CHAPTER FOUR

WHEEL-BASED UPDATES PROPAGATION PROTOCOL

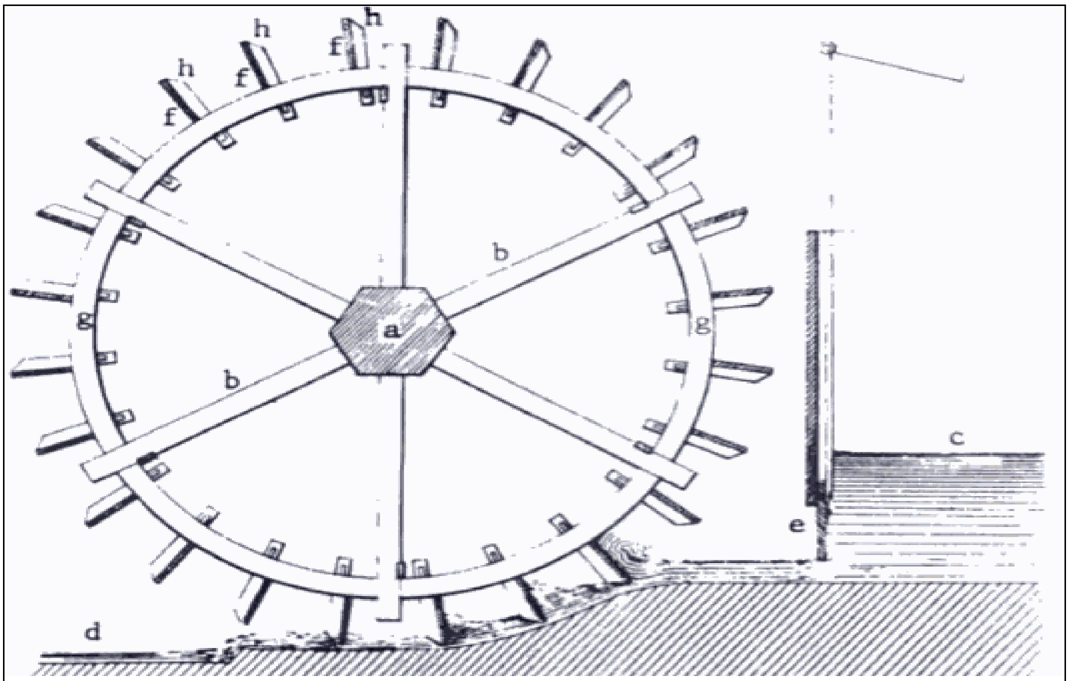
4.0 Overview

This chapter provides the details of the proposed protocol for updates propagation through the components of replication architecture. The protocol consists of a 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 the replicas for propagating recent updates from their sources to other replicas that are distributed over the wheel. Accordingly, the resulting protocol is called Wheel-Based updates propagation protocol.

4.1 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 application of the water wheel structure here is arising from its general design (see Figure 4.1.1) and functionality. The water wheel structure (Terry, 1983) 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 center 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 4.1.1 depicts water wheel features that applied and mapped to the proposed architecture.



(*a*) axle (*b*) spokes or arms; (*c*) head race; (*d*) tail race; (*e*) sluice gate or chute, the device which regulates the admission of water onto the water wheel; (*f*) floats, floatboards, blades, or paddles; (*g*) rim (the circular built-up felloes to which the arms are mortised and the floatboards attached); (*h*) starts or supports, pieces of wood or metal projecting from the rims to which the floatboards or blades are secured.

Figure 4.1.1 Water Wheel Structure (adapted from Terry, 1983)

Table 4.1.1 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 Center	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 Figure 4.1.2. The following definition will formally define the propagation wheel (PW).

Definition 4.1.1 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 center points where a sub set of *F*, *H*, or *S* can be connected to each center 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 that share same data items.

4.1.1 Center Points

As depicted in Figure 4.1.2, the different types of hosts are represented by circles in the propagation wheel. Some hosts act as center 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 center points or ordinary points (i.e. points act as either fixed or mobile hosts), which are located on virtual circular rims as follows:

1. Master server: It acts as the main center point that is linked with secondary center 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 center 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 center and ordinary points represent the different types of hosts of the replication system, while the spokes between them represent the network connections (channels).

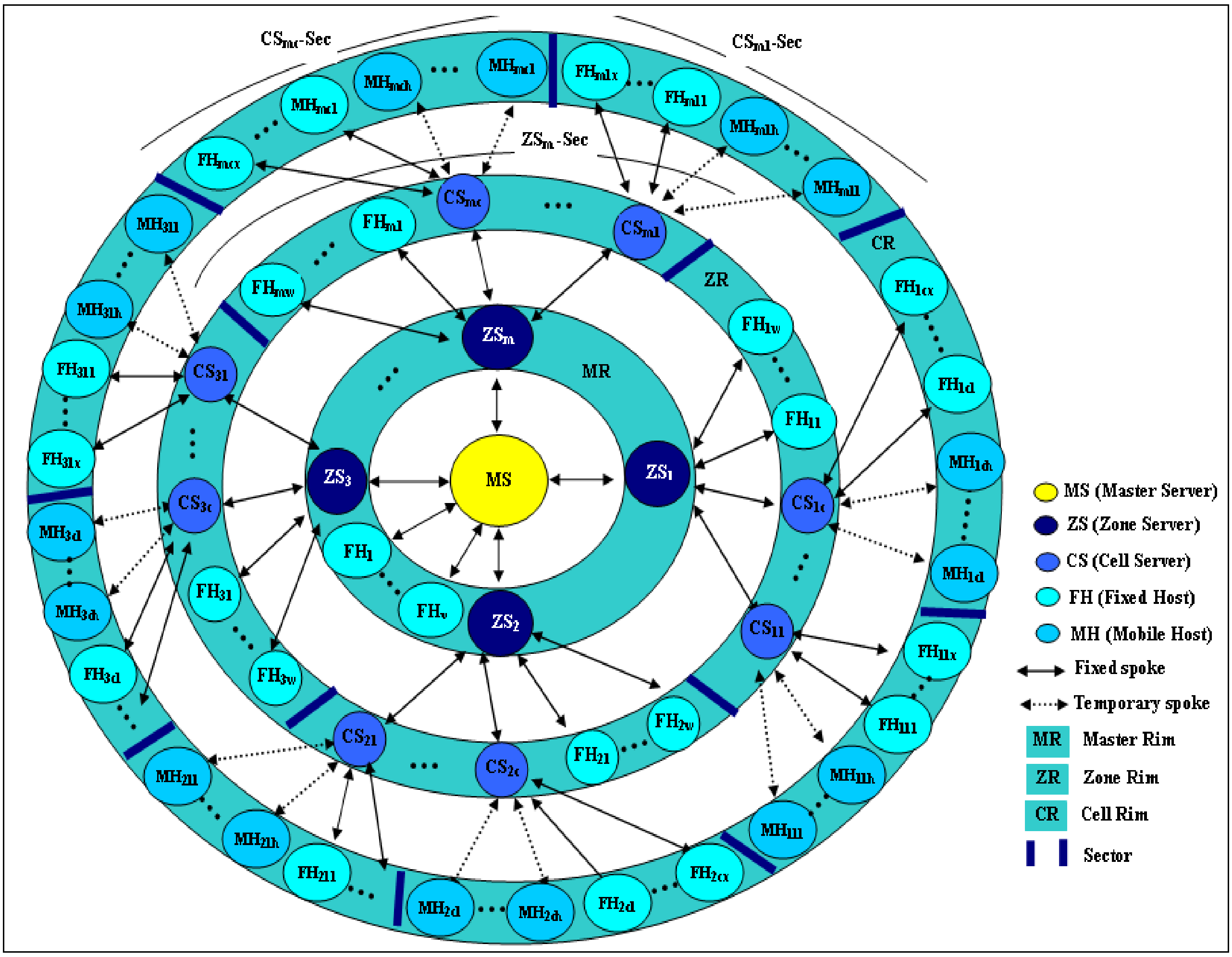


Figure 4.1.2 Updates Propagation Wheel

4.1.2 Rims

Rims 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 4.1.2.1 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 center 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).

4.1.3 Sectors

Both 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 center 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 4.1.3.1 A sector (*S*) is a subset of replicas in either zone rim or cell rim as follows.

- $S\text{-}Sec = \{FH_1,\ldots, FH_w\} \cup \{CS_1, \ldots, CS_c\}$ 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

- $S\text{-}Sec = \{FH_1,\ldots, FH_x\} \cup \{MH_1, \ldots, MH_y\}$ 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 $S_{Z2}\text{-}Sec = \{CS_{21},\ldots, CS_{2c}\} \cup \{FH_{21},\ldots, FH_{2v}\}$ represents a part of the zone rim under responsibility of zone server number 2.

4.1.4 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.

4.1.5 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.

4.1.6 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 (see Figure 4.1.6.1 (a)), updates flow through the rims in the direction of the wheel center from their sources in an outer rim into an inner rim until they pour into the master center 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 center 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 modeled by the cell rim, while the highest level represents the master server and it is modeled by the main center point. Accordingly, this mechanism can be called Bottom-Up propagation (BU).

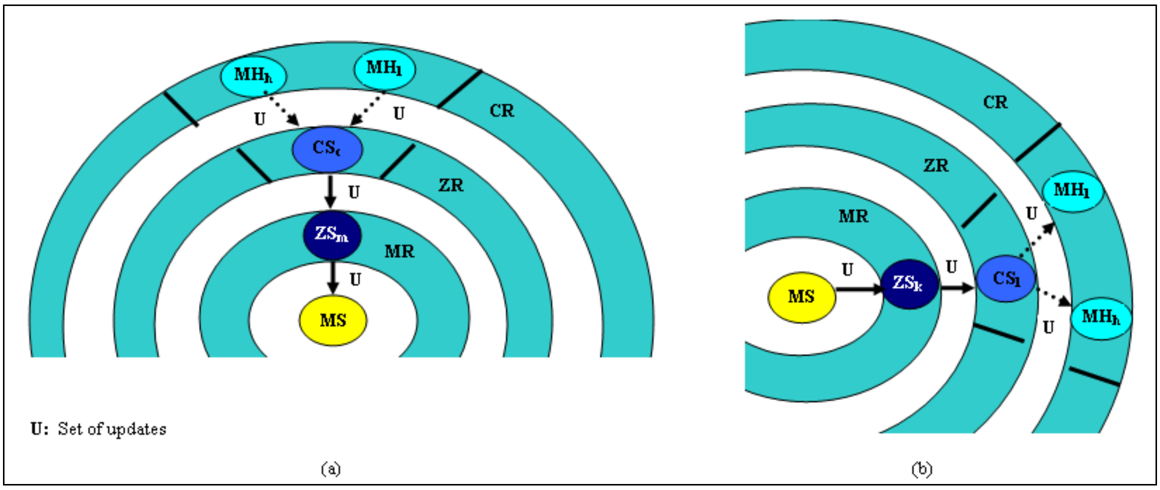


Figure 4.1.6.1 (a) Outer-to-Inner Propagation (b) Inner-to-Outer Propagation

2. **Inner-to-Outer Propagation.** In this mechanism (see Figure 4.1.6.1 (b)), totally ordered updates by the main center 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 center point are pumped into the secondary center points that populate the master rim.
    - Each secondary center point in the master rim pushes those updates to its underlying secondary center points that populate the zone rim.
    - Each center 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 the 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.

The three basic propagation mechanisms represent the possible alternatives for exchanging recent updates between the different types of hosts. Bottom-Up propagation mechanism is necessary to ensure that all recent updates, which occurred in the lower level hosts reach the servers that are located in the higher levels. This is because these servers cover different areas, and are required to provide their underlying hosts with the recent updates that occurred in the other areas. Top-Down propagation mechanism is required for each server to disseminate the received updates from other areas to the hosts that are located in its area. P2P Propagation mechanism is provided to enable the hosts, which exist in the same area to exchange their updates directly without needing to send these updates to the server of area where they are located. For example, some hosts may be involved in performing duties that are associated with their area. Accordingly, they can share their updates using this mechanism without needing to send these updates to the other areas.

4.2 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 replication system. This means that the propagation wheel incorporates multiple wheels that are formed by the secondary center 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 4.2.1 Hidden wheel is a wheel in which following specifications are satisfied:

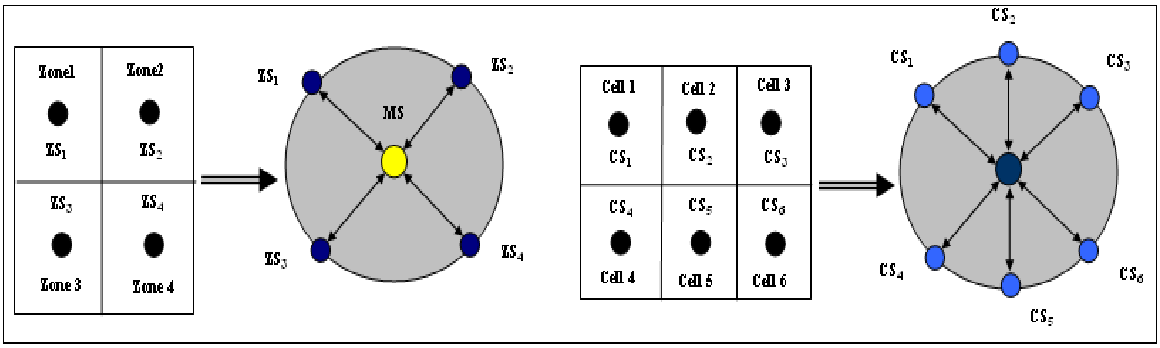
1. The center point inhabits either a master or zone rim in the propagation wheel.
2. The rim is incorporated as additional sector in an outer rim in the propagation wheel from that its center point exists.

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 replication 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 center of the master area. This mapping is depicted in Figure 4.2.1 (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 center of the zone area.

Figure 4.2.1 (b) depicts this mapping by assuming that the zone area is divided into 6 cells.



(a) (b)

Figure 4.2.1 (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 center 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 center 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 4.2.2 illustrates the mapping of the area wheels that are described in Figure 4.2.1 into the propagation wheel.

Accordingly, a new replica is added to the wheel by placing it according to its type and direction (in the 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.

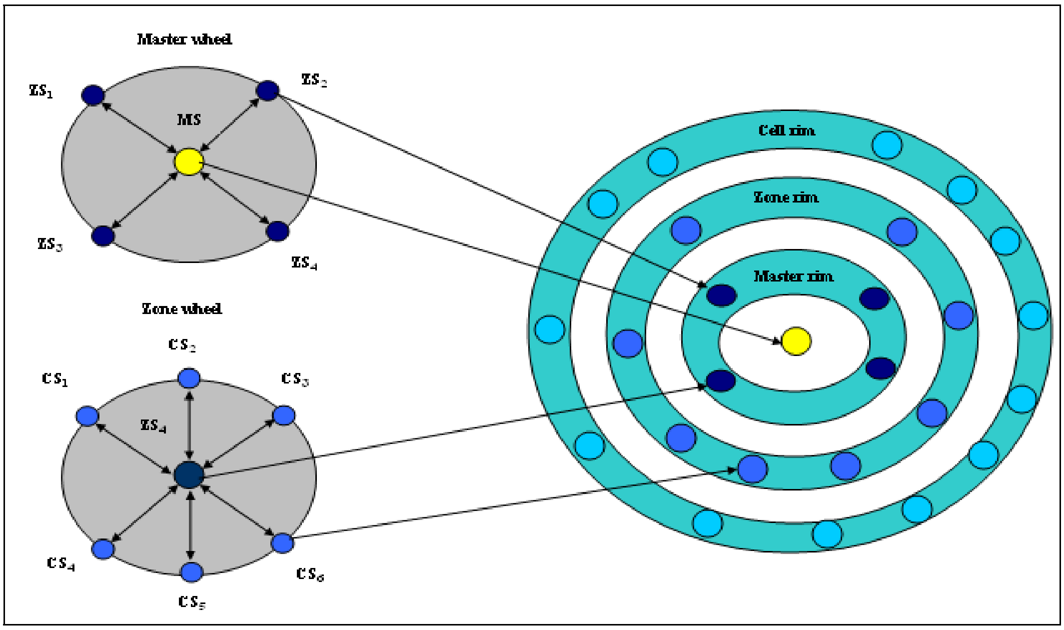


Figure 4.2.2 Mapping of Hidden Wheels in Figure 4.2.1 into the Propagation Wheel

In Figure 4.2.2, the master wheel is mapped into the propagation wheel as a hidden wheel by placing its center point as the main center 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 center point on the master rim and points on its rim (i.e. cell servers) on the zone rim in the propagation wheel.

In the case of the replication system covers only one master area, it scales up by adding new hosts to the different rims and their corresponding spokes. In the case of the replication 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 replication system covers new master areas. Also, in the latter case, the former case is applied by considering inside wheel scalability.

4.3 Hybrid Propagation Mechanisms

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

4.3.1 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 center 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 replication system (i.e. zone servers). The resolution of updates conflict 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 totally ordered updates are propagated from the highest level to the lower levels using TDP propagation.

4.3.2 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 propagate 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 (see Figure 4.3.2.1).

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 the 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.

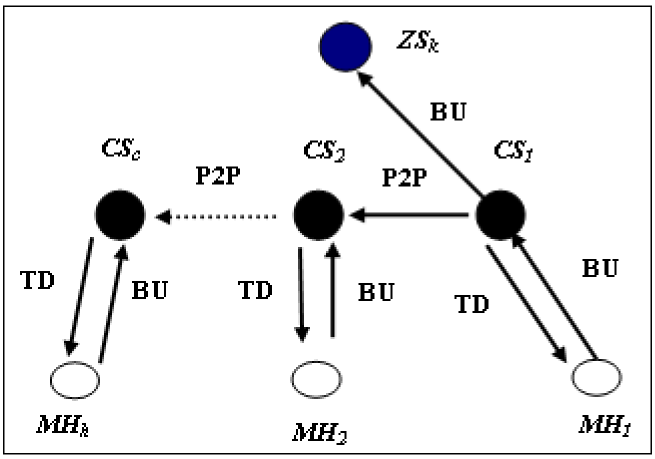


Figure 4.3.2.1 P2P-Concentrate in the Zone Area

The two hybrid mechanisms do not provide guarantee of availability of recent updates always. This is because both cases guaranteed availability and non guaranteed availability depend directly on the propagation of recent updates from their sources in order to be available in the servers of the different areas. The lower level hosts can then obtain these updates when they are synchronized with these servers. Accordingly, the guaranteed availability is achieved only if all sources of updates in the different areas propagated their updates to the fixed network, and these updates reached the server of the area where the host exists before its synchronization with this server. On the other hand, the non guaranteed availability is resulted from the delay that is associated with the propagation of updates from their sources. This delay results in missing some of recent updates due to their unavailability in the server of the area.

4.4 Performance Evaluation

The main objective of the proposed replication strategy is to maintain the consistency and improve the availability 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 analysis 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.

4.4.1 Update Propagation Delay (UPD)

An important requirement in a replication system with a 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 can not rely on the actual propagation times and delays from a host to another.

Definition 4.4.1.1 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 1 illustrates this definition.

Definition 4.4.1.2 The hop is a host that participates in propagating updates from its source to the destination.

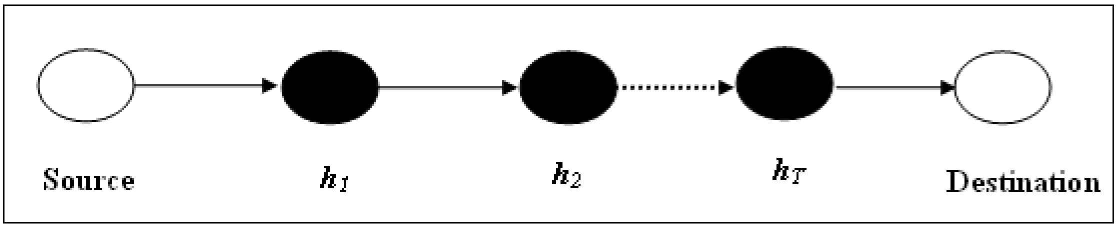


Figure 4.4.1.1 Hops that are involved in propagating an update from the source to the destination

In Figure 4.4.1.1, updates are propagated from the source to the destination through the hops $h_1, h_2, \ldots, h_T$, where T is the total number of hops. The hops are determined according to the type of propagation technique.

4.4.1.1 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.

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

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 at the appendix Two**.**

Same values of UPD are held in despite of 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 UPD for P2P-CONCENTRATE

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

(i) Worst case

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

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 at the appendix Two**.**

(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. 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

4.4.1.2 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 the number of cells and propagation techniques have on the UPD?

2. Which is 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 4.4.1.2.1.

Table 4.4.1.2.1 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 (see appendix Three).

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.

  $Y_{ijk} = \mu + \alpha_i + \beta j + \alpha\beta_{ij} + \varepsilon_{ijk}$ (i=1, 2, 3; j=1, 2, 3,…, 100; k=1,2) (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 4.4.1.2.2. 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 4.4.1.2.2 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 4.4.1.2.2, 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 (Montgomery, 1991).

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 4.4.1.2.3.

Table 4.4.1.2.3 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:

$$S = ({}^{MSE}/n)^{1/2} \qquad (7)$$

   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= $(210.85/200)^{1/2}$ = 1.027

3. Finding the critical value $q_\alpha(k, v)$ from the table of significant ranges (Montgomery, 1991)

   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):

$$R_k = q_\alpha(k, v) * \mathrm{S} \qquad (8)$$

Accordingly, the least significant ranges are:

$R_2 = q_{0.05}\ (2,200)$ * S = 3.787

$R_3 = q_{0.05}\ (3,200)$ * S = 3.947

5. Calculating the actual differences between the different pairs of means (see Figure 4.4.1.2.1) and comparing them with the corresponding least significant ranges.

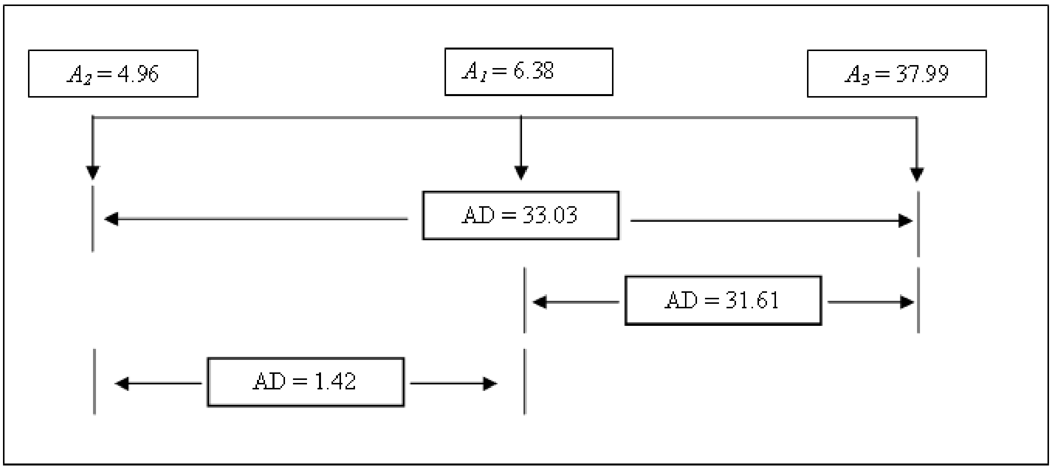


Figure 4.4.1.2.1 Actual Difference between the Different Pairs of Means

According to Figure 4.4.1.2.1, the actual differences between the different pairs of means are:

$A_3 - A_2 = 33.03 > R_3$

$A_3 - A_1 = 31.61 > R_2$

$A_1 - A_2 = 1.42 < R_2$

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 technique (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 the 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.

4.4.2 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 4.4.2.1 The relation between UPD and T can be defined using the following equation.

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

Proof. It is Straightforward and in same manner as for computing UPD (see the appendix Two) 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 Figure 4.4.2.1 and Figure 4.4.2.2 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 Figure 4.4.2.1, 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.

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.

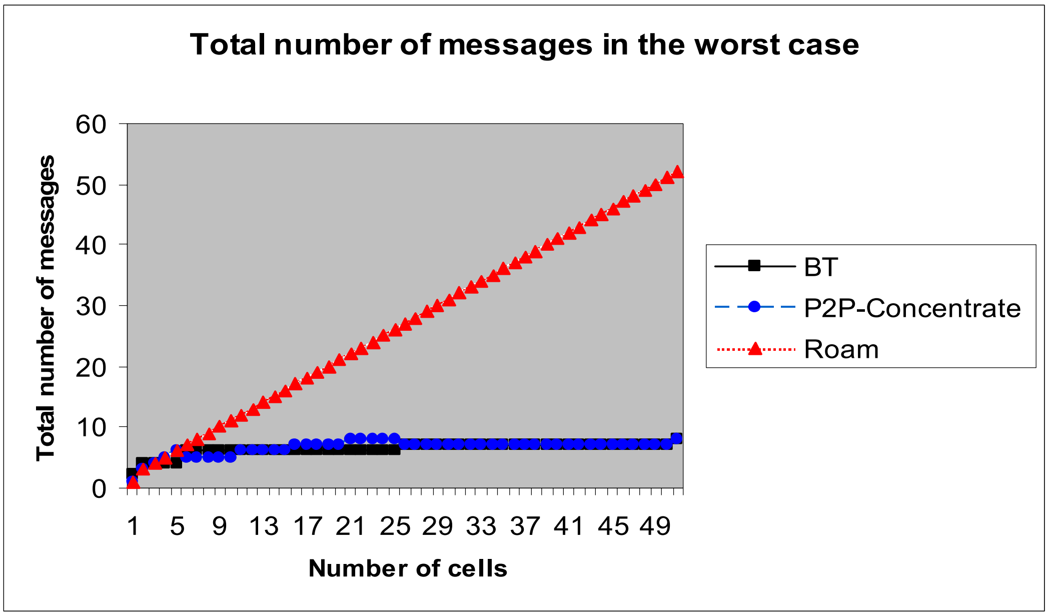


Figure 4.4.2.1 Comparison of the Three Techniques Based on the Total Number of Messages for the Worst Case

As shown in Figure 4.4.2.2, 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).

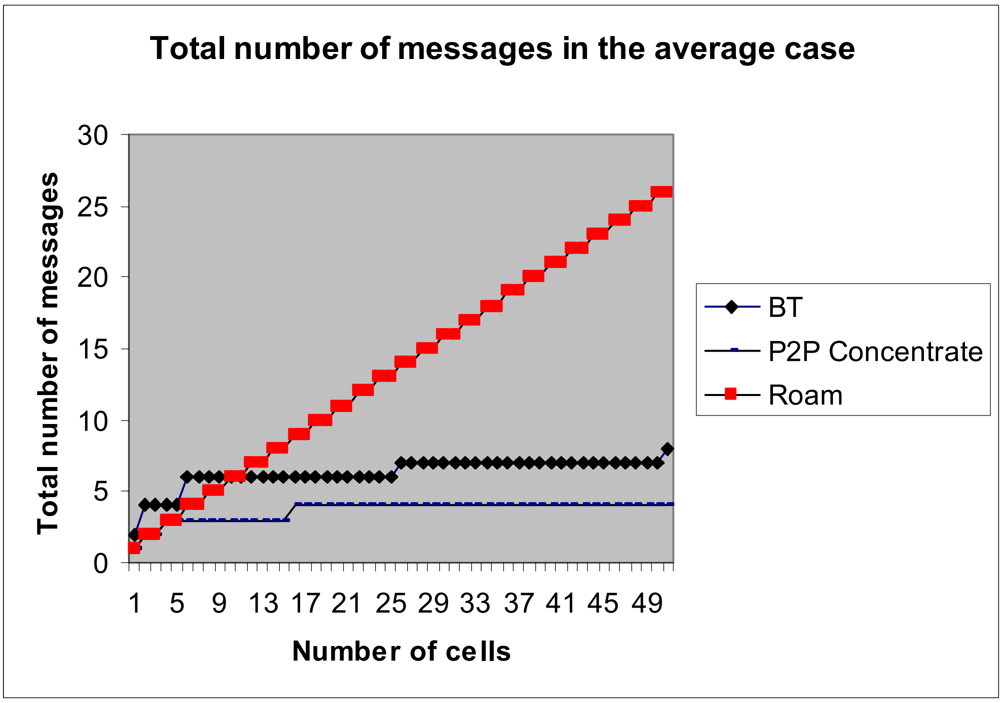


Figure 4.4.2.2 Comparison of the Three Techniques Based on the Total Number of Messages for the Average Case

To enhance the appearance of the details that are included in Figure 4.4.2.1 and Figure 4.4.2.2, we limit the number of cells to 50.

4.4.3 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 4.4.3.1.

Table 4.4.3.1 Parameters for Performance Evaluation Based on ALB

Parameter	Description	Remarks
N	Total number of replicas	$N = 100*n$ (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.
$\bar{C}$	Total number of cells in the replication system	
S	Total number of replicas in the different servers (i.e. CSs, ZSs, and MS)	$S = 1 + Z + \bar{C}$
$\bar{N}$	The average total number of mobile hosts	Estimated based on: $\bar{N} = N - S$
H	Average number of mobile hosts in each cell	Estimated based on: $H = \bar{N} / \bar{C}$
U	Number of updates	1

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

1. The replication 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 4.4.3.1 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):

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

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

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

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

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

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

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

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 the 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 the case of BT or the nearby peer in the case of P2P- CONCENTERATE.

- In equation (13), the mobile host propagates updates to either the cell server in the case of BT or the nearby peer in the 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 (see appendix Four) and a comparative study is performed by considering different values for both the number of replicas and D as follows.

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 Figure 4.4.3.1. 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.

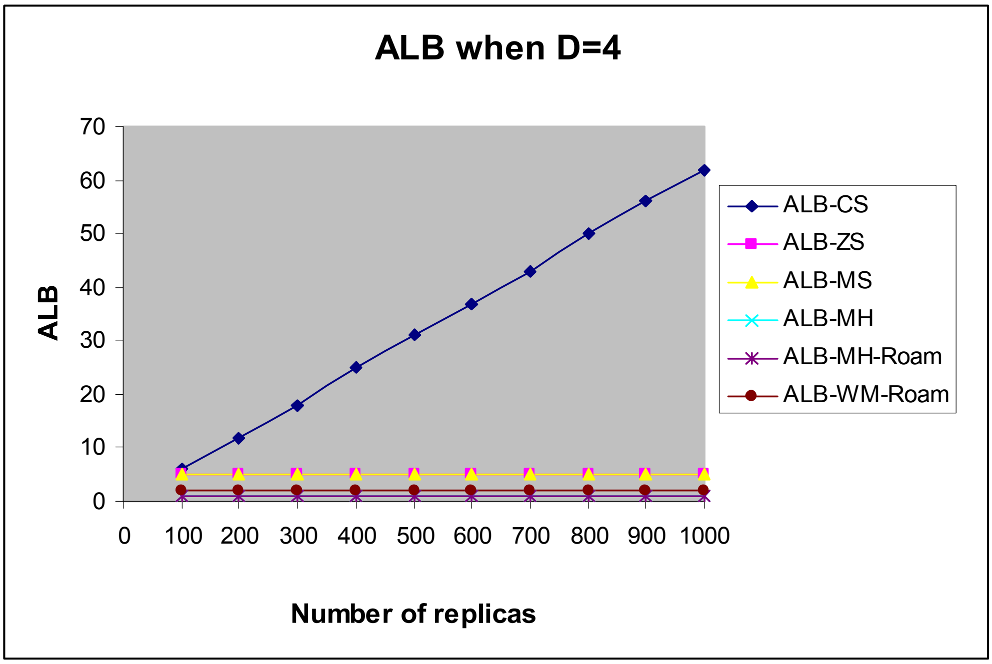


Figure 4.4.3.1 Average Load Balance When D=4

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, …, 10) and the value of D. This variation has a greater impact on ALB-CS as shown in Figure 4.4.3.2. 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 the case of D=5, 6, 7, and 8 the increasing amount is <= 4).

The actual load of 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.

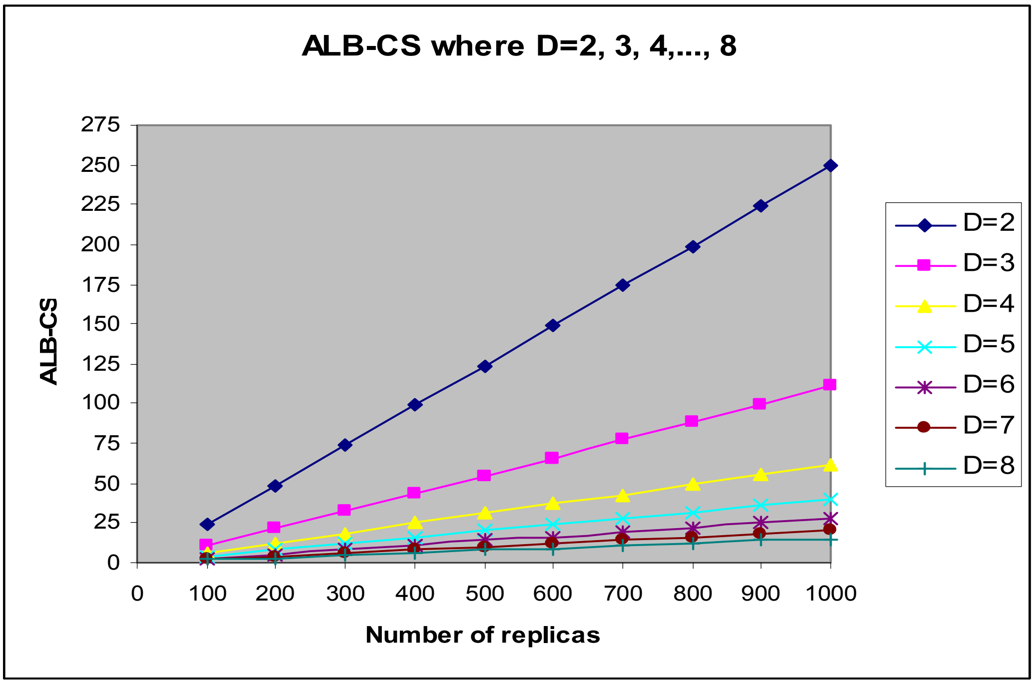


Figure 4.4.3.2 ALB for CS Where D=2, 3, 4,…, 8

Figure 4.4.3.3 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.

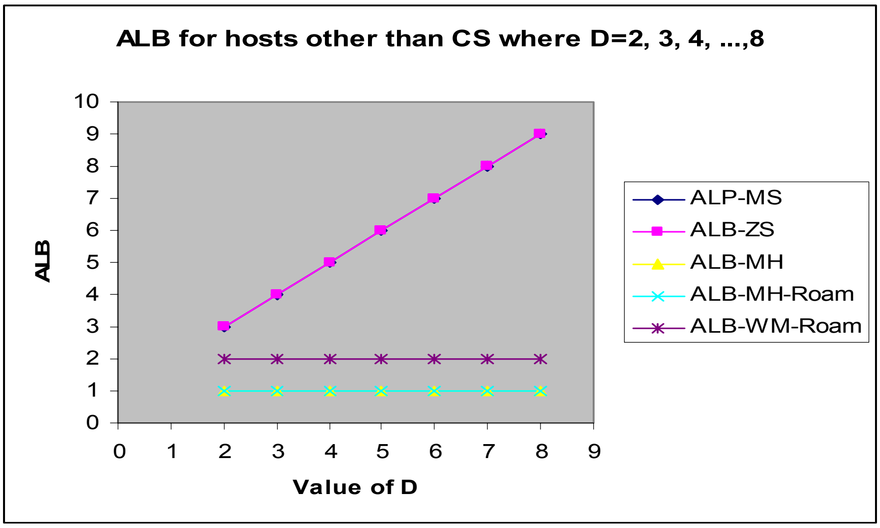


Figure 4.4.3.3 ALB for Hosts other than CS for Different Values of D (D=2, 3, …, 8)

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.

4.4.4 Comparison with N-ary Tree Based Updates Propagation Protocol

The purpose of this comparison is to determine which mode of extension (that characterizes the propagation protocol) is suitable for LMDDBSs in that it achieves lower values of UPD. The extension here refers to the method that the propagation protocol follows for accommodating new replicas and servicing them through providing them with recent updates in the case of the replication system scales up in terms of its size and number of replicas. Two modes of extensions are compared, which are: vertical extension that characterizes N-ary Tree Based Propagation Protocol (NTPP) and horizontal extension that characterizes Wheel-Based Propagation Protocol (WPP). The vertical extension requires adding new replicas to the last level of the *N*-ary tree or placing them in new levels in the case of the last level is full (i.e. there are no empty places for accommodating new replicas). The horizontal extension implies creating new propagation wheels when the replication system covers new distant areas. The vertical extension also characterizes HARP protocol, since it is based on a tree structure. However, here we concern on comparing with NTPP because it is most recent than HARP and same conclusions are applied for both protocols.

In this comparison, the worst case is assumed for both protocols. For NTPP, this case implies that MHj (the host that receives updates occurred in MHi) exists in the last level, while MHi (where update occurred) exists in the root of the tree as this protocol implies. For WPP, the worst case implies that both MHi and MHj exist in the last level of the propagation wheel (most outer rim) and the location of MHj is assumed in the last cell. BT propagation technique is considered for propagation updates between the different propagation wheels, since it is equivalent to P2P-CONCENTRATE in the case of the number of propagation wheels is greater than 1. We assume the propagation technique that is used in NTPP is Top-Down propagation according to its characteristics.

Based on the worst case, the number of hops that is imposed by NTPP for propagating an update from the root of the tree to a host in the last level is L -2, where L is the number of levels of N-ary tree. In contrast, in WPP, the number of hops that are required for propagating an update from MHi to MHj in another cell is fixed (as proved above), which equals 5 in the case of the number of master servers is 1 and the number of zones is greater than or equals 2. Accordingly, if we consider the propagation wheel is equivalent to all levels of N-ary tree in that it encompasses all hosts distributed over the N-ary tree, we find that the number of hops in NTPP for sending updates to MHj is affected by the value of the number of levels, while it does not exceed 5 for different values of L for WPP as shown in Figure 4.4.4.1.

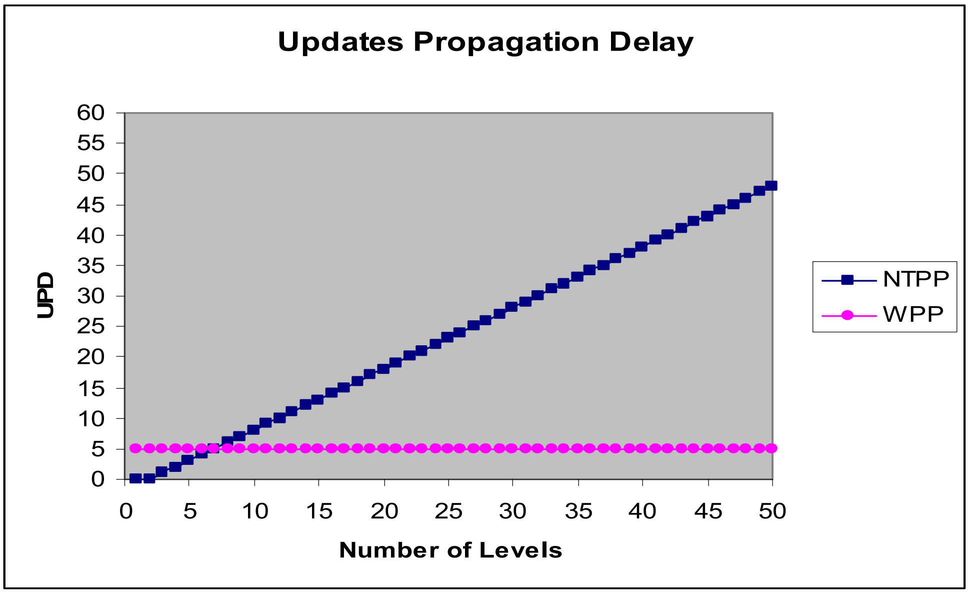


Figure 4.4.4.1. Comparison of the Two Protocols Based on the Updates Propagation Delay for the Worst Case

As shown in figure 4.4.4.1, the value of *L* is varied from 1 to 50. The total number of hops in NTPP is proportional to the value of *L* in that UPD increases as the number of levels that construct the tree gets higher, while it is fixed at 5 for different values of *L* in WPP.

However, the assumption of one propagation wheel that is equivalent to all levels of *N*-ary tree is not always valid, since the replication system may scale to encompass new hosts, especially when it covers new areas. Accordingly, for the sake of equivalent conditions for both NTPP and WPP, we assume that the number of propagation wheels increases as the levels of *N*-ary tree increase. This means that each propagation wheel is equivalent to a certain number of levels of *N*-ary tree. Accordingly, the number of propagation wheels is varied by considering the following cases:

(i) 1 propagation wheel is equivalent to 4 levels of *N*-ary tree.

(ii) 1 propagation wheel is equivalent to 5 levels of *N*-ary tree.

(iii) 1 propagation wheel is equivalent to 6 levels of *N*-ary tree.

(iv) 1 propagation wheel is equivalent to 7 levels of *N*-ary tree.

This variation is based on the aforementioned characteristic of the WPP that it extends in a horizontal manner as the replication system covers new areas, while NTPP extends in a vertical manner to encompass new hosts. By considering this variation, UPD values for each case of the above four cases are calculated based on the following equations:

- For NTPP:

$$UDP = L - 2 \qquad (14)$$

where L is the number of levels

- For WPP:

$$UDP = 4 + m \qquad (15)$$

Where m is the number of propagation wheels as in equation 1 and 2 for measuring UPD for both BT and P2P-CONCENTRATE (Note that one propagation wheel means one master server).

Based on the calculated values of UPD for both protocols, the following conclusions can be reached which are shown in Figure 4.4.4.2 by considering 50 observations for each case, which correspond to 50 levels (see appendix Five).

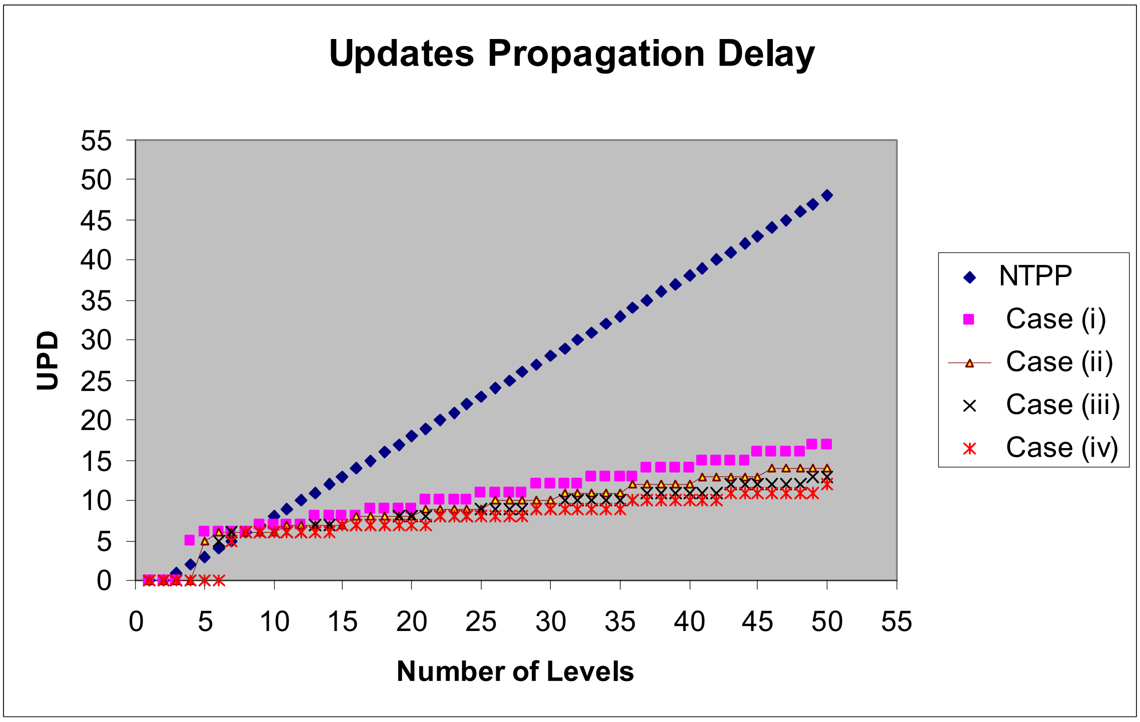


Figure 4.4.4.2 Comparison of the NTPP and the Four Cases of WPP Based on the Updates Propagation Delay for the worst case

In Figure 4.4.4.2, NTPP has higher values for UPD than WPP for any case of the four cases when the number of levels gets higher. The value of this metric is slightly higher in case (i) than the other three cases when the number of levels gets higher. Also, it is slightly higher in case (ii) than the other two cases (i.e. case (iii) and case (iv)), and it is slightly higher in case (iii) than case (iv). This means that when the number of levels that are represented by one propagation wheel increases, the values of UPD get lower. Accordingly, the horizontal extension of WPP affects the value of UPD with a small amount of increasing (i.e. equals to 1) when a new wheel is constructed, while the vertical extension of NTPP increases the value of UPD with amount that is proportional to the number of new levels. Thus, the horizontal exertion is more suitable for LMDDBSs, since these systems extend by covering new areas. Moreover, placing new hosts of those areas in the last level of the *N*-ary tree is not reasonable due to the large values of UPD that are resulted by sending updates to these new hosts.

4.5 Summary

This chapter provided a description of the Wheel-Based updates propagation protocol, which is proposed for LMDDBSs for propagating recent updates between the replicas. The two components of the protocol are described. The first component is a propagation wheel, which represents a water wheel inspired structure that organizes replicas based on their geographical areas and types. A description of how this wheel is constructed is given. The second component is propagation mechanisms for exchanging updates among these replicas. Three basic mechanisms are identified for propagating updates from their sources to a set of other hosts in the propagation wheel, and two hybrid propagation techniques are proposed based on these basic mechanisms. The proposed protocol is compared with Roam with regard to achieving load balance, propagation delay reduction and less communication cost. The required equations that characterize the updates propagation are developed analytically. The results showed that the proposed protocol performs better than Roam for propagating updates in LMDDBSs. Moreover, the proposed protocol is compared with *N*-ary tree based propagation protocol and is concluded that the horizontal extension provided by the proposed protocol is more suitable than the vertical extension for LMDDBSs.

The performance of the proposed updates propagation mechanisms are compared with Roam and NTPP, since they are most recent representatives of the best known replication strategies that are proposed for large scale mobile environments.