# ABIDE: A Bid-based Economic Incentive Model for Enticing Non-cooperative Peers in Mobile-P2P Networks

Anirban Mondal[1] Sanjay Kumar Madria[2] Masaru Kitsuregawa[1]

[1] Institute of Industrial Science
University of Tokyo, JAPAN
{anirban, kitsure}@tkl.iis.u-tokyo.ac.jp

[2] Department of Computer Science
University of Missouri-Rolla, USA
madrias@umr.edu

**Abstract.** In mobile ad-hoc peer-to-peer (M-P2P) networks, a large percentage of the mobile peers typically do not provide any service to the network, thereby motivating incentive schemes for enticing non-cooperative mobile peers to provide service. We propose ABIDE, a novel bid-based economic incentive model for enticing non-cooperative mobile peers to provide service in M-P2P networks. The main contributions of ABIDE are three-fold. First, it encourages relay peers to act as brokers for performing value-added routing (i.e., pro-actively search for query results) due to bid-based incentives. Second, it integrates newly joined peers in the system seamlessly by sharing the loads with the neighbouring brokers. This helps the new peers to earn revenues in order to be able to obtain services. Third, it considers both effective data sharing and resource sharing among the peers. ABIDE also considers quality of service, load, energy and network topology. Our performance study indicates that ABIDE is indeed effective in increasing the number of service-providers in M-P2P networks, thereby improving query response times and data availability.

## 1 Introduction

In a Mobile Ad-hoc Peer-to-Peer (M-P2P) network, mobile peers (MPs) interact with each other in a peer-to-peer (P2P) fashion. Proliferation of mobile devices (e.g., laptops, PDAs, mobile phones) coupled with the ever-increasing popularity of the P2P paradigm [13] strongly motivate M-P2P network applications. Some application scenarios, which would facilitate mobile users in sharing information with each other *on-the-fly* in a P2P manner, are as follows:

– A pedestrian could issue a request for an available taxi.
– A mobile user could look for plumbing services or book appointments with doctors concerning non-emergency medical services.
– A car driver could search for a restaurant nearby his current location or he could request traffic information about how to go from point A to point B.

Such P2P interactions among mobile users are generally not freely supported by existing wireless communication infrastructures. The inherently ephemeral nature of M-P2P environments suggests that *timeliness* of data delivery is of paramount importance in these applications. For example, if a pedestrian looking for an available taxi receives an answer after 20 minutes have already elapsed since he issued the query, he may no longer find the answer to be useful. Furthermore, *data quality* is also a major concern e.g., a car driver requesting traffic information of a few miles ahead from other car drivers would be interested in obtaining data from a driver, whose traffic data has been updated recently. In the same vein, a mobile user requesting an image could be interested in a high-resolution image. Notably, our application scenarios do not require an absolute threshold of data quality, hence we also consider tolerance to lower data quality, depending upon users' requirements.

Incidentally, existing incentive schemes [21, 22] for M-P2P networks do not address the issue of creating pro-active mobile peers to provide value-added routing service. Moreover, they do not entice the non-cooperative peers in providing service (e.g., providing data to other MPs) to the network by allowing load-sharing so that peers can generate revenues, thereby encouraging seamless participation of peers in the system. Moreover, the existing schemes in [21, 22] deal with data dissemination, while we consider on-demand services. Notably, most peers in P2P systems do not provide any data [6, 9, 12, 18]. (Nearly 90% of the peers in Gnutella [20] were free-riders [1].) Increased MP participation in providing service to the network would lead to better data availability, likely better data quality, higher available bandwidth and multiple paths to answer a given query. Moreover, these schemes do not provide incentives to the relay MPs for *pro-actively* searching for query results or even for simply forwarding queries. Given the typically limited energy resources of the MPs and the fact that relaying messages requires energy, the relay MPs may not always be willing to forward the queries in the absence of any incentives, let alone search pro-actively for query results.

The role of the relay peers becomes even more important in case of M-P2P networks due to frequent network partitioning arising from user movement and/or users switching 'on'/'off' their mobile devices. Thus, it is of paramount importance to ensure that at least those MPs, which have connectivity, actually perform their relay tasks to ensure that most of the data in the network is reachable from most of the MPs. Furthermore, the ephemeral nature of M-P2P networks suggests that queries should generally be forwarded quickly by the relay MPs to ensure timeliness of data delivery. In the absence of incentives, the relay MPs may not necessarily forward the queries quickly. Furthermore, existing schemes do not consider the issue of data quality, which is of considerable importance for M-P2P users.

Given the requirement of timeliness in answering queries, relay MPs should pro-actively perform *value-added routing* by trying to identify the paths in which the query result could be found quickly and maintain the freshness of the paths. Hence, we propose ABIDE (A BID-based Economic model), which is a novel bid-based incentive model for enticing non-cooperative relay peers to participate in providing service in M-P2P networks. We designate our proposed model as *'ABIDE'* because as we shall see later, every MP would benefit in terms of obtaining better service, if they *abide* by the model.

In ABIDE, an MP may provide *'service'* by providing data to other MPs, performing value-added routing by pro-actively searching for targetted peers for query results and deploying its resources to perform computational tasks for others (e.g., an MP may issue a request for converting a certain file to a PDF file format). Each service in ABIDE is associated with a *price* (in terms of a *virtual currency*). ABIDE requires a service-requesting MP to pay the *price* of the service to the service-providing MP, thereby encouraging MPs to become service-providers. As a single instance, a user requesting a data item would need to pay the price of the data item to the MP serving its request. Data item price depends upon several factors such as access frequency, data quality and estimated response time for accessing the data item. Similarly, when an MP $M$ deploys its resources to perform computational tasks, the price of such resource sharing depends on the energy consumption of $M$.

In our bid-based model, brokers collect bids from data/service providers and then create a summary of recommendation based on the query preferences specified by the users. Based on the bids and the application, users selects a single bid, depending upon the price that a user wants to pay. Broker-based bidding model also protects the privacy of the requester and the bidder. After a bid is accepted, the requesting peers can either directly request the data from the service-providing peer or they can obtain the data using the same broker. In the latter case, the privacy of peers is maintained and all future services can only be performed using the brokers. In the former case, the requesting peer and service-providing peer can negotiate a better price for future services as the broker's commission may be reduced.

In ABIDE, the relay MPs maintain indexes of the services available at other MPs such as data stored at those MPs or computational tasks that can be performed by those MPs. The index at different MPs could be different. Using its index, a relay MP can act as a *broker* to pro-actively search for targetted peers for query results. The service-requesting MP needs to pay a *broker's commission* (based on bidding) to the relay MPs, which act as brokers, thereby encouraging them to pro-actively search for query results.(If the relay MP's index does not contain any information concerning the queried service, it selectively forwards the query to its neighbours to earn a relay commission.) Moreover, brokers could cache the paths of frequently queried services, thereby reducing the communication traffic for querying. In the absence of such brokerage, queries would always need to be broadcast (which would flood the network) because there would be little incentive for any MP to cache the paths associated with frequently queried services. Furthermore, a broker MP may also replicate data items that are frequently queried in order to reduce the traffic.

ABIDE also facilitates load-sharing among the MPs as follows. When a broker MP $M$ becomes overloaded with too many requests, it transmits its index to relay MPs, who are willing to store its index. We shall designate such relay MPs as **sub-brokers**. $M$ identifies the sub-brokers by sending a message to its neighbours. Observe that newly joined peers (which are likely to have zero revenue) and existing relay peers would be willing to store the replica of $M$'s index because it would provide them an opportunity to earn some revenue by performing broker-related functions using $M$'s index replicated at themselves. Thus, they would be able to actively participate in the network and obtain better service from the network. In essence, the system dynamically creates brokers and sub-brokers based on load and network performance to effectively convert non-cooperative relay MPs into broker MPs.

We define the **revenue** of an MP as the difference between the amount of virtual currency that it earns (by providing services) and the amount that it spends (by requesting services). ABIDE provides an incentive for MPs to provide service to the network so that they can earn more in order to be able to issue their own requests for services. The main contributions of ABIDE are three-fold:

1. It encourages relay peers to act as brokers and sub-brokers for performing value-added routing (i.e., pro-actively search for query results) due to bid-based incentives.
2. It integrates newly joined peers in the system seamlessly by sharing the loads with the neighbouring brokers. This helps the new peers to earn revenues in order to be able to obtain services.
3. It considers both effective data sharing and resource sharing among the peers.

ABIDE also considers quality of service, load, energy and network topology. Our performance study indicates that ABIDE is indeed effective in increasing the number of service-providers in M-P2P networks, thereby improving query response times and data availability.

## 2 Related Work

Economic models have been discussed in [5, 8, 14] primarily for resource allocation in distributed systems. A competitive micro-economic auction-based bidding model with support for load-balancing has been proposed in [5]. The proposal in [8] uses game-theoritic and trust-based ideas. The work in [14] examines economy-based optimal file allocation. Incidentally, none of these works address the unique issues associated with the M-P2P environment such as frequent network partitioning and mobile resource constraints. Moreover, they do not address free-riding and incentives for peer participation.

Works concerning free-riding include [6, 7, 9, 12, 15, 16, 18]. P2P-related free-riding has been discussed in [6]. The works in [7, 12, 16] propose incentive schemes to combat free-riding. The works in [9, 18] discuss utility functions to capture user contributions, while trust issues are examined in [15]. However, these works do not consider economic models and brokerage to combat free-riding.

Incentive mechanisms for static peer-to-peer networks have been discussed in [17]. However, predefined data access structures (e.g., distributed hash tables and searching routing tables), which are used for static P2P networks [20], are too static in nature to be practically viable for mobile ad-hoc networks. As a single instance, distributed hash tables [19] are not adequate for M-P2P networks because they assume the peers' availability and fixed topology since they are designed for static P2P systems. In essence, these data access structures have not been designed to handle mobility of peers and frequent network partitioning, which are characteristic of mobile ad-hoc networks. Incentive mechanisms have also been investigated for mobile ad-hoc networks [3, 4, 23], the main objective being to encourage a mobile peer in forwarding information to other mobile peers. However, the works in [3, 4, 23] do not consider brokerage model, bids and M-P2P architecture. Data replication has been discussed for mobile ad-hoc networks [10], but without considering incentives and prices of data items.

Economic ideas in the context of M-P2P networks have been discussed in [22, 21]. While the proposal in [22] addresses issues concerning spatio-temporal data in M-P2P networks, the work in [21] proposes opportunistic dissemination of data in M-P2P networks, the aim being to ensure that the data reaches more people. In contrast, we disseminate data on-demand because transmitting data to MPs, who may not actually require the data, significantly taxes the generally limited energy resources of the MPs. Furthermore, the proposals in [22, 21] do not consider brokerage and bidding issues.

## 3 Data and Resource sharing in ABIDE

Each MP maintains recent read-write logs (including timestamps) of its own data items and the read-logs of the replicas stored at itself. As we shall see shortly, each MP uses this information for computing the prices of the data items and replicas stored at itself. In ABIDE, each data item $d$ is owned by only *one* MP, which can update $d$ *autonomously* anytime; other MPs cannot update $d$. Memory space of MPs, bandwidth and data item sizes may vary. **Load** $L_{i,j}$ of an MP $M_i$ at time $t_j$ equals ( $J_{i,t_j}/B_i$ ), where $J_{i,t_j}$ represents the job queue length of $M_i$ at time $t_j$. Since job queue length is a function of time, load is also a function of time. $B_i$ is the normalized value of the available bandwidth of $M_i$. $B_i$ = ( $B_{M_i}$ / $B_{min}$ ), where $B_{M_i}$ represents the available bandwidth of $M_i$ and $B_{min}$ is a low bandwidth e.g., we have used $B_{min}$ = 56 Kbps.

Each query in ABIDE is either a request for a data item or a request for a computational task. Queries are of the form ($Q_{id}$, $\tau_S$, $\tau_H$, $\epsilon$), where $Q_{id}$ is the unique identifier of the query, while $\tau_S$ and $\tau_H$ are the user-specified soft and hard deadlines for answering the query. The significance of $\epsilon$ is that the query issuing MP stops accepting bids after $\epsilon$ time units have elapsed since the time of query issue (see Section 4). Given that a query $Q$ for a request $S$ is issued at time $t_0$, if $Q$ is answered within time ($t_0$ + $\tau_S$) (i.e., within the soft deadline), the query issuing MP $M_I$ pays the price $\mu$ of $S$ to the query serving MP $M_S$. However, if $Q$ is answered within the time interval [$t_0$ + $\tau_S$, $t_0$ + $tau_S$ + $tau_H$], $M_I$ pays a reduced price for $S$ to $M_S$, thereby penalizing $M_S$ for delayed service. As we shall see later, the value of the reduced price depends upon the time delay after the soft deadline $\tau_S$ i.e., more delay implies more reduction in price. Finally, if $Q$ is answered after the hard deadline $\tau_H$, $M_I$ does not pay any currency to $M_S$. Notably, such deadlines for answering queries are necessary due to the inherently ephemeral nature of the M-P2P environment because queries, which are answered after a certain threshold of time has already elapsed, are generally not useful to the user.

**Sharing data items in ABIDE:** In ABIDE, each data item $d$ has a *price* $\mu$ (in terms of a *virtual currency*) that quantitatively reflects its relative importance to the M-P2P network. We assume that there could be one original version of $d$ and multiple replicas of $d$ stored at different MPs. When an MP issues a query for a data item $d$, it pays the price of $d$ to the MP serving its request. The price $\mu$ of $d$ depends upon $d$'s (recent) access frequency, average query response times (w.r.t. deadlines) for queries on $d$ and data quality of $d$. An MP $M_S$ computes the price of a data item (or replica) $d$ stored at itself in two steps: (a) $M_S$ first computes the price $\mu_{rec}$ of $d$ based on accesses to $d$ during the most recent time period. (We divide time into equal intervals called *periods*, the size of a period being application-dependent.) (b) $M_S$ computes the moving average price $\mu$ of $d$ based on the previous $N$ time periods. The moving average price is necessary to take spurious spikes in accesses to $d$ into consideration to ensure that $d$'s price actually reflects $d$'s importance. $M_S$ computes $\mu_{rec}$ of $d$ as follows:

$$\mu_{rec} = \int_{t_1}^{t_2} \int_{0}^{\delta} ( \; \eta \;\; dt \times (1/\delta^2) \, d\delta \times \tau \times DQ \times BA_{M_S} \times PA_{M_S} \; ) \; / \; J_{M_S,t_j} \qquad (1)$$

where [$t_2 - t_1$] represents a given time period and $\delta$ is the distance between the query issuing MP $M_I$ and the query serving MP $M_S$ (i.e., the MP which stores $d$ and serves the query on $d$). Given that the positions of $M_I$ and $M_S$ during the time of query issue[1] are $(x_I, y_I)$ and $(x_S, y_S)$ respectively, $\delta$ =

[1] We assume that the positions of $M_I$ and $M_S$ do not change significantly between the time of query issue and the time of query retrieval.

$\sqrt((x_S - x_I)^2 + (y_S - y_I)^2)$ i.e., $\delta$ is *Euclidean distance*. Observe how $\mu_{rec}$ decreases as $\delta$ increases. This is because when the distance between $M_I$ and $M_S$ increases, the response time for queries on $d$ also increases, hence $d$'s price should decrease. In Equation 1, $\eta$ is the access frequency of the given data item $d$ during the most recent time period. $\tau$ reflects the price reduction (i.e., penalty) due to delayed service. Given that $t_0$ is the time of query issue, and $t_q$ is the time when the query results reached the query issuing MP, $\tau$ is computed as follows.

$$\begin{aligned} \tau &= \mu && if\ \ t_0 \ \geq \ t_q \ \geq \ (\ t_0 + \tau_S\ ) \\ &= \mu \times e^{-(t_q - \tau_S)} && if\ \ (\ t_0 + \tau_S\ ) \geq \ t_q \ \geq (\ t_0 + \tau_S + \tau_H\ ) \\ &= 0 && otherwise \end{aligned} \tag{2}$$

where $\tau_S$ and $\tau_H$ are the soft and hard deadlines of a given query respectively. $DQ$ reflects the quality of data provided by $M_S$ for queries on $d$. $DQ$ is essentially application-dependent. For example, for applications in which image sharing is involved, image resolution would determine data quality. Similarly, for applications in which (replica) consistency is of considerable importance, data quality should be based on data consistency. In general, each MP maintains a copy of the table $T_{\epsilon,DQ}$, which contains the following entries: (x%, high), (y%, medium), (z%, low), where x, y, z are error-bounds, whose values are application-dependent and pre-specified by the system at design time. Essentially, we consider three discrete levels of $DQ$ i.e., *high*, *medium* and *low*, and their values are 1, 0.5 and 0.25 respectively.

In Equation 1, $BA_{M_S}$ is the bandwidth allocated by $M_S$ for $d$'s download. $BA_{M_S}$ equals $(\sum B_i)/n_d$, where $B_i$ is the bandwidth that $M_S$ allocated for the $i^{th}$ download of $d$ from itself during the most recent time period, while $n_d$ is the number of downloads of $d$ from $M_S$. As $BA_{M_S}$ increases, $\mu_{rec}$ increases because higher bandwidth implies reduced response times for queries on $d$. $PA_{M_S}$ is the probability of availability of $M_S$. When $PA_{M_S}$ is high, the implication is that other MPs can rely more on $M_S$ to provide $d$, hence $\mu_{rec}$ increases with increase in $PA_{M_S}$. $J_{M_S,t_j}$ is the job queue length at $M_S$ during time $t_j$. $\mu_{rec}$ decreases with increase in the job queue of $M_S$ because when $M_S$ is overloaded with too many requests, $M_S$'s response time in answering queries on $d$ can be expected to increase.

After computing $\mu_{rec}$, $M_S$ computes the moving average price $\mu$ of $d$. Notably, we use the Exponential Moving Average (EMA), which is capable of reacting quickly to changing access patterns of data items since it gives higher weights to recent access patterns relative to older access patterns. This is consonance with the dynamically changing access patterns that are characteristic of M-P2P networks. $M_S$ computes the price $\mu$ of $d$ as follows:

$$\mu = (\mu_{rec} - EMA_{prev}) \times 2/(N+1)) + EMA_{prev} \tag{3}$$

where $EMA_{prev}$ represents the EMA that was computed for the previous time period, and $N$ represents the number of time periods over which the moving average is computed. Our preliminary experiments suggest that $N$ = 5 is a reasonably good value for our application scenarios.

An MP $M_S$ earns virtual currency from accesses to its own data items and replicas of others that are stored at itself, and through sharing its computational power with others. $M_S$ spends currency when it queries for services stored at other MPs. The revenue of an MP $M$ is simply the difference between the amount of virtual currency that $M$ earns and $M$ spends. When an MP joins the M-P2P network for the first time, it has zero currency, hence it first needs to serve other MPs' requests or share some load with neighbouring MPs and in lieu, earn some revenues before it can start issuing its own queries, thereby preventing free-riding. Observe how ABIDE's economy-based paradigm of load-sharing, and replication of data and indexes encourages MPs to increase their revenues, thereby ensuring that they obtain better service from the M-P2P network.

**Sharing computational power of MPs in ABIDE:** ABIDE also facilitates the sharing of computational power among the MPs. Sharing of computational power becomes important because many users may not have the software for running an application that they need to run. For example, an MP may issue a query requesting a service to convert a certain file format into the PDF format, or to convert from one song format to another, or to run any application on a different platform. Note that there could be multiple MPs that are capable of performing the same computational task. As in the case of data sharing in ABIDE, the service-requesting MP $M_I$ pays the price of the service to the service-providing MP $M_S$. The price $\mu_C$ of a computational power sharing service is determined by the amount of energy expended by the service-providing MP for performing the requested computational task. The service-providing MP $M_S$ computes $\mu_C$ as follows:

$$\mu_C = (\ CPU_{cycles} \times E \times \tau \times B_{M_S} \times PA_{M_S}\ )\ /\ J_{M_S,t_j} \tag{4}$$

where $CPU_{cycles}$ is the number of CPU cycles required by $M_S$ to perform the computational task, while $E$ is the energy needed per cycle. Incidentally, $E$ is device-dependent and is fixed for a given device. As a single instance, the MICA2 sensor device uses 4 nano-Joules per cycle [11]. In Equation 4, the significance of $\tau$ is the same as in Equation 1 (the case for data sharing) i.e., $\tau$ reflects the penalty due to delayed service. Consequently, $\tau$ is computed by means of Equation 2. $B_{M_S}$ refers to the bandwidth allocated by $M_S$ to transmit the results of the computational task to $M_I$. Finally, $PA_{M_S}$ and $J_{M_S,t_j}$ have the same significance as discussed for Equation 1.

## 4 Value-added routing by relay MPs in ABIDE

This section discusses value-added routing by the relay MPs in ABIDE. We shall henceforth refer to a query issuing MP and a service-providing MP as $M_I$ and $M_S$ respectively.

**Basic model of ABIDE:** ABIDE provides an incentive to the relay MPs to pro-actively search for the query results as opposed to just forwarding queries. Each MP maintains an index of the services (i.e., data items stored at other MPs and computational tasks that other MPs are capable of performing.) This index is built by each MP on-the-fly in response to queries that are issued to it. Hence, different MPs have different indexes. An MP $M_I$ issues a query $Q$ using a broadcast mechanism. When any given MP receives the broadcast query, it checks its index. If its index does not contain the identifier of at least one MP that is associated with the query result, it just forwards the query to earn a small amount of revenue as the relay commission. Otherwise, it acts as a broker by issuing a new query for finding the route to locate MPs that can answer the query.

Incidentally, the broker MP's commission is significantly higher than that of the relay MP's commission, which encourages a larger number of non-cooperative relay MPs to index more services, thereby providing them with a higher likelihood of being able to act as brokers. Broker MPs also cache paths for frequently requested services. Hence, after the system has run for a certain period of time, the need for broadcasting queries can be expected to be significantly reduced. A broker MP may also replicate data items that are frequently queried in order to reduce the querying traffic. A given service-providing MP $M_S$ may also allow a broker MP to store a replica of some of its 'hot' data items. In this manner, even if $M_S$ is disconnected, it can still earn revenues. Notably, this also leads to better data availability.

ABIDE also facilitates load-sharing among broker MPs and relay MPs as follows. When a broker MP $M$ becomes overloaded[2] with too many requests, it sends a message to its neighbours to enquire which of its neighbouring relay MPs would be willing to store a replica of its index. $M$'s neighbouring relay MPs, which are willing to store a replica of $M$'s index, become the sub-brokers of $M$. The incentive for these sub-brokers to store a replica of $M$'s index is that they would be able to earn revenue by performing broker-related functions using $M$'s index replicated at themselves. This would facilitate newly joined MPs and existing relay MPs to seamlessly integrate themselves in the system by actively participating in the network. This effectively converts non-cooperative relay MPs into broker MPs.

Once a given broker MP obtains the route to one or more MPs that can serve the query, it acquires information about the price of the service at each of these MPs. Thus, the broker MP stores information of the form ($S$, $MP_{id}$, $\mu$, $Path$), where $S$ is the service being requested, $MP_{id}$ is the unique identifier

[2] A broker MP considers itself to be overloaded when its capacity utilization is 60% of its maximum capacity.

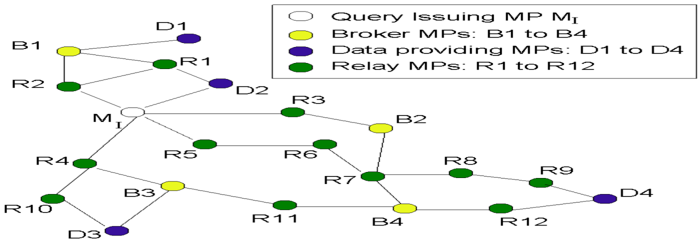



**Fig. 1.** Illustrative example of an instance of network topology

of the MP that can serve the query, and $\mu$ is the price of $S$. $Path$ is simply a linked list data structure containing the list of MPs, which fall in the path between the broker MP and the service-providing MP. In case of multiple paths between the broker MP and the service-providing MP, $Path$ could be a pointer to a set of linked lists (or a two-dimensional array).

**Illustrative example for the network topology in ABIDE:** Figure 1 depicts an illustrative example of the topology of the M-P2P network at a certain point of time. In this example, assume that a data item $d$ is being requested as service. Using Figure 1, we shall now make certain key observations concerning the network topology in ABIDE. As indicated in Figure 1, the query issuing MP $M_I$, broker MPs $B1$ to $B4$, the service-providing MPs (i.e., $M_S$) $D1$ to $D4$ and the relay MPs $R1$ to $R12$ are indicated by the white, yellow, blue and green circles respectively. Suppose $D1$ to $D4$ all contain some copy of $d$ albeit possibly with varying quality of data. Observe that the number of relay nodes between $M_I$ and a broker MP can vary. For example, the path $\{M_I, R2, B1\}$ has only one relay MP, while the path $\{M_I, R5, R6, R7, B4\}$ has three relay MPs. Furthermore, the number of relay MPs between broker MPs and a given data providing MP $M_S$ can vary e.g., the number of relay MPs in the paths $\{B4, R12, D4\}$ and $\{B2, R7, R8, R9, D4\}$ are 1 and 3 respectively. Thus, the number of hops in the path from $M_I$ to a given $M_S$ can differ.

Interestingly, it is also possible for a given $M_S$ to be a one-hop neighbour of $M_I$ e.g., $M_I$ and $D2$ are one-hop neighbours. However, some other $M_S$ such as $D1$ may be able to provide better data quality and/or lower response time than $D2$ (e.g., due to low bandwidth between $D2$ and $M_I$). Hence, the role of the broker MPs would still be relevant in such cases. In essence, the broker MPs provide $M_I$ with different paths for accessing $M_I$'s requested data item $d$ or its replica. This allows $M_I$ to choose the copy of $d$, which best suits $M_I$'s requirements in terms of response time and data quality.

In Figure 1, observe that there can be multiple paths from $M_I$ to the same $M_S$ and these paths may pass through different brokers. As a single instance, $D4$ may be accessed by means of multiple paths such as $\{M_I, R3, B2, R7, R8, R9, D4\}$ and $\{M_I, R5, R6, R7, B4, R12, D4\}$ and $\{M_I, R4, B3, R11, B4, R12, D4\}$. Incidentally, it is possible for a path between $M_I$ and a given $M_S$ to have multiple brokers e.g., the path $\{M_I, R4, B3, R11, B4, R12, D4\}$ contains two brokers, namely $B3$ and $B4$. In such cases, the broker that occurs first in the traversal starting from $M_I$ (i.e., $B3$ in this example) would make the bid, while the other brokers (i.e., $B4$) in the path would only act as relay MPs. This is necessary to avoid conflicts among brokers.

**Privacy considerations in ABIDE:** Based on the way in which the query result is sent from a given service-provider $M_S$ to a query issuing MP $M_I$, we define two auction models, namely the *Privacy-Preserving Auction model (PPA)* and the *Non-Privacy preserving Auction model (NPA)*. In PPA, the query result is sent via the broker MP, thereby ensuring that the data providing MP $M_S$ and the query issuing MP $M_I$ remain anonymous to each other. Thus, PPA has the advantage of preserving the privacy of both $M_S$ and $M_I$. However, given that there could be multiple paths between $M_S$ and $M_I$, it is possible that there exists a shorter path than the path via the given broker MP. The likelihood of the existence of the shorter route between $M_S$ and $M_I$ can be intuitively expected to increase with the number of routes between $M_S$ and $M_I$. Thus, PPA could incur relatively higher communication overhead in sending the query results.

**Algorithm *ABIDE_Query_Issuing_MPs***
Inputs: (a) $Q$: Query (b) $d$: Queried data item
(1) Broadcast its query Q for a data item d
(2) Receive all the bids that arrive at itself within $\epsilon$ time units of issuing the query
(3) Evaluate the score $\gamma$ for each bid
(4) Select the bid for which the value of $\gamma$ is highest and **select** the corresponding broker MP $Sel$
(5) Send message to **selected** broker MP $Sel$
(6) if ModelType is PPA
(7) Receive the data item from the selected broker MP $Sel$
(8) Send the broker commission to the selected broker MP $Sel$
(9) else if ModelType is NPA
(10) Receive the route to the selected $M_S$ from the broker MP
(11) Obtain data item from the selected $M_S$
(12) Send the broker commission to the selected broker MP $Sel$
**end**

**Fig. 2.** ABIDE algorithm for Query Issuing MP $M_I$

On the other hand, NPA requires that the broker MP should reveal the identity of $M_I$ and $M_S$ to each other, as well as the route between $M_S$ and $M_I$. Hence, in NPA, the query results do not need to pass via the broker MP. While NPA could potentially lower the communication cost between $M_S$ and $M_I$, it does not ensure the confidentiality of $M_S$ and $M_I$ since it does not preserve privacy. We believe that choosing whether to use PPA or NPA is not only application-dependent, but also depends upon the privacy requirements of the user. Thus, we allow the user to specify in the query whether he wishes to use PPA or NPA. Notably, in both PPA and NPA, $M_I$ pays the commission to the broker MP after the query results have reached $M_I$. It is possible for a malicious $M_I$ to avoid paying the commission to the broker MP. In such cases, the broker MP blacklists $M_I$ and informs its neighbours regarding the malicious behaviour of $M_I$, thereby deterring $M_I$ from indulging in such malicious behaviour.

**Algorithms in ABIDE:** Figure 2 depicts the algorithm executed by a query issuing MP, while Figure 3 indicates the algorithm executed by the other MPs, which can either be broker MPs or relay MPs. For the sake of convenience, we describe the algorithms of ABIDE from the perspective of data sharing. However, these algorithms also hold good for sharing of computational power. As Lines 1-2 of Figure 2 indicate, the query issuing MP $M_I$ broadcasts[3] its query and waits until $\epsilon$ time units have elapsed (since the time of query issuing) to collect the bids from all the brokers. Then $M_I$ determines which bid to accept by computing a score $\gamma$, based on the estimated query response time and the data quality (see Line 3). $M_I$ computes $\gamma$ as follows.

$$\gamma \ = \ a \times RT + b \times DQ \tag{5}$$

where $RT$ and $DQ$ represent the estimated query response time and data quality respectively. The values of $RT$ and $DQ$ are provided to $M_I$ by the broker MP. In Equation 5, $a$ and $b$ are weight coefficients which determine the relative weights of $RT$ and $DQ$, such that $0 \leq a, b \leq 1$ and $a + b = 1$. The values of $a$ and $b$ must be specified by the user because different users have different preferences concerning the relative importance of query response time and data quality essentially due to varying user requirements. In Equation 5, $DQ$ is computed in the same manner as discussed for Equation 1. $RT$ equals the data item size divided by the sum of the bandwidths at the intermediate hops between $M_S$ and $M_I$.

[3] After a period of time, if $M_I$ knows a broker MP that can serve the query, broadcast would not be necessary.

**Algorithm *ABIDE_Brokers_and_Relay_MPs***

Inputs: (a) $Q$: Query (b) $d$: Queried data item

(1) Receive the broadcast query $Q$ for data item $d$ from query issuing MP $M_I$

(2) Check own index to list the identifier of all the MPs that store $d$ into a set $Set_{M_S}$

(3) if $Set_{M_S}$ is empty

(4) Forward $Q$ to its one-hop neighbours

(5) else

(6) for each $M_S$ $M$ in $Set_{M_S}$

(7) Issue a query to find the route(s) to $M$

(8) List all the routes from itself to $M$ into a set $Set_{Route}$

(9) if $Set_{Route}$ is empty

(10) Forward $Q$ to the one-hop neighbours

(11) else

(12) Select the shortest route $R$ from itself to $M$ based on bandwidths at the intermediate hops

(13) Obtain price and data quality information from $M$

(14) Collate all the price, $M_S$, response time and data quality information with the value of its bid $\beta$, and send to $M_I$

(15) Wait for $M_I$'s reply

(16) if $M_I$ accepts bid

(17) Obtain identifier of **selected** $M_S$ from $M_I$

(18) if ModelType is PPA

(19) Obtain data item from $M_S$ and send data item to $M_I$

(20) Receive broker commission from $M_I$

(21) else if ModelType is NPA

(22) Send a message to **selected** $M_S$ to send the data item to $M_I$

(23) Receive broker commission from $M_I$

**end**

**Fig. 3.** ABIDE algorithm for broker MPs and relay MPs

$M_I$ selects the bid with highest value of $\gamma$, and selects the broker MP $Sel$ who made that bid (see Line 4). As Lines 5-12 indicate, $M_I$ initiates conversation with selected broker to obtain the query results using either the PPA model or the NPA model. Lines 6-12 in Figure 2 and Lines 18-23 in Figure 3 indicate how ABIDE works differently for the PPA model and the NPA model. In essence, query results must pass through the broker MP for the PPA model, while for the NPA model, the query results are transmitted from the query serving MP to the query issuing MP via the route suggested by the broker MP.

The algorithm in Figure 3 is executed by MPs, which are either broker MPs or relay MPs. As indicated by Lines 3-14, if the index of a given MP contains the identifier of the queried data item, it acts as a broker, otherwise it just forwards the query. In Line 14, observe that different brokers may bid different amounts of currency for the same data item (or its replica). The amount $\beta$ of currency that a broker MP bids depends upon the quality of the data item that it is able to provide and the estimated response time for the query issuing MP $M_I$ to receive the data item. Given a data item $d$ of price $\mu$, a given broker MP computes $\beta$ as ( $\mu \times \alpha$ ), where $\alpha$ is a percentage of the data item price, hence $0 \leq \alpha \leq 1$. $\alpha$ depends upon the urgency of $M_I$. Thus, we compute $\alpha$ as $e^{-\tau_S}$, where $\tau_S$ is the soft deadline of the query. Observe that increase in $\tau_S$ implies decrease in $\beta$ due to less urgency.

## 5 Performance Evaluation

This section discusses our performance evaluation. In our experiments, MPs move according to the *Random Waypoint Model* [2] within a region of area 1000 metres $\times$1000 metres. The *Random Waypoint*

*Model* is appropriate for our application scenarios, which involve random movement of users. As a single instance, pedestrians (calling a taxi) generally move randomly i.e., they do not follow any specific mobility pattern. A total of 200 data items are uniformly distributed among 50 MPs i.e., each MP owns 4 data items. Each query is a request for a data item. In all our experiments, 20 queries/second are issued in the network, the number of queries directed to each MP being determined by the Zipf distribution. Communication range of all MPs is a circle of 100 metre radius. Table 1 summarizes the parameters used in our performance evaluation.

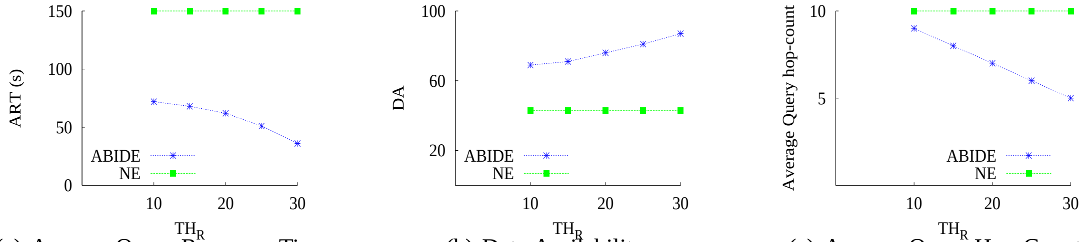
| Parameter | Default value | Variations |
|---|---|---|
| No. of MPs ($N_{MP}$) | 50 | |
| Zipf factor (ZF) | 0.9 | 0.1, 0.3, 0.5, 0.7 |
| Queries/second | 20 | |
| Bandwidth between MPs | 28 Kbps to 100 Kbps | |
| Probability of MP availability | 50% to 85% | |
| Size of a data item | 50 Kb to 350 Kb | |
| Memory space of each MP | 1 MB to 1.5 MB | |
| Speed of an MP | 1 metre/s to 10 metres/s | |
| Size of message headers | 220 bytes | |

**Table 1.** Performance Study Parameters

Our performance metrics are **average response time** (**ART**) of a query, **data availability** (**DA**) and **average querying traffic**. ART equals ( $(1/N_Q)\sum_{i=1}^{N_Q}(T_f - T_i)$ ), where $T_i$ is the time of query issuing, $T_f$ is time of the query result reaching the query issuing MP, and $N_Q$ is the total number of queries. $DA$ is computed as ( $(N_S/N_Q)\times 100$ ), where $N_S$ is the number of queries that were answered successfully and $N_Q$ is the total number of queries. In ABIDE, queries can fail because MPs, which store queried data items, may be unavailable due to being switched 'off' or owing to network partitioning. Average querying traffic is the average number of hops required for query processing in ABIDE. Incidentally, none of the existing proposals for M-P2P networks address economic auction-based revenue models. Hence, as reference, we adapt a non-economic model **NE**, in which querying occurs by means of the broadcast mechanism. NE does not provide any incentive for the MPs to contribute to the M-P2P network. NE does not perform replication and it does not cache query paths. Notably, the performance of the PPA model and the NPA model were comparable in all our experiments, hence here we present the performance of ABIDE w.r.t. the PPA model.

**Effect of variations in the number of MPs above threshold revenue:** Threshold revenue $TH_R$ is defined as the ratio of the total revenue of the system to the total number of MPs. In other words, $TH_R$ is the average revenue in the system. Figure 4 depicts the results concerning the effect of variations in the number of MPs above $TH_R$. The results indicate that when the revenue of more MPs exceed $TH_R$, ART decreases and data availability increases. This is due to more MPs participating in providing service as their revenues increase, thereby implying more memory space for holding data items and replicas and more available bandwidth. Moreover, increase in the number of MPs acting as brokers and sub-brokers provide multiple paths for locating a given queried data item. Thus, ABIDE outperforms NE essentially due to the economic incentive nature of ABIDE (which encourages higher MP participation) and load-sharing among brokers and sub-brokers. NE shows relatively constant ART and DA since NE is independent of revenue. The presence of brokers and sub-brokers also reduces the number of hops required for accessing data items because they maintain index of data items and they cache the paths of frequently queried data items, which explains the results in Figure 4c.

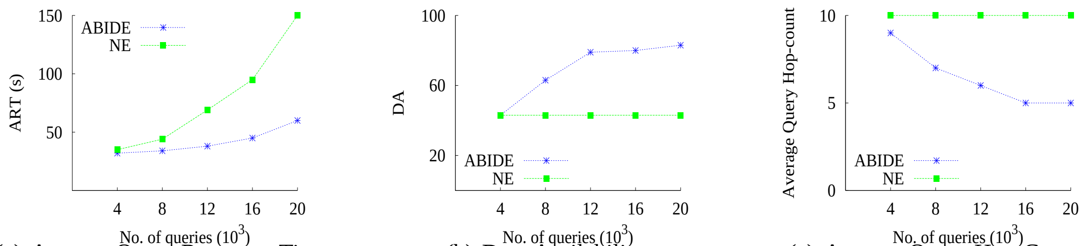
**Performance of ABIDE:** We conducted an experiment using default values of the parameters in Table 1. Figure 5a indicates that the ART of both ABIDE and NE increases with time due to the skewed workload ($ZF$ = 0.9), which overloads some of the MPs that store 'hot' data items, thereby forcing queries to incur high waiting times and consequently high ART. However, over time, more MPs start




(a) Average Query Response Time (b) Data Availability (c) Average Query Hop-Count

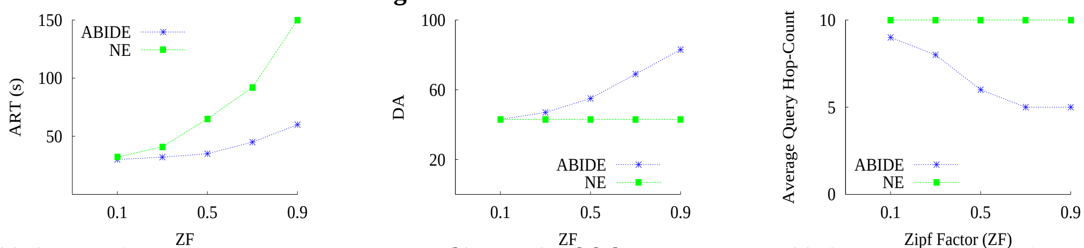
**Fig. 4.** Effect of revenue threshold

participating as brokers and sub-brokers in case of ABIDE, thereby providing more memory space and more bandwidth for replication of ‘hot’ data items, which facilitates load-balancing. This explains the increasing performance gap between ABIDE and NE in terms of ART and DA. In Figure 5b, DA eventually plateaus due to reasons such as network partitioning and unavailability of some of the MPs. Furthermore, unlike ABIDE, NE does not maintain the cached routes to the ‘hot’ data items and it does not perform replication, hence ABIDE outperforms NE in terms of query hop-counts. The query hop-counts decrease over time for ABIDE essentially due to replication at the brokers and sub-brokers, and path caching.




(a) Average Query Response Time (b) Data Availability (c) Average Query Hop-Count

**Fig. 5.** Performance of ABIDE




(a) Average Query Response Time (b) Data Availability (c) Average Query Hop-Count

**Fig. 6.** Effect of variations in the workload skew

**Effect of variations in the workload skew:** Figure 6 depicts the results when the zipf factor (ZF) is varied. The performance gap between ABIDE and NE decreases with decreasing skew since lowly skewed workloads do not necessitate replication. For high ZF values (i.e., high skew), ABIDE significantly outperforms NE in terms of ART and DA due to more replications performed by the brokers and the sub-brokers in response to load-imbalance conditions. Moreover, ABIDE exhibits lower query hop-counts at high skews essentially due to replication at the brokers and sub-brokers, and path caching.

## 6 Conclusion

We have proposed ABIDE, a novel economic bid-based incentive model for enticing non-cooperative mobile peers to provide service in M-P2P networks. The main contributions of ABIDE follow. First, it encourages relay peers to act as brokers for performing value-added routing (i.e., pro-actively search for query results) due to bid-based incentives. Second, it integrates newly joined peers in the system seamlessly by sharing the loads with the neighbouring brokers. This helps the new peers to earn revenues in order to be able to obtain services. Third, it considers both effective data sharing and resource sharing among the peers. ABIDE also considers quality of service, load, energy and network topology. Our performance study indicates that ABIDE is indeed effective in increasing the number of service-providers in M-P2P networks, thereby improving query response times and data availability.

## References


1. E. Adar and B. A. Huberman. Free riding on Gnutella. *Proc. First Monday*, 5(10), 2000.
2. J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocol. *Proc. MOBICOM*, 1998.
3. L. Buttyan and J. Hubaux. Nuglets: a virtual currency to stimulate cooperation in self-organized mobile ad hoc networks. *Technical Report DSC/2001/001, Swiss Federal Institute of Technology,Lausanne*, 2001.
4. L. Buttyan and J.P. Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. *Proc. ACM/Kluwer Mobile Networks and Applications*, 8(5), 2003.
5. D.F. Ferguson, Y. Yemini, and C. Nikolaou. Microeconomic algorithms for load balancing in distributed computer systems. *Proc. ICDCS*, pages 491–499, 1988.
6. M. Fischmann and O. Gunther. Free riders: Fact or fiction? Sep 2003.
7. P. Golle, K.L. Brown, and I. Mironov. Incentives for sharing in peer-to-peer networks. *Proc. Electronic Commerce*, 2001.
8. C. Grothoff. An excess-based economic model for resource allocation in peer-to-peer networks. *Proc. Wirtschaftsinformatik*, 2003.
9. M. Ham and G. Agha. ARA: A robust audit to prevent free-riding in P2P networks. *Proc. P2P*, pages 125–132, 2005.
10. T. Hara and S.K. Madria. Data replication for improving data accessibility in ad hoc networks. *To appear in IEEE Transactions on Mobile Computing*, 2006.
11. Crossbow Technology Inc. Mpr400/410/420 mica2 mote. datasheet 2005.
12. S. Kamvar, M. Schlosser, and H. Garcia-Molina. Incentives for combatting free-riding on P2P networks. *Proc. Euro-Par*, 2003.
13. Kazaa. http://www.kazaa.com/.
14. J. F. Kurose and R. Simha. A microeconomic approach to optimal resource allocation in distributed computer systems. *Proc. IEEE Trans. Computers*, 38(5):705–717, 1989.
15. S. Lee., R. Sherwood, and B. Bhattacharjee. Cooperative peer groups in NICE. *Proc. INFOCOM*, 2003.
16. N. Liebau, V. Darlagiannis, O. Heckmann, and R. Steinmetz. Asymmetric incentives in peer-to-peer systems. *Proc. AMCIS*, 2005.
17. First Workshop on the Economics of P2P Systems. http://www.sims.berkeley.edu/research/conferences/p2pecon. 2003.
18. L. Ramaswamy and L. Liu. Free riding: A new challenge to P2P file sharing systems. *Proc. HICSS*, page 220, 2003.
19. I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. *Proc. ACM SIGCOMM*, 2001.
20. Gnutella website. http://gnutella.wego.com/.
21. O. Wolfson, B. Xu, and A.P. Sistla. An economic model for resource exchange in mobile Peer-to-Peer networks. *Proc. SSDBM*, 2004.
22. B. Xu, O. Wolfson, and N. Rishe. Benefit and pricing of spatio-temporal information in Mobile Peer-to-Peer networks. *Proc. HICSS*, 2006.
23. S. Zhong, J. Chen, and Y.R. Yang. Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. *Proc. IEEE INFOCOM*, 2003.