Processing Distributed Mobile Queries with Interleaved Remote Mobile Joins

Chang-Hung Lee and Ming-Syan Chen, *Senior Member*, *IEEE*

Abstract—The query processing in a mobile computing environment involves join processing among different sites which include static servers and mobile computers. Because of the presence of asymmetric features in a mobile computing environment, the conventional query processing for a distributed database cannot be directly applied to a mobile computing system. In this paper, we first explore some unique features of a mobile environment and then, in light of these features, devise query processing methods for both join and query processing. Remote mobile joins are said to be effectual if they are, when being interleaved into a join sequence, able to reduce the amount of data transmission cost required for distributed mobile query processing. Since mobile relations are employed as reducers in our proposed query processing cost model, more mobile joins in the query processing lead to less data transmitted through the network. With proper scheduling, interleaving effectual remote mobile joins into a query scheduling can significantly reduce the total amount of data transmission among different sites. A simulator is developed to evaluate the performance of algorithms devised. Our results show that the approach of interleaving the processing of distributed mobile queries with effectual remote mobile joins is not only efficient, but also effective in reducing the total amount of data transmission cost required to process distributed mobile queries.

Index Terms—Distributed query processing, mobile computing, remote mobile joins, query scheduling.

1 Introduction

Recently, the need for accessing information from anywhere at any time has been a driving force for a variety of portable devices and mobile applications. As the number of mobile applications increases rapidly, there has been a growing demand for the use of distributed database architectures for various applications [10], [25], [38], [43], [57], [67]. Applications such as stock activities, traffic reports, and weather forecasts have become increasingly popular [58]. Various wireless data networking technologies, including IS-136 [54], CDMA2000 [37], Wireless Application Protocol (WAP) [59], and third generation mobile phone [6], have been developed. Among others, with the rapid advances in palm computer technologies, a mobile computer is envisioned to be equipped with more powerful capabilities, including the storage of a small database and the capacity of data processing [44]. Consequently, the query processing in a mobile computing system which involves fixed hosts and several mobile computers has emerged as an issue of growing importance.

Generally, there are three primary types of wireless mobile networks [26], [36], [42], [52], [56]. The first one is known as an infrastructured network, i.e., a network with fixed and wired base stations. These base stations act as the gateways between high-speed wired networks and low-bandwidth wireless networks. A mobile device within these networks connects to the nearest base station with a wireless connection when the device is inside the service area of the base station. A handoff occurs when a mobile device moves from one service area to another. Examples of this type of networks include GPRS [11] and 3G [1]. The second type of networks is known as infrastructureless networks, which is also known as mobile ad hoc networks (referred to as MANETs). MANETs do not have fixed nodes and all nodes are capable of movement and can be connected dynamically. In addition to end hosts, mobile nodes of these networks also function as routers which discover and maintain routes, and forward packets to other nodes. A number of standards have been developed to support MANETs, including IEEE 802.11 [36], HomeRF [42], and Bluetooth [26]. Example applications of MANETs include digital battlefield communications, personal area networks [52], [56], and sensor networks [34]. Third, several hybrid network architectures [56], [66], e.g., IEEE 802.16, have been proposed to integrate heterogeneous networks to provide high availability and high bandwidth mobile computing environments.

On the other hand, a considerable amount of research effort on mobile database issues has been elaborated upon in recent years. These studies cover a broad spectrum of topics including:

1. data replication in infrastructured networks [31], [51] and MANETs [28], [62];
2. data broadcasting [2], [3] and dissemination strategy [39], [55];
3. caching design [35], [47], [63];
4. mobility management [4], [33], [46], [56];
5. location-dependent data query processing [50], [68] and caching [48]; and
6. transaction management [19], [21].

Conventionally, as pointed out in [64], the processing of a distributed query is composed of the following three

• *The authors are with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, ROC. E-mail: chlee@arbor.ee.ntu.edu.tw, mschen@cc.ee.ntu.edu.tw.*

Manuscript received 15 July 2001; revised 15 May 2002; accepted 20 May 2002.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 116590.

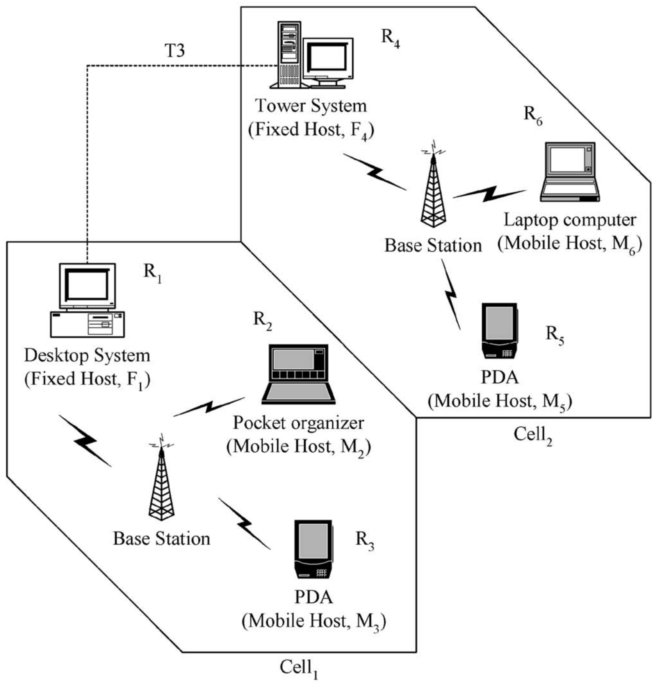


Fig. 1. An illustrative mobile computing environment with mobile hosts and fixed hosts.

phases: 1) *local processing phase,* 2) *reduction phase,* and 3) *final processing phase*. Significant research efforts have been focused on the problem of reducing the amount of data transmission required for phases 2) and 3) of distributed query processing [9], [13], [14], [22]. The semijoin and join operations have received a considerable amount of attention and have been extensively studied in the literature [15], [38], [41], [49], [61]. In addition, relevant works on query processing include client-server-based query processing [22], [32], mobile computing query processing [30], [32], [53], [60], query processing on the Web [25], and network-based ones [10], [38], [43], [57], [65], [67], to name a few. However, as will be explained later, without considering network characteristics and asymmetric computing capability, the conventional approach for distributed query processing cannot be directly applied to the mobile computing environment nowadays.

Consider an inventory application, for example, where a salesperson uses, for his/her work, a mobile computer device in which a fragment of database contains the information of his/her customer records. In Fig. 1, a portable computer, such as M_2, is hand-carried by this salesperson and is located at $Cell_1$ while F_1 and M_3 are also located at the same region $Cell_1$. On the other hand, F_4, M_5, and M_6 with different data sets are allocated at $Cell_2$. F_1 and F_4 represent fixed hosts and M_2, M_3, M_5, and M_6 are mobile hosts. Note that, depending on the corresponding coherency control mechanism employed, the data copy in the fixed host server could be obsolete [16]. Since the most up-to-date data is stored in the mobile computers, a query generated by a salesperson could be a sequence of joins to be performed across the relations residing in the server and several mobile computers, resulting in a very different execution scenario from the one for query processing in a traditional distributed system. Furthermore, mobile computers use small batteries for their operations without directly connecting to any power source and the bandwidth of wireless communication is, in general, limited. As a result, how to conserve the computing capability and communication bandwidth of a mobile unit while allowing mobile users of the ability to access information from anywhere at any time has become an important design issue in a mobile system [7], [18], [32].

Consequently, we shall explore in this paper three important asymmetric features of a mobile computing system and, in light of these features, develop corresponding query processing schemes for mobile computing systems. The first asymmetric feature is on the computing capability between fixed hosts and mobile hosts [45]. Usually, mobile computers have limited resources for their computing operations and the server is certainly much more powerful than a portable computing device. Note that, in traditional distributed query processing, the sites involved in a query processing are usually assumed to have the same level of processing capability, which is, however, not valid in a mobile environment. The second asymmetric feature is on the transmission bandwidth between fixed hosts and mobile hosts. Clearly, the transmitting capability among mobile hosts is smaller than that among fixed hosts since the transmission bandwidth of fixed hosts is, in general, much larger than that of mobile hosts. The third asymmetric feature is on the transmission cost coefficients among local hosts and remote hosts. The transmission cost required for transmitting one unit of data among local hosts is much smaller than the corresponding cost required among remote hosts. These features distinguish the query processing in a mobile environment from the one in a traditional distributed system and, hence, have to be considered when the costs of the corresponding operations are modeled.

Due to the presence of asymmetric features in a mobile computing environment, the conventional query processing for a distributed database cannot be directly applied to a mobile computing system. In view of this, we shall explicitly devise query processing methods for both joins and query processing. Remote mobile joins are said to be *effectual* if they are, when being interleaved into a join sequence, able to reduce the amount of data transmission cost required for distributed mobile query processing. Since mobile relations are employed as reducers in our proposed query processing cost model, more mobile joins in the query processing lead to less data transmitted through the network. Instead of processing queries by performing the minimum-cost joins sequentially, as with conventional methodologies [9], [13], [14], [22], interleaving effectual remote mobile joins into a query scheduling can significantly reduce the total amount of data transmission among different cells. It can be verified that the total data transmission cost of the processing in a distributed mobile query can be reduced by the algorithms devised in this paper by using effectual remote joins. Performance studies on the sensitivity of various important parameters, including the number of mobile relations in a cell architecture, the density of query, the number of relation tuples, the amount of an attribute cardinality, and network transmission coefficients in a mobile computing model, are also conducted. It is shown by our simulation results that, by exploiting three asymmetric features, the effectual remote mobile joins proposed are very powerful in reducing the amount of data transmission cost incurred and can lead to the design of an efficient and effective query processing procedure for a mobile computing environment.

We mention in passing that, without dealing with query processing, the authors in [5] studied the issues of optimization between energy consumption and server workload in a mobile environment. Several research efforts have elaborated upon developing a location dependent query mechanism [20], [53], [60]. The authors in [29] presented the concept of queries with location constraints, i.e., constraints which involve location of mobile users. In [53], the authors proposed a spatial-temporal data model for the query of moving data in mobile environments. The position update policy is addressed in [60]. Without exploiting the network characteristics and asymmetric features of computing capability, the attention of prior studies was mainly paid to the query mechanisms with location constraints and query processing in traditional distributed databases [12], [13], [22], [23], but not to the specific cost model and the query processing for a mobile computing system explored in this paper. As mentioned above, due to these asymmetric features of a mobile computing system, the cost model and the design of query processing schemes are different from those in a traditional distributed database. In this paper, we not only formulate a new cost model which takes these asymmetric features into consideration, but also explicitly investigate the join methods and develop the corresponding query processing schemes. These features distinguish this paper from others.

This rest of this paper is organized as follows: Preliminaries are given in Section 2. Two join schemes and query processing schemes for multijoin queries are proposed in Section 3. Performance studies are conducted in Section 4. This paper concludes with Section 5.

2 PRELIMINARIES

As in most previous works in distributed databases [64], we assume a query is in the form of conjunctions of *equi-join* predicates and all attributes are renamed in such a way that two join attributes have the same attribute name if and only if they have a join predicate between them. $|K|$ is used to denote the cardinality of a set K. For notational simplicity, the width of an attribute A and that of a tuple in R_i are assumed to be one unit. The size of the total amount of data in R_i can then be denoted by $|R_i|$. $|A|$ is used to denote the cardinality of the domain of an attribute A. Define the selectivity $\rho_{i,a}$ of attribute A in R_i as $\frac{|R_i(A)|}{|A|}$, where $R_i(\mathrm{A})$ is the set of distinct values for the attribute A in R_i. $R_i - A \rightarrow R_j$ means a semijoin from R_i to R_j on attribute A. After the semijoin $R_i - A \rightarrow R_j$, the cardinality of R_j can be estimated as $|R_j|\rho_{i,a}$. To simplify the notation, $R_i \rightarrow R_j$ is used to mean a semijoin from R_i to R_j in the case that the semijoin attribute does not have to be specified. Also, the notation $R_i \Rightarrow R_j$ is used to mean that R_i is sent to the site of R_j and a join operation is performed with R_j there. We use R_i' to denote the resulting relation after joins/semijoins are applied to an original relation R_i.

TABLE 1
An Example for Semijoin Operation

R_1	
A	B
a_1	b_1
a_2	b_1
a_2	b_3
a_2	b_4
a_3	b_3

R_2	
B	C
b_1	c_1
b_1	c_2
b_2	c_1
b_5	c_2
b_6	c_4
b_7	c_2
b_8	c_3

R_2'	
A	B
b_1	c_1
b_1	c_2

Consider the relations in Table 1. Suppose $|A| = 5$, $|B| = 10$, and the width of each attribute is one unit. In addition, we have $\rho_{1,b} = 0.3$ and $\rho_{2,b} = 0.6$. Also, $|R_1| = 5$, $|R_2| = 7$, $R_1(B) = \{b_1, b_3, b_4\}$, and $R_1.B = R_2.B$.

Conventionally, a function of the form $C(X) = c_0 + c_1 \cdot X$ is used to characterize communication cost, where X is the amount of data shipped from one site to another, c_1 is the communication cost per data unit [8], and the start-up connection cost c_0 is usually less significant. However, if the network topology is taken into consideration, the notion of identifying a *profitable* semijoin that prior work relied upon [64] is incomplete and, in fact, might be misleading in some cases. Explicitly, c_1 is not a constant when network characteristics are considered and its value is dependent upon the network topology.

In general, it is very difficult to determine a network cost model since the practical transmission bandwidth for a network traffic is in fact time-dependent. Hence, statistical values of transmission bandwidth of the network are employed to provide a proper solution. Note that, even though the temporal traffic is not a constant value and almost unpredictable in the present network, utilizing a statistical average to optimize the scheduling of query processing in a mobile environment will limit the error of scheduling to an acceptable range. Nevertheless, due to the fast development of QoS techniques in the next generation mobile units, IEEE 802.11a/b and IEEE 802.16, the network traffic is envisioned to become more stable in coming years [40]. As a consequence, the transmission coefficient $c_{m\rightarrow n}$ is used to serve as the statical average value in each network edge. We define an *effectual semijoin* as follows.

Definition 1 (Effectual Semijoin). *A semijoin, $R_1(S_1) - B \rightarrow R_2(S_2)$, is called* effectual *if its cost of sending $R_1(B)$, i.e., $c_{1\rightarrow 2}(|R_1(B)| = |B|\rho_{1,b})$, is smaller than its benefit, i.e., $c_{2\rightarrow 1}(|R_2| - |R_2|\rho_{1,b} = |R_2|(1-\rho_{1,b}))$, where R_1 and R_2 are located at sites S_1 and S_2, respectively, and $|R_2|$ and $|R_2|\rho_{1,b}$ represent, respectively, the sizes of R_2 before and after the semijoin. Thus, $|R_1(B)|c_{1\rightarrow 2}$ is used to denote the cost of a semijoin $R_1 - B \rightarrow R_2$.*

Note that $|R_1(B)| = 1 \times 3$ and

$$|R_2|(1 - \rho_{1,b}) = 1 \times 7 \times 0.7 = 4.9,$$

TABLE 2
Description of Symbols for the Cost Model in a Mobile Computing System

Symbol	Description
c^L_{FF}	Local transmission cost coefficient among fixed hosts
c^L_{MF}	Local transmission cost coefficient between mobile hosts and fixed hosts
c^L_{MM}	Local transmission cost coefficient among mobile hosts
c^R_{FF}	Remote transmission cost coefficient among fixed hosts
c^R_{MF}	Remote transmission cost coefficient between mobile hosts and fixed hosts
c^R_{MM}	Remote transmission cost coefficient among mobile hosts
r^{RL}_{FF}	Transmission cost ratio between remote fixed hosts and local fixed hosts
r^{RL}_{MM}	Transmission cost ratio between remote mobile hosts and local mobile hosts
r^L_{MF}	Transmission cost ratio between local mobile hosts and local fixed hosts
r^R_{MF}	Transmission cost ratio between remote mobile hosts and remote fixed hosts

as illustrated in the example above. If $R_1 - B \rightarrow R_2$ is effectual, then $c_{1\rightarrow 2}$ should be smaller than $\frac{4.9}{3} \times c_{2\rightarrow 1}$. Otherwise, if

$$c_{2\rightarrow 1}(|R_2(B)|) < c_{1\rightarrow 2}(|R_1|(1 - \rho_{2,b})),$$

then $R_2(S_2) - B \rightarrow R_1(S_1)$ is an effectual semijoin. Different transmission paths with different transmission coefficients will lead to different transmission costs though the amount of data transmission is the same. Thus, the scheduling of query processing will be significantly influenced by the transmission coefficients among network characteristics. In general, the path with higher bandwidth and lower communication costs, such as the local communication with fixed hosts, is associated with a lower transmission cost coefficient. The remote mobile communication, in contrast, is certainly more expensive than the local one.

Furthermore, we assume that the values of attributes are uniformly distributed over all tuples in a relation and that the values of one attribute are independent from each other. The cardinalities of the resulting relations from join operations can thus be estimated according to the formula in [14]. Note that this assumption is not essential, but will simplify our presentation. In the presence of certain database characteristics and data skew, we only have to modify the formula for estimating the cardinalities of resulting relations from joins accordingly [24], [27].

2.1 Cost Model

Consequently, we derive a cost model which considers these three asymmetric features of a mobile computing system. Our model consists of two distinct sets of entities: mobile hosts and fixed hosts [7]. Furthermore, we use *local* and *remote* to indicate two different communication modes. Local communication means that the transmission is among hosts in the same cell, whereas remote communication means that the transmission is among different cells. For ease of our discussion, symbols used are shown in Table 2. c^L_{FF} denotes local transmission cost coefficient among fixed hosts and we assume c^L_{FF} is a basic coefficient and its value is given as one unit for transmitting one unit of data among local fixed hosts. The local transmission cost coefficient among mobile hosts is denoted by c^L_{MM}. Analogously, we use c^L_{MF} to indicate the local transmission cost coefficient between mobile hosts and fixed hosts. For remote communication, we have three parameters to model the transmission costs among mobile and fixed hosts, i.e., c^R_{FF}, c^R_{MM}, and c^R_{MF}. In addition, several transmission cost ratios are used to represent the relationship among these transmission coefficients, i.e., $r^{RL}_{FF} = \frac{c^R_{FF}}{c^L_{FF}}$, $r^{RL}_{MM} = \frac{c^R_{MM}}{c^L_{MM}}$, $r^L_{MF} = \frac{c^L_{MM}}{c^L_{FF}}$, and $r^R_{MF} = \frac{c^R_{MM}}{c^R_{FF}}$. Note that the processing time in each computing host may vary and its system dependent optimization is a challenging issue itself [9] and is beyond the scope of this paper.

3 Query Processing in a Mobile Computing System

Join processing in a mobile computing system is discussed in Section 3.1. The query processing scheme with a divide-and-conquer technique based on the cell architecture (to be referred to as scheme QP_C) is discussed in Section 3.2. The scheme that is devised with effectual remote mobile joins (to be referred to as scheme QP_R) is described in Section 3.3. Moreover, the solution searching space is analyzed in Section 3.4.

3.1 Join Processing in a Mobile Computing System

We now derive the solution procedure for minimizing the cost of join methods in a mobile computing system. Consider the scenario of join processing in Fig. 2, where the fixed host F_1 has relation R_1 and the fixed hosts F_2 has relation R_2. R_3 is located at the mobile host M_3. Suppose that the mobile user M_3 submits a query that performs a join operation of R_1, R_2, and R_3 on their common attribute A and B, $R_1.A = R_3.A$ and $R_2.B = R_3.B$, with the corresponding selectivity factors ρ_A and ρ_B, respectively. We will select F_1 as the location for storing the join result. With this given model, we shall examine two join methods. To simplify our presentation, $TC(J)$ is used to represent the data transmission cost of the join method J.

In what follows, we examine a join sequence which preforms the joins based on cell architecture with a divide-and-conquer technique in Section 3.1.1. Section 3.1.2 describes the effectual remote mobile join method. Analysis of these join methods is given in Section 3.1.3.

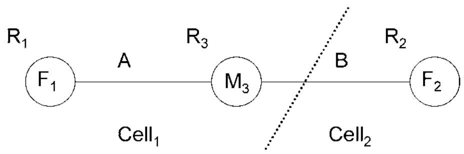


Fig. 2. An example scenario for join processing.

3.1.1 Processing Joins with Divide-and-Conquer (denoted by J_C)

Consider a query in Fig. 2 as an example. Traditionally, the query processing is performed based on the minimum-cost join in a forward-scheduling manner. Since the transmission cost among local communication paths is more inexpensive than that among remote communication paths, the query will be naturally divided into two separated subqueries based on the cell architecture and processed independently. This is how the notion of divide and conquer comes out. One is the subquery belonging to the communication cell Cell_1 and the other is belonging to Cell_2. After the join results of each subquery are merged into a fixed host, the residue relations can be processed with the new query. Such a processing scenario is shown in Fig. 3. Note that, with the forward scheduling method, the join processing, merging the partial database R_3 on M_3 to R_1 of F_1, will be the most efficient processing. As a result, a cost of $TC(R_3 \Rightarrow R_1) = c^L_{MF} * |R_3|$ is incurred and a new relation R'_1 is generated in F_1, where $|R'_1| = \frac{|R_1||R_3|}{|A|}$.

After all of the local join sequences in each subquery are finished, two separated subqueries are merged to be a new query, i.e., $R'_1.B = R_2.B$ between F_1 and F_2. Since the amount of tuples storing in the fixed host database is much larger than the number of an attribute cardinality, i.e., both $|R_1|$ and $|R_2|$ are much larger than $|B|$ in a mobile environment, an effectual semijoin occurs between these two residual relations in fixed hosts. Because of $|R_1| \gg |B|$, $\rho_{1,B}$ is assumed to be unchanged after the join processing. In other words, a semijoin $R'_1 - B \rightarrow R_2$ and a join $R'_2 \Rightarrow R'_1$ will be processed in this merged query, which leads to a cost of

$$TC(R'_1 - B \rightarrow R_2) + TC(R'_2 \Rightarrow R'_1) \\ = c^R_{FF} * |R'_1(B)| + c^R_{FF} * \rho_{1,B} * |R_2|.$$

Then, the corresponding costs is summarized as follows:

$$TC(J_C) = c^L_{MF} * |R_3| + c^R_{FF} * \rho_{1,B} * (|B| + |R_2|).$$

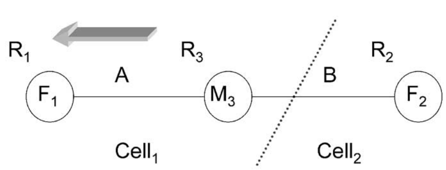


Fig. 3. An illustrative scenario of the join processing with divide-and-conquer.

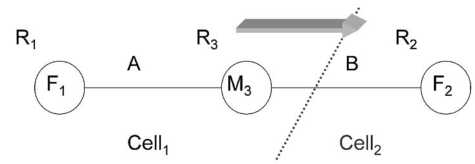


Fig. 4. An illustrative scenario of the join processing with remote mobile joins.

3.1.2 Processing Joins with Remote Mobile Join (denoted by J_R)

Next, consider the case of join processing with remote mobile joins. Instead of merging the join operation between F_1 and M_3, R_3 is merged to R_2, followed by the join processing between F_1 and F_2. Even though the remote transmission cost coefficient between mobile hosts and fixed hosts, i.e., c^R_{MF}, is much larger than the local transmission cost between mobile hosts and fixed hosts, i.e., c^L_{MF}, it can be still profitable with a high reduction ratio leading to the use of an effectual remote mobile join. As shown in the execution scenario in Fig. 4, the total transmission cost will be $TC(R_3 \Rightarrow R_2) + TC(R_1 - B \rightarrow R'_2) + TC(R''_2 \Rightarrow R_1)$, where $TC(R_3 \Rightarrow R_2) = c^R_{MF} * |R_3|$, $TC(R_1 - B \rightarrow R'_2) + TC(R''_2 \Rightarrow R_1) = c^R_{FF} * (\rho_{1,B} * |B| + |R''_2|)$ and $|R''_2| = \rho_{1,B} * \frac{|R_1||R_3|}{|A|}$. Consequently, we have corresponding costs below.

$$TC(J_R) = c^R_{MF} * |R_3| + c^R_{FF} * \rho_{1,B} * \left(|B| + \frac{|R_1||R_3|}{|A|}\right).$$

3.1.3 Analysis of Join Processing

To examine the amount of data transmission cost incurred by $\mathbf{J}_C$ and $\mathbf{J}_R$. Specifically, the criterion of identifying an effectual remote mobile join to reduce the amount of data transmission cost is derived. In practice, the local transmission cost coefficient between local mobile hosts and local fixed hosts c^L_{MF} is very close to the value among local mobile hosts c^L_{MM}. To simplify our discussion, $c^L_{MF} = c^L_{MM}$ and $c^R_{MF} = c^R_{MM}$ are assumed in this paper. Note that such as assumption is made for ease of discussion and is not essential for the use of remote joins we propose in this paper. For better readability, proofs of lemmas and theorems are given in the Appendix for interested readers.

Lemma 1. $c^R_{FF} = \frac{r^{RL}_{MM}}{r^R_{MF}} * c^L_{MF}$.

Lemma 2. *With* $\frac{r^R_{MF} \times (r^{RL}_{MM} - 1)}{r^{RL}_{MM}} < \rho_{1,B} * (\frac{|R_2|}{|R_3|} - \frac{|R_1|}{|A|})$, *the amount of data transmission cost incurred by method* J_R *is smaller than that by method* J_C, *i.e.,* $TC(J_R) < TC(J_C)$, *where* R_2 *is a remote fixed host and* R_3 *is an example of the local mobile host.*

With Lemma 2, an effectual remote mobile join is defined as follows:

Definition 2. *A remote mobile join is called* effectual *if and only if* $TC(J_R)$ *is smaller than* $TC(J_C)$.

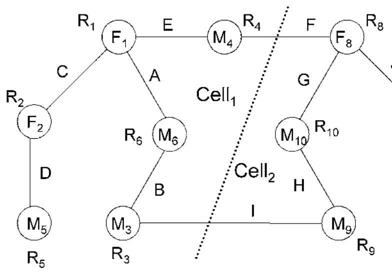


Fig. 5. Division of a query.

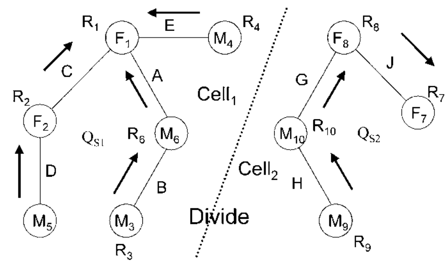


Fig. 6. Query processing with QP_C methodology.

With Definition 2, we can derive the following theorem. According to Theorem 1, effectual remote mobile joins can be interleaved into the query scheduling to reduce the data transmission cost of multijoin processing.

Theorem 1. *A remote mobile join is* effectual *if and only if* $\frac{r^R_{MF}\times(r^{RL}_{MM}-1)}{r^{RL}_{MM}} < \rho_{1,B} * (\frac{|R_2|}{|R_3|} - \frac{|R_1|}{|A|})$, *where* $|R_3|$ *is the size of relations in a remote fixed host,* $\rho_{1,B}$ *denotes the selectivity of a relation in the local fixed host, and* $|R_2|$ *is the size of a relation in the local mobile host.*

It can be verified that, by judiciously applying effectual remote mobile joins, method J_R can reduce the amount of data transmission cost as a whole. As can been seen later, Theorem 1 derived above can be employed to determine the threshold for whether method J_R should be utilized.

3.2 Query Processing with Divide-and-Conquer (denoted by QP_C)

Consider the illustrative query in Fig. 5 as an example where the destination site is F_1. In scheme QP_C, the J_C method is utilized. First, the query is divided into two subqueries and each subquery is processed with forward scheduling algorithm. In Fig. 6, Q_{S1} and Q_{S2} belong to Cell_1 and Cell_2, respectively. R_1, R_2, R_3, R_4, R_5, and R_6 are located at Q_{S1} and R_7, R_8, R_9, and R_{10}, in contrast, belong to subquery Q_{S2}. After each partial result of subquery is generated, we merge these residue relations to be a new query. Then, the forward scheduling algorithm is utilized again for the new query processing. Note that, since the amount of $|\mathrm{R}_F|$, where R_F denotes the relation in fixed hosts, is usually much larger than $|\mathrm{R}_M|$, that is the relation in mobile host, the partial result of each subquery will be naturally located at the fixed host. Therefore, we assume that the query result R'_1 of Q_{S1} is located in F_1 and the result R'_7 of Q_{S2} is located in F_7. Consequently, the query result can be generated in F_1 by the final merging processing from R'_7 to R'_1. Such an adaptive version of conventional procedure denoted by QP_C can be outlined below. The concept of algorithm FS (standing for forward scheduling) is also presented.

Procedure $\mathbf{QP_C}$: Determine the scheduling of multijoin queries based on the cell architecture.

Step 1: Based on cell architecture, divide the original query into several subqueries.

Step 2: Process each subquery with algorithm forward scheduling.

Step 3: Merge residue relations from each subquery into a new query, which is referred to as a conquer query.

Step 4: Do the query processing of the conquer query with forward scheduling algorithm again and generate the query result.

Step 5: Send the query result to the needed destination.

Algorithm *Forward Scheduling* (algorithm FS): Determine the join sequence starting from performing the minimum-cost join.

Step 1: Perform effectual semijoins in the query.

Step 2: With join processing, merge relations from the path of minimum transmission cost.

Step 3: Reorganize the query.

Step 4: If the query is empty, go to Step 5. Otherwise, go back to Step 2.

Step 5: End

3.3 Query Processing with Effectual Remote Mobile Joins (denoted by QP_R)

Clearly, scheme QP_C does not exploit the relationship among remote relations and may thus consume much valuable communication cost for the join processing in the merged query Q_M. Instead of partitioning the query into several subqueries based on the cell architecture, as in scheme QP_C, the concept of the effectual remote mobile join will be employed in algorithm QP_R. According to Theorem 1, an effectual remote mobile join can successfully reduce the transmission cost. The corresponding figures of each step in QP_R procedure are illustrated in Fig. 7. For ease of exposition, $L_d()$ denotes a set of local joins in the destination cell and $L_r()$ is the set of local joins in a remote cell. In addition, $R()$ represents a set of the remote joins across different cells. For example, $L_d(R_M, R_M)$ denotes the set of joins among local mobile relations in the destination cell.

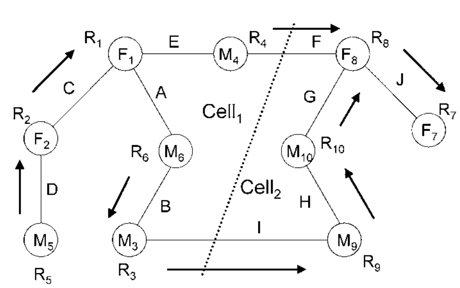


Fig. 7. Multiquery processing with QP_R methodology.

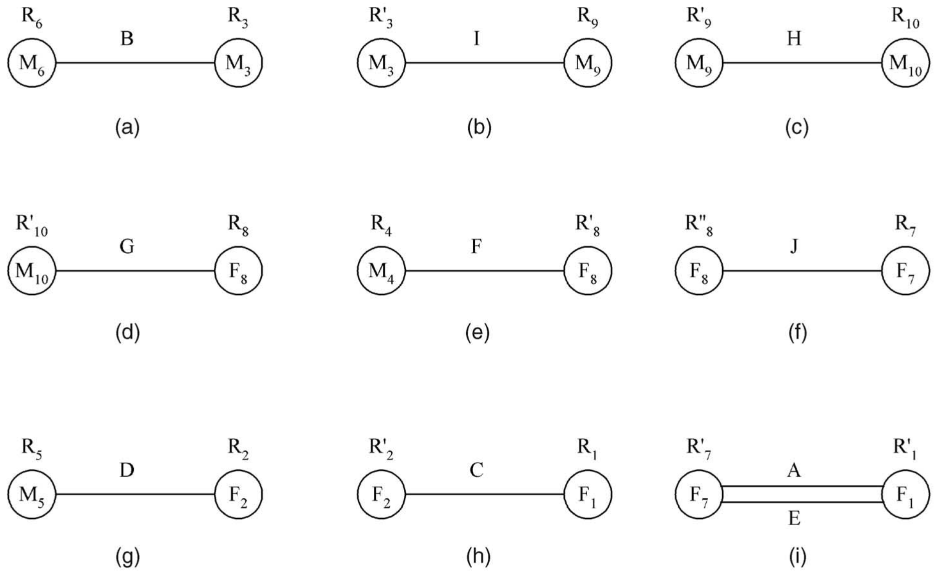


Fig. 8. Query processing with QP_R methodology. (a) Step 1: $\mathrm{L_d(R_M, R_M)}$. Step 2: $\mathrm{R(R_M, R_M)}$. (c) Step 3: $\mathrm{L_r(R_M, R_M)}$. (d) Step 4: $\mathrm{L_r(R_M, R_F)}$. (e) Step 5: $\mathrm{R(R_M, R_F)}$. (f) Step 6: $\mathrm{L_f(R_F, R_F)}$. (g) Step 7: $\mathrm{L_d(R_M, R_F)}$. (h) Step 8: $\mathrm{L_d(R_F, R_F)}$. (i) Step 9: $\mathrm{R(R_F, R_F)}$.

First, in Step 1 of Fig. 8a, connected relations among fixed hosts and mobile hosts in the cell of query destination are merged with algorithm FS. For ease of our discussion, we assume that the join result of $R_6.B = R_3.B$ is merged to M_3. The relationship $R'_3.I = R_9.I$ among mobile hosts located in different cells is exploited by the join processing in Step 2. The result is in M_9, as shown in Fig. 8b, if $R(R_M, R_M)$ can induce effectual remote mobile joins. In Step 3 of Fig. 8c, we merge $R'_9.H = R_{10}.H$ of the connected mobile hosts in remote cells to the mobile host M_{10}. Then, Fig. 8d shows that R'_{10} is merged to the fixed host $\mathrm{F_8}$ in Step 4. Using effectual remote mobile joins $R(R_M, R_F)$ in Step 5, mobile relations in the local cell are merged into fixed hosts in the remote cell. Fig. 8f indicates the operation of merge relations in remote fixed hosts to $\mathrm{F_7}$ in Step 6. Furthermore, the merge operations among local mobile hosts and local fixed hosts are performed in Step 7, as shown in Fig. 8g. Similarly, the merged result R'_2 is assumed to be located in F_2. Then, we merge relations of the fixed hosts in the local cell to F_1 with $L_d(R_F, R_F)$ of Step 8 in Fig. 8h. Finally, Fig. 8i illustrates the final step of merging the relations in remote fixed hosts to the local fixed host F_1. The final result is generated in Step 9 of QP_R. Procedure QP_R is outlined below. Note that, in each step, the merging processing is based on algorithm FS.

Procedure $\mathrm{QP_R}$: Determine the scheduling of multijoin queries with remote mobile joins

Step 1: Merge relations in mobile hosts which are connected with each other in the destination cell of query. That is, perform the joins in the set of $L_d(R_M, R_M)$;

Step 2: If there exist effectual remote mobile joins among relations in mobile hosts, merge those relations to the mobile hosts in remote cell. That is, perform the joins in the set of $R(R_M, R_M)$;

Step 3: Merge relations in mobile hosts which are connected with each other in remote cells. That is, perform the joins in the set of $L_r(R_M, R_M)$;

Step 4: Merge relations from mobile hosts to fixed hosts, where mobile hosts and fixed hosts are connected with each other in remote cells. That is, perform the joins in the set of $L_r(R_M, R_F)$;

Step 5: If there exist effectual remote mobile joins among mobile hosts and fixed hosts, merge relation in mobile hosts of the destination cell to the fixed hosts in remote cells. That is, perform the joins in the set of $R(R_M, R_F)$;

Step 6: Merge relations in fixed hosts which are connected with each other in remote cells. That is, perform the joins in the set of $L_r(R_F, R_F)$;

Step 7: Merge relations from mobile hosts to fixed hosts, where mobile hosts and fixed hosts are in the destination cell of query. That is, perform the joins in the set of $L_d(R_M, R_F)$;

Step 8: Merge relations in fixed hosts which are in the destination cell of query. That is, perform the joins in the set of $L_d(R_F, R_F)$;

Step 9: Merge residue relations in fixed hosts to the fixed host of the destination cell. That is, perform the joins in the set of $R(R_F, R_F)$;

3.4 Analysis of Solution Space

Assume that there are N_{Cell} cells in a mobile network and each cell is of N_{Mobile} mobile hosts and N_{Fixed} hosts. In essence, according to the traditional query processing technique, i.e., FS-like algorithm as mentioned above, the size of solution space could be up to $O(((N_{Mobile} + N_{Fixed}) \times N_{Cell})!)$. On the other hand, algorithm QP_C merges those relations in each cell separately by algorithm FS in advance, followed by the employment of another FS process to merge those sub-query results of each cell as the final query solution. The size of solution space of QP_C is therefore $O((N_{Mobile} + N_{Fixed})! \times N_{Cell}!)$. It is noted that the QP_C is more efficient than those traditional query proces-

TABLE 3
Default Values of Model Parameters

Symbol	Description	Default
N_M	Number of mobile relations in a cell	2
p_{QG}	Intensity of a query graph	0.5
$\|R_M\|$	The average amount of tuples for relations in mobile hosts	500
$\|R_F\|$	The average amount of tuples for relations in fixed hosts	5×10^5
$\|K\|$	The average size of cardinality for attributes	2,500
c_{FF}^L	Local transmission cost coefficient among fixed hosts	1
r_{FF}^{RL}	Transmission cost ratio between remote fixed hosts and local fixed hosts	30
r_{MF}^L	Transmission cost ratio between local mobile hosts and local fixed hosts	10
r_{MF}^R	Transmission cost ratio between remote mobile hosts and remote fixed hosts	1.5

sing algorithms in the wireless mobile computing environment. As compared to algorithm QP_C, QP_R utilizes a larger searching space. However, as will be seen in our experimental studies, judiciously applying effectual remote mobile joins, algorithm QP_R can significantly reduce the amount of data transmission cost as a whole.

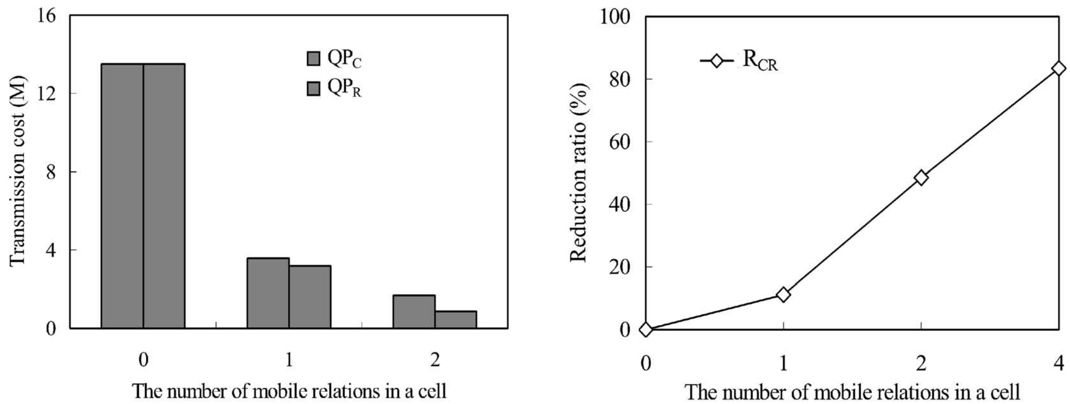
4 Experimental Studies

As shown in our previous analysis, in such mobile environments, the query processing, enhanced with useful features of wireless technology and mobility of mobile units, provides a new interesting dimension beyond traditional distributed computing systems. The applications of processing distributed mobile queries with interleaved remote mobile joins can be well developed, for example, in a telecommunication alarm system. With wireless communication technologies, the newly explored information in remote mobile devices can also be applied to online services.

For obtaining reliable experimental results, the method to generate synthetic query processing we employed in this study is similar to the ones used in prior works [14], [17]. Simulations were performed to evaluate the effectiveness of join processing methods and query processing schemes. The simulation program was coded in C++ and input queries were generated as follows: The number of relations in a query was predetermined. The occurrence of an edge between two relations in the query graph was determined according to a given probability, denoted by p_{QG}. Without loss of generality, only queries with connected query graphs were deemed valid and used for our study. Based on the above, the cardinalities of relations and attributes were randomly generated from a uniform distribution within some reasonable ranges. These settings are similar to those prior works in query processing [14], [17]. To concentrate our evaluation, the number of cells to be evaluated is assumed to be two and only one fixed server host is located in each communication cell. In addition to two mobile hosts in each cell, we also assume that each host only contains one relation. With merge operations, we can merge several fixed hosts in the same cell together and combine several remote cells to be one unit of cell. As such, despite its simplicity, our model can still reflect the reality. For ease of exposition, unless mentioned otherwise, the default value of each parameter is given in Table 3. The selectivity of relation attributes in mobile hosts is randomly generated in the range of 0.1 to 0.2, while that in fixed hosts is in the range of 0.8 to 0.95. In addition, the communication costs across remote hosts are more expensive than those across local hosts. Thus, r_{FF}^{RL} and r_{MM}^{RL} are, in general, larger than one, e.g., $r_{FF}^{RL} = 30$ and $r_{MM}^{RL} = 10$ in Taiwan telecommunication service. Similarly, $r_{MF}^{R} = 1.5$ and $r_{MF}^{L} = 4.5$ are larger than one due to the asymmetry features between mobile hosts and fixed hosts. Moreover, the density of query is given as $p_{QG} = 0.5$ and each execution cost is the result of the average from 20 query executions. To simplify our presentation, the execution cost of algorithms A denoted by $\mathrm{Cost}(A)$, where A can be QP_C or QP_R. To exhibit the benefit of relation replication, the reduction ratio $R_{CR} = |\frac{Cost(QP_C)-Cost(QP_R)}{Cost(QP_C)}|$ is used as a metric to compare QP_C and QP_R.

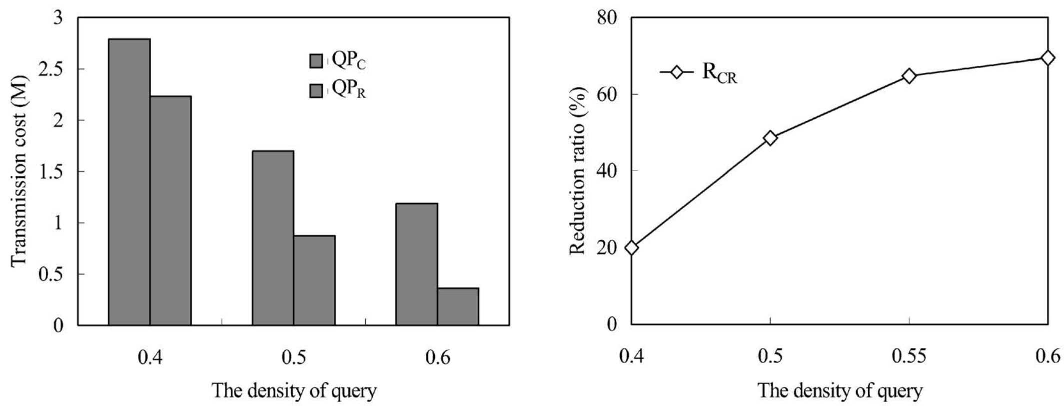
Even though many prior studies have developed several efficient algorithms for join or semijoin processing, little work has taken both the network topology and the limitation on network bandwidth into consideration. In accordance with the cost model proposed in this paper, the algorithm QP_C, our proposed algorithm, can be taken as one kind of the extended schemes from the conventional query processing. Furthermore, as in most previous works in distributed query processing, averages are taken over absolute query execution costs. Performance comparison on execution costs of queries originating from different sites is, in fact, a system-dependent issue and is beyond the scope of this paper. Without loss of generality, we assume the temporal-final query result will be located at a dedicated fixed host. Then, the final query result will be transmitted to the original host of the query.

Our results demonstrate the effectiveness of our effectual remote mobile joins in the distributed mobile query processing as taking the network topology into consideration. Extensive performance studies are conducted. Sensitivity analysis on various parameters, including number of mobile hosts in a cell, the density of query, the amount of tuples in a relation, the size of relation cardinality, and transmission cost coefficients in a mobile computing network is conducted.



(a) (b)

Fig. 9. Performance studies on various values of N_M in each cell. (a) Transmission cost of $\mathrm{QP_C}$ and $\mathrm{QP_R}$. (b) Reduction ratio between $\mathrm{QP_C}$ and $\mathrm{QP_R}$.



(a) (b)

Fig. 10. Performance studies on the density of query. (a) Transmission cost of $\mathrm{QP_C}$ and $\mathrm{QP_R}$. (b) Reduction ratio between $\mathrm{QP_C}$ and $\mathrm{QP_R}$.

4.1 Experiment One: Evaluating Number of Mobile Relations in Each Cell

Fig. 9 shows the performance results for the number of mobile relations N_M in each cell. Explicitly, since mobile relations are employed as reducers in our proposed query processing cost model, more mobile joins in the query processing lead to less data transmitted through the network. In other words, more mobile relations in a cell will lead to a higher likelihood of having the effectual mobile joins as reducers in the query processing. As a result, with the growth of N_M in each cell, the transmission costs required by both algorithms QP_C and QP_R decrease, as shown in Fig. 9a. In Fig. 9b, it can be seen that, with the presence of effectual remote mobile joins, QP_R outperforms QP_C. A higher reduction ratio R_{CR} is observed for large numbers of N_M.

4.2 Experiment Two: Performance Studies for Density of Query

In this experiment, we analyze the contribution of the density of query p_{QG} in algorithms QP_C and QP_R. In Fig. 10a, it can be seen that the execution results of both algorithms improves when the connected probability among relations increases. Statistically, a larger value of p_{QG} leads to a higher possibility of having effectual mobile joins, including local and remote mobile joins. Thus, both QP_C and QP_R improve with the growth of query density. However, QP_R performs better with the extra benefit from effectual remote mobile joins, as shown in Fig. 10b.

4.3 Experiment Three: Evaluation on the Attribute Cardinalities

Fig. 11 shows the performance results for the ratio of attribute cardinalities over the amount of relation tuples in the mobile hosts. Consequently, with the growth of attribute cardinalities, both of the transmission costs of QP_C and QP_R decrease, as shown in Fig. 11a. Fig. 11b shows that, due to the use of the remote mobile joins, the advantage of QP_R over QP_C increases as the number of attribute cardinalities increases. However, once the size of attribute cardinality grows over a threshold ratio of the amount of relation tuples in mobile hosts, the effect of cost reduction achieved by using remote mobile joins will become saturated.

4.4 Experiment Four: Evaluating Tuples Ratio between Fixed Hosts and Mobile Hosts

The horizontal axis in Fig. 12 indicates the value of $\frac{|R_F|}{|R_M|}$. With fixed size of the relation tuples in mobile hosts, the increase of the number of tuples in fixed hosts will lead to more transmission costs required in the query processing of both QP_C and QP_R, as shown in Fig. 12a. Specifically, as

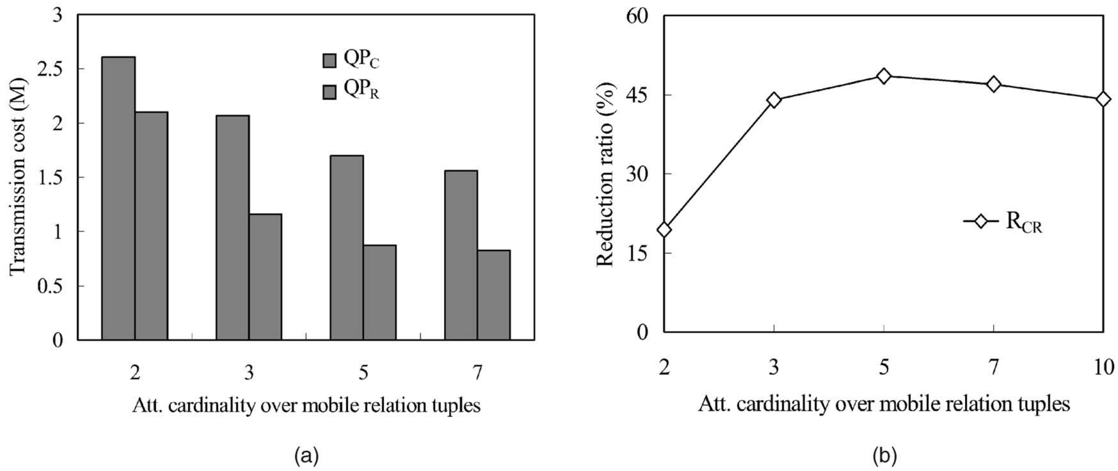


Fig. 11. Performance studies on the size of attribute cardinalities over the amount of relation tuples in mobile hosts. (a) Transmission cost of $\mathrm{QP_C}$ and $\mathrm{QP_R}$. (b) Reduction ratio between $\mathrm{QP_C}$ and $\mathrm{QP_R}$.

shown in Fig. 12b, QP_R exhibits a better scheduling than QP_C for a multijoin query processing with the growth of $\frac{|R_F|}{|R_M|}$. Note that effectual remote mobile joins are more powerful for dealing with the large amount of relation tuples in remote fixed hosts, thereby reducing the amount of data transmission costs incurred. Consequently, QP_R can lead to the design of an efficient and effective query processing procedure for a mobile computing environment.

4.5 Experiment Five: Evaluation on the Transmission Cost Ratio

Several parameters, as known, are used to be the transmission cost coefficients in a mobile computing environment. In this experiment, we will show that these assumptions have less influence on the efficiency of our algorithms. Fig. 13 shows the experimental results with various values of r^{RL}_{FF} while Fig. 14 shows the performance studies with various values of r^{L}_{MF}. Moreover, the performance studies about r^{R}_{MF} are also given in Fig. 15. Since $r^{R}_{MF} * r^{RL}_{FF} = r^{RL}_{MM} * r^{L}_{MF}$, as discussed in the cost model, the transmission cost ratio between remote mobile hosts and local mobile hosts, i.e., r^{RL}_{MM}, can be derived and $r^{RL}_{MM} = \frac{r^{R}_{MF} * r^{RL}_{FF}}{r^{L}_{MF}}$. Similar scenarios were observed when r^{RL}_{MM} was evaluated.

With the increase of r^{RL}_{FF}, it can be seen that the transmission coefficient among remote fixed hosts, i.e., $c^{R}_{FF} = r^{RL}_{FF} * c^{L}_{FF}$, gets higher. Thus, the total transmission costs required by both of QP_C and QP_R increase, as shown in Fig. 13a. However, since $|\mathrm{R}_F|$ is, in general, large, the reduction ratio R_{CR} is less affected by r^{RL}_{FF}. Thus, R_{CR} just slightly increases with the growth of r^{RL}_{FF} in Fig. 13b. Similarly, because the number of relation tuples in mobile hosts is small as compared to that in fixed hosts, the reduction ratio R_{CR} remains unchanged with the growth of r^{L}_{MF} as shown in Fig. 14b. Even though higher r^{L}_{MF} will also lead to the increase of total transmission costs caused by QP_C and QP_R, the total transmission cost of query processing is orthogonal to the value of r^{L}_{MF}, as shown in Fig. 14. Furthermore, due to the advantage of using remote mobile joins in QP_R, R_{CR} increases slightly in Fig. 15b.

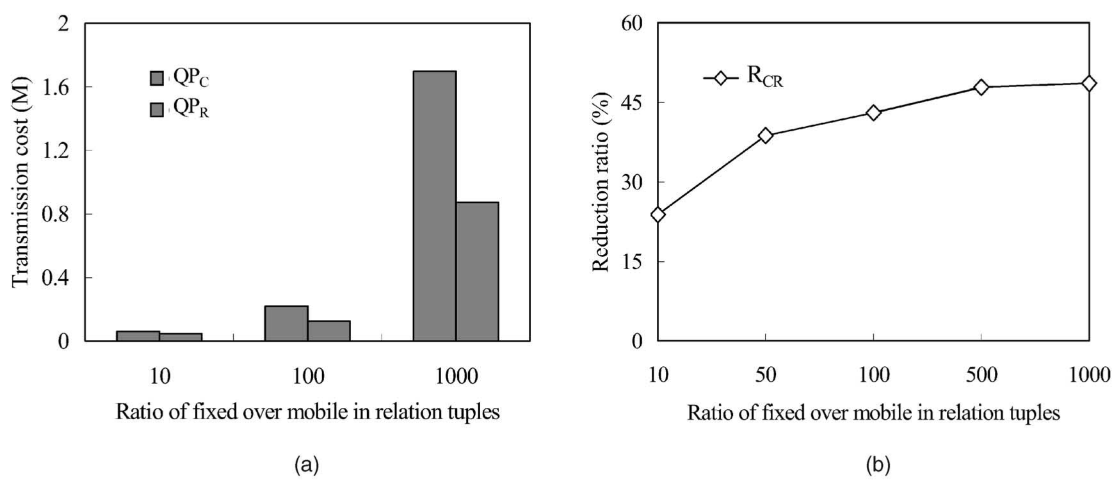


Fig. 12. Performance studies on the ratio of relation tuples in fixed hosts over that in mobile hosts. (a) Transmission cost of $\mathrm{QP_C}$ and $\mathrm{QP_R}$. (b) Reduction ratio between $\mathrm{QP_C}$ and $\mathrm{QP_R}$.

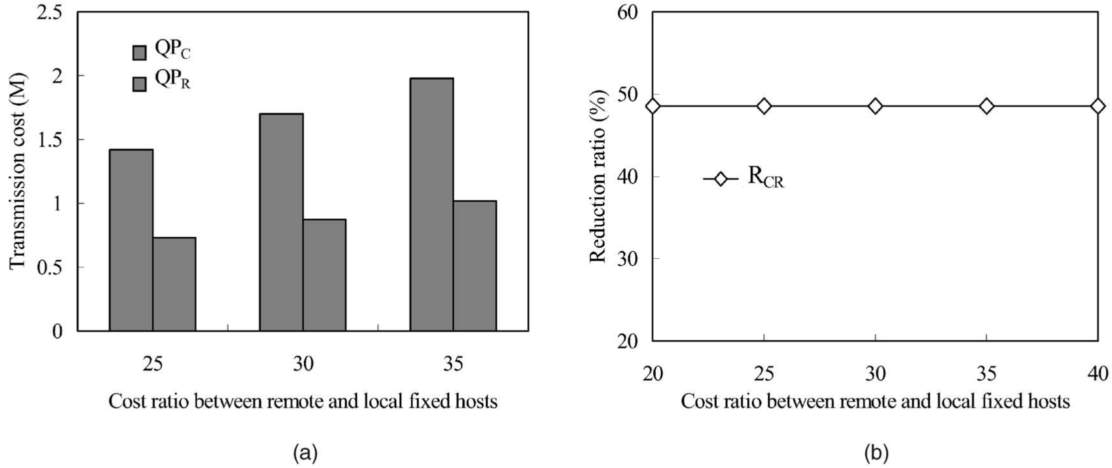


Fig. 13. Performance studies on transmission cost ratios between remote fixed hosts and local fixed hosts. (a) Transmission cost of $\mathrm{QP_C}$ and $\mathrm{QP_R}$. (b) Reduction ratio between $\mathrm{QP_C}$ and $\mathrm{QP_R}$.

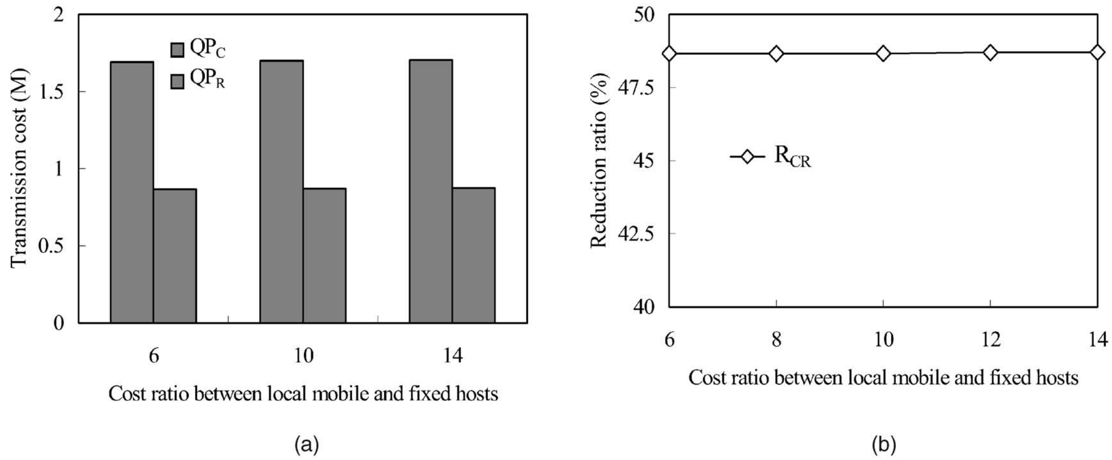


Fig. 14. Performance studies on transmission cost ratios between local mobile hosts and local fixed hosts. (a) Transmission cost of $\mathrm{QP_C}$ and $\mathrm{QP_R}$. (b) Reduction ratio between $\mathrm{QP_C}$ and $\mathrm{QP_R}$.

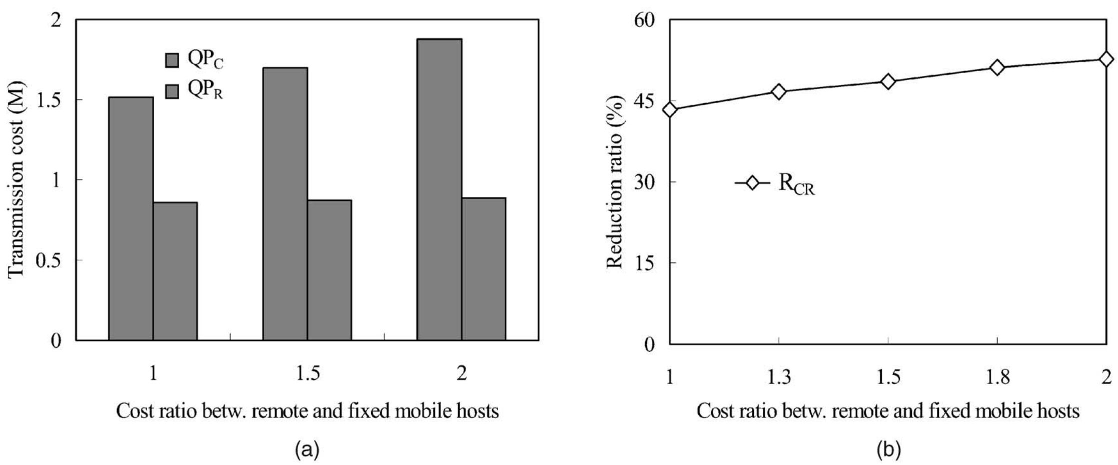


Fig. 15. Performance studies for transmission cost ratio betwen remote mobile hosts and local mobile hosts. (a) Transmission cost of $\mathrm{QP_C}$ and $\mathrm{QP_R}$. (b) Reduction ratio between $\mathrm{QP_C}$ and $\mathrm{QP_R}$.

5 Conclusions

In this paper, we have explored some unique features of a mobile environment and then, in light of these features, we devised query processing methods for both join and query processing. Remote mobile joins were said to be effectual if they were, when interleaved into a join sequence, able to reduce the amount of data transmission cost required for distributed mobile query processing. Since mobile relations were employed as reducers in our proposed query processing cost model, more mobile joins in the query processing led to less data transmitted through the network.

Judiciously interleaving effectual remote mobile joins into a query scheduling can significantly reduce the total amount of data communication among different cells. It was verified that the total data transmission cost of the processing in a distributed mobile query was reduced by the algorithms designed by using effectual remote joins. Performance studies on the sensitivity of various important parameters, including the number of mobile relations in a cell architecture, the density of query, the size of relation tuples, attribute cardinalities, and network transmission coefficients in a mobile computing model were also conducted.

APPENDIX

PROOFS OF LEMMA 1 AND THEOREM 1

Proof of Lemma 1. According to the definition of $c^L_{MF} = c^L_{MM}$, $c^R_{MF} = c^R_{MM}$, $c^R_{MM} = r^R_{MF} * c^R_{FF}$, and $c^R_{MM} = r^{RL}_{MM} * c^L_{MM}$, it can be seen that $c^R_{MF} = r^{RL}_{MM} * c^L_{MF}$ and $c^R_{MF} = r^R_{MF} * c^R_{FF}$. As a consequence, with $r^R_{MF} * c^R_{FF} = r^{RL}_{MM} * c^L_{MF}$, we have $c^R_{FF} = \frac{r^{RL}_{MM}}{r^R_{MF}} * c^L_{MF}$. □

Proof of Theorem 1. It follows from Sections 3.1.1 and 3.1.2 that $TC(J_R) = c^R_{MF} * |R_3| + c^R_{FF} * \rho_{1,B} * (|B| + \frac{|R_1||R_3|}{|A|})$ and $TC(J_C) = c^L_{MF} * |R_3| + c^R_{FF} * \rho_{1,B} * (|B| + |R_2|)$. In order to let $TC(J_R) < TC(J_C)$, $\{c^R_{MF} * |R_3| + c^R_{FF} * \rho_{1,B} * (|B| + \frac{|R_1||R_3|}{|A|})\}$ should be smaller than

$$\{c^L_{MF} * |R_3| + c^R_{FF} * \rho_{1,B} * (|B| + |R_2|)\}.$$

From Lemma 1,

$$TC(J_R) = r^{RL}_{MM} * c^L_{MF} * |R_3| + \frac{r^{RL}_{MM}}{r^R_{MF}} * c^L_{MF} * \rho_{1,B} * \left(|B| + \frac{|R_1||R_3|}{|A|}\right).$$

Similarly, we have

$$TC(J_C) = c^L_{MF} * |R_3| + \frac{r^{RL}_{MM}}{r^R_{MF}} * c^L_{MF} * \rho_{1,B} * (|B| + |R_2|).$$

$TC(J_R) < TC(J_C)$ will lead to

$$r^{RL}_{MM} * c^L_{MF} * |R_3| + \frac{r^{RL}_{MM}}{r^R_{MF}} * c^L_{MF} * \rho_{1,B} * \left(|B| + \frac{|R_1||R_3|}{|A|}\right) < c^L_{MF} * |R_3| + \frac{r^{RL}_{MM}}{r^R_{MF}} * c^L_{MF} * \rho_{1,B} * (|B| + |R_2|).$$

We next get

$$r^{RL}_{MM} * |R_3| + \frac{r^{RL}_{MM}}{r^R_{MF}} * \rho_{1,B} * \left(|B| + \frac{|R_1||R_3|}{|A|}\right) < |R_3| + \frac{r^{RL}_{MM}}{r^R_{MF}} * \rho_{1,B} * (|B| + |R_2|).$$

Then, $\frac{r^R_{MF} \times (r^{RL}_{MM} - 1)}{r^{RL}_{MM}} < \rho_{1,B} * (\frac{|R_2|}{|R_3|} - \frac{|R_1|}{|A|})$. Explicitly,

$$\frac{r^R_{MF} \times (r^{RL}_{MM} - 1)}{r^{RL}_{MM}} < \rho_{1,B} * \left(\frac{|R_2|}{|R_3|} - \frac{|R_1|}{|A|}\right)$$

leads to $TC(J_R) < TC(J_C)$. □

ACKNOWLEDGMENTS

The authors are supported in part by the National Science Council, Project Nos. NSC 90-2213-E-002-086 and NSC 90-2213-E-002-116, Taiwan, Republic of China.

REFERENCES

[1] 3GPP, http://www.3gpp.org/, 2002.
[2] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, "Broadcast Disks: Data Management for Asymmetric Communication Environments," *Proc. ACM SIGMOD Conf.*, pp. 198-210, Mar. 1995.
[3] D. Aksoy and M.J. Franklin, "Scheduling for Large-Scale On-Demand Data Broadcasting," *Proc. IEEE INFOCOM Conf.*, pp. 651-659, Mar. 1998.
[4] I. Akyildiz, J. McNair, J. Ho, H. Uzunalioglu, and W. Wang, "Mobility Management in Next-Generation Wireless Systems," *Proc. IEEE*, vol. 87, pp. 1347-1384, Aug. 1999.
[5] R. Alonso and S. Ganguly, "Query Optimization in Mobile Environments," *Proc. Fifth Workshop Foundations of Models and Languages for Data and Objects*, pp. 1-17, Sept. 1993.
[6] "Applications of Mobile Computing," http://www.nokia.com/3g/index.html, 2002.
[7] D. Barbara, "Mobile Computing and Databases—A Survey," *IEEE Trans. Knowledge and Data Eng.*, vol. 11, no. 1, pp. 108-117, Jan./Feb. 1999.
[8] P.A. Bernstein, N. Goodman, E. Wong, C. Reeve, and J.B. Rothnie, "Query Processing in a System for Distributed Databases," *ACM Trans. Database Systems*, vol. 6, no. 4, pp. 602-625, Dec. 1981.
[9] G. Bojan and M.M. Qutaibah, "Combinatorial Optimization of Distributed Queries," *IEEE Trans. Knowledge and Data Eng.*, vol. 7, no. 6, pp. 915-927, Dec. 1995.
[10] P. Bonnet, J. Gehrke, and P. Seshadri, "Querying the Physical World," *IEEE Personal Comm.*, vol. 75, pp. 10-15, 2000.
[11] J. Cai and D.J. Goodman, "General Packet Radio Service in GSM," *IEEE Comm. Magazine*, pp. 122-131, Oct. 1997.
[12] S. Ceri and G. Pelagatti, *Distributed Databases Principles and Systems.* McGraw-Hill, 1984.
[13] M.-S. Chen and P.S. Yu, "Interleaving a Join Sequence with Semijoins in Distributed Query Processing," *IEEE Trans. Parallel and Distributed Systems*, vol. 3, no. 5, pp. 611-621, Sept. 1992.
[14] M.-S. Chen and P.S. Yu, "Combining Join and Semijoin Operations for Distributed Query Processing," *IEEE Trans. Knowledge and Data Eng.*, vol. 5, no. 3, pp. 534-542, June 1993.
[15] M.-S. Chen and P.S. Yu, "A Graph Theoretical Approach to Determine a Join Reducer Sequence in Distributed Query Processing," *IEEE Trans. Knowledge and Data Eng.*, vol. 6, no. 1, pp. 152-165, Feb. 1994.
[16] M.-S. Chen, P.S. Yu, and T.-H. Tang, "On Coupling Multiple Systems with a Global Buffer," *IEEE Trans. Knowledge and Data Eng.*, vol. 8, no. 2, pp. 339-344, Apr. 1996.
[17] M.-S. Chen, P.S. Yu, and K.-L. Wu, "Optimization of Parallel Execution for Multi-Join Queries," *IEEE Trans. Knowledge and Data Eng.*, vol. 8, no. 3, pp. 416-428, June 1996.
[18] M.H. Dunham, "Mobile Computing and Databases," *Tutorial Int'l Conf. Data Engi.*, Feb. 1998.
[19] M.H. Dunham, A. Helal, and S. Balakrishnan, "A Mobile Transaction Model that Captures Both the Data and Movement Behavior," *ACM Mobile Networks and Applications*, vol. 2, no. 2, pp. 149-162, Oct. 1997.
[20] M.H. Dunham and V. Kumar, "Location Dependent Data and Its Management in Mobile Databases," *Proc. Ninth International Workshop Database and Expert Systems Applications*, pp. 26-29, Aug. 1998.
[21] M.H. Dunham and V. Kumar, "Impact of Mobility on Transaction Management," *Proc. First ACM Int'l Workshop Data Eng. for Wireless and Mobile Access*, pp. 14-21, Aug. 1999.
[22] M.J. Franklin, B.T. Jonsson, and D. Kossmann, "Performance Tradeoffs for Client-Server Query Processing," *Proc. ACM SIGMOD*, pp. 149-160, June 1996.

[23] S. Ganguly, "Design and Analysis of Parametric Query Optimization Algorithms," *Proc. 24th Int'l Conf. Very Large Data Bases,* pp. 228-238, 1998.

[24] D. Gardy and C. Puech, "On the Effect of Join Operations on Relation Sizes," *ACM Trans. Database Systems,* vol. 14, no. 4, pp. 574-603, Dec. 1989.

[25] N. Gupta, J.R. Haritsa, and M. Ramanath, "Distributed Query Processing on the Web," *Proc. 2000 Int'l Conf. Data Eng.,* 2000.

[26] J.C. Haartsen, "The Bluetooth Radio System," *IEEE Personal Comm.,* vol. 7, no. 1, pp. 28-36, Feb. 2000.

[27] P. Haas and A. Swami, "Sequential Sampling Procedures for Query Size Estimation," *Proc. ACM SIGMOD,* pp. 341-350, June 1992.

[28] T. Hara, "Effective Replica Allocation in Ad Hoc Networks for Improving Data Accessibility," *Proc. IEEE INFOCOM Conf.,* Apr. 2001.

[29] T. Imielinski and B.R. Badrinath, "Querying in Highly Mobile and Distributed Environment," *Proc. 18th Int'l Conf. Vary Large Data Bases,* pp. 41-52, Aug. 1992.

[30] T. Imielinski, S. Viswanathan, and B.R. Badrinath, "Data on Air: Organization and Access," *IEEE Trans. Knowledge and Data Eng.,* vol. 9, no. 3, pp. 353-372, June 1997.

[31] J. Jannink, D. Lam, N. Shivakumar, J. Widom, and D.C. Cox, "Efficient and Flexible Location Management Techniques for Wireless Communication Systems," *ACM Wireless Networks,* vol. 3, no. 5, pp. 361-374, Oct. 1997.

[32] J. Jing, A. Helal, and A. Elmagarmid, "Client-Server Computing in Mobile Environments," *ACM Computing Surveys,* vol. 31, no. 2, pp. 117-157, June 1999.

[33] P. Krishnamurthy, K. Pahlavan, and A. Hatami, "Handoff in Hybrid Mobile Data Networks," *IEEE Personal Comm.,* vol. 7, no. 2, Apr. 2000.

[34] J.M. Kahn, R.H. Katz, and K.S.J. Pister, "Mobile Networking for Smart Dust," *Proc. Fifth ACM/IEEE Int'l Conf. Mobile Computing and Networking,* pp. 17-19, Aug. 1999.

[35] A. Kahol, S. Khurana, S.K.S. Gupta, and P.K. Srimani, "A Strategy to Manage Cache Consistency in a Disconnected Distributed Environment," *IEEE Trans. Parallel and Distributed Systems,* vol. 12, no. 7, pp. 686-700, July 2001.

[36] S. Kapp, "802.11: Leaving the Wire Behind," *IEEE Internet Computing,* vol. 6, no. 1, Feb. 2002.

[37] D.N. Knisely, S. Kumar, S. Laha, and S. Nanda, "Evolution of Wireless Data Services: IS-95 to CDMA2000," *IEEE Comm. Magazine,* pp. 140-149, Oct. 1998.

[38] D. Kossmann, "The State of the Art in Distributed Query Processing," *ACM Computing Surveys,* Sept. 2000.

[39] W.-C. Lee, Q.L. Hu, and D.L. Lee, "A Study on Channel Allocation for Data Dissemination in Mobile Computing Environments," *ACM/Baltzer Mobile Networks and Applications,* vol. 4, no. 5, pp. 117-129, May 1999.

[40] R. Mase and S. Nogami, "An Internet Paradigm and Scheduling Algorithm for Quaranteeing End-to-End QoS," *Proc. Internet Workshop 1999 (IWS '99),* pp. 107-111, 1999.

[41] P. Mishra and M.H. Eich, "Join Processing in Relational Databases," *ACM Computing Surveys,* vol. 24, no. 1, pp. 63-113, Mar. 1992.

[42] K.J. Negus, A.P. Stephens, and J. Lansford, "HomeRF: Wireless Networking for the Connected Home," *IEEE Personal Comm.,* vol. 7, no. 1, pp. 20-27, Feb. 2000.

[43] E. Pacitti, P. Minet, and E. Simon, "Fast Algorithms for Maintaining Replica Consistency in Lazy Master Replicated Databases," *Proc. 1999 Int'l Conf. Very Large Data Bases,* pp. 126-137, 1999.

[44] Palm Pilots of 3com, http://www.3com.com/palm/index.html, 2002.

[45] W.-C. Peng and M.-S. Chen, "Exploiting the Features of Asymmetry for Query Processing in a Mobile Computing Environment," *Proc. Fifth IFCIS Conf. Cooperative Information Systems (CoopIS 2000),* 2000.

[46] E. Pitoura and G. Samaras, "Locating Objects in Mobile Computing," *IEEE Trans. Knowledge and Data Eng.,* vol. 13, no. 4, pp. 571-592, 2001.

[47] Q. Ren and M.H. Dunham, "Using Clustering for Effective Management of a Semantic Cache in Mobile Computing," *Proc. First ACM Int'l Workshop Data Eng. for Wireless and Mobile Access,* pp. 94-101, Aug. 1999.

[48] Q. Ren and M.H. Dunham, "Using Semantic Caching to Manage Location Dependent Data in Mobile Computing," *Proc. Sixth ACM/IEEE Int'l Conf. Mobile Computing and Networking,* pp. 210-221, Aug. 2000.

[49] N. Roussopoulos and H. Kang, "A Pipeline N-Way Join Algorithm Based on the 2-Way Semijoin Program," *IEEE Trans. Knowledge and Data Eng,m* vol. 3, no. 4, pp. 461-473, Dec. 1991.

[50] A.Y. Seydim, M.H. Dunham, and V. Kumar, "Location Dependent Query Processing," *Proc. Second ACM Int'l Workshop Data Eng. for Wireless and Mobile Access,* pp. 47-53, May 2001.

[51] N. Shivakumar, J. Jannink, and J. Widom, "Per-User Profile Replication in Mobile Environments: Algorithms, Analysis, and Simulation Results," *ACM Mobile Networks and Applications,* vol. 2, no. 2, pp. 129-140, Oct. 1997.

[52] T.M. Siep, I.C. Gifford, R.C. Braley, and R.F. Heile, "Paving the Way for Personal Area Network Standards: An Overview of the IEEE P802.15 Working Group for Wireless Personal Area Networks," *IEEE Personal Comm.,* pp. 6-13, Feb. 2000.

[53] A.P. Sistla, O. Wolfson, S. Chamberlain, and S. Dao, "Modeling and Querying Moving Objects," *Proc. 13th Int'l Conf. Data Eng.,* pp. 422-432, Apr. 1997.

[54] N.R. Sollenberger, N. Seshadri, and R. Cox, "The Evolution of IS-136 TDMA for Third-Generation Wireless Services," *IEEE Personal Comm.,* pp. 8-18, June 1999.

[55] K. Stathatos, N. Roussopoulos, and J.S. Baras, "Adaptive Data Broadcast in Hybrid Networks," *Proc. 23rd Int'l Conf. Very Large Data Bases,* pp. 326-335, 1997.

[56] M. Stemm and R.H. Katz, "Vertical Handoffs in Wireless Overlay Networks," *ACM Mobile Networks and Applications,* vol. 3, no. 4, pp. 319-334, 1998.

[57] A.B. Stephens, Y. Yesha, and K.E. Humenik, "Optimal Allocation for Partially Replicated Database Systems on Ring Networks," *IEEE Trans. Knowledge and Data Eng.,* vol. 6, pp. 975-982, 1994.

[58] "WAP Application in Nokia," http://www.nokia.com/corporate/wap/future.html, 2002.

[59] WAP Forum, http://www.wapforum.org/, 2002.

[60] O. Wolfson, S. Chamberlain, S. Dao, L. Jiang, and G. Mendez, "Cost and Imprecision in Modeling the Position of Moving Objects," *Proc. 14th Int'l Conf. Data Eng.,* pp. 588-596, Feb. 1998.

[61] Z. Xie and J. Han, "Join Index Hierarchies for Supporting Efficient Navigations in Object-Oriented Databases," *Proc. 1994 Int'l Conf. Very Large Data Bases,* pp. 522-533, Sept. 1994.

[62] B. Xu, O. Wolfson, S. Chamberlain, and N. Rishe, "Cost Based Data Dissemination in Satellite Networks," *ACM Mobile Networks and Applications,* vol. 7, no. 1, Jan. 2002.

[63] J.L. Xu, Q.L. Hu, D.L. Lee, and W.-C. Lee, "SAIU: An Efficient Cache Replacement Policy for Wireless On-Demand Broadcasts," *Proc. Ninth ACM Int'l Conf. Information and Knowledge Management,* pp. 46-53, Nov. 2000.

[64] C.T. Yu and C.C. Chang, "Distributed Query Processing," *ACM Computing Surveys,* vol. 16, no. 4, pp. 399-433, Dec. 1984.

[65] C.T. Yu and W. Meng, *Principle of Database Query Processing for Advanced Applications.* San Francisco: Morgan Kaufmann, 1997.

[66] T.B. Zahariadis, K.G. Vaxevanakis, C.P. Tsantilas, and N.A. Zervos, "Global Roaming in Next-Generation Networks," *IEEE Comm.,* vol. 40, no. 2, Feb. 2002.

[67] Z. Zhang and W. Perrizo, "Distributed Query Processing Using Active Networks," *Proc. 2000 ACM Symp. Applied Computing,* pp. 374-380, 2000.

[68] B. Zheng and D.L. Lee, "Processing Location-Dependent Queries in a Multi-Cell Wireless Environment," *Proc. Second ACM Int'l Workshop Data Eng. for Wireless and Mobile Access,* pp. 54-65, May 2001.

Chang-Hung Lee received the BS and PhD degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1996 and 2002, respectively. Dr. Lee has published in ACM SIGKDD, IEEE ICDCS, IEEE ICDM, and SIAM SDM. He is currently a research staff member at BenQ Inc., Taipei, Taiwan, leading projects in wireless multimedia applications. His research interests include distributed database systems, data mining, mobile computing systems, wireless multimedia, and flat panel display.

Ming-Syan Chen received the BS degree in electrical engineering from National Taiwan University, Taipei, Taiwan, and the MS and PhD degrees in computer, information, and control engineering from the University of Michigan, Ann Arbor, in 1985 and 1988, respectively. Dr. Chen is currently a professor in the Electrical Engineering Department, National Taiwan University, Taipei, Taiwan. He was a research staff member at the IBM T.J. Watson Research Center, Yorktown Heights, New York, from 1988 to 1996. His research interests include database systems, data mining, mobile computing systems, and multimedia networking, and he has published more than 135 papers in his research areas. In addition to serving as a program committee member for many conferences, Dr. Chen served as an associate editor of the *IEEE Transactions on Knowledge and Data Engineering* on data mining and parallel database areas from 1997 to 2001, an editor of the *Journal of Information Science and Engineering*, a distinguished visitor of the IEEE Computer Society for Asia-Pacific from 1998 to 2000, and program chair of PAKDD-02 (Pacific Area Knowledge Discovery and Data Mining), program vice-chair of VLDB-2002 (Very Large Data Bases), general chair of the Real-Time Multimedia System Workshop in 2001, program chair of the IEEE ICDCS Workshop on Knowledge Discovery and Data Mining in the World Wide Web in 2000, and program cochair of the International Conference on Mobile Data Management in 2003, International Computer Symposium (ICS) on Computer Networks, Internet, and Multimedia in 1998 and 2000, and ICS on Databases and Software Engineering in 2002. He was a keynote speaker on Web data mining at the International Computer Congress in Hong Kong, 1999, a tutorial speaker on Web data mining at DASFAA-1999, and on parallel databases at the 11th IEEE International Conference on Data Engineering in 1995, and also a guest coeditor for the *IEEE Transactions on Knowledge and Data Engineering* for a special issue for data mining in December 1996. He holds, or has applied for, 18 US patents and seven ROC patents in the areas of data mining, Web applications, interactive video playout, video server design, and concurrency and coherency control protocols. He received the Outstanding Innovation Award from IBM Corporation in 1994 for his contribution to parallel transaction design and implementation for a major database product and numerous awards for his research, teaching, inventions, and patent applications. Dr. Chen is a senior member of the IEEE and a member of the ACM.