Data Management in Mobile Peer-to-Peer Networks[1]

Bo Xu and Ouri Wolfson

Department of Computer Science, University of Illinois at Chicago
{boxu, wolfson}@cs.uic.edu

Abstract. In this paper we examine the database management of spatio-temporal resource information in mobile peer-to-peer networks, where moving objects communicate with each other via short-range wireless transmission. Several inherent characteristics of this environment, including the dynamic and unpredictable network topology, the limited peer-to-peer communication throughput, and the need for incentive for peer-to-peer cooperation, impose challenges to data management. In this paper we propose our solutions to these problems. The proposed system has the potential to create a completely new information marketplace.

1 Introduction

A mobile peer-to-peer network is a set of moving objects that communicate via short-range wireless technologies such as IEEE 802.11, Bluetooth, or Ultra Wide Band (UWB). With such communication mechanisms, a moving object receives information from its neighbors, or from remote objects by multi-hop transmission relayed by intermediate moving objects. A killer application of mobile peer-to-peer networks is resource discovery in transportation. For example, the mobile peer-to-peer network approach can be used to disseminate the information of available parking slots, which enables a vehicle to continuously display on a map to the driver, at any time, the available parking spaces around the current location of the vehicle. Or, the driver may use this approach to get the traffic conditions (e.g. average speed) one mile ahead. Similarly, a cab driver may use this approach to find a cab customer, or vice versa. Safety information (e.g. a malfunctioning brake light in a vehicle) can also be disseminated in this fashion.

A mobile peer-to-peer network can also be used in matching resource producers and consumers among pedestrians. For example, an individual wishing to sell a pair of tickets for an event (e.g. ball game, concert), may use this approach right before the event, at the event site, to propagate the resource information. For another example, a passenger who arrives at an airport may use this approach to find another passenger for cab-sharing from the airport to downtown, so as to split the cost of the cab. Furthermore, the approach can be used in social networks; when two singles whose profiles match are in close geographic proximity, one can call the other's cell phone and suggest a short face-to-face meeting.

[1] Research supported by NSF Grants 0326284, 0330342, ITR-0086144, and 0209190.

The approach can also be used for emergency response and disaster recovery, in order to match specific needs with expertise (e.g. burn victim and dermatologist) or to locate victims. For example, scientists are developing cockroach-sized robots or sensors that are carried by real cockroaches, which are able to search victims in exploded or earthquake-damaged buildings [4]. These robots or sensors are equipped with radio transmitters. When a robot discovers a victim, it can use the data dissemination among mobile sensors to propagate the information to human rescuers. Sensors can also be installed on wild animals for endangered species animal assistance. A sensor monitors its carrier's health condition, and it disseminates a report when an emergency symptom is detected. Thus we use the term moving objects to refer to all, vehicles, pedestrians, robots, and animals.

We would like to comment at this moment that in our model a peer does not have to be a moving object, and databases residing on the fixed network may be involved. In many cases there are both moving peers and fixed peers, and they collaborate in data dissemination. For example, a sensor in the parking slot (or the meter for the slot) monitors the slot, and, while unoccupied, transmits the availability information to vehicles nearby. Or all the slots in a parking lot may transmit the information to a fixed 802.11 hotspot via a wired network, and the hotspot announces the information. In either case, the vehicles that receive the information may propagate it to a wider area via the mobile peer-to-peer network approach. In such an environment the mobile peer-to-peer network serves as a supplement/extension to the fixed-site based solution.

Compared to static peer-to-peer networks and static sensor networks, mobile peer-to-peer networks have the following characteristics that present challenges to data management.

1. Dynamic, unpredictable, and partitionable network topology. In our environment the peers are physically mobile, and sometimes can be highly mobile (consider vehicles that move in opposite directions at 120 miles/hour relative speed). The traffic density can vary in a big range from rush hours to midnight. The underlying communication network is thus subject to topology changes and disconnections. Peer-to-peer or sensor network approaches that require pre-defined data access structures such as search routing tables used in Gridella [12], Chord [14] and spanning trees used in Cougar [15] and TinyDB [13, 21] are impractical in such an environment.

2. Limited peer-to-peer communication throughput. The communication throughput between two encountered peers is constrained by the wireless bandwidth, the channel contention, and the limited connection time. For example, previous investigations into Bluetooth links have suggested 2 seconds as a typical setup time between two unknown devices [22]. This gives less than 2 seconds for data transfer when two vehicles encounter each other at 120 miles/hour relative speed (assuming that the transmission range is 100 meters). The limited throughput requires that the communication be selective such that the most important data are communicated.

3. Need for incentive for both information supplier and information propagators. Like many other peer-to-peer systems or mobile ad-hoc networks, the ultimate success of mobile peer-to-peer networks heavily relies on cooperation among users. In P2P systems, incentive is provided for peers to participate as suppliers of data, compute cycles, knowledge/expertise, and other resources. In mobile ad-hoc networks, incentive is provided for mobile hosts to participate as intermediaries/routers. In mobile peer-to-peer networks, the incentive has to be provided for participation as both suppliers and intermediaries (namely brokers).

The objective of our Dissemination of Resource Information in Vehicular Environments (DRIVE) project is to build a software platform that addresses the above issues and can be embedded within a hardware device attached to moving objects such as vehicles, personal digital assistants (PDAs), and sensors. The DRIVE platform consists of the following components:

1. Data model. We introduce a unified data model for spatio-temporal resources in mobile peer-to-peer applications related to transportation, disaster recovery, mobile electronic commerce, and social networks. We illustrate how the data model can be used to represent various resource types even though these resource types are utilized in quite different ways.

2. Data Dissemination. We propose an *opportunistic* approach to dissemination of reports regarding availability of resources (parking slot, taxi-cab customer, dermatologist, etc.). In this approach, a moving object propagates the reports it carries to encountered objects, i.e. objects that come within transmission range; and it obtains new reports in exchange. For example, a vehicle finds out about available parking spaces from other vehicles. These spaces may either have been vacated by these encountered vehicles or these vehicles have obtained this information from other previously encountered ones. We call this paradigm *opportunistic peer-to-peer* (or OP2P).

3. Total ordering of resources by relevance. With OP2P, a moving object constantly receives reports from the objects it encounters. If not controlled, the number of availability reports saved by an object will continuously increase, which will in turn increase the communication volume in future exchanges. Thus, to deal with the throughput challenge, we investigate techniques that prioritize the reports exchanged. These techniques provide a total rank in terms of relevance for all the reports across all the resource types stored in a moving object's reports database. The key issue is how to quantify the tradeoffs between the contributions of different attributes to the utility of a report.

4. Query Language and Query Processing. With OP2P, each peer *m* maintains a local reports database. The collection of the local databases of all the peers forms a virtual database to the database application in *m*. So the query language component and the query processing component deal with how to query this virtual database and how the query is processed.

5. Economic model. Our incentive mechanisms are based upon virtual currency [5]. Each peer carries virtual currency in the form of a coin counter that is protected from illegitimate manipulation by a trusted and tamper resistant hardware module [6]. Each coin is bought for a certain amount of real money but it cannot be cashed for real money. We analyze the requirements to the economic model and propose possible solutions.

6. Information Usage Strategy. This component deals with how a resource consumer should use the received reports to take possession of a resource. This is important when the resource can only be exclusively used by one object at one time. Consider for example a driver who is looking for a parking slot. The driver may receive reports of multiple parking slots, and these parking slots may be in different orientation and distance with respect to the driver's current location. Then the question is which parking slot the driver should go to (namely, pursue).

7. Transaction Management. This component aims to study a spectrum of solutions to transactional and consistency issues that arise in report dissemination, and minimize dependence on any centralized structure.

All the components are divided into three layers as shown in Figure 1. The bottom is the *data* layer, which implements the data model for the spatio-temporal resources. Above the data layer is the *support* layer. This layer defines how the data is disseminated and how queries are processed. It also contains transaction management. The top is the *utility* layer, which contains the modules relevant to utilization of the resource information, including relevance evaluation, query language, economic model, and usage strategies.

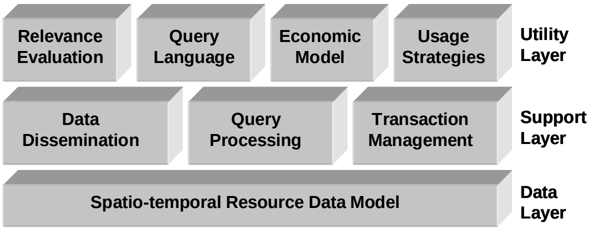

Figure 1: The architecture of DRIVE

The rest of the paper is organized as follows. Section 2 introduces the data model and report ordering. Section 3 discusses OP2P data dissemination. Section 4 presents the query language and discusses query processing. Section 5 discusses the economic model. Section 6 discusses information usage strategies and transaction management. Section 7 discusses relevant work. Section 8 concludes the paper.

2 Data Model

2.1 Resource Model

In our system, resources may be spatial, temporal, or spatio-temporal. Information about the location of a gas station is a spatial resource. Information about the price of a stock on 11/12/03 at 2pm is temporal. There are various types of spatio-temporal resources, including parking slots, car accidents (reports about such resources provide traffic-jam information), taxi-cab requests, ride-sharing invitations, demands of expertise in disaster situations, and so on. Formally in our model there are *N resource types* T_1, T_2, ..., T_N . At any point in time there are *M resources* R_1, R_2, ..., R_M, where each resource belongs to a resource type. Each resource pertains to a particular point location and a particular time point, e.g. a parking slot that is available at a certain time, a cab request at a street intersection, invitation of cab-sharing from airport to downtown from a passenger wishing to split the cost of the cab, or the demand of certain expertise at a certain location at a certain time. We assume that resources are located at points in two-dimensional geospace. The location of the resource is referred to as the *home* of the resource. For example, the home of an available parking space is the location of the space, and the home of a cab request or a cab-sharing invitation is the location of the customer. For each resource there is a *valid duration*. For example, the valid duration of the cab request resource is the time period since the request is issued, until the request is satisfied or canceled. The valid duration of the cab-sharing invitation starts when the invitation is announced and ends when an agreement is reached between the invitation initiator and another passenger. A resource is *valid* during its valid duration.

Let us comment further about spatial resources, such as gas stations, ATM machines, etc. In these cases the valid duration is infinite. Opportunistic dissemination of reports about such resources is an alternative paradigm to geographic web searching (see e.g. [7]). Geographic web searching has generated a lot of interest since many search-engine queries pertain to a geographic area, e.g. find the Italian restaurants in the town of Highland Park. Thus instead of putting up a web site to be searched geographically, an Italian restaurant may decide to put a short-range transmitter and advertise via opportunistic dissemination. In mobile systems, this also solves some privacy concerns that arise when a user asks for the closest restaurant or gas station. Traditionally, the user would have had to provide her location to the cellular provider; but she does not need to do so in our scheme. In our scheme, the transmission between two vehicles can be totally anonymous.

2.2 Peers and Validity Reports

The system consists of two types of peers, namely fixed hotspots and moving objects. Each peer m that senses the validity of resources produces *validity reports*. Denote by $a(R)$ a report for a resource R. For each resource R there is a single peer m that produces validity reports, called the *report producer* for R. A peer may be the report producer for multiple resources. Each report $a(R)$ contains at least the following information, namely *resource-id*, *create-time*, and *home-location*. Resource-id is the identification of R that is unique among all the resources of the same type in the system; create-time is the time when report $a(R)$ is created (it is also the time when R is sensed valid); home-location is the home of R.

In the parking slots example, a sensor in the parking slot (or the meter for the slot) monitors the slot, and, when the slot becomes free, it produces a validity report. In the car accident example, the report is produced by the sensor that deploys the air-bag.

$a(R)$ may contain other information depending on the resource type of R. For example, a parking slot report may include the time limit of the parking meter; a single-matching request may include the sender's personal information such as occupation and age; and so on. We say that $a(R)$ is a type T_i report if R is a type T_i resource.

Let $a(R)$ be a type T_i report. At any point in time, a peer m is either a *consumer* or a *broker* of $a(R)$. m is a consumer of $a(R)$, and $a(R)$ is a *consumer report* to m, if m is attempting to discover or find a type T_i resource. m is a broker of $a(R)$ and $a(R)$ is a *broker report* to m, if m is not attempting to discover/find T_i but is brokering $a(R)$, i.e. the only purpose of m storing $a(R)$ is to relay it to other peers.

2.3 Reports Relations

There are two relations in the reports database of a peer m. One is the *consumer relation*, which stores all the reports that m knows about and for which m is a consumer. Another is the *broker relation*, which stores all the reports that m knows about and for which m is a broker. The two relations have a common object-relational schema. The schema contains three columns: (i) resource-type which indicates the type of the reported resource; (ii) resource-id; (iii) report-description, which is an abstract data type that encapsulates all the attributes of a report. All the report description data types inherit from a single data type called *AbstractReport*. AbstractReport contains two attributes, namely create-time and home-location. Thus every report description data type has these two attributes.

2.4 Report Relevance

Given the memory and communication-throughput constraints, it is desirable that the most important or useful reports are communicated during an encounter. One possible approach that appears to achieve this goal is that the receiver explicitly expresses the criteria for the reports it is interested in receiving. For example, "Give me all the reports $a(R)$ such that the distance between R and me is smaller than 1 mile and the age of $a(R)$ (i.e. the length of the time-period since the creation of $a(R)$) is less than 1 minute." However, this does not guarantee a total order of the reports; on the other hand such a total order is necessary to ensure that most relevant reports are exchanged first (such that if disconnection occurs before the exchange completes, the loss is minimal), and that the less relevant reports are purged from memory before more relevant ones.

Our approach is to rank all the reports in a peer's reports database in terms of their relevance or expected utility, and then the reports are communicated and saved in the order of their relevance. Or, the reports requested and communicated are the ones with a relevance above a certain threshold. The notion of relevance quantifies the importance or the expected utility of a report to a peer at a particular time and a particular location.

Let $a(R)$ be a type T_i report. The relevance of $a(R)$ to a consumer at time q and location p represents the importance or the expected utility of $a(R)$ to the consumer at q and p. The relevance of $a(R)$ to a broker at time q and location p represents the importance or the expected utility of $a(R)$ to future consumers of the report that the broker estimates it will encounter. The question is how to evaluate the relevance such as to provide a total order of all the reports across all the reports relations within a peer.

We consider reports ranking a multiple attribute decision making (MADM) problem [11]. We adopt a hierarchical weighting structure. At the first level of the weighting hierarchy, each resource type T_i is assigned a weight (priority) that represents the importance of T_i relative to other resource types. At the second level, each attribute of T_i is assigned a weight that represents the importance of that attribute relative to other attributes of T_i. When ordering reports, each report is assigned a score that is a weighted aggregation of the normalized values of each attribute. Then the reports are sorted based on their scores.

3 Data Dissemination

We assume that each peer is capable of communicating with the neighboring peers within a maximum of a few hundred meters. One example is an 802.11 hotspot or a PDA with Bluetooth support. The underlying communication module provides a mechanism to resolve interference and conflicts. Each peer is also capable of discovering peers that enter into or leave out of its transmission range. Finally, each peer is equipped with a GPS system so that (i) the peer knows its location at any point in time and (ii) the clock is synchronized among all the peers.

When two peers m_1 and m_2 encounter each other, m_1 first requests consumer reports from m_2 with relevance (according to the relevance metadata of m_1) above the lowest relevance in m_1's consumer relation. Then m_1 requests broker reports from m_2 with relevance above the lowest relevance in m_1's broker relation.

We would like to emphasize that in our model, the interactions among peers are completely self-organized. The association between a pair of peers is established when they encounter each other and is ended when they finish the exchange or when they are out of the transmission range of each other. Other than this there is no other procedure for a peer to join or leave the network.

4 The Economic Model

In this section we introduce an economic model that stimulates peers to participate in report dissemination even if they are not interested in using a resource. The economic model needs to satisfy the following requirements:

It should handle two categories of reports, depending on whether the producer or the consumer pays for the reports. Reports that the owner is interested in advertising are *producer-paid*. Reports that the consumer is interested in knowing are *consumer-paid*. A resource may have both producer-paid and consumer-paid reports, if both the producer and the consumer are willing to pay for the reports. For example, reports that include the location of a gas station may be producer-paid because the gas station wishes to advertise them to neighboring vehicles. They may also be consumer-paid because a consumer may be willing to pay for a gas station report if he really needs one. Similarly for taxi-cab requests and reports of available parking slots.

1. It should consider peers that may be producers, consumers, and brokers. For consumer paid reports, both producers and brokers should be incentivized. For producer paid reports, brokers should be incentivized.

2. It should allow any peer to turn-off the spatio-temporal information module. But if it turns on the spatio-temporal information module, then the module behaves according to the economic model.

3. It should protect from the following attacks: (i) A peer creates and sells fictitious validity reports; (ii) A propagator modifies a report; (iii) A consumer-paid report is overheard by an intruding-consumer that that does not pay; in other words, an intruder overhears the legitimate transfer of the report to a consumer; (iv) A peer illegitimately increases its virtual currency counter.

Now we present our solution that satisfies the above requirements. Section 4.1 introduces two fundamental components of our economic model, namely virtual currency and the security module. Section 4.2 discusses producer-paid reports. Section 4.3 discusses consumer-paid reports.

4.1 Virtual Currency and the Security Module

The system circulates a virtual currency called *coins*. The coins owned by each peer is represented by a coin counter that is physically stored in that peer. The coin counter is decreased when the peer pays out for buying validity reports and increased when the peer earns in for selling. Each peer has a trusted and tamper resistant hardware module called the *security module*. A common example of a low-cost security module is smart card with an embedded one-chip computer [6]. The coin counter is stored in the security module and thus is protected from illegitimate manipulation. Each coin is bought for a certain amount of real money but it cannot be cashed for real money, and therefore the motivation for breaking into the security module is significantly reduced. The validity reports database, including the consumer relation and the broker relation, are stored in the security module.

When two moving objects m_1 and m_2 encounter each other, if both m_1 and m_2 have their security module open, then m_1 and m_2 start a secure session to trade validity reports[2]. The trading policy is implemented in the security module. For each resource type *T*, the owner/user of a moving object may decide not to participate in the exchange of type *T* reports. The owner/user may also turn off the security module. However, if it participates in the game, then security module behaves according to the economic model.

4.2 Producer-paid Reports

In our prior work [19], we studied producer-paid reports. At a high level, the producer-paid model works as follows. When a resource *R* is announced by its producer, the producer loads with the report *a*(*R*) a certain number of coins, *C*, called the *initial budget* of *a*(*R*). When *a*(*R*) is received by a peer, it carries a certain budget

[2] The secure session is established based on some public key infrastructure that is omitted in this paper due to space limitations.

C_0. A peer earns a flat commission fee f each time it transmits the report to another peer. The remaining budget of the report is divided between the sender and receiver (in order for both to keep propagating the report).

Intuitively, the higher the initial budget, the more peers can be reached. In [19] we determined the tradeoff between the initial budget and the effect of advertisement (i.e. the percentage of peers reached by the advertisement).

4.3 Consumer-paid Reports

Each report $a(R)$ is acquired by the security module of a peer in one of the following two modes:

1. *Consumer*. In consumer mode, the report $a(R)$ is saved in the consumer relation. The consumer relation is accessible by the user so the user can read $a(R)$. The consumer buys new reports according to some strategy (see section 6) but cannot sell them. The price of a report is a function of the relevance of the report.

2. *Broker*. In broker mode the report $a(R)$ is saved in the broker relation. The broker relation is not accessible by the user so the user cannot read $a(R)$. Thus the user in broker mode cannot directly benefit from $a(R)$. The security module simply stores $a(R)$ and forwards it to other peers. A broker pays a percentage of the price of the report. It is paid the same percentage when selling the report to another broker, and it is paid the full price when selling the report to a consumer. How to setup the percentage to maximize the incentive is a subject of our future work. The received payment constitutes the incentive of the broker to participate in the game. A broker may sell $a(R)$ to multiple consumers or brokers. A producer always operates in broker mode for the reports it transmits.

Validity reports acquired in consumer mode are consumer reports, and reports acquired in broker mode are broker reports. At a particular peer a report cannot switch between broker and consumer.

For reports which both the producer and the consumer are willing to pay for, the producer-paid policy and the consumer-paid policy can be combined. For example, initially the report is producer-paid. After the carried budget is used out, the report becomes consumer-paid.

5 Query and Query Processing

With OP2P, each peer m maintains a local reports database. The collection of the local databases of all the peers forms a virtual database to the database application in m. In this section we discuss the query interface to this virtual database and the query processing issue.

5.1 Query Language

In order to motivate the design of our query language, first let us give several typical example queries a user may issue in our environment. These queries are expressed in natural language.

Example 1: Consider a transportation application where a passenger needs to transfer from one bus route to another. Assume that buses can wait for transfer passengers for certain amount of time. Now a transfer passenger Bob wants to transfer to route #8 at a certain intersection *P*. Bob expects to arrive at *P* at 10:10. Usually a bus driver is willing to wait at a stop for a transfer passenger for at most 2 minutes. So Bob wants to notify a route #8 bus to wait him if the bus arrives at *P* between 10:08 and 10:10.

Example 2: A hotspot collects the average traffic speed on the inbound 2-miles stretch of the I-290 highway that is centered at the hotspot.

Example 3: Alert when more than 50 taxi cabs are within a certain area at the same time.

Example 4: A driver wants to know all the parking slots located inside the downtown area and the relevance of which is higher than 0.5.

We believe that declarative languages like SQL are the preferred way of express such queries. DRIVE uses the following query template.

SELECT *select-list* [FROM **reports**] WHERE *where-clause*
[GROUP BY *gb-list* [HAVING *having-list*]]
[EPOCH DURATION *epoch* [FOR *time*]]
[REMOTE *query-destination-region* [BUDGET]]

The SELECT, FROM, WHERE, GROUP BY and HAVING clauses are very similar to the functionality of SQL. The relation name **reports** represents the virtual database.

The optional EPOCH DURATION clause specifies the time between successive samplings and for how long the query lasts (see [21]).

The optional REMOTE clause specifies whether the query is to be answered by the local database or is to be evaluated in a remote geographic region. If the REMOTE clause is used, then a query-destination-region should be further specified; it indicates that the query should be disseminated to all the peers in the specified region. If the REMOTE clause is omitted, then the query is processed locally.

BUDGET specifies how much budget in virtual currency the user is willing to spend for disseminating the query and collecting the answers. If BUDGET is omitted, then the database system automatically sets a budget based on the distance to the query-destination-region, the size of the query-destination-region, the peer density, and so on.

Our query template is similar to that provided by TinyDB [21] or Cougar [15]. The difference is that we have the REMOTE…BUDGET clause discussed above.

Finally, we define a member function Rel() for each report description data type. This function takes as input a set of attributes and it returns the relevance using the input as the relevance attributes.

Now we illustrate how our query template can be used to express the query examples given at the beginning of section 4.1 (Queries for examples 2-4 are omitted due to space limitations).

Example 1: The following query notifies a route #8 buses to wait if the bus arrives at *P* between 10:08 and 10:10.

SELECT resource_id

FROM **reports**

WHERE resource-type=BUS and report-description.route_no=8 and WITHIN_DISTANCE_ SOMETIME_BETWEEN(report-description.Traj, P, 0, 10:08, 10:10)

REMOTE route_of_bus_ #8

route_no and Traj are two attributes of a bus report. Traj is the *trajectory* of the bus moving object; it defines the object's future location as a piece-wise linear function from time to the two-dimensional geography. WINTIN_DISTANCE_SOMETIME_BETWEEN(*a*,*b*,*c*,*d*,*e*) is a predicate introduced in [17]. It is true iff the distance between moving object *a* and point location *b* is within *c* some time between *d* and *e*. In our example it is true iff the bus arrives at *P* some time between 10:08 and 10:10.

If a route #8 bus receives the query and it will wait, then the bus sends Bob an answer to the query.

5.2 Query Processing

We focus on remote query processing. A remote query from moving object *m* is processed in three steps. First, the trajectory of the querying moving object is attached to the query, so that the answering objects know where to return answers. As explained earlier, the trajectory defines the object's future location as a piecewise linear function from time to the two-dimensional geography. It may be constructed based on the shortest path between the origin and the destination of the object, and the traffic speeds on each road segment along the path. The origin and destination are provided, for example, by the car navigation system. In the second step, the query is disseminated from *m* to the moving objects in the query-destination-region (given in the REMOTE clause). Finally the answers are returned to *m*. We concentrate on the query dissemination step and the answer delivery step in the rest of this section.

Query dissemination. Simple flooding can always be used for query dissemination, but this may unnecessarily incur a high communication cost. For example, if the receiving object is moving away from the query destination region, then propagating the query to it may be wasteful. So the question is: when a moving object m_1 that is carrying a query encounters another moving object m_2, how m_1 decides whether to forward the query to m_2? The objective is to reach an optimal tradeoff between communication cost and accuracy of answers. We postulate that the decision should depend on the location and the moving direction of m_2 relative to the query-destination-region, the shape of the query-destination-region, the density of moving objects, and the budget of the query.

Answer Delivery. There can be several strategies to propagate the answer back to the query originator *m*. First, each moving object can send *m* the answers it is aware of; in turn, *m* consolidates the results (e.g. eliminates duplicates). The second possibility is that a leader is elected in the query-destination-region; the leader collects and consolidates the answers of the responding objects, before delivering them to *m*. The third possibility is a hybrid, hierarchical solution, in which leaders of small sub-areas propagate to leaders of larger areas.

6 Information Usage Strategies and Transactional Issues

Information Usage Strategies

When multiple consumers hear about the same competitive resource (such as a parking slot or a cab customer), they may all head to that resource, leading to contention. In order to address this phenomenon of ″herding″, a consumer needs to be selective when buying and acting on reports. In our prior work [1] we proposed an approach called Information Guided Searing (IGS) strategy to address this issue. In this approach, a consumer goes to a resource only when the relevance of the report is higher than an adaptive threshold. We compared by simulations the above information usage with the naive resource discovery approach where information is not used. The results showed that in some cases IGS cuts discovery time by more than 75%. We are studying strategies for using information to capture (i.e. reach before other competitors) geospatially distributed resources.

Transactional Issues

The transaction between two peers consists of a handshake initiation that includes the types of resources each one is interested in consuming/brokering, followed by the report exchange and coin charge/credit for each report. Observe that these operations must be executed as a distributed atomic transaction. For example, the credit of one account should be committed only if the debit of the other account is committed; and in turn, this should occur if and only if the corresponding report was received properly. Therefore, the transaction must be followed by a commit protocol. The problem is that, due to the high mobility at which the transaction occurs, the commit protocol between two peers may not begin or may not complete.

We propose to resolve this problem by a Mobile Peer-to-Peer Transaction (MOPT) mechanism which is a combination of an audit trail (or log) maintained online in the security module, and a central bank to which the audit trails of all peers are transmitted periodically, e.g. once a day. Our proposed MOPT mechanism has an online component that executes at the security module for each transaction, and an offline component.

The online component of MOPT at a security module *S* performs the following functions. It keeps a log of the reports that have been exchanged and the credit/debit charged for each one. The records of this log correspond to the log records in database transaction recovery. When a transaction completes unsuccessfully, then the user of *S* is still charged and can use the reports it received, and gets credit for the reports it (thinks it) sold. So if a broker *B* sent a report to a consumer *C*, but didn't receive the commit message, it still gets (temporary) credit.

The offline component of MOPT, at the end of the day sends to a central bank the logs of the transactions that completed unsuccessfully during the day. After receiving all the logs from all the peers, the central bank does the following for the transactions that completed unsuccessfully at one or both participants (thus it ignores transactions that completed successfully at both participants). If the same transaction completed unsuccessfully at both participants, then the traces from the respective security modules are used to settle the credit/charge to both accounts. In the example above, if *C* didn't receive the report, *B*'s credit will be reversed. If the transaction completed unsuccessfully at only one of the participants, i.e. the transaction is absent from the other security module trace, this fact indicates how the account at the unsuccessful participant should be settled. In this case, in the example above, *B*'s credit will be made permanent.

Observe that our MOPT mechanism needs to remember only the logs of unsuccessfully completed transactions, but can forget successfully completed transactions. Considering that peers may execute thousands of transaction per day, this is an important property.

Observe that this offline banking mechanism violates to some extent our principle of a completely decentralized economy. We will examine the framework/principles that can be enforced for a given level of decentralization. For example, assume that it is tolerable that occasionally peers may receive reports without paying, and some other peers may transmit resources without being paid. However, the system should provide integrity for the total amount of virtual currency in the system, namely virtual currency should not be lost or created. What is the maximum amount of decentralization allowed by this framework? Can the central bank be eliminated by doing so? In other words, we consider the semantic properties of our mobile peer-to-peer application to enable maximum decentralization; and this distinguishes our research from the extensive body of existing work on transactions/serializability issues.

7 Relevant Work

Traditional peer-to-peer approaches

A traditional peer-to-peer approach like Gnutella [20] could be used to search spatio-temporal resources, the problem addressed in this paper. In Gnutella, a query for a resource type (expressed by key words) is flooded on the overlay network (within predefined hops), and replies are routed back to the querying node along the same path as the query message. In other words, resource information is *pulled* by the querying node from the resource producer. This generates two problems in our context. First, since resources are transient and consumers do not know when they are generated, a consumer will have to constantly flood its query in order to catch resource information. Second, this does not work if there is not a path between the querying node and the resource producer. In our approach, a resource report is *pushed* by the resource producer to consumers via opportunistic dissemination and the dissemination area is automatically bounded by information prioritization. Gridella [12] and DHT systems such as Chord [14] have similar problems as Gnutella in that they use a *pull* model. In addition, Gridella and DHT systems require that the complete identifier (or key) of the searched data item be provided in a query, whereas in our case a consumer does not know a priori the keys of the searched resources.

Resource discovery and data dissemination in mobile distributed environments

Resource discovery and data dissemination in mobile distributed environments have been repeatedly studied (see e.g. [3, 16, 8]). Some use the gossiping/epidemic paradigm [16, 8] which is similar to our OP2P approach. All this work considers dissemination of regular data items rather than spatio-temporal information. None of them discusses information prioritization and incentive mechanisms.

Static sensor networks

A database approach has been applied to static sensor networks in Cougar [15], TinyDB [13], and direct diffusion [2]. All these methods require that a certain graph structure such as a tree be established in the network such that each node aggregates the results returned by its downstream nodes and its own result, and forwards the aggregation result to its upstream node. However, in our environment, due to the dynamic and unpredictable network topology, such a graph structure is hard to maintain. Our distributed query processing relies on opportunistic interactions between mobile nodes and therefore is totally different than Cougar and TinyDB.

Incentive mechanisms for P2P and MANET

Our economic model, including virtual currency, security module, and consumer-paid policy, is inspired by the work of Buttyan and Hubaux [5] on stimulating packet forwarding in MANET. In their work, a node receives one unit of virtual currency for forwarding a message of another node, and such virtual currency units (nuglets) are deducted from the sender (or the destination). In our model, however, the amount of virtual currency charged by an intermediary node (broker) for forwarding a report is proportional to the expected benefit of the report, the latter depending on the dynamic spatio-temporal properties of the report (age and distance) as well as various system environmental parameters.

To the best of our knowledge, our work is the first one that attempts to quantify the relevance of spatio-temporal information and to price based on the benefit of information to the consumer rather than the cost of forwarding it. This distinguishes our work from many other incentive mechanisms (see e.g. [9, 10]) which concentrate on compensating forwarding cost in terms of battery power, memory, CPU cycles. In a vehicular network such cost is negligible.

8 Conclusion

In this paper we devised a platform for dissemination of spatial and temporal resource-information in a mobile peer-to-peer network environment, in which the database is distributed among the moving objects. The moving objects also serve as routers of queries and answers. The platform includes spatio-temporal resource data model, database maintenance via opportunistic peer-to-peer interactions, relevance evaluation for information prioritization, query language and query processing, economical model that provides incentive for peers to participate as information suppliers and intermediaries, information usage strategies, and transaction management.

In general, we feel that the P2P paradigm is a tidal wave that has tremendous potential, as Napster and Gnutella have already demonstrated for entertainment resources. Mobile P2P is the next step, and it will revolutionize dissemination of spatial and temporal resources. For example, location based services have been considered a hot topic for quite some time, and it has been assumed that they have to be provided by a separate commercial entity such as the cellular service providers. The approach outlined in this paper can provide an alternative that bypasses the commercial entity.

References

1. O. Wolfson, B. Xu, Y. Yin, Dissemination of Spatial-Temporal Information in Mobile Networks with Hotspots, *DBISP2P* 2004.
2. C. Intanagonwiwat, et al. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. *Proceedings of MobiCOM*, 2000.
3. A. Helmy. Efficient Resource Discovery in Wireless AdHoc Networks: Contacts Do Help. Book Chapter in *Resource Management in Wireless Networking* by Kluwer Academic Publishers, May 2004.
4. http://firechief.com/ar/firefighting_roborescuers_increase_disaster/
5. L. Buttyan and J.P. Hubaux. Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks. ACM/Kluwer Mobile Networks and Applications (MONET), 8(5), October 2003.
6. A. Pfitzmann, B. Pfitzmann, and M. Waidner. Trusting Mobile User Devices and Security Modules. *IEEE Computer*, February 1997.
7. A. Markowetz, et al. Exploiting the Internet As a Geospatial Database, *International Workshop on Next Generation Geospatial Information*, 2003.
8. M. Papadopouli et al. Effects of Power Conservation, Wireless Coverage and Cooperation on Data Dissemination Among Mobile Devices. *MobiHoc 2001*, Long Beach, California.
9. S. Zhong, et al. Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks. In *Proceedings of IEEE INFOCOM* 2003.
10. R. Krishnan, et al. The economics of peer-to-peer networks, Carnegie Mellon University, 2002.
11. K. Yoon and C. Hwang. *Multiple Attribute Decision Making: An Introduction*. Sage Publications, 1995.
12. K. Aberer, et al. Improving Data Access in P2P Systems, *Internet Computing*, 6(1), 2002.
13. J. Hellerstein, et al. Beyond Average: Toward Sophisticated Sensing with Queries. *The Second International Workshop on Information Processing in Sensor Networks*, 2003.
14. I. Stoica, R. Morris, et al. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. In *Procs. ACM SIGCOMM*, 2001.
15. Y. Yao, J. Gehrke. Query Processing in Sensor Networks. *First Biennial Conference on Innovative Data Systems Research*, 2003.
16. K. Rothermel, C. Becker, and J. Hahner. Consistent Update Diffusion in Mobile Ad Hoc Networks. *Technical Report 2002/04*, CS Department, University of Stuttgart, 2002.
17. M. Vazirgiannis, O. Wolfson. A Spatiotemporal Query Language for Moving Objects. *Proceedings of the 7th International Symposium on Spatial and Temporal Databases*, 2001.
18. T. Michael. *Machine Learning*. McGraw-Hill, 1997.
19. O. Wolfson, B. Xu, P. Sistla. An Economic Model for Resource Exchange in Mobile Peer-to-Peer Networks. *Proceedings of SSDBM 2004*.
20. Gnutella website. http://gnutella.wego.com
21. S. Madden, et al. *TinyDB: In-Network Query processing in TinyOS*, Intel Research, October 15, 2002.
22. B. Wilcox-O'Hearn. Experiences Deploying a Large Scale Emergent Network. *International Workshop on Peer-toPeer Systems*, 2002.