Wireless Personal Communications **24:** 71–95, 2003.

HLR Replication Protocols for Global Mobility Management in PCS Networks ⋆

GWO-CHUAN LEE [1], TSAN-PIN WANG [2] ⋆⋆ and CHIEN-CHAO TSENG [1]

[1] *Department of Computer Science and Information Engineering, National Chiao-Tung University, HsinChu 30050, Taiwan, R.O.C.*
[2] *Department of Computer Science and Information Management, Providence University, Shalu 433, TaiChung Taiwan, R.O.C.*
E-mail: tpwang@pu.edu.tw

Abstract. The new trend for PCS networks is to provide mobile users with large-scale mobile capability across many service areas. In this scenario, global database management for PCS networks has become an increasingly important research issue. In this paper, we examined two replicated database strategies, single-replica (SR) and multiple-replica (MR), for large-scale mobility of per-user data management in personal communication networks. The SR strategy uses a single replica approach of HLR. The MR strategy replicates the per-user data of HLR in many regions. The two strategies are based a partial replication scheme, and a primary copy method is used to maintain replicas' consistency. Our numerical results show that the MR strategy outperforms the SR strategy in most situations; however, it may be worse when the probability of a mobile user visiting a foreign region is high and the query rates from other foreign regions are low. Additionally, the number of replicas should be compact in the MR strategy in order to achieve a reasonable query response time. Therefore, we propose an adaptive multiple replication protocol to choose a suitable replication strategy and to decide the optimized number of replicas.

Keywords: personal communication services, mobility management, mobility databases, HLR replication.

1. Introduction

In recent years, personal communication services (PCS) have become popular with people in many countries. People can communicate with one another easily using mobile handsets everywhere at any time. Today, some people may wish to travel across many countries and take a handset to aid in their communications. It is therefore essential to support global mobility services for mobile users in PCS networks.

There has been much research involving mobility database design [2, 3, 6, 7, 11]. However, to support global mobility services efficiently in a wireless environment, database designs with the capability for large-scale mobility have become an important research issue [1, 8, 15, 16]. In this paper, we focus on a mobility database system for PCS networks. The current approach for a PCS database system is a two-level hierarchical system. Two different types of databases are required: HLR (Home Location Register) and VLR (Visitor Location Register) [5, 14, 13, 17]. The HLR is a home database that contains all of the customer records of mobile users, and the VLR is a remote database that will cache the profiles of callees located in a remote region. Thus, when a new call is generated, the query message of the caller will be sent to the HLR to find the location of the VLR where the callee is located. The VLR is then used to retrieve

⋆ This paper was supported by the NSC Projects under Grant No. 90-2213-E-126-009.
⋆⋆ Corresponding author.

information for handling the calls to or from the callee. Due to high terminal mobility and call demands in future PCS networks, the HLR-VLR architecture may perform inefficiently for a heavy traffic load on the signal links of networks close to the centralized HLR [4]. Particularly, when a mobile user travels with large-scale mobility over numerous service areas, the average query response time will increase quickly because of the long searching path to the HLR. The current approach is thus not well suited for global PCS mobility services.

Replication is a popular technique that is commonly used to minimize the traffic load on networks in traditional large-scale databases [18]. The scale of mobility database systems is always larger than the traditional database systems because of mobile hosts' mobility. Therefore, the need for reducing the traffic load and response time of querying on mobility databases is more significant and replication may increasingly become the desired technology. Many studies on replicating data in mobile databases have been discussed [2, 3, 6, 11], and can be classified into two types of replication strategies. The first is the static replication strategy that allocates the replicated data to every replicated database. Each replica contains a complete set of data. If a mobile user alters any data on a local replica, updates must be propagated to all the other replicas to maintain the consistency of the replicated data. The update is always caused by the fast handoff of a mobile host in PCS networks. However, the traffic load for each update is heavy because of the amount of propagation. Thus, the number of replicas and the mobility patterns will effect the performance of mobility database systems. In contrast, a dynamic replication strategy does not replicate the data onto all databases. It replicates the data onto the specified service regions. The allocation for these replicas is dynamic in different time periods. Thus, this strategy will reduce the traffic load by eliminating redundant message flows propagating to the redundant replicas. Some research [3] used this approach to determine the locations of replicas according to the maximum number of callers in given regions.

Leung [11] classified the replication strategies into two kinds: partial replication and full replication. He proposed a partial replication protocol that belongs to a dynamic replication strategy. In this approach, the replicated data always reside in the region where a mobile user is located. When a mobile user alters any data, an update is sent to the replicated database, and forwarded to the mobile user's home database by a delay message. Thus, the replicated database contains a full set of user data, and the home database contains a partial set because of weak consistency. This may reduce the undesirable traffic load on the home database if portions of the data are not requested frequently. He showed that the partial replication protocol is efficient for PCS mobility databases. Nevertheless, this approach is not always suitable for large-scale mobility services. In his design, when a mobile user leaves a service region, the replicated user data in the replicated database will be removed. This reveals that only one replica for each user profile exists in the mobile systems and we view this as a single-replica (SR) strategy. This strategy performs well using the assumption that most queries are generated in the same region where the callee is located. However, the query time is still large if there are many queries generated from other remote sites where the callee is not located. Therefore, a multiple-replica (MR) strategy is necessary for improving the performance of the SR strategy, and the MR strategy should be dynamic. In this paper, we propose an MR strategy to determine whether a profile should be replicated on a service region. The decision depends on the query rate of the service region. We also model the MR strategy based on a partial replication approach to compare with the SR strategy. Moreover, we propose an adaptive multiple replication (AMR) protocol to improve the performance of the MR strategy.

The rest of this paper is organized as follows. In Section 2, we briefly describe the SR and MR strategies in mobility database systems. In Section 3, we analyze their performance. The

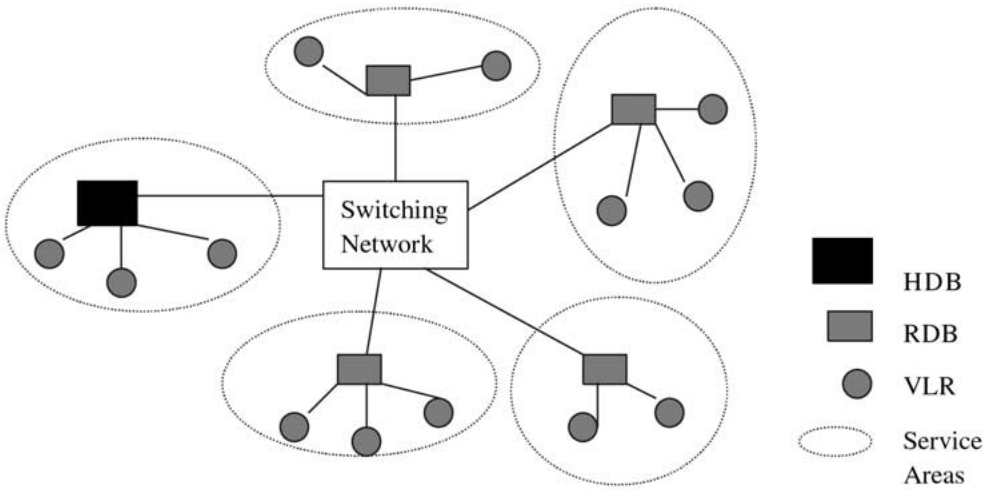


Figure 1. Replicated HLRs database systems.

performance results of the two strategies are discussed in Section 4. Next we propose an AMR protocol in Section 5. Finally, we conclude this paper in Section 6.

2. HLR Replication Strategies

In this section, we describe the two replication strategies, SR and MR strategies, for mobility database systems. In PCS networks, the VLR needs to cache the replication profile generated from HLR. Thus, the traditional HLR-VLR architecture can be viewed as a partial replication scheme. In this paper, we focus on the replicated HLRs of PCS networks. First, we describe the infrastructure of the replicated mobility database systems shown in Figure 1. The *home database* (HDB) and *replicated databases* (RDBs) involve the HLR functions. HDB contains the same customer data that the traditional HLR does. RDB is a database that may contain a replica of user data from HDB. In a service area, there are many VLRs managed by the HDB or RDB. For simplicity, we assume that a service area contains only an HDB or an RDB. The service area that contains the HDB is called the *home region*, and the service area that contains an RDB is called a *foreign region*. The sizes of the service areas may be different according to some cost constraints, for example, the population of customers.

2.1. Single-Replica Strategy

The SR strategy is similar to the partial replication approach proposed by Leung [11]. In this approach, only one replica of a user profile is created along with the movement of a mobile user. When the mobile user leaves the HDB service area and enters a foreign region, it will create a replica of its profile in the RDB. However, if the mobile user continues to move to another foreign region, the replica in the previous RDB is removed and a new replica is created in the new RDB. The replica can be built using a registration and a download message sent by the mobile user. The user can then immediately update all data on the visiting RDB, but forward the data to the HDB only at each registration time. The RDB can thus be viewed as a primary site that contains a full replica of data, and the HDB is a secondary site that owns the partial data. The partial data are the *infrequently* updated information that may not contain

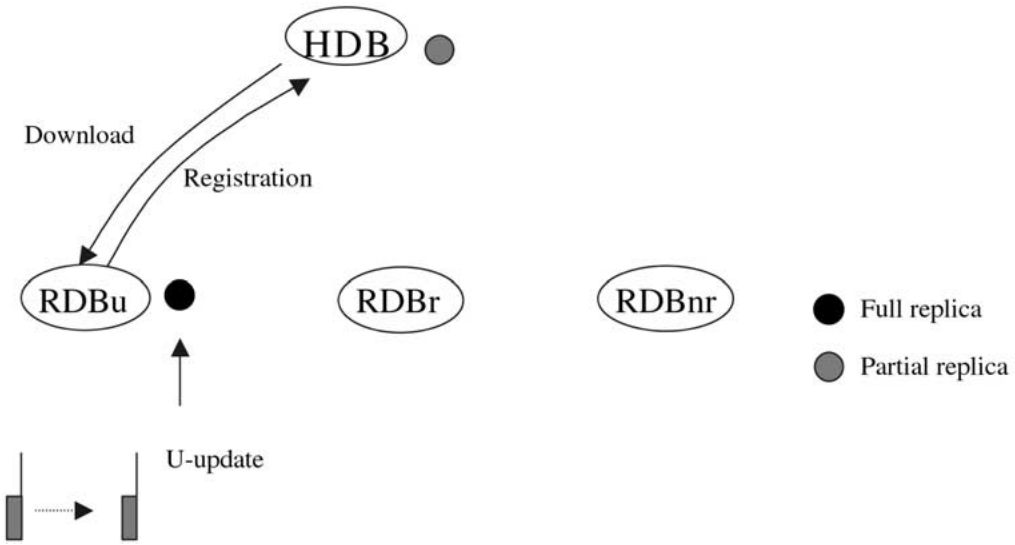


Figure 2. Traffic of registrations/downloads when a mobile user visits a foreign region in SR strategy.

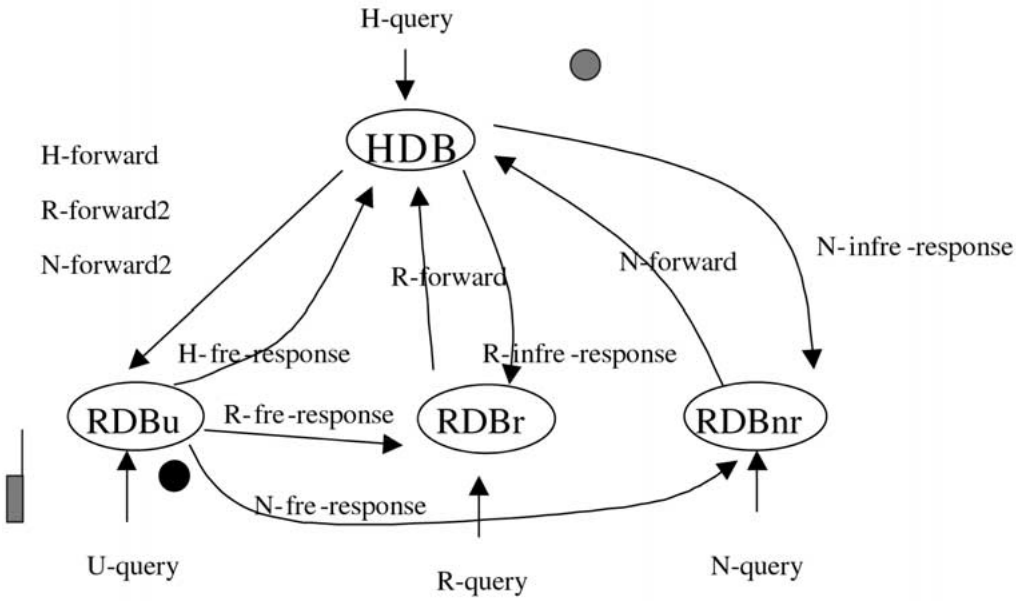


Figure 3. Traffic of queries/responses when a mobile user visits a foreign region in SR strategy.

the current location for high mobility hosts. The primary copy method such as PWP protocol [12, 19] can be used to maintain the consistency of replicated data. The SR strategy is shown in Figures 2, 3 and 4. The RDB_u denotes the replicated databases in a foreign region where a callee is located. The RDB_r and RDB_{nr} both represent the replicated databases without any replica of the callee's profile.

As Figure 2 shows, when a mobile user visits a foreign region, it first sends a registration message to the HDB and downloads its profile onto the RDB_u as a replica. The user can thus change any data on the local replica without any delay. In the scenario, the replica can be viewed as a full replica. When the user needs to send a registration to HDB again, it sends

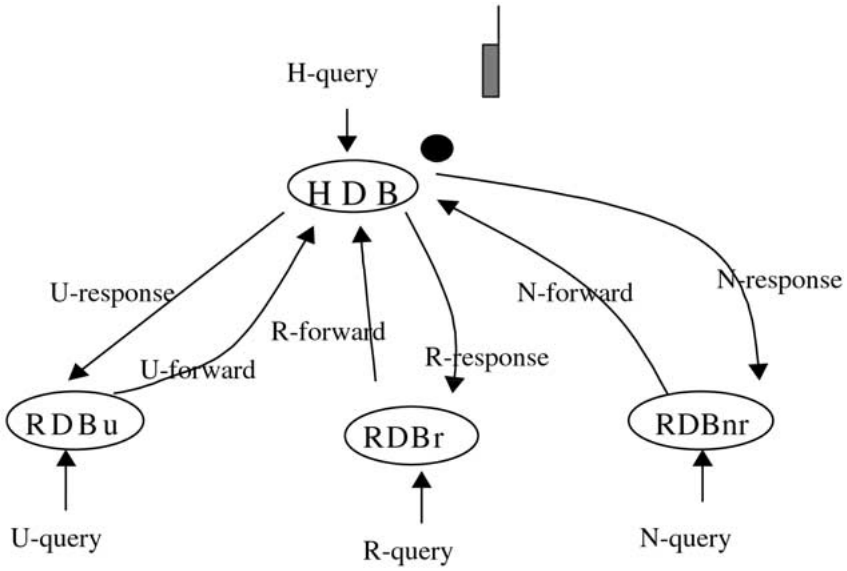


Figure 4. Traffic of queries/responses when a mobile user visits the home region in SR strategy.

all the updated data including the frequently and infrequently updated data to the HDB. Therefore, the data on HDB can be viewed as a partial replica because the registration to HDB may be delayed after the change of data on RDB_u.

In Figure 3, the queries (marked as U-query) generated in the foreign region are processed first by the RDB_u, all updated data can be obtained immediately because of the full replicas in the RDB_u. The queries (marked as N-query) generated in other foreign regions with no replicas are processed by the RDB_{nr} first. When failing to access the user profile, RDB_{nr} forwards a message (marked as N-forward) to HDB to query the lost data. Thus, infrequently updated data can be found at the HDB and returned using a response (marked as N-infre-response). Additionally, the RDB_{nr} sends another forward message (marked as N-forward2) to RDB_u to query the frequently updated data, and the results are returned using N-fre-response directly to the user. We assume that there are some network links from RDB_u to both the RDB_r and RDB_{nr}. The queries marked R-query have a similar protocol for N-query, because in the SR model the RDB_r is a database without any replica. If there are some queries (marked as H-query) generated in the home region, HDB sends a forward message (marked as H-forward) to RDB_u when failing to access the frequently updated data. The results are returned using the response marked as H-fre-response.

As indicated in Figure 4, when a mobile user visits the home region, there are no replicas in the databases of the foreign regions. If there are some queries (marked as U-query, R-query and N-query) generated in foreign regions, the foreign regions will send the forward messages (marked as U-forward, R-forward and N-forward) to HDB to query the required data including frequently and infrequently updated data. These results are returned using responses marked as U-response, R-response and N-response.

2.2. Multiple-Replica Strategy

The MR strategy may replicate many copies of user data in some RDBs, and use the partial replication policy and the primary copy method to manage the replicas. When a mobile user

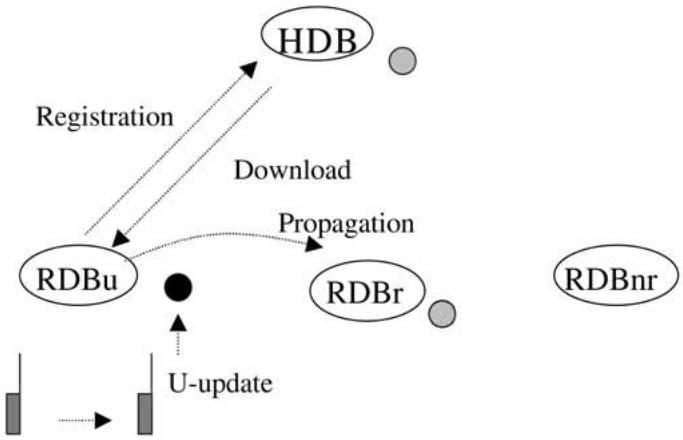


Figure 5. Traffic of registrations/downloads and propagations when a mobile user visits a foreign region in MR strategy.

visits a foreign region collocated with the RDB_u, it downloads its profile from HDB immediately, in the same way as the SR strategy. The decision for replicating a profile on an RDB where the callee is not located is based on the query rate of callers in that region. If the query rate is high, the RDB will replicate the profile, and is referred to as RDB_r. If the query rate is low, the RDB has no replica, and is referred to as RDB_{nr}. The RDB_u that the mobile user visits is viewed as a primary site, and the RDB_rs with replicas are the secondary sites. When the mobile user needs to change data, all updated data are written to the primary RDB_u and the *frequently* updated data are propagated to the secondary RDB_rs immediately. Intuitively, the cache scheme [9] will have a good performance in handling the *infrequently* updated information on the RDB_rs. Hence, we don't need to propagete the infrequently updated information to the HDB immediately. The updated data are written to HDB only at the registration time; thus, the HDB keeps a partial replica because of weak consistency. The underlying principle of the MR strategy is to reduce the overhead of querying the frequently updated data. This strategy is described from Figures 5 to 8.

As Figure 5 shows, when a mobile user visits a foreign region, it sends a registration to HDB and downloads its profile onto RDB_u as a replica. The user can change all the data on the replica without any delay, but it needs to propagate the frequently updated data to those foreign regions marked as RDB_r with the same replicas for consistency. Similar to the SR model, the user alters the frequently and infrequently updated data in the HDB only at the registration time.

In Figure 6, if there are queries (marked as U-query) generated in the foreign region where a mobile user is located, these queries are processed by RDB_u first. Then all updated data can be obtained immediately because of the full replica on RDB_u. If there are queries (marked as R-query) generated in other foreign regions with the replicas of a callee, these queries are processed by RDB_r first. When failing to access the infrequently updated data that is not replicated on the RDB_r, RDB_r sends a forward message (marked as R-forward) to HDB. The HDB will then generate a response (marked as R-infre-response) to carry the infrequently updated data to the user. Additionally, it forwards another query (marked as R-forward2) to RDB_u to request the lost information of frequently updated data. Finally, RDB_u sends the R-fre-miss-response message to return the lost data directly to the user. If there are queries (marked as N-query) generated in other foreign regions with no replicas, these queries are

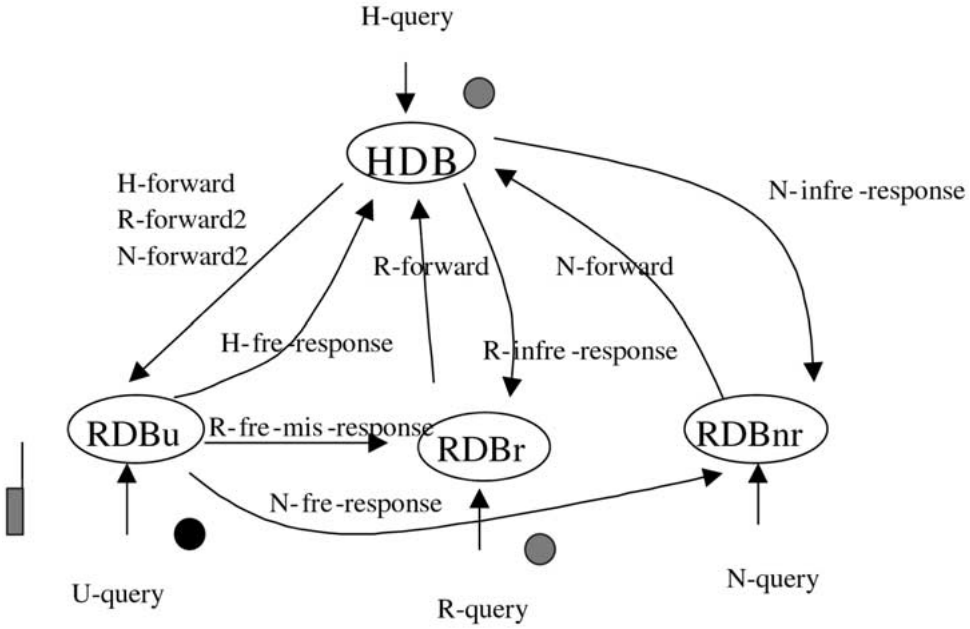


Figure 6. Traffic of queries/responses when a mobile user visit a foreign region in MR strategy.

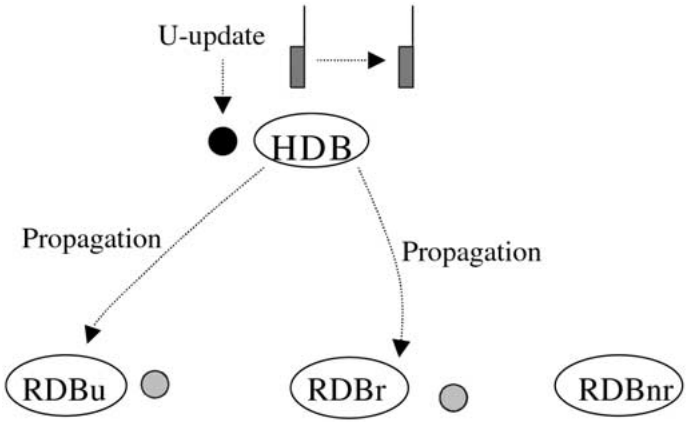


Figure 7. Traffic of propagations when a mobile user visits the home region in MR strategy.

processed by RDB_{nr} first. Then RDB_{nr} sends a forward message (marked as N-forward) to HDB to query the user profile. The HDB will return the infrequently updated data to the user using the N-infre-response message. Additionally, HDB sends a forward message (marked as N-forward2) to RDB_u to query the frequently updated data, and the results are returned directly to the user using N-fre-response message. If there are some queries (marked as H-query) coming from the home region, the infrequently updated data can be directly obtained from HDB. The HDB also sends a forward message (marked as H-forward) to RDB_u because the frequently updated data were not accessed from HDB. Finally, the lost data are returned to the user using the H-fre-response message.

When a mobile user visits the home region, as indicated in Figure 7, the operations of registration and download are performed locally on the HDB. If the user wishes to update his profile, only the frequently updated data need to be propagated to both RDB_u and RDB_r.

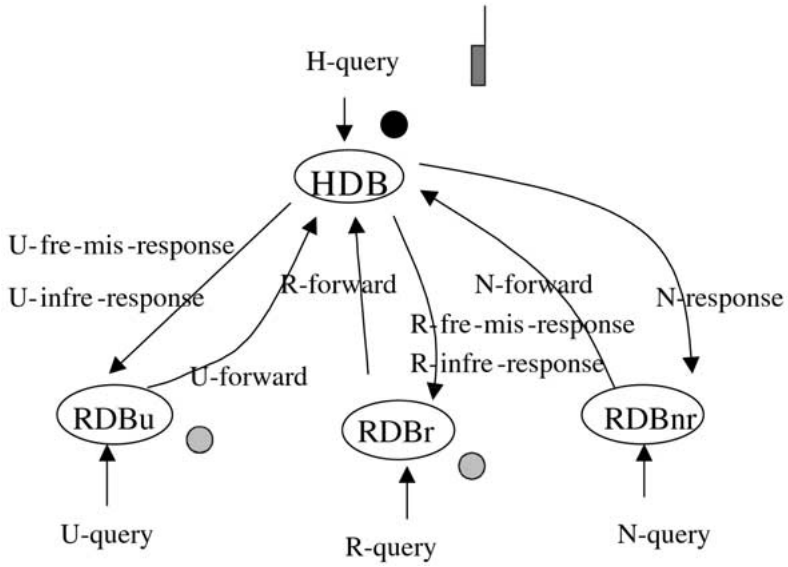


Figure 8. Traffic of queries/responses when a mobile user visits the home region in MR strategy.

As Figure 8 shows, if there are queries (marked as R-query) generated in the foreign region with a replica of the user profile, these queries are first processed by RDB_r. The frequently updated data can be obtained immediately. RDB_r then sends a forward message (marked as R-forward) to HDB to query the infrequently updated data and the possibly obsolete replica of frequently updated data. These results are returned using the responses marked as R-infre-response and R-fre-mis-response. The U-query has the same behavior as R-query. If there are some queries (marked as N-query) sent to the RDB_{nr}, RDB_{nr} will send a forward message (marked as N-forward) to HDB to query all the needed information because of no replica on the databases. The results are directly returned to the callers using the N-response message.

3. Performance Analysis of the Replication Strategies

The performance evaluation of the two replication strategies is based on queuing models. For simplicity, the databases and the network links are also modeled as independent $M/M/1$ queues in our assumption.

3.1. Definitions of Key Parameters

To develop our model, first we define the following parameters. Some of the notations are consulted from Leung's paper [11].

- ϕ represents the probability that a callee visits at a foreign region.
- $1 - \phi$ represents the probability that a callee visits at the home region.
- φ represents the missed rate of querying the frequently updated data on RDB_r. The miss is caused by the network link transmission delay.
- α represents the probability that a call is generated from a foreign region collocated with RDB_u.
- β represents the probability that a call is generated from a foreign region collocated with RDB_r.

Table 1. Fraction of call-types.

From	To HDB	To RDB_u
HDB	$(1-\alpha-k\beta-s\gamma)(1-\phi)$	$(1-\alpha-k\beta-s\gamma)\phi$
RDB_u	$\alpha(1-\phi)$	$\alpha\phi$
RDB_r	$k\beta(1-\phi)$	$k\beta\phi$
RDB_{nr}	$s\gamma(1-\phi)$	$s\gamma\phi$

- γ represents the probability that a call is generated from a foreign region collocated with RDB_{nr}.
- R_q represents the query to update rate ratio, which describes the average number of data queries per update.
- R_g represents the query to registration rate ratio, which describes the average number of data queries per registration. In general, the registration rate is less than the update rate; thus, the ratio R_g is larger than R_q.
- N_q represents the average number of queries for each call. We also assume that only $\frac{1}{N_q}$ queries for each call will access to the frequently updated data.
- F_u represents the probability of updating the frequently updated data for each update.
- N represents the number of total service regions.
- k represents the number of service regions which contains a replica of the profile for a callee.
- s represents the number of service regions without any replica of the profile for a callee.
- ρ represents the server loads on all databases of the mobile systems.
- X_q, $\overline{X_q^2}$ represent average and second moment of query processing time on databases.
- X_u, $\overline{X_u^2}$ represent average and second moment of update processing time on databases.
- Y_q, $\overline{Y_q^2}$ represent average and second moment of transmission time for a query or response on the network links.
- Y_u, $\overline{Y_u^2}$ represent average and second moment of transmission time for an update on the network links.
- Y_d, $\overline{Y_d^2}$ represent average and second moment of transmission time for a downloaded data on the network links.
- τ represents the average delay of transmission or propagation time on the network links.

According to our model, there are fractions of eight call-types shown in Table 1. For example, a fraction $\alpha(1-\phi)$ of call rates represents the calls generated from the foreign region collocated with RDB_u and destined for a callee at the home region.

For fair comparison of the two replication strategies, we fix the server load ρ. That is, the server loads on all databases of SR and MR models are the same. We can derive the arrival rates of queries and updates from the processing times and the ratio of query-to-update. Thus, the arrival query rate λ_q and the arrival update rate λ_u on all databases can be obtained as follows.

$$\lambda_q = \frac{\rho}{X_q + \frac{X_u}{R_q}}, \qquad \lambda_u = \frac{\lambda_q}{R_q}. \tag{1}$$

The performance metric for comparison is the average query response time. In the followings, we derive the average query response time of the two models by conditioning the traffic load from different types of replicated databases and network links.

3.2. SINGLE-REPLICA MODEL

In our model, we assume that there are N service areas in total, and each service area contains an RDB except the home region, which contains an HDB. Then we obtain that the number s of RDB_{nr}s is equal to $N-k-2$, because there are k RDB_rs and an RDB_u in the $N-1$ RDBs.

Load of RDB_u: The query load on the RDB_u includes all queries marked as U-query and the forward messages marked as R-forward2, N-forward2 and H-forward. Therefore, the probability P_u^q of query rate on RDB_u can be derived as follows according to Table 1.

$$\begin{aligned} P_u^q &= \alpha + k\beta\frac{\phi}{N_q} + s\gamma\frac{\phi}{N_q} + (1-\alpha-k\beta-s\gamma)\frac{\phi}{N_q} \\ &= \alpha + (1-\alpha)\frac{\phi}{N_q}\,. \end{aligned} \tag{2}$$

The fraction $\frac{1}{N_q}$ means that only one of the N_q queries on the HDB will access to the frequently updated data. These queries need to be forwarded to RDB_u for further processing because of the partial replication scheme. Additionally, the updates on RDB_u are all generated from the callee of a foreign region. Thus, combining the query and update load, the server load ρ_u of RDB_u is

$$\rho_u = P_u^q\lambda_q X_q + \phi\lambda_u X_u\,. \tag{3}$$

According to the queuing theory [10], the average query response time T_{uq} on RDB_u is equal to the summation of the query processing time X_q and average waiting time. Then we obtain

$$T_{uq} = X_q + \frac{P_u^q\lambda_q\overline{X_q^2} + \phi\lambda_u\overline{X_u^2}}{2(1-\rho_u)}\,. \tag{4}$$

Load from RDB_u to HDB: The traffic load of the network link from RDB_u to HDB includes the forward query marked as U-forward, the query response marked as H-freq-response , and the registration. Thus, the probability Q_{uh}^q of the query rate on the link is derived using

$$Q_{uh}^q = \alpha(1-\phi) + (1-\alpha-k\beta-s\gamma)\frac{\phi}{N_q}\,. \tag{5}$$

The registrations are all generated from the callee of the foreign region collocated with RDB_u. $\frac{\lambda_q}{R_g}$ is the registration rate. Thus, the traffic load β_{uh} from RDB_u to HDB is

$$\beta_{uh} = Q_{uh}^q\lambda_q Y_q + \phi\frac{\lambda_q}{R_g}Y_u\,. \tag{6}$$

The average query response time R_{uh}^q on the link is equal to the summation of the transmission time Y_q and average waiting time. Then we can obtain

$$R_{uh}^q = Y_q + \frac{Q_{uh}^q \lambda_q \overline{Y_q^2} + \phi \frac{\lambda_q}{R_g} \overline{Y_u^2}}{2(1-\beta_{uh})}. \tag{7}$$

Load on HDB: There are four types of queries, an update and a registration on the database HDB. These queries are H-query, U-forward, R-forward, and N-forward. Thus the probability P_h^q of query rate on HDB can be derived using

$$\begin{aligned} P_h^q &= (1-\alpha-k\beta-s\gamma)+\alpha(1-\phi)+k\beta+s\gamma \\ &= 1-\alpha\phi\,. \end{aligned} \tag{8}$$

The update comes from the home region of the callee, and the registration is generated from the callee of a foreign region. Combining these messages, the server load ρ_h on HDB is given using

$$\rho_h = P_h^q \lambda_q X_q + \left[(1-\phi)\lambda_u + \phi\frac{\lambda_q}{R_g}\right] X_u\,, \tag{9}$$

and the average query response time T_{hq} is given using

$$T_{hq} = X_q + \frac{P_h^q \lambda_q \overline{X_q^2} + [(1-\phi)\lambda_u + \phi\frac{\lambda_q}{R_g}]\overline{X_u^2}}{2(1-\rho_h)}\,. \tag{10}$$

Load from HDB to RDB$_u$: The traffic load from HDB to RDB$_u$ includes the forward messages marked as H-forward, R-forward2 and N-forward2, and the query response marked as U-response. Then the probability Q_{hu}^q of query rate on the link is given using

$$\begin{aligned} Q_{hu}^q &= (1-\alpha-k\beta-s\gamma)\frac{\phi}{N_q} + k\beta\frac{\phi}{N_q} + s\gamma\frac{\phi}{N_q} \\ &\quad +\alpha(1-\phi) \\ &= (1-\alpha)\frac{\phi}{N_q} + \alpha(1-\phi)\,. \end{aligned} \tag{11}$$

Whenever a mobile user visits a foreign region, the profile needs to be downloaded at registration time. Thus, the load β_{hu} on the link is derived as

$$\beta_{hu} = Q_{hu}^q \lambda_q Y_q + \phi\frac{\lambda_q}{R_g} Y_d\,. \tag{12}$$

The average query response time of R_{hu}^q is

$$R_{hu}^q = Y_q + \frac{Q_{hu}^q \lambda_q \overline{Y_q^2} + \phi\frac{\lambda_q}{R_g}\overline{Y_d^2}}{2(1-\beta_{hu})}\,. \tag{13}$$

Loads on RDB$_r$ and RDB$_{nr}$: In this model, RDB$_r$ and RDB$_{nr}$, which do not contain any replicas of the callee can be viewed as the same type of replicated databases. For simplicity, we combine the two loads into one for calculation. The server loads of the two databases include the R-query and N-query separately. Thus, we derive the probability P_n^q of query rate and the load ρ_n on the two databases as follows.

$$P_n^q = k\beta + s\gamma \,, \tag{14}$$

$$\rho_n = P_n^q \lambda_q X_q \,. \tag{15}$$

Then the average query response time T_{nq} on RDB$_r$/RDB$_{nr}$ is obtained using

$$T_{nq} = X_q + \frac{P_n^q \lambda_q \overline{X_q^2}}{2(1-\rho_n)} \,. \tag{16}$$

Load from RDB$_r$/RDB$_{nr}$ to HDB: Considering the uplink traffic loads from RDB$_r$/RDB$_{nr}$ to HDB, the corresponding messages include R-forward and N-forward. For simplicity, the two links are also combined into one for calculation. Thus, the probability Q_{nh}^q of query rate and the load β_{nh} on the two links can be derived using

$$Q_{nh}^q = k\beta + s\gamma \,, \tag{17}$$

$$\beta_{nh} = Q_{nh}^q \lambda_q Y_q \,, \tag{18}$$

and the average query response time R_{nh}^q is obtained using

$$R_{nh}^q = Y_q + \frac{Q_{nh}^q \lambda_q \overline{Y_q^2}}{2(1-\beta_{nh})} \,. \tag{19}$$

Loads from HDB to RDB$_r$/RDB$_{nr}$: The downlink traffic loads from HDB to RDB$_r$/RDB$_{nr}$ include four types of query responses marked as R-infre-response, R-response, N-infre-response and N-response. Thus, the probability Q_{hn}^q of query rate can be derived using

$$\begin{aligned} Q_{hn}^q \;=\; & k\beta \frac{N_q - 1}{N_q}\phi + k\beta(1-\phi) + s\gamma \frac{N_q - 1}{N_q}\phi \\ & + s\gamma(1-\phi) \,. \end{aligned} \tag{20}$$

The fraction $\frac{N_q-1}{N_q}$ means that only $(N_q - 1)$ of the N_q queries on HDB will access the infrequently updated data successfully if the user visits at a foreign region. Then the traffic load β_{hn} on the two links is

$$\beta_{hn} = Q_{hn}^q \lambda_q Y_q \,, \tag{21}$$

and the average query response time R_{hn}^q is given using

$$R_{hn}^q = Y_q + \frac{Q_{hn}^q \lambda_q \overline{Y_q^2}}{2(1-\beta_{hn})} \,. \tag{22}$$

Loads from RDB_u to RDB_r/RDB_{nr}: The traffic loads of network links directly from RDB_u to RDB_r and RDB_{nr} are caused by the messages marked as R-fre-response and N-fre-response, respectively. Therefore, the probability Q^q_{un} of query rate and the load β_{un} on the links is derived using

$$Q^q_{un} = k\beta\frac{\phi}{N_q} + s\gamma\frac{\phi}{N_q}\,, \tag{23}$$

$$\beta_{un} = Q^q_{un}\lambda_q Y_q\,, \tag{24}$$

and the average query response time R^q_{un} is given using

$$R^q_{un} = Y_q + \frac{Q^q_{un}\lambda_q\overline{Y^2_q}}{2(1-\beta_{un})}\,. \tag{25}$$

By combining all of traffic loads derived above, we can obtain the average query response time T^{SR}_q for SR model as follows.

$$\begin{aligned} T^{\mathrm{SR}}_q \;=\; & P^q_u T_{uq} + Q^q_{uh}(R^q_{uh}+\tau) + Q^q_{hu}(R^q_{hu}+\tau) \\ & + P^q_h T_{hq} + P^q_n T_{nq} + Q^q_{nh}(R^q_{nh}+\tau) \\ & + Q^q_{hn}(R^q_{hn}+\tau) + Q^q_{un}(R^q_{un}+\tau)\,. \end{aligned} \tag{26}$$

3.3. Multiple-Replica Model

We proceed to analyze the query response time of the MR model by conditioning the traffic load from the different types of replicated databases and network links.

Load on RDB_u: The query load of RDB_u is caused by all the U-query and forward messages marked as R-forward2, N-forward2 and H-forward similar to the SR model. The only difference is that the R-forward2 message represents the missed requests of R-query for accessing the frequently updated data on the RDB_r. Parameter φ describes the missed rate caused by the propagating delay between RDB_u and RDB_r. Thus, the probability P^q_u of query rate on the database can be derived.

$$\begin{aligned} P^q_u \;&=\; \alpha + k\beta\frac{\phi}{N_q}\varphi + s\gamma\frac{\phi}{N_q} + (1-\alpha-k\beta-s\gamma)\frac{\phi}{N_q} \\ &=\; \alpha + k\beta\frac{\phi}{N_q}\varphi + (1-\alpha-k\beta)\frac{\phi}{N_q}\,. \end{aligned} \tag{27}$$

Additionally, the update load on RDB_u includes all updates sent from the callees at the foreign region and the updates propagated from the callees at the home region. The propagating rate is $\lambda_u F_u$ which is the rate for updating the frequently updated data. Therefore, the server load ρ_u is obtained using

$$\rho_u = P^q_u\lambda_q X_q + [\phi\lambda_u + (1-\phi)\lambda_u F_u]X_u\,, \tag{28}$$

and the average query response time T_{uq} is equal to the summation of the query processing time X_q and average waiting time. Then we obtain

$$T_{uq} = X_q + \frac{P^q_u\lambda_q\overline{X^2_q} + [\phi\lambda_u + (1-\phi)\lambda_u F_u]\overline{X^2_u}}{2(1-\rho_u)}\,. \tag{29}$$

Load from RDB$_u$ to HDB: The traffic load from RDB$_u$ to HDB includes the query response marked as H-fre-response and the forward message marked as U-forward. Thus, the probability Q_{uh}^q of query rate on the link is

$$Q_{uh}^q = (1-\alpha-k\beta-s\gamma)\frac{\phi}{N_q} + \alpha(1-\phi)\left(\frac{N_q-1}{N_q}+\frac{\varphi}{N_q}\right). \tag{30}$$

The fraction $\frac{N_q-1}{N_q}$ means that the portion for accessing the infrequently updated data, and $\frac{\varphi}{N_q}$ means the failed queries for accessing the frequently updated data. U-forward message includes the two querying messages. Besides, the load also includes the registration message sent from the callees of the foreign region with RDB$_u$. Then the traffic load β_{uh} on the link can be derived as follows.

$$\beta_{uh} = Q_{uh}^q \lambda_q Y_q + \phi\frac{\lambda_q}{R_g} Y_u\,. \tag{31}$$

The average query response time R_{uh}^q is given using

$$R_{uh}^q = Y_q + \frac{Q_{uh}^q \lambda_q \overline{Y_q^2} + \phi\frac{\lambda_q}{R_g}\overline{Y_u^2}}{2(1-\beta_{uh})}\,. \tag{32}$$

Load on HDB: From Figures 5 to 8, the server load on HDB is caused by four types of queries and two types of updates. One type of the queries is the U-forward message described above. The others include the query load on HDB (marked as H-query) generated from the home region, the R-forward message, and the N-forward message. The R-forward includes the messages of querying for the infrequently updated data and the messages missed at accessing to the replica of the frequently update data. Thus, the probability P_h^q of query rate on HDB can be derived in the following

$$\begin{aligned} P_h^q &= \alpha(1-\phi)\left(\frac{N_q-1}{N_q}+\frac{\varphi}{N_q}\right) + (1-\alpha-k\beta-s\gamma) \\ &\quad + k\beta\left(\frac{N_q-1}{N_q}+\frac{\varphi}{N_q}\right) + s\gamma \\ &= \alpha(1-\phi)\left(\frac{N_q-1}{N_q}+\frac{\varphi}{N_q}\right) + (1-\alpha-k\beta) \\ &\quad + k\beta\left(\frac{N_q-1}{N_q}+\frac{\varphi}{N_q}\right). \end{aligned} \tag{33}$$

The two types of updates include all of the updates generated from the callees at home, and the registrations generated from the callees at the foreign region. Therefore, the load ρ_h on HDB can be obtained by conditioning the queries, updates and registration loads.

$$\rho_h = P_h^q \lambda_q X_q + \left[(1-\phi)\lambda_u + \phi\frac{\lambda_q}{R_g}\right] X_u\,. \tag{34}$$

The average query response time T_{hq} is given using

$$T_{hq} = X_q + \frac{P_h^q \lambda_q \overline{X_q^2} + [(1-\phi)\lambda_u + \phi\frac{\lambda_q}{R_g}]\overline{X_u^2}}{2(1-\rho_h)} . \tag{35}$$

Load from HDB to RDB$_u$: The downlink traffic load from HDB to RDB$_u$ is effected by the forward messages including H-forward, R-forward2 and N-forward2, and the response messages including U-fre-mis-response and U-infre-response. These forward messages are described in Equation (27). U-fre-mis-response and U-infre-response messages return all the data queried by U-forward, which is described in Equation (30). Thus, the probability Q_{hu}^q of query rate can be derived as follows.

$$\begin{aligned} Q_{hu}^q &= (1-\alpha-k\beta-s\gamma)\frac{\phi}{N_q} + k\beta\frac{\phi}{N_q}\varphi + s\gamma\frac{\phi}{N_q} \\ &\quad + \alpha(1-\phi)\left(\frac{N_q-1}{N_q} + \frac{\varphi}{N_q}\right) \\ &= (1-\alpha-k\beta)\frac{\phi}{N_q} + k\beta\frac{\phi}{N_q}\varphi + \alpha(1-\phi) \\ &\quad \left(\frac{N_q-1}{N_q} + \frac{\varphi}{N_q}\right) . \end{aligned} \tag{36}$$

Other types of messages occurred on the link are the download messages and the propagation messages. Combining these messages, the load β_{hu} is given using

$$\beta_{hu} = Q_{hu}^q \lambda_q Y_q + \phi\frac{\lambda_q}{R_g} Y_d + (1-\phi)\lambda_u F_u Y_u , \tag{37}$$

and the average query response time R_{hu}^q is

$$R_{hu}^q = Y_q + \frac{Q_{hu}^q \lambda_q \overline{Y_q^2} + \phi\frac{\lambda_q}{R_g}\overline{Y_d^2} + (1-\phi)\lambda_u F_u \overline{Y_u^2}}{2(1-\beta_{hu})} . \tag{38}$$

Load on RDB$_r$: The server load of RDB$_r$ includes the R-query and all the propagation from the primary site (RDB$_u$ or HDB). Thus, the probability P_r^q of query rate on the database is

$$P_r^q = k\beta , \tag{39}$$

and the server load ρ_r on RDB$_r$ can be obtained

$$\begin{aligned} \rho_r &= P_r^q \lambda_q X_q + [\phi\lambda_u F_u + (1-\phi)\lambda_u F_u] X_u \\ &= P_r^q \lambda_q X_q + \lambda_u F_u X_u . \end{aligned} \tag{40}$$

The average query response time T_{rq} is given

$$T_{rq} = X_q + \frac{P_r^q \lambda_q \overline{X_q^2} + \lambda_u F_u \overline{X_u^2}}{2(1-\rho_r)} . \tag{41}$$

Load on RDB$_{nr}$: The load on RDB$_{nr}$ is caused only by the N-query message. The probability P_n^q of query rate and the load ρ_n are obtained

$$P_n^q = s\gamma \,, \tag{42}$$

$$\rho_n = P_n^q \lambda_q X_q \,. \tag{43}$$

The average query response time T_{nq} is

$$T_{nq} = X_q + \frac{P_n^q \lambda_q \overline{X_q^2}}{2(1-\rho_n)} \,. \tag{44}$$

Load from RDB$_r$ to HDB: The traffic load from RDB$_r$ to HDB is caused by the R-forward message described in Equation (33). Therefore, the probability Q_{rh}^q of query rate and the load β_{rh} on the link is easily derived in the following.

$$Q_{rh}^q = k\beta \left(\frac{N_q - 1}{N_q} + \frac{\varphi}{N_q} \right) \tag{45}$$

$$\beta_{rh} = Q_{rh}^q \lambda_q Y_q \,. \tag{46}$$

The average query response time R_{rh}^q is given

$$R_{rh}^q = Y_q + \frac{Q_{rh}^q \lambda_q \overline{Y_q^2}}{2(1-\beta_{rh})} \,. \tag{47}$$

Load from RDB$_{nr}$ to HDB: The uplink traffic load from RDB$_{nr}$ to HDB can be derived directly by the N-forward message. Therefore, the load β_{nh} is calculated as follows.

$$Q_{nh}^q = s\gamma \,, \tag{48}$$

$$\beta_{nh} = Q_{nh}^q \lambda_q Y_q \,. \tag{49}$$

The average query response time is given

$$R_{nh}^q = Y_q + \frac{Q_{nh}^q \lambda_q \overline{Y_q^2}}{2(1-\beta_{nh})} \,. \tag{50}$$

Load from HDB to RDB$_r$: The downlink traffic from HDB to RDB$_r$ includes the query responses marked as R-infre-response and R-fre-mis-response. R-infre-response returns the infrequently updated data no matter where the callee is located, whereas R-fre-mis-response returns the missed frequently updated data when the calee is at home. Thus, the probability Q_{hr}^q of query rate on the link is derived as follows.

$$Q_{hr}^q = k\beta \frac{N_q - 1}{N_q} + k\beta(1-\phi)\frac{\varphi}{N_q} \,. \tag{51}$$

The traffic also consists of the propagation messages generated from the callees in the home region. Then the load can be obtained

$$\beta_{hr} = Q_{hr}^{q}\lambda_q Y_q + (1-\phi)\lambda_u F_u Y_u \,, \tag{52}$$

and the average query response time R_{hr}^{q} is given

$$R_{hr}^{q} = Y_q + \frac{Q_{hr}^{q}\lambda_q \overline{Y_q^2} + (1-\phi)\lambda_u F_u \overline{Y_u^2}}{2(1-\beta_{hr})} \,. \tag{53}$$

Load from HDB to RDB$_{nr}$: The downlink traffic load from HDB to RDB$_{nr}$ includes the query responses marked as N-response and N-infre-response. The two messages carry all of the required data when the callee is at home, and carry the infrequently updated data when the callee is at the foreign region. Thus, the load β_{hn} is derived in the following.

$$Q_{hn}^{q} = s\gamma(1-\phi) + s\gamma\phi\frac{N_q - 1}{N_q} \,, \tag{54}$$

$$\beta_{hn} = Q_{hn}^{q}\lambda_q Y_q \,. \tag{55}$$

The average query response time R_{hn}^{q} is given

$$R_{hn}^{q} = Y_q + \frac{Q_{hn}^{q}\lambda_q \overline{Y_q^2}}{2(1-\beta_{hn})} \,. \tag{56}$$

Load from RDB$_u$ to RDB$_r$: The traffic load from RDB$_u$ to RDB$_r$ includes the R-fre-mis-response message, and thus the probability for the query rate is

$$Q_{ur}^{q} = k\beta\phi\frac{\varphi}{N_q} \,. \tag{57}$$

The load also includes the propagation messages sent from the foreign region with RDB$_u$. Then the load β_{ur} is obtained

$$\beta_{ur} = Q_{ur}^{q}\lambda_q Y_q + \phi\lambda_u F_u Y_u \,, \tag{58}$$

and the average query response time is given

$$R_{ur}^{q} = Y_q + \frac{Q_{ur}^{q}\lambda_q \overline{Y_q^2} + \phi\lambda_u F_u \overline{Y_u^2}}{2(1-\beta_{ur})} \,. \tag{59}$$

Load from RDB$_u$ to RDB$_{nr}$: The traffic load from RDB$_u$ to RDB$_{nr}$ is caused only by the N-fre-response message. The load β_{un} can be derived directly

$$Q_{un}^{q} = s\gamma\frac{\phi}{N_q} \,, \tag{60}$$

$$\beta_{un} = Q_{un}^{q}\lambda_q Y_q \,, \tag{61}$$

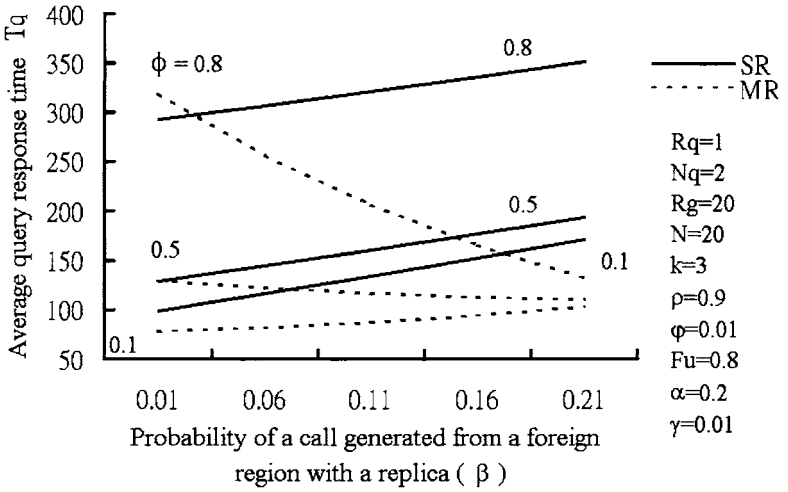


Figure 9. Query response times vs. query rates for various ϕs.

and the average query response time R_{un}^q is given

$$R_{un}^q = Y_q + \frac{Q_{un}^q \lambda_q \overline{Y_q^2}}{2(1-\beta_{un})}. \tag{62}$$

Finally, we combine the above conditional terms to derive the average query response time T_q^{MR}.

$$\begin{aligned} T_q^{\mathrm{MR}} \;=\; & P_u^q T_{uq} + Q_{uh}^q (R_{uh}^q + \tau) + P_h^q T_{hq} \\ & + Q_{hu}^q (R_{hu}^q + \tau) + P_r^q T_{rq} + P_n^q T_{nq} \\ & + Q_{rh}^q (R_{rh}^q + \tau) + Q_{nh}^q (R_{nh}^q + \tau) \\ & + Q_{hr}^q (R_{hr}^q + \tau) + Q_{hn}^q (R_{hn}^q + \tau) \\ & + Q_{ur}^q (R_{ur}^q + \tau) + Q_{un}^q (R_{un}^q + \tau). \end{aligned} \tag{63}$$

4. Numerical Results

In this section, we present the performance results by comparing the average query response time of the two replication strategies. The results are shown from Figures 9 to 14, where the SR curve represents the single-replica model, and the MR curve represents the multiple-replica model. T_q is the average query response time.

We assumed that the processing times for queries and updates are exponentially distributed with means $X_q = 5$ ms and $X_u = 10$ ms in our performance study. The transmission times for queries, updates and downloads on the network links were also assumed to be exponentially distributed with means $Y_q = Y_u = 6.25$ ms and $Y_d = 253$ ms, respectively. The average transmission delay τ on the network links is 50 ms. These assumptions were also used in [11].

Figure 9 reveals the average query response times of the two models with various ϕs. When ϕ is large (more than about 0.414), and β exceeds a threshold value, the average query response time using the MR model is smaller than the time using the SR model. It is also true that when ϕ is small, no matter the value β is. But if ϕ is large and β is less than the threshold,

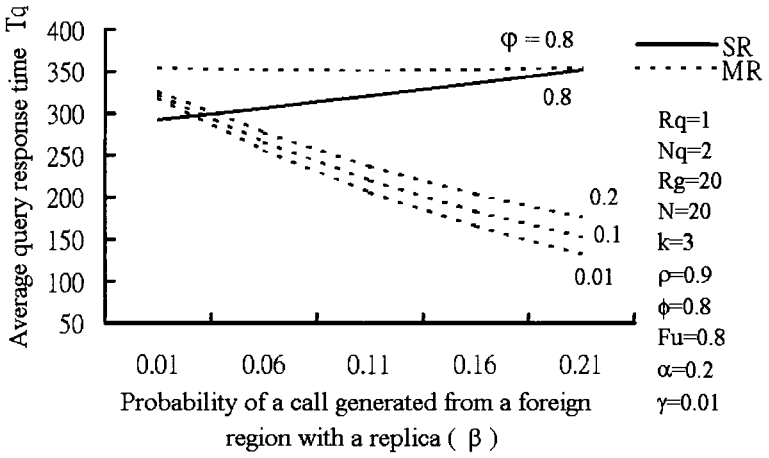


Figure 10. Query response times for various miss rates.

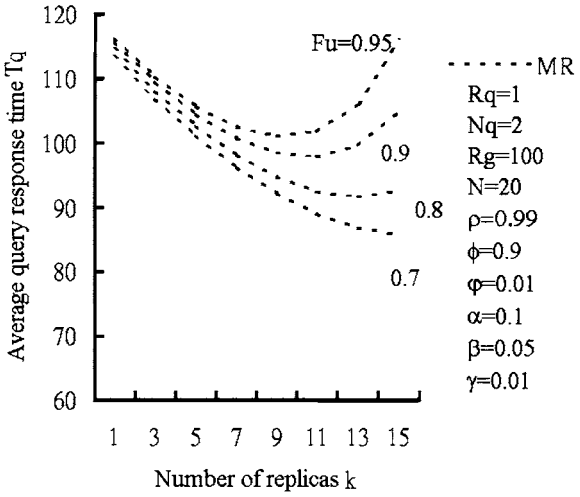


Figure 11. Query response times for different F_us.

the MR model becomes worse. The phenomenon is also consistent with an intuitive result that the MR model performs well in most situations. However, there exists a worst case situation, when a mobile user visits at a foreign region, propagating replicas to other foreign regions with small query rates for the user will enlarge the query time. The worse case may happen possibly because the query rates in foreign regions are always small. Moreover, we find that the query response time of the MR model decreases much faster as β increases slightly if ϕ is large. This result shows that if the query rates of other foreign regions with replicas are large, the benefit of the MR model will be obvious.

Figure 10 illustrates the average query response times with various miss rates for querying in foreign regions with replicas. The miss rate is caused by the propagation delay. As one would expect, the MR model always performs well if the miss rate is small.

Figure 11 describes a comparison of the average query response times with different F_us in the MR model. We find T_q increases as F_u increases. F_u is the propagating rate that can be used to model the mobility of the callees. The result can be explained as follows. When a

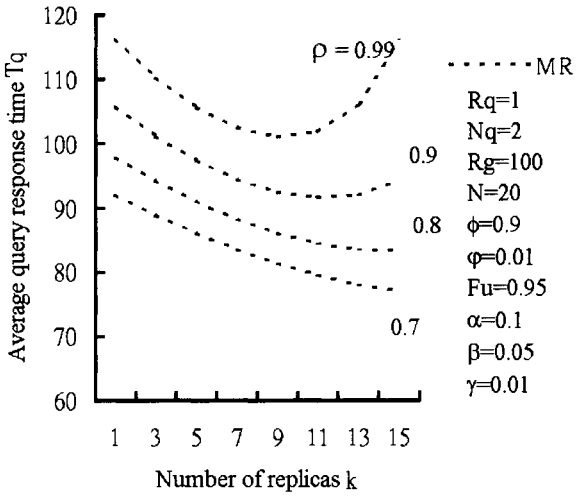


Figure 12. Impacts of traffic loads on the number of replicas.

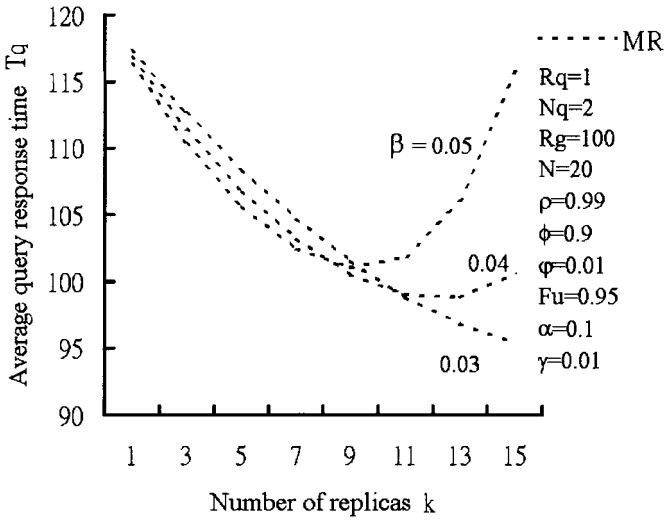


Figure 13. Impacts of query rates on the number of replicas.

mobile host moves fast, the propagating load will increase; thus, the average query response time also increases. Additionally, the figure also shows that if F_u is large, an increase in k may make T_q increase. This phenomenon indicates that k cannot increase unlimitedly. That is, the propagation overhead will be obvious when k or the speed of the mobile hosts increases.

Figure 12 shows the impacts of traffic loads on MR. We find that the query response time increases as the traffic load increases. The traffic load is directly proportional to the propagation rate if the load does not get saturated. Hence the query time is low for low mobility users. Besides, we find that the best k, which minimizes the query time, decreases as the traffic load increases. As one would expect, the higher the traffic load, the lower the best number of replicas should be. Figure 13 depicts the query times as a function of the number of replicas for various βs. It shows that the best k decreases as the query rate β increases. If k is small, an increase in the query rate β makes the query time T_q decrease; however, if k is large, an increase in the query rate β makes T_q increase fast. This is because the propagation benefit

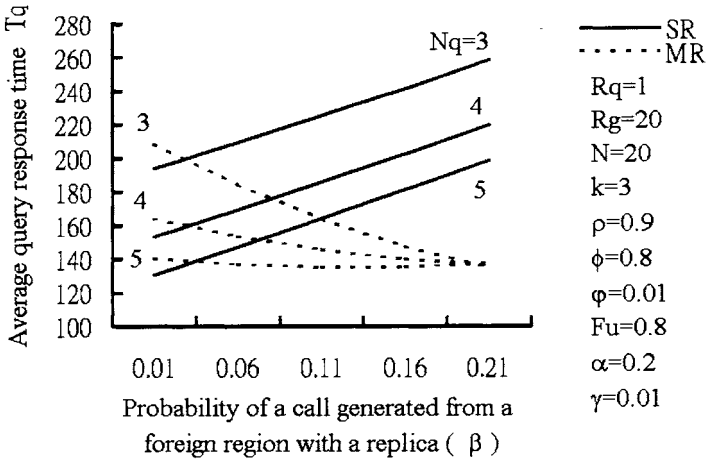


Figure 14. Query response times for different N_qs.

will be reduced by an increase in the traffic load, which is caused by an increase in the query rate β from the replication sites. In other words, the number of replicas is also bounded by the query rates of foreign regions with replicas. From Figures 11, 12 and 13, we can conclude that the number of replicas cannot enlarge arbitrarily, it should be bounded according to the total traffic load ρ, the query rate β, and the propagating rate F_u.

In Figure 14, the average query response times of two models for different N_qs are presented. We find that the query time decreases as N_q increases in both strategies. In other words, the query time is low when the number of queries in a call is large. In SR model, the query time is inversely proportional to N_q. Nevertheless, in MR model if the query rate β is high, the query time will converge no matter how large the number of queries in a call is. We can conclude that the SR model prefers batch processing. In contrast, the MR model prefers interactive processing for the queries. Moreover, $\frac{1}{N_q}$ represents the rate of accessing the frequently updated data; thus, the decrease of N_q will increase the benefit of the partial replication scheme. Figure 14 shows that the query time will decrease quickly when N_q is small in the MR model. This result indicates that the MR model will achieve more benefits from partial replication than the SR model.

5. Adaptive Multiple Replication Protocol

The numerical results show that the MR strategy cannot always perform well. First, there is a worse case situation when ϕ is large and β is small, the average query response time of the MR strategy is larger than that of the SR strategy. Further, the number of replicas cannot be too large to achieve a reasonable query response time. Therefore, we propose an adaptive multiple replication (AMR) protocol based on the MR strategy to solve these problems. The protocol is composed of two components.

The first part of the AMR protocol is to prevent the MR strategy from getting into the worse case situation. Therefore, we designed the first part as follows.

> When ϕ is larger than a threshold value (denoted by T_ϕ) and β is smaller than another threshold value (denoted by T_β), we use the SR strategy to replicate the profile of the

mobile user. In other situations, we use the MR strategy to increase the number of replicas.

First, we need to estimate the threshold T_ϕ. T_ϕ can be derived using the assumption that T_q^{SR} is always larger than T_q^{MR} no matter how large β is. For example, T_ϕ is about 0.414 in Figure 9. Then we derive the threshold T_β that can be calculated from the Equations (34) and (83) when $T_q^{\mathrm{SR}} = T_q^{\mathrm{MR}}$. T_q^{SR} and T_q^{MR} are functions of parameters ϕ, α, γ, etc. If we measure these parameters, we can obtain T_β easily. However, the measurement and computation of these parameters at once are not simple tasks; thus, we suggest that they should be performed periodically. In this manner, the average query response time is equal to the query time of the SR strategy when β is less than T_β, and equal to the query time of the MR strategy when β is larger than T_β.

The second part of the AMR protocol involves avoiding the number of replicas exceeding an optimized value. When the query rate of a foreign region for a callee increases, the database in that region may attempt to replicate the callee's profile to reduce the query response time. However, an increase in replicas makes the propagating load increase; thus, the query time also increases. Additionally, an increase in the query rate will also increase both the traffic load and the query time; thus, it will reduce the replication benefit. These factors will affect the allowable maximum number of replicas. Therefore, we use a 2-phase methodology combining a distributed request scheme and a centralized decision scheme to decide the number of replicas. We describe the second part of the protocol including the two phases as follows.

- *Phase one: The distributed request scheme.* If the call rate of a foreign region for a callee is high, the replicated database of the region will send a request message to RDB_u to require attaching a replica of the user profile. If the request is accepted, the RDB_{nr} of that region becomes an RDB_r.
- *Phase two: The centralized decision scheme.* The replicated database (RDB_u) of the foreign region where a callee visits will estimate the allowable maximum number of replicas (denoted by T_k) for the callee's profile according to the total traffic load. When RDB_u receives a request message, it compares the number of replicas for the profile and the T_k. If the number of replicas is less than T_k, the RDB_u will allow the profile propagating to the requesting region; otherwise, the RDB_u will reject the request. Additionally, when the total traffic load increases, the RDB_u can decide to discard a replica from other foreign regions if the number of replicas is larger than a newly estimated T_k.

6. Conclusions

We modeled the single-replica (SR) and multiple-replica (MR) strategies of mobility databases for PCS networks. The two strategies are based on a partial replication scheme, and a primary copy method is our assumption to maintain the consistency of all replicas. In the SR strategy, only one replica is created along with the movement of the callee, while in the MR strategy, the replica allocation is dynamic according to the query rate of the callers. We compared the two strategies in a global mobility environment in which the query rates of other service areas for a callee may be high.

The numerical results show some important phenomena in our performance study. First, the MR strategy prefers interactive processing for the call queries, whereas the SR strategy prefers batch processing. Therefore, we can easily design an interactive mobile system using the MR strategy. Moreover, we find that the MR strategy will get the benefit of partial replication more

than the SR strategy does. Second, the MR strategy performs well than the SR strategy in most situations except that the probability of a mobile user visiting a foreign region is high and the query rates from other foreign regions are low. The worse case situation for the MR strategy may possibly happen for some callees. The third phenomenon is that the number of replicas in the MR strategy should be compact in order to achieve a reasonable query response time. This is because the propagation overhead will overtake the replication benefit which reduces the traffic load for queries and updates. Consequently, a mechanism is necessary for controlling the number of replicas for a callee. The decision for the number of replicas depends upon the traffic load, the propagating load and the query rate. We proposed an adaptive multiple replication (AMR) protocol to solve the above problems. The AMR protocol measures the distribution of the callee and the query rates of other service areas for the callee in choosing the SR strategy or the MR strategy. This protocol combines a distributed request scheme and a centralized decision scheme to obtain the optimized number of replicas.

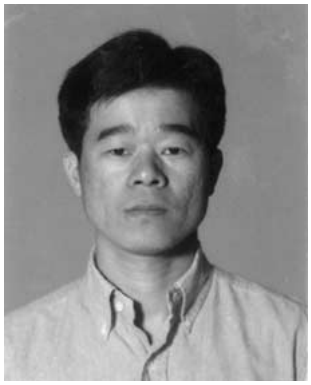
References

1. I.F. Akyildiz, J. McNair, J.S.M. Ho, H. Uzunalioglu and W. Wang, "Mobility Management in Next-generation Wireless Systems", *Proceedings of the IEEE*, Vol. 87, No. 8, pp. 1347–1384, 1999.
2. B.R. Badrinath and T. Imielinski, "Replication and Mobility", in *Proceedings of 2nd IEEE Workshop on Management of Replicated Data*, Monterey, CA, November 1992, pp. 9–12.
3. B.K. Harumoto, M. Tsukamoto, S. Nishio and T. Takine, "On Strategies for Allocating Replicas of Mobile Databases", *IEICE Trans. Inf. and Syst.*, Vol. E81-D, No. 1, pp. 37–46, 1998.
4. B. Gabelgaard, "The (GSM) HLR – Advantages and Challenges", in *Proc. of Third IEEE Int. Conf. on Universal Personal Communications*, Sept. 1994, pp. 335–339.
5. D.J. Goodman, G. Pollini and K.S. Meier-Hellstern, "Network Control for Wireless Communications", *IEEE Commun. Mag.*, pp. 116–124, 1992.
6. Y. Huang, P. Sistla and O. Wolfson, "Data Replication for Mobile Computers", in *Proceedings of the ACM-SIGMOD*, May 1994, pp. 13–24.
7. Y. Huang and O. Wolfson, "Object Allocation in Distributed Databases and Mobile Computers", in *IEEE Proceedings of the 10th International Conference on Data Engineering*, Feb. 1994, pp. 20–29.
8. G.C. Lee, T.-P. Wang and C.-C. Tseng, "Resetting Forwarding Pointers with Delay Propagation Schemes in a Distributed HLR Environment", *IEICE Transactions on Communications*, Vol. E84-B, No. 4, 2001.
9. R. Jain, Y.B. Lin, C. Lo and S. Mohan, "A Caching Strategy to Reduce Network Impacts of PCS", *IEEE JSAC*, Vol. 12, pp. 1434–1444, 1994.
10. L. Kleinrock, *Queueing Systems – Volume I: Theory*, Wiley-Interscience, New York, U.S.A., 1975.
11. K.K. Leung and Y. Levy, "Global Mobility Management by Replicated Databases in Personal Communication Networks", *IEEE JSAC*, Vol. 15, No. 8, pp. 1582–1596, 1997.
12. K.K. Leung, "An Update Algorithm for Replicated Signaling Databases in Wireless and Advanced Intelligent Networks", *IEEE Trans. on Computers*, Vol. 46, No. 3, pp. 362–367, 1997.
13. Y.B. Lin, "Failure Restoration of Mobility Databases for Personal Communication Networks", *Wireless Networks*, Vol. 1, No. 3, 365–372, 1995.
14. Y.B. Lin and I. Chlamtac, *Wireless and Mobile Network Architectures*, John Wiley & Sons, 2000.
15. Y.B. Lin, "Overflow Control for Cellular Mobility Databases", *IEEE Transactions on Vehicular Technology*, Vol. 49, No. 2, pp. 520–530, 2000.
16. Y.B. Lin, "Eliminating Overflow for Large-Scale Mobility Databases in Cellular Telephone Networks", *IEEE Transactions on Computers*, Vol. 50, No. 4, pp. 356–370, 2001.
17. Mohan, S. and Jain, R., "Two User Lacation Strategies for Personal Communications Services", *IEEE Personal Comunications*, Vol. 1, No. 1, pp. 42–50, 1994.
18. O. Wolfson, S. Jajodia and Y. Huang, "An Adaptive Data Replication Algorithm", *ACM Transactions on Database Systems*, Vol. 22, No. 2, pp. 255–314, 1997.

19. A. Zaslavsky, M. Faiz, B. Srinvasan, A. Rasheed and S. Lai, "Primary Copy Method and Its Modifications for Database Replication in Distributed Mobile Computing Environment", in *IEEE 15th Symposium On Reliable Distributed Systems*, 1996, pp. 178–187.

Gwo-Chuan Lee is currently a Ph.D. candidate in the Department of Computer Science and Information Engineering at National Chiao-Tung University, Hsin-Chu, Taiwan. He received his B.Sc. degree in computer science and information engineering from National Chiao-Tung University, Hsin-Chu, Taiwan, in 1988; M.Sc. degree in computer science from National Taiwan University, Taipei, Taiwan, in 1990. His research interests include mobile computing, wireless Internet, and computer networks.

Tsan-Pin Wang is currently an associate professor in the Department of Computer Science and Information Management at Providence University, Shalu, Taiwan. He received the B.Sc. degree in applied mathematics; M.Sc. and Ph.D. in computer science and information engineering, from National Chiao Tung University, Taiwan, ROC, in 1990, 1992, and 1997, respectively. From 1992 to 1993, he was a system engineer in the R&D Division of Taiwan NEC Ltd. From 1997 to 2001, he was an assistant professor at Providence University. His research interests include mobile computing, mobile communications, and computer networks.



Chien-Chao Tseng is currently a professor in the Department of Computer Science and Information Engineering at National Chiao-Tung University, Hsin-Chu, Taiwan. He received his B.Sc. degree in industrial engineering from National Tsing-Hua University, Hsin-Chu, Taiwan, in 1981; M.Sc. and Ph.D. degrees in computer science from the Southern Methodist University, Dallas, Texas, U.S.A., in 1986 and 1989, respectively. His research interests include mobile computing, and wireless Internet.