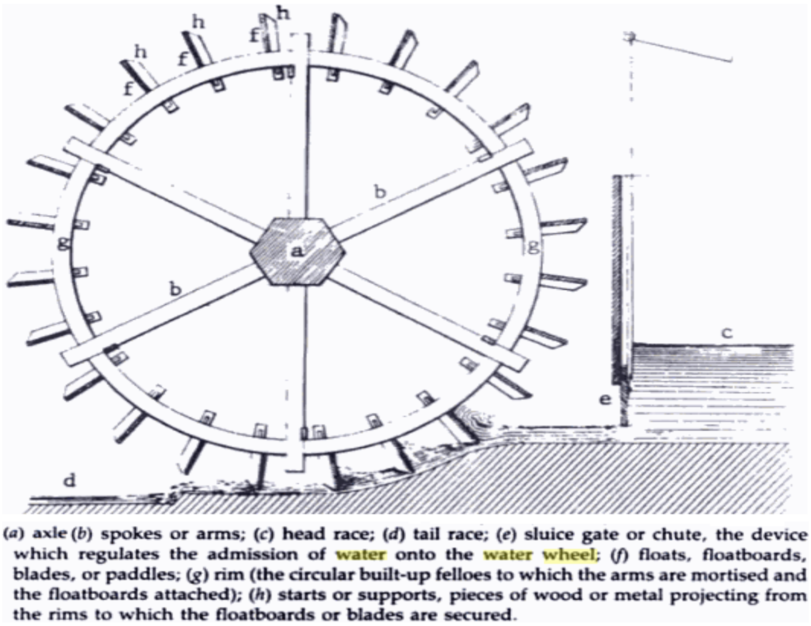


Fig. 2. Water Wheel Structure (Adapted from [20]).

Table1. Mapping Water Wheel Architecture to Updates Propagation Architecture.

Characteristic	Water wheel	Propagation wheel
Resource Handled	Water	Updates
Source	The sources include River.	Miniworld (the part of the external world that its data are represented in the database)
Service	Include transferring water to river strand	Transfer updates to another host in the wheel
Wheel Centre	Axle (shaft)	The master server
Spokes	Wooden or metal arms	Network links
Rim	The circular built-up felloes to which the arms are mortised and buckets attached	Virtual circular paths on which the hosts from same type are located
No of rims	1 physical rim	3 virtual rims
Wheel rotation	In one direction	Randomly on two directions
Transferring facility	buckets	Servers, Fixed hosts, Mobile hosts
Facility location	Buckets are arranged on the outside rim forming the driving surface	Hosts are arranged on virtual rims. The most outer rim contains MHs, which form the driving surface

Given N replicas of the database, the propagation protocol organizes them logically into wheel structure based on their areas and types as shown in Fig. 3. The following definition will formally define the propagation wheel (PW).

Definition 3.3.1.1 PW is 9-tuple $<F, H, S, L, R, P, U, M, T>$, where:

$F= \{FH_1,\ldots, FH_n\}$ is a finite set of fixed hosts that act as fixed points (i.e., buckets) that are distributed over the wheel.

$H= \{MH_1,\ldots, MH_h\}$ is a finite set of mobile hosts that act as mobile buckets over the wheel.

$S= \{s_1,\ldots, s_s\}$ is a finite set of servers that act as fixed centre points where a sub set of F, H, or S can be connected to each centre point.

$L = \{l_1,\ldots, l_l\}$ is a finite set of communication links that act as spokes for linking the different points distributed over the wheel.

$R = \{r_1, r_2, r_3\}$ is a finite set of virtual circular rims that act as a collection of points that have same area.

$P= \{p_1, \ldots, p_p\}$ is a finite set of parts that constitute each rim. Each part is called sector.

$U: F \cup H \cup S \rightarrow \{1, 2, 3,\ldots, k\}$ is a function for assigning a unique identifier serially for each host in the wheel according to its type.

$M = \{m_1, m_2, m_3\}$ is a finite set of mechanisms for exchanging updates between the different points in the wheel.

$T= \{t_1,\ldots, t_{n+h+s}\}$ is a finite set of total number of updates that are currently stored in each host (such as water in each bucket) which measures the consistency of updates on that host by comparing it with the other hosts. A propagation mechanism in M is required to make this total number to be identical in all hosts share same data items.

Centre points. As depicted in Fig. 3, the different types of hosts are represented by circles in the propagation wheel. Some hosts act as centre points where multiple spokes are collected on them. These points represent the servers of the different areas. Accordingly, these points can be classified into master server, zone servers, and cell servers according to their areas. Such points are linked through spokes to a set of either other centre points or ordinary points (i.e., points act as either fixed or mobile hosts), which located on virtual circular rims as follows:

1. Master server: It acts as the main centre point that is linked with secondary centre points, which represent the zone servers and ordinary points that represent the fixed hosts on the master area.
2. The zone servers are linked with secondary centre points that represent the cell servers and ordinary points that represent the fixed hosts on the zone area.
3. The cell servers are linked with ordinary points that represent the mobile hosts and fixed hosts on the cell area.

In this wheel, both centre and ordinary points represent the different types of hosts of the replicated system, while the spokes between them represent the network connections (channels).

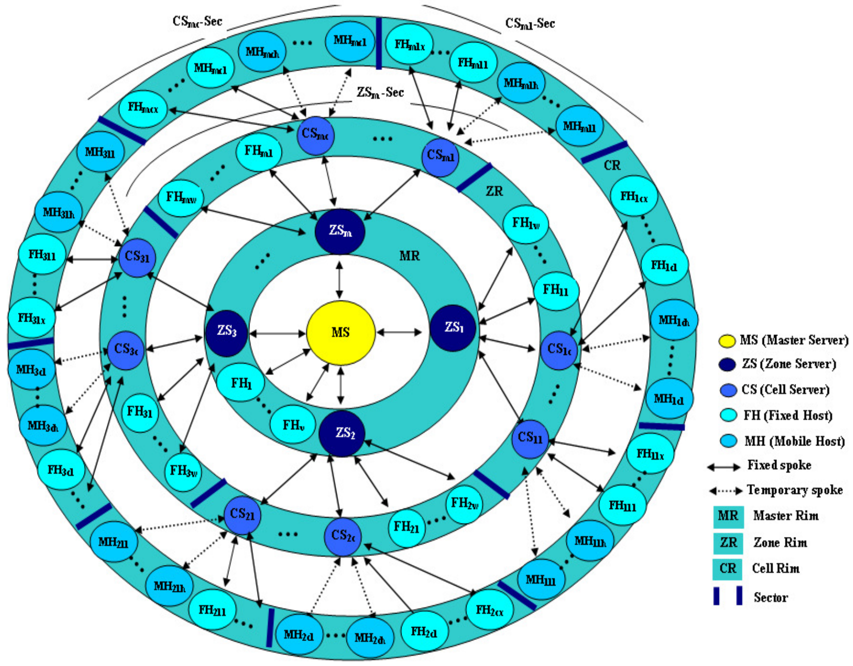


Fig. 3. Updates Propagation Wheel.

Rims. They are formed by the hosts that have same area despite their directions. Accordingly, we have three rims as follows:

(i) Master Rim: It contains all zone servers as well as fixed hosts on the master area. The master server is responsible for all hosts exist in this rim in that it receives their updates and sends their missed updates to them.

(ii) Zone Rim: It contains all cell servers as well as fixed hosts on the zone area. The zone server is responsible for a part of this rim called sector, which represents the cell servers and fixed hosts that are located in its area.

(iii) Cell Rim: It contains all mobile hosts and fixed hosts in the cell area. The cell server is responsible for a part of the cell rim which represents the fixed hosts and mobile hosts that are located in its cell.

Thus, we called the relation between the hosts on the three rims as Responsible-For and it is defined as follows.

Definition 3.3.1.2 A host H_i Responsible-For another host H_j, iff the following statements are true:

1. H_i inhabits an inner rim to the rim where H_j inhabits.

2. H_i passes H_j's updates to the next inner rim and provides it with updates that it receives from the next inner rim.

According to this definition, the Responsible-For is one-to-many relationship because it associates multiple hosts that exist in an outer rim with one centre point in the next inner rim.

The rotation of the MHs in both clockwise and anticlockwise directions in the cell rim can be envisioned as a motivation for the revolution of the wheel since the MHs are located here on the most outer rim (i.e., cell rim).

Sectors. Both the zone and cell rims have multiple sectors (i.e., they are divided into multiple parts). Each sector consists of a set of hosts that have same area (either zone or cell) and are connected to same centre point in the next inner rim (i.e., their area's server). For example, the fixed hosts and cell servers that belong to specific zone form a sector on the zone rim and they connect to the server of this zone in the master rim. Accordingly, the sector can be defined formally as follows:

Definition 3.3.1.3 A sector (S) is a subset of replicas in either zone rim or cell rim as follows.

- $S\text{-}Sec = \{FH_1,\ldots, FH_w\}$ U $\{CS_1, \ldots, CS_c\}$ iff :
  (i) Each FH_i and CS_j inhabits the zone rim
  (ii) Each FH_i and CS_j is under responsibility of same secondary master point in the master rim.

or

- $S\text{-}Sec = \{FH_1,\ldots, FH_x\}$ U $\{MH_1, \ldots, MH_y\}$ iff:
  (i) Each FH_i and FH_j is a part of the cell rim
  (ii) Each FH_i and CS_j is under responsibility of same secondary master point in the master rim.

The sector is named using the name of the responsible secondary point in the next inner rim. For example, the sector $S_{Z2}\text{-}Sec= \{CS_{21},\ldots, CS_{2c}\}$ U $\{FH_{21},\ldots, FH_{2v}\}$ represents a part of the zone rim under responsibility of zone server number 2.

Spokes. The hosts in a given rim are linked to their related hosts in another rim or nearby hosts in the same rim through spokes. Two categories of spokes exist in the propagation wheel as follows.

a. **Fixed spokes.** This category links the servers in a given rim with their related servers and fixed hosts in the next outer rim.
b. **Temporary spokes.** They link the cell server with mobile hosts that are currently roaming in its cell (i.e., its sector). Also, this category links two nearby hosts from the same type in same level. For example, it links two nearby cell serves in the same zone or two mobile hosts in the same cell.

Naming schema. The hosts are named using the schema: $Host\text{-}Type_{\text{Zone-No Cell-N0 Host-Serial}}$ (e.g. FH_{212} is the name of the fixed host number 2 in cell number 1, which belongs to zone number 2). MHs are named by considering the zone and cell areas where they have been registered for the first time. The cell servers are named using the following schema $CS_{\text{Zone-No Cell-Serial}}$ (e.g. CS_{41} is the name of the cell server number 1 in zone number 4). The zone servers are identified serially.

Propagation mechanisms. Three basic mechanisms are identified for propagating updates from their sources to a set of other hosts in the propagation wheel as follows.

1. **Outer-to-Inner Propagation.** In this mechanism and as shown in Fig. 4(a), updates flow through the rims in the direction of the wheel centre from their sources in an outer rim into an inner rim until they pour into the master centre point. Each intermediate rim keeps the poured updates for a certain period for the purpose of accumulating them before pouring them into the next inner rim. Accordingly, the steps that are carried out for this type of propagation are as follows:
    - Updates on the hosts (i.e., MHs and FHs) that populate the cell rim flow into their responsible secondary center points (i.e., CSs) in the zone rim.
    - The secondary center points in the zone rim accumulate the poured updates from the cell rim for further processing that implies the ordering of these updates.
    - Processed updates on the CSs of zone rim flow into their responsible secondary center points (i.e., ZSs) that populate the master rim.
    - The secondary center points in the master rim accumulate the poured updates from the zone rim for processing them in a total manner.
    - All accumulated and processed updates on the zone rim flow to the master centre point.

    This type models the propagation of updates from the lowest level in the replication architecture to the highest level. The lowest level represents the cell level, which is modelled by the cell rim, while the highest level represents the master server and it is modelled by the main centre point. Accordingly, this mechanism can be called Bottom-Up propagation (BU).

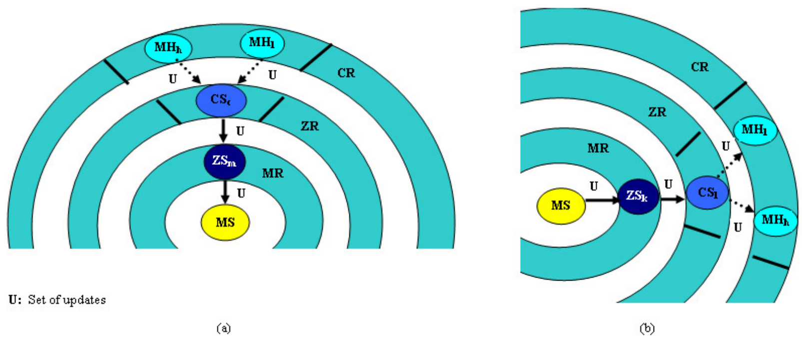


Fig. 4. (a) Outer-to-Inner Propagation (b) Inner-to-Outer Propagation.

2. **Inner-to-Outer Propagation.** In this mechanism, Fig. 4(b), totally ordered updates by the main centre point are pumped from an inner rim into an outer rim in the direction of the most outer rim. Each intermediate rim

contributes the pumping by pushing those updates to reach the most outer rim. Accordingly, the steps that are carried out for this type of propagation are as follows.

- Totally processed updates on the main centre point are pumped into the secondary centre points that populate the master rim.
- Each secondary centre point in the master rim pushes those updates to its underlying secondary centre points that populate the zone rim.
- Each centre point in the zone rim pushes those updates to underlying points that populate the cell rim.

This type models the propagation of updates from the highest level (i.e., master level) in the replication architecture to the lowest level (i.e., cell level). Thus, this mechanism can be called Top-Down propagation (TD).

3. **Inside-Sector propagation.** In this propagation, updates are exchanged inside the rim between two nearby hosts that have same type and sector (i.e., they populate same area). Accordingly, this mechanism is also called P2P propagation. Each peer pumps its received updates (either from other rim or generated on it) into the other peer. The peers form a ring in order to push updates to all peers in the sector. In case of existing of more than one master area, this implies exchanging of updates between the master servers of the wheels that represent these master areas in a peer-to-peer manner. This is because there is no higher level than the master server.

3.3.2. Wheel construction

The propagation wheel is resulted from mapping multiple wheels into one wheel with three rims. These wheels represent the different zone areas and cell areas in the replicated system. This means that the propagation wheel incorporates multiple wheels that are formed by the secondary centre points. Incorporated wheels are called hidden wheels because although they physically exist, their components are incorporated in the three rims of the propagation wheel. Accordingly, the hosts are located on the three rims of the propagation wheel by mapping their locations in their hidden wheels (original areas) into the equivalent rims. The following definition will formally define the hidden wheel.

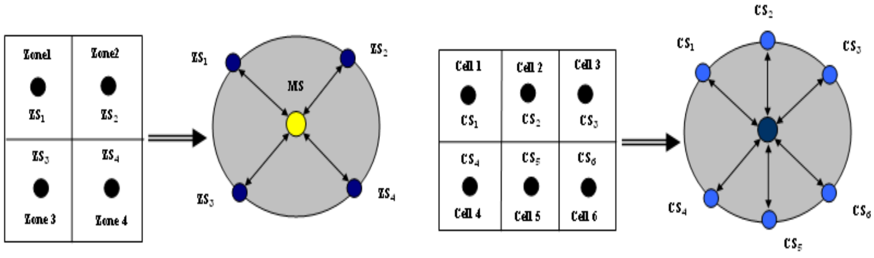
Definition 3.3.2.1. Hidden wheel is a wheel in which following specifications are satisfied:

1. The centre point inhabits either a master or zone rim in the propagation wheel.
2. The rim is incorporated as additional sector in an outer rim in the propagation wheel from that its centre point exists.

Now, the steps of structuring the propagation wheel are as follows:

Step 1. The replicas are placed into wheels (i.e., will be called hidden wheels) according to their cardinal or intermediate geographical directions in their areas or sub areas that are resulted from the replication architecture. The number of directions depends on the locations that the replicated system covers inside the area or sub area. For example, if the master area is divided into four zones, the replicas that represent servers for these zones are located (mapped) into a wheel in four different directions according to their locations in the master area by considering the location of the master server in the centre of the master area. This mapping is depicted in Fig. 5(a) by assuming that the master area is divided into four zones. The resulted wheel from the mapping represents the hidden wheel.

Similarly, when the zone area is divided into multiple cells, the replicas that represent servers for these cells are mapped into a wheel in multiple different directions according to their locations in the zone area by considering the location of the zone server in the centre of the zone area. Figure 5(b) depicts this mapping by assuming that the zone area is divided into 6 cells.



(a) (b)

Fig. 5. (a) Mapping of the Master Area into a Wheel
(b) Mapping of the Zone Area into a Wheel.

Step 2. The area's wheel is mapped into the propagation wheel as a hidden wheel by placing its centre point (area's server) and the points (i.e., underlying servers and fixed hosts) in its rim in specific rims of the propagation wheel according to the type of the hosts and area that is represented by the hidden wheel. The centre point is placed in an inner rim according to the type of the area's server, while the points are placed in the next outer rim. Figure 6 illustrates the mapping of the area wheels that are described in Fig. 5 into the propagation wheel.

Accordingly, a new replica is added to the wheel by placing it according to its type and direction (in case of a server) into the corresponding rim. The most outer rim (i.e., cell rim) has a variable number of replicas, since this number is changed frequently as MHs move from a sector in this rim to another. Replicas can be removed from the wheel as follows:

- If the replica represents either FH or MH, then the removing is straightforward by deleting its information from the Hosts object.
- If the replica represents a server, then each child will be attached to another area. Accordingly, the information of each child replica under it is changed to the new parent.

In Fig. 6, the master wheel is mapped into the propagation wheel as a hidden wheel by placing its centre point as the main centre point and points on its rim (i.e., zone servers) on the master rim in the propagation wheel. The zone wheel is mapped by placing its centre point on the master rim and points on its rim (i.e., cell servers) on the zone rim in the propagation wheel.

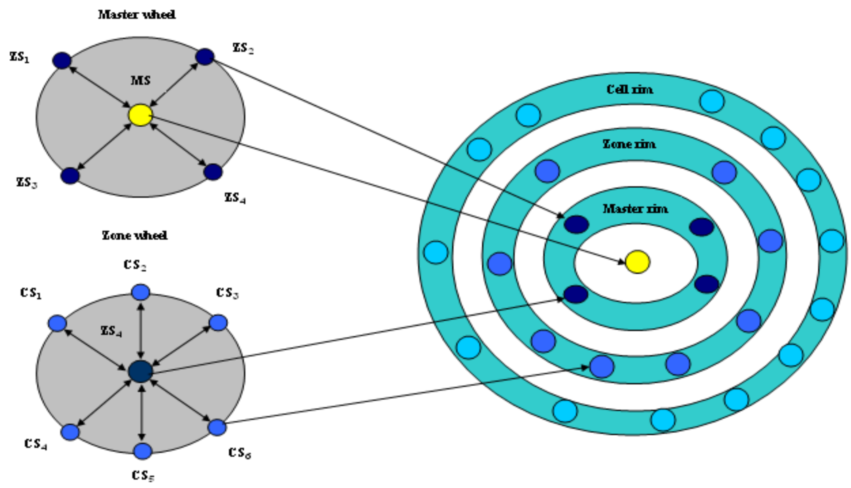


Fig. 6. Mapping of Hidden Wheels in Fig. 5 into the Propagation Wheel.

In case of the replicated system covers only one master area, it scales up by adding new hosts to the different rims and their corresponding spokes. In case of the replicated system covers more than one master area, the scalability is achieved by adding more propagation wheels as the number of master areas. Thus, our propagation wheel extends in a horizontal manner when the replicated system covers new master areas. Also, in the latter case, the former case is applied by considering inside wheel scalability.

3.3.3. Hybrid propagation mechanisms

The following mechanisms act as a hybrid of two or all basic mechanisms for propagating updates from their sources to all hosts:

1. **Bottom-UP_Top-Down Propagation (BT).** It represents a hybrid of both Outer-to-Inner and Inner-to-Outer Propagation mechanisms. In this mechanism, updates are propagated to all hosts by delegating the responsibility of propagation to the main centre point, which represents the server that exists in the highest level (i.e., master server) in the replication architecture. This is because this server has a stable connectivity with the servers that cover all areas in the replicated system (i.e., zone servers). The resolution of updates conflicts through updates ordering process is carried out at the server in the higher level. The steps are as follows:
   - The hosts in the lower levels propagate their updates using BUP to the server in the higher level till they reach the server in the highest level.
   - The collected updates are propagated from the highest level to the lower levels using TDP propagation.
2. **Bottom-UP_P2P_Top-Down Propagation (P2P Concentrate).** It represents a hybrid of the three basic mechanisms for exchanging

updates in the same area (i.e., same cell, same zone, or same master area). In this hybrid, the role of the server of the area where peers inhabit (i.e., the center point in the next inner rim) is eliminated to allow the peers to exchange their updates without needing to send them to the higher level. However, peers need to propagate their updates to this server when these updates should be propagated to the other areas of the replication architecture. The steps are as follows.

- The lower level hosts propagates their updates to the servers in the higher level of their region using BU propagation.
- Each server propagates those updates to its nearby peer until they reach the last peer in the same region (i.e., last peer in the ring) using P2P propagation.
- Each server propagates these updates to the lower level hosts using TD Propagation.

In this technique, the responsibility of the resolution of updates conflicts is delegated to the next nearby peer in the ring.

As an example, in the zone area, this mechanism is applied as follows (Fig. 7).

i. The hosts in each cell propagate their updates to the cell server using BU propagation.
ii. The cell servers exchange those updates using P2P propagation.
iii. Each cell server propagates these updates to its underlying hosts using TD propagation.

However, updates are propagated to the zone server only when they should be propagated to other zone. This case implies exchanging of these updates between the zone servers and their underlying hosts using this mechanism and eliminating the role of the master server.

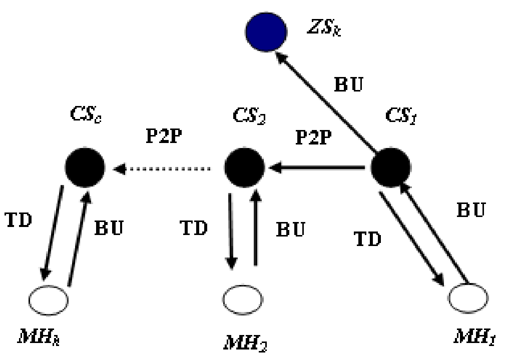


Fig. 7. P2P-Concentrate in the Zone Area.

4. Performance Evaluation

The main objective of the proposed replication strategy is to maintain the consistency through obtaining recent updates. This objective is achieved through the update propagation process. Accordingly, in this section, the two proposed propagation techniques, which are BT and P2P-Concentrate are evaluated and compared with Roam propagation technique with regard to achieving load balance, propagation delay reduction and less communication cost. The required equations that characterize the updates propagation are developed analytically in this section for computing the update propagation delay, communication cost, and average load balance. In the analysis, we start from a consistent state and analyse a single update request. The description of those performance metrics and the required equations for analyzing them as well as the evaluation are as follows.

4.1. Update propagation delay (UPD)

An important requirement in a replicated system with large number of replicas is ensuring fast propagation of updates from their sources to all other replicas. Therefore, reduction of propagation delay is a characteristic of scalable replication strategies.

UDP is measured based on the number of hops that are required for propagating an update from a replica to another replica. This is because measuring the exact time that is consumed in the updates propagation depends on many complicated factors such as connectivity (bandwidth and network delays) and availability of hosts. Moreover, mobile environments suffer from inherited frequent disconnections. Accordingly, we cannot rely on the actual propagation times and delays from a host to another.

Definition 4.1.1 Update propagation delay is the total number of hops from the host that represents the source of update to another host that is either in the same area or in different area. Figure 8 illustrates this definition.

Definition 4.1.2 The hop is a host that participates in propagating updates from its source to the destination.

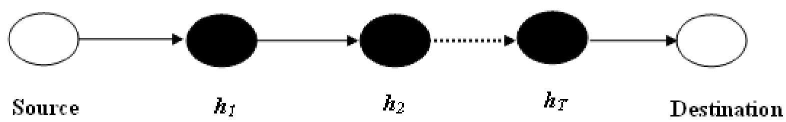


Fig. 8. Hops that are Involved in Propagating an Update from the Source to the Destination.

In the Fig. 8, updates are propagated from the source to the destination through the hops h_1, h_2,..., h_T, where T is the total number of hops. The hops are determined according to the type of propagation technique.

4.1.1. Measuring UPD

To measure the propagation delay, we analytically developed the required equations that are based on the following assumptions:

(i) Two replicas: a replica on MHi, which generates an update that must be propagated to all other hosts. The other replica is MHj, which acts on behalf of all other hosts in that the same results are applied as they have been examined.

(ii) Two cases for the location of the destination, which are as follows.

- Worst case: The purpose of this case is to determine the maximum number of hops that is required to propagate an update to all hosts. Therefore, the location of MHj (i.e., the destination) is assumed in the last cell, which exists in the last zone, or last master area, or it represents the last mobile host in the same cell of the MHi.
- Average case: In this case, UPD is calculated on average in despite of the location of the MHj.

The required equations are developed by considering both worst and average cases for each propagation technique in a separate manner as follows.

(a) **Measuring UPD for BT**

In BT, the following equation is applied for both worst case and average case.

$$\text{UPD} = \begin{cases} 1, & m = 0,\ z = 0,\ c = 1 \\ 3, & m = 0,\ z = 1,\ c \geq 2 \\ 5, & m = 1,\ z \geq 2 \\ m + 4, & m \geq 2 \end{cases} \tag{1}$$

where:
- m is the number of master servers.
- z is the number of zone servers.
- c is the number of cell servers.

Proof. As provided in *Appendix A*.

Same values of UPD are used despite the number of the cell where the update occurs (MHi exists) or the number of the other cell where MHj exists. This means that values do not change for different number of cells in the zone and different number of zones in the master area. This is in contrast with Roam.

(b) **Measuring UDP for P2P-CONCENTRATE**

In this technique, different equations are used for the worst case and the average case as follows.

(i) **Worst case**

$$\text{UPD} = \begin{cases} n - 2, & \text{MHi and MHj in same cell},\ n \geq 2 \\ c, & \text{MHi and MHj in different cells in same zone},\ c \geq 2 \\ z + 2, & \text{MHi and MHj in different zones in same master area},\ z \geq 2 \\ m + 4, & \text{MHi and MHj in different master areas},\ m \geq 2 \end{cases} \tag{2}$$

where:
- n is the number of MHs in the cell.
- c is the number of CSs in the zone.

- z is the number of ZSs in the master area.
- m is the number of MSs.

- **Proof.** As provided in *Appendix A*.

(ii) **Average case**

$$\text{UPD} = \begin{cases} \frac{1}{2}(n-2), \text{MHi and MHj in same cell}, n \geq 2 \\ \frac{c}{2}, \text{MHi and MHj in different cells in same zone}, c \geq 2 \\ \frac{1}{2}(z+2), \text{MHi and MHj in different zones in same master area}, z \geq 2 \\ \frac{1}{2}(m+4), \text{MHi and MHj in different master areas}, m \geq 2 \end{cases} \quad (3)$$

(c) **Measuring UPD for Roam**

In Roam, propagating an update from a replica MH_i to MH_j requires first sending it from MH_i to MH_j's ward master, then sending it from MH_i's ward master to MH_j's ward master, and then finally to MH_j [14]. Accordingly, UPD is calculated as follows:

(i) **Worst case**

$$\text{UPD} = \begin{cases} n-2, \text{MHi and MHj in same ward} \\ w, \text{MHi and MHj in different wards} \end{cases} \quad (4)$$

(ii) **Average case**

$$\text{UPD} = \begin{cases} \frac{1}{2}(n-2), \text{MHi and MHj in same ward} \\ \frac{w}{2}, \text{MHi and MHj in different wards} \end{cases} \quad (5)$$

where:
- n is the number of mobile hosts in the ward
- w is the number of wards

4.1.2. Comparative study using ANOVA and Duncan's Test based on UPD

In this section, a comparative study for the three updates propagation techniques (i.e., BT, P2P-CONCENTRATE, and Roam) is performed based on UPD as a performance metric. The purpose of the study is to answer the following questions:

1. What effects do number of cells and propagation techniques have on the UPD?
2. Which the best technique among the three that can be used to propagate updates in large scale mobile distributed database system?

The techniques are compared by varying the number of cells (i.e., equivalent to wards in Roam) and computing UPD based on the developed equations.

We assume that the number of cells in each zone is 5 and similarly, the number of zones in each master area is 5 (same conclusions are drawn when the number of cells or zones is greater than or equal 5 as already examined for different values). This means that in this comparison, varying the number of cells leads to the variation of the number of both zones and master servers in our strategy.

In this comparison, if MHi and MHj in the same cell, we consider there are no any MHs between them. This because the number of MHs that act as hops between them cannot be estimated, since this depends on the number of MHs roaming at that cell on that time instant. Therefore, we consider UPD = 0 in this case as the best case for Roam.

Two replications for each cell number are taken into consideration for the calculation of UPD using the different techniques. Accordingly, for this comparison, the following factors are considered:

1. Different techniques for updates propagation (Factor A)
2. Number of cells (Factor B)

A summary of the factors and their levels in the experimentation is presented in Table 2.

Table 2. Levels of two Factors A and B.

Serial No.	Factors	Values	Number of Levels
1	Propagation techniques	---	3
2	Number of cells	1,2,3,…,100	100

Based on these factors, the experimental combination contains the number of the cell and the corresponding UPD values for the three techniques. Since two replications for each cell number (factor B) are taken into consideration for the calculation of UPD using the different levels of techniques (factor A), this means that the total number of experimental combinations is equal to 200.

The UPD values are analyzed in two stages using ANOVA and Duncan's multiple range tests. The summary of these analyses is as follows:

Stage 1. ANOVA

The problem (i.e., comparing three techniques) is treated as two ANOVA.

- **Factor A:** Techniques

  Levels: 3
- **Factor B:** Number of cells

  Levels: 100
- **Response Variable:** Performance metric (measure or value) namely UPD.
- **Number of observations (n):** 600 (3*100*2)
- **Model:**

  The model of 2-factor experiment is as follows:

$$Y_{ijk} = \mu + \alpha_i + \beta j + \alpha\beta_{ij} + \varepsilon_{ijk} \quad (i=1, 2, 3; j=1, 2, 3,\dots, 100; k=1,2) \qquad (6)$$

where:

- Y_{ijk} is the performance measure namely UPD of the k^{th} replicate under the i^{th} and j^{th} treatments of the factors A and B respectively
- μ is the overall mean effect.
- α_i is the effect of the performance measure namely UPD due to the i^{th} treatment of Factor A.
- βj is the effect of the performance measure namely UPD due to the j^{th} treatment of Factor B.
- $\alpha\beta ij$ is the effect of the performance measure namely UPD due to the i^{th} treatment of Factor A and j^{th} treatment of Factor B.
- ε_{ijk} is the random error (the effect of random experimental error)

- **Null hypotheses:**

  H_0^1 **:** $\alpha_1 = \alpha_2 = \alpha_3 = 0$

  Three techniques (Factor A) do not have significant effect on UPD.

  H_0^2**:** $\beta_1 = \beta_2 = \ldots = \beta_{100} = 0$

  Number of cells (Factor B) does not have significant effect on UPD.

  H_0^3 **:** $(\alpha\,\beta)_{ij} = 0$ for all i, j

  Interaction between techniques (Factor A) and number of cells (Factor B) does not have significant effect on UPD.

- **Alternative hypotheses:**

  H_1^1 **:** at least one $\alpha_i \neq 0$

  H_1^2 **:** at least one $\beta_i \neq 0$

  H_1^3 **:** at least one $(\alpha\,\beta)_{ij} \neq 0$

- **Level of Significance:** It is assumed as 0.05.
- **ANOVA Table:** It is as shown in Table 3. From this table, if the calculated value of F of a particular source of variation is greater than the corresponding tabulated value of F (F_T), then it can be concluded that the above source of variation is having significant effect on the performance measure namely UPD (i.e., the null hypothesis corresponding to the source of variation is rejected). Otherwise, it can be concluded that the source of variation is not having any significant effect on the performance measure namely UPD (i.e., the null hypothesis corresponding to the source of variation is accepted).

Table 3. Two Way ANOVA Table.

Source of Variation	Sum of Squares (SS)	Degrees Of Freedom (v)	Mean Sum of Squares (MS)	F (Calculated)	F_T	$F > F_T$ Yes or No
A	139479.29	2	69739.65	496.1322	3.025846	YES
B	37930.74	99	383.14	2.725673	1.296908	YES
AB	56554.04	198	285.63	2.031964	1.234578	YES
E	42170	300	140.57			
Total	276134.1	599				

- **Results:** From the ANOVA statistics shown in Table 3, the following conclusions can be arrived at:

a. Techniques (Factor A) are having significant effect on the performance measure namely UPD (i.e., H_0^1 is rejected).

b. Number of cells (Factor B) is having significant effect on the performance measure namely UPD (i.e., H_0^2 is rejected).

c. Interaction between Factor A and Factor B is having significant effect on the performance measure namely UPD (i.e., H_0^3 is rejected).

In accordance with ANOVA results, the model components A, B, and AB are statistically significant.

Stage 2. Test of means using Duncan's multiple range test

The first stage of the analysis concludes that the factor "Techniques" (Factor A) is having significant effect on UPD resulted into rejecting the null hypothesis. Accordingly, the next stage is the test of means to check whether the difference between any pair of treatment means is significant at a given confidence level. This stage is performed using Duncan's multiple range test [21].

Now, the steps that are carried out for this test are as follows.

1. Arranging the means in the ascending order of their respective values as shown in Table 4.

Table 4. Ordering of Mean Values.

Technique	Mean Symbol	Mean Value	Order
BT	A1	6.38	2
P2P-CONCENTRATE	A2	4.96	1
Roam	A3	37.99	3

2. Calculation of the standard error of each average:

$$S = \sqrt{MSE \ / \ n} \tag{7}$$

where:
- MSE is mean sum of square error from ANOVA (i.e., MSE= 210.85)
- n is the sample size (i.e., n= 200)

Thus $S = \sqrt{210.85 / 200} = 1.027$

3. Finding the critical value $q_\alpha(k, v)$ from the table of significant ranges [21] where:
- α is the significance level
- k the number of means being compared, and all means in-between (k=2,3)
- v is the degrees of freedom for error from the ANOVA table.

Accordingly, the critical values are: $q_{0.05}(2,200)$ =3.687 and $q_{0.05}(3,200)$ =3.843

4. Calculating the value of the least significant range (R_k):

$$R_k = q_\alpha(k, v)S \tag{8}$$

Accordingly, the least significant ranges are:
$R_2 = q_{0.05}(2,200)\times S = 3.787$
$R_3 = q_{0.05}(3,200)\times S = 3.947$

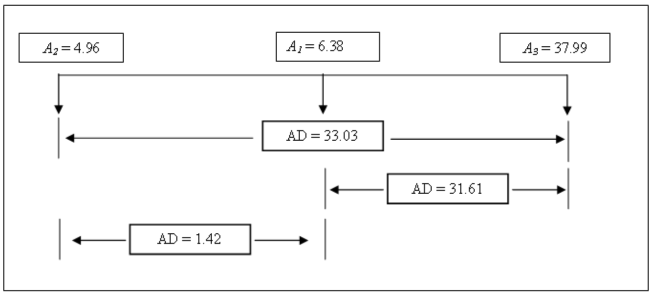


Fig. 9. Actual Difference between the Different Pairs of Means.

5. Calculating the actual differences between the different pairs of means (Fig. 9) and comparing them with the corresponding least significant ranges.

According to Fig. 9, the actual differences between the different pairs of means are:

$A_3 - A_2 = 33.03 \quad > R_3$
$A_3 - A_1 = 31.61 \quad > R_2$
$A_1 - A_2 = 1.42 \quad < R_2$

Duncan's test results. According to the previous analysis it can be concluded that there are significant differences between two pairs of means. The remaining pair is not significantly different. Accordingly, one can come to the following conclusions:

- The Roam propagation technique (corresponding to the mean value: A_3 = 37.99) performs most badly than the other two techniques, which are BT (corresponding to the mean value: A_1 = 6.38) and P2P-CONCENTRATE (corresponding to the mean value: A_2 = 4.96).
- There is no significant difference between BT and P2P-CONCENTRATE techniques.
- P2P-CONCENTRATE techniques (corresponding to the mean value: A_2 = 4.96) performs better than BT (corresponding to the mean value: A_1 = 6.38), but this in case that the ordering process is not important or can be delegated from a peer to another peer which leads to heavy work load on the last peer for the ordering process.

Since there is no significant difference between BT and P2P-CONCENTRATE, we conclude that the BT technique can be used mainly for propagating updates within the same master area in order to perform the ordering process in a hierarchical manner. This achieves fair conflict resolution for all updates that are generated on the lower levels by delegating the responsibility of resolution to the server in the higher level, while we use the P2P-CONCENTRATE technique for propagating updates between the master areas, since there is no higher level than the master area. In this case, update conflicts resolution is performed in a peer-to-peer manner by delegating the responsibility of ordering to the next nearby peer in the ring.

4.2. Communication cost

In this section, the comparison is performed based on the communication cost that is incurred by propagating updates between the different hosts. In the three techniques, update information is propagated in a form of a message from a host to another until reaching the destination. Therefore, the communication cost that is incurred by propagating an update from the source to the destination is directly proportional to the total number of messages (T) that are involved in this propagation. Accordingly, the total number of messages depends on the number of hops between the two hosts. Thus far, there is a relation between UPD and T as follows.

Assertion. The relation between UPD and T can be defined using the following equation.

$$T = UPD + 1 \quad (9)$$

Proof. It is straightforward and in same manner as for computing UPD (*Appendix A*) by considering a message flows from the source to the first hop and messages that are exchanged between the hops till reaching the destination.

Based on the results that are obtained by considering UPD as the performance metric and the relation between T and UPD, the following conclusions, which are shown in the Figs. 10 and 11 can be reached for both worst and average cases

- The Roam propagation technique has the highest cost for propagating updates.
- There is no significant difference between BT and P2P-CONCENTRATE techniques.
- P2P-CONCENTRATE technique has the lowest cost but it can not be performed between hosts that exist in different areas.

In Fig. 10, we observe that the total number of messages of Roam and P2P-CONCENTRATE is same for small number of cells (around 1-5 cells) because updates are propagated between two hosts that either in same cell or same zone, but for Roam it is linearly gets higher. The total number of messages of BT and P2P-CONCENTRATE is same for large number of cells that exist on more than one master area. This is because when the number of master servers exceeds one, updates are propagated between these servers using P2P-CONCENTRATE, since there is no higher level to perform BT. Thus, in this case as we mentioned P2P-CONCENTRATE is equivalent to BT.

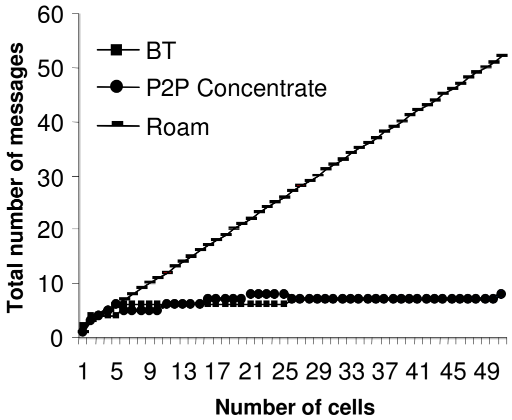


Fig. 10. Comparison of the Three Techniques Based on the Total Number of Messages for the Worst Case.

The value of this metric is slightly lower in P2P-CONCENTRATE than BT for small number of cells (around 1 message lower for around 1-10 cells) due to small number of peers. And it is slightly lower in BT than P2P-CONCENTRATE for a number of cells that ranges from16 to 25 (and it is around 1-2 messages lower) because the number of peers increases in this range and the number of the master server is 1.

According, we can conclude that both BT and P2P-CONCENTRATE are more scalable than the Roam with regard to the communication cost.

As shown in Fig. 11, the total number of messages of BT and P2P-CONCENTRATE is far better than roam because it increases in Roam as the number of cells increases. P2P-CONCENTRATE has the lower values than BT because in the latter same values are hold for both worst and average cases (i.e., it does not differentiate between the worst and average cases).

To enhance the appearance of the details included in the Figs. 10 and 11, we limit the number of cells to 50.

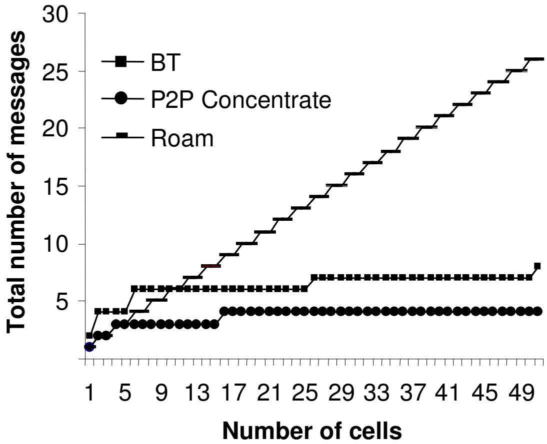


Fig. 11. Comparison of the Three Techniques Based on the Total Number of Messages for the Average Case.

4.3. Average Load balance (ALB)

An important requirement for improving the performance of large scale database systems with large number of updates is the distributing of the overhead of the updates propagation over many hosts. This overhead is measured by average load balance, which is defined as the average number of hosts to which each host propagates the update information.

To evaluate ALB of the three techniques, we consider the parameters that are listed in Table 5.

Table 5. Parameters for Performance Evaluation Based on ALB.

Parameter	Description	Remarks
N	Total number of replicas	$N = 100\times n$ (n=1,2,…,15)
Z	Number of zone servers	The master area is divided into a different number of zones in each trial.
C	Number of cell servers in each zone	The zone area is divided into a different number of cells in each trial.
C^{-}	Total number of cells in the replicated system	
S	Total number of replicas in the different servers (i.e., CSs, ZSs, and MS)	$S = 1 + Z + C^{-}$
N^{-}	The average total number of mobile hosts	Estimated based on: $N^{-} = N - S$
H	Average number of mobile hosts in each cell	Estimated based on: $H = N^{-} / C^{-}$
U	Number of updates	1

The following assumptions are considered for the simplification of the analysis based on ALB:

1. The replicated system covers one master area. This is because we interest in the load of the master server rather than the propagation to other areas. Moreover, same results are applied in case of existing of more than one master area.
2. Symmetric distribution of cells in the different zones as follows.
   (i) The zones have same number of cells.
   (ii) The number of cells in each zone is equal to the number of zones in the master area. That is Z=C=D, where D is the number of directions in the updates propagation wheels.
3. Each cell contains the same number of mobile hosts.

Accordingly, in this comparison, we vary the number of directions (D), which leads to the variation of both the number of zone servers and the number of cell servers in each zone.

Based on the above parameters and assumptions, the ALB can be computed for each server and mobile host using the following equations.

Assertion. ALB for different types of hosts and for both BT and P2P-CONCENTERATE techniques is computed as follows.

a. ALB for the master server (ALB-MS):

$$\text{ALB-MS} = Z + 1 \quad (10)$$

b. ALB for the zone server (ALB-ZS):

$$\text{ALB-ZS} = C + 1 \quad (11)$$

c. ALB for the cell server (ALB-CS):

$$\text{ALB-CS} = H + 1 \quad (12)$$

d. ALB for the mobile host (ALB-MH):

$$\text{ALB-MH} = 1 \quad (13)$$

Proof. It is straightforward for both BT and P2P-CONCENTERATE as follows:

- In equation (10), the master server propagates updates to underlying zone servers in addition to the nearby peer in case of existing more than one master area (i.e., the case of BT).
- In equation (11), the zone server propagates updates to its underlying cell servers in addition to either the master server in case of BT or the nearby peer in case of P2P- CONCENTERATE.
- In equation (12), the cell server propagates updates to mobile hosts that are located in its cell in addition to either the zone server in case of BT or the nearby peer in case of P2P-CONCENTERATE.
- In equation (13), the mobile host propagates updates to either the cell server in case of BT or the nearby peer in case of P2P-CONCENTERATE.

On the other hand, the ALB for Roam propagation technique is 1 for the mobile host (ALB-MH-Roam) and 2 for the ward master (ALB-WM-Roam) because it propagates to the nearby peer and to a mobile host in its ward.

The values of ALB are generated based on these equations by a comparative study is performed by considering different values for both the number of replicas and D as follows.

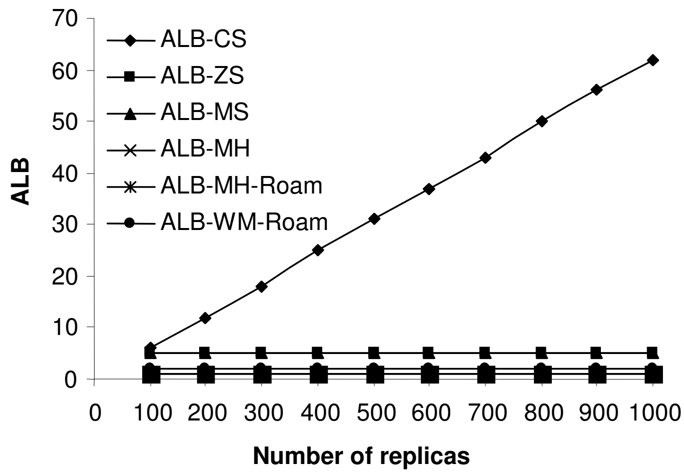


Fig. 12. Average Load Balance when *D*=4.

1. Comparing ALB values when D=4

   The three techniques are compared based on ALB values by varying the number of replicas (N), where N= 100×n (n=1, 2…10) and considering D=4. The impact of this variation on ALB is as shown in Fig. 12. The load of CS gets higher as the number of replicas increases. ALB values for both ZS and MS are not affected by the changing of the number of replicas for same value of D. ALB values for MH, MH in Roam, and ward master are not affected at all for different values of the number of replicas.

2. The effect of *D* on ALB where *D* = 2, 3, 4,…,8

The impact of *D* on ALB for different types of hosts is studied in order to characterize the optimal value of *D*. For this purpose, we varied the number of replicas to be 100**n* (*n*= 1, 2, …, 8) and value of *D*. This variation has a greater impact on ALB-CS as shown in Fig. 13. When the value of *D* gets higher, ALB-CS decreases as the highest value is when *D* = 2 and the Lowest value is when *D* = 8. Accordingly, increasing the value of D will result in a decreased average load balance for each CS. As the number of replicas gets higher by 100, this leads to:

(i) Increasing the load of the CS according to a value ≤ 4 for *D* ≥ 5 and not more than 25 for *D* = 2, 11 for *D* = 3, 6 for *D* = 4.

(ii) Increasing the value of ALB-CS by small amount than the previous value of *N* (e.g. in case of *D* =5, 6, 7, and 8 the increasing amount is ≤ 4).

The actual load for CS is less than the calculated value according to the fact that the MHs in a given cell do not stay connected to the CS at all times. Accordingly, the higher values for ALB-CS are justified by that fact.

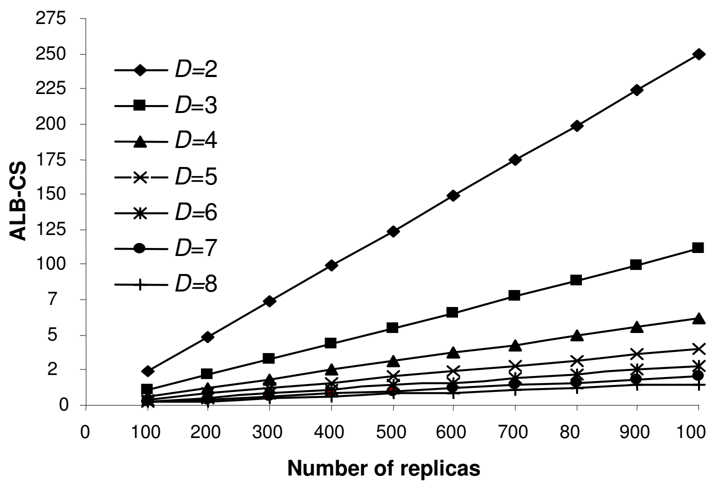


Fig. 13. ALB for CS where *D* = 2, 3, 4,…, 8.

Figure 14 shows the impact of the variation of *D* on ALB values of other hosts than CS. This variation leads to increasing the load of ZS and MS by only 1, but it does not affect the values of MH load and the load in roam for MHs and ward masters. Thus, MHs are having same load in our strategy and roam.

Accordingly, this section can be concluded as that the proposed strategy places the overhead (much of the load involved in the updates propagation) of updates propagation to be performed by the servers that exist in the fixed network, since they have more storage and processing capabilities than mobile hosts.

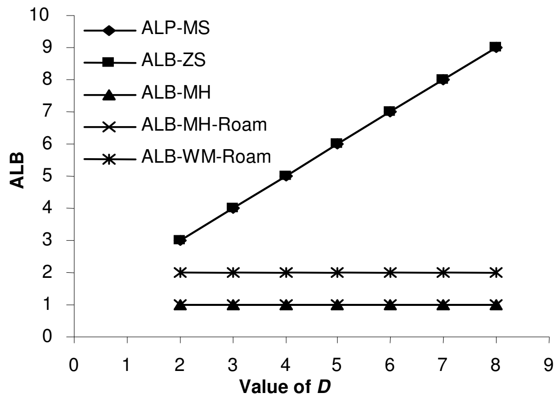


Fig. 14. ALB for hosts other than CS for different values of *D* (*D*=2, 3, …, 8).

5. Conclusions

In this paper, our research has focused on proposing a new replication strategy to maintain the consistency of replicated data in large scale mobile environments. The replication strategy encompassed three-level replication architecture and wheel-based updates propagation protocol as a binary combination that is needed to achieve such a goal. The strategy supports frequent disconnections and mobility of hosts by enabling the users to perform their updates in a disconnected mode and then synchronizing their updates with the higher levels.

To exploit the features of both optimistic and pessimistic replication, the new strategy is based on a hybrid approach that divides data into frequently changed data and infrequently changed data, and then updates are restricted or allowed according to these types.

The effectiveness of the proposed strategy is verified through the comparative study with Roam. The results show that our strategy achieves better propagation delay and lesser total number of messages than Roam replication system. Moreover, the proposed propagation protocol achieves load balance in both propagation and ordering processes because these processes are shared by multiple hosts.

As part of our future research, a plan will be provided to develop the required tools and interfaces to implement the proposed strategy in mobile healthcare environments to provide healthcare practitioners with an efficient access to healthcare data.

References

1. Barbara, D. (1999). Mobile computing and databases – a Survey. *IEEE Transactions on Knowledge and Data Engineering*, 11(1), 108-117.
2. Budiarto; S.N., S.; and Tsukamoto, M. (2002). Data management issues in mobile and peer-to-peer environments. *Data & Knowledge Engineering*, 41(2-3), 183-204.
3. Phatak, S.H.; and Nath, B. (2004). Transaction-centric reconciliation in disconnected client-server databases. *Journal of Mobile Networks and Applications*, 9(5), 459-471.

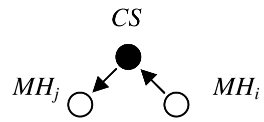
4. Connolly, T.M.; and Begg, C.E. (2004). *Database systems: a practical approach to design, implementation and management.* (4th Ed.). Addison-Wesley.
5. Holliday, J.; Agrawal, D.; and Abbadi, A.E. (2002). Disconnection modes for mobile databases. *Wireless Networks*, 8(4), 391-402.
6. Madria, S.; and Bhowdrick, S. (2001). Mobile data management. Potentials, *IEEE*, 20(4), 11-15.
7. Imielinksi, T.; and Badrinath, B. (1994). Wireless mobile computing: challenges in data management. *Communications of ACM*, 37(10), 18-28.
8. Saito, Y.; and Shapiro, M. (2005). Optimistic replication. *ACM Computing Surveys (CSUR)*, 37(1), 42-81.
9. Gray, J.; Helland, P.; and O'Neil, P. (1996). The dangers of replication and a solution. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 25(2), 173-182.
10. Wiesmann, M.; Pedone, A.; Schiper, B.; Kemme, G.; and Alonso, G. (2000). Understanding replication in databases and distributed systems. *Proceedings of the 20th International Conference on Distributed Computing Systems (ICDCS 2000),* 464.
11. Bernstein, P. (1997). *Principles of transaction processing*. Morgan Kaufmann Publishers Inc.
12. Helal, A.; Heddaya, A.; and Bhargava, B. (1996). *Replication Techniques in Distributed Systems*. Kluwer Academic Publishers.
13. Ratner, D.; Reiher, P.; and Popek, G.J. (2004). Roam: a scalable replication system for mobility. *Mobile Network and Applications*, 9(5), 537-544.
14. Ratner, D. (1998). *Roam: a scalable replication system for mobile and distributed computing*. PhD Dissertation, University of California, Los Anglos.
15. Ratner, D.; Reiher, P.; Popek, G.J.; and Kuenning, G.H. (2001). Replication requirements in mobile environments. *Mobile Networks and Applications*, 6(6), 525-533.
16. Monteiro, J.; Brayner, A.; and Lifschitz, S. (2007). A mechanism for replicated data consistency in mobile computing environments. *Proceedings of the ACM symposium on Applied computing, Seoul, Korea*, 914-919.
17. Abawajy, J.; Deris, M.; and Omer, M. (2006). A novel data replication and management protocol for mobile computing systems. *Mobile Information Systems*, 2(1), 3-19.
18. Tolia, N.; Satyanarayanan, M.; and Wolbach, A. (2007). Improving mobile database access over wide-area networks without degrading consistency. *Proceedings of the 5th international conference on Mobile systems,* applications and services, San Juan, Puerto Rico, 71-84.
19. Zhiming, D.; Xiaofeng, M.; and Shan, W. (2002). A transactional asynchronous replication scheme for mobile database systems. *Journal of Computer Science and Technology*, 17(4), 389-396.
20. Terry, S. (1983). *Stronger than a hundred men: A History of the vertical water wheel*. Johns Hopkins University Press, 9-46.
21. Montgomery, D.C. (1991). *Design and analysis of experiments*. (3rd Ed.). New York: John Wiley and Sons Inc.

Appendix A

1. Proof for equation of Measuring UPD for BT

Assume that h is the number of hops (they are represented with the black circles). The following cases are considered:

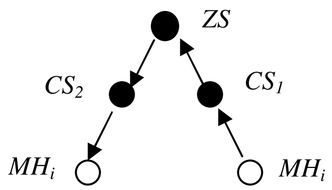
a. If $c = 1 \rightarrow h = 1$



b. If $c = 2 \rightarrow h = 3$

This is because if c = 2, this implies existing of one zone server according to our assumption that two or more cell servers need a zone server for resolving their conflicts.

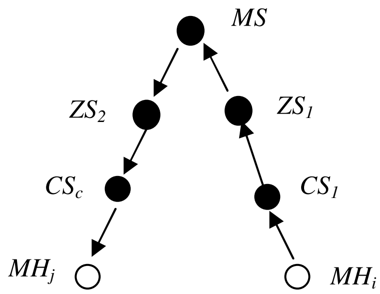
Thus, $c = 2 \leftrightarrow z = 1$



c. If $z = 2 \rightarrow h = 5$

Also, if z = 2, this implies existing of one master server for resolving their conflicts.

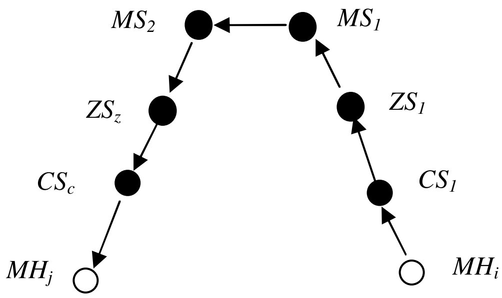
Thus, $z = 2 \leftrightarrow m = 1$



d. For $m \geq 2$, we use mathematical induction as follows:

If $m = 2 \rightarrow h = 6$

This is because updates should be propagated in P2P manner in case of existing more than one master server since there is no higher level than the master level in our strategy. Update conflicts are resolved by delegating the responsibility of resolving to the next peer.



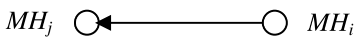
Accordingly, the If $m = k \rightarrow h=k+4$
Thus, if $m = k + 1 \rightarrow h=(k+1)+4$

2. Proof for equation of measuring UPD for P2P-CONCENTRATE

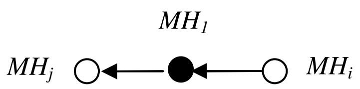
By using mathematical induction and assuming h is the number of hops, we consider the following cases:

a. MH_i and MH_j in the same cell

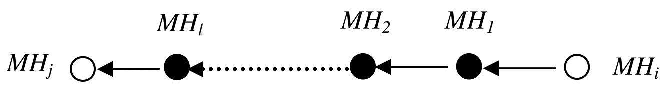
If $n = 2 \rightarrow h = 0$ (There are no hops between MH_i and MH_j)



If $n = 3 \rightarrow h = 1$



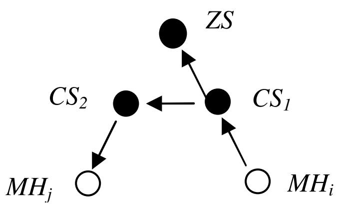
Accordingly, If $n = k \rightarrow h = k\text{-}2$



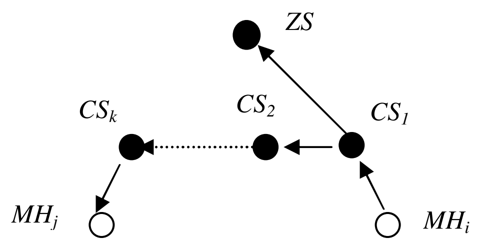
Since the equation holds for n=k, this implies that:
If $n = k+1 \rightarrow h = (k\text{-} 2) +1= (k+1)\text{-}2$

b. MH_i and MH_j in different cells in the same zone

If $c = 2 \rightarrow h = 2$



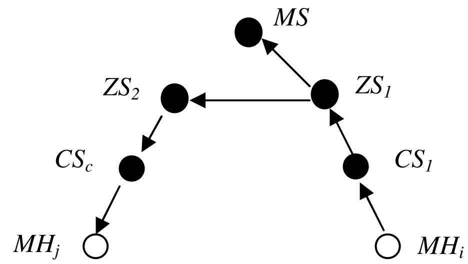
If $c = k \rightarrow h = k$



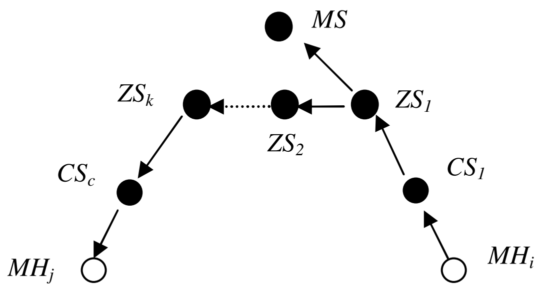
Thus, if $c = k+1 \rightarrow h = k+1$

c. MH_i and MH_j in different zones in the same master area

If $z = 2 \rightarrow h = 4$



If $z = k \rightarrow h = k+2$



Thus, if $z = k+1 \rightarrow h = (k+1) +2$.

d. MH_i and MH_j in different master areas.
The proof is performed in same manner as above.