Computer Communications 36 (2013) 861–874

Contents lists available at SciVerse ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Economic incentive-based brokerage schemes for improving data availability in mobile-P2P networks

Nilesh Padhariya [a], Anirban Mondal [a], Sanjay Kumar Madria [b,*], Masaru Kitsuregawa [c]

[a] Indraprastha Institute of Information Technology, Delhi, India
[b] Missouri University of Science and Technology, Rolla, USA
[c] University of Tokyo, Japan

ARTICLE INFO

Article history:
Received 23 September 2011
Received in revised form 19 January 2013
Accepted 26 January 2013
Available online 14 February 2013

Keywords:
Economic incentive schemes
Brokerage
Data availability
Mobile-P2P networks

ABSTRACT

In mobile ad hoc peer-to-peer (M-P2P) networks, data availability is typically low due to rampant free-riding, frequent network partitioning and mobile resource constraints. This work proposes the E-Broker system for improving data availability in M-P2P networks. The main contributions of E-Broker are three-fold. First, it proposes the EIB (Economic Incentive-based Brokerage) scheme, which incentivizes relay peers to act as information brokers for performing value-added routing and replication in M-P2P networks, thereby effectively improving data availability. Second, it proposes the EIB+ (enhanced Economic Incentive-based Brokerage) scheme, which extends the EIB scheme by incorporating three different broker scoring strategies for providing additional incentives to brokers towards providing better service. Moreover, EIB+ facilitates load-sharing among the peers. Third, it experimentally determines the number of brokers, beyond which the mobile peers are better off without a broker-based architecture i.e., they can directly access data from the data-providing peers. Our performance evaluation indicates that the proposed schemes are indeed effective in improving query response times, data availability and query hop-counts at reasonable communication traffic cost in M-P2P networks as compared to a recent existing scheme.

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 [1]. Proliferation of mobile devices (e.g., laptops, PDAs, mobile phones) coupled with the ever-increasing popularity of the P2P paradigm (e.g., Kazaa, Gnutella) strongly motivate M-P2P network applications, which facilitate MPs in sharing information *on-the-fly*. For example, an application could involve an MP looking for an available parking slot within 1 km of its current location. MPs in the vicinity can collect information about available parking slots and charges, and then they can inform the brokers. The broker can then provide the available parking slots to the query-issuing MP in terms of price or distance (from the user's current location). Note that the parking slot availability information has to be current. Incidentally, although we consider brokers, the nature of the networking environment is still ad hoc in the sense that the peers can move and they can change their brokers. Hence, the presence of brokers does not make our environment completely structured.

In a similar vein, a user could look for a restaurant with "happy hours" (or "manager's special hours") within 1 km of her current location. A broker can facilitate such queries by soliciting information from the peers moving in the vicinity of the query location. Similarly, an MP may want to find nearby shops selling Levis jeans in a shopping mall with criteria such as (low) price during a specific time duration. Observe that such ad hoc queries are spatio-temporal in nature (e.g., parking slot availability information), hence they cannot be answered by the broker without obtaining information from other MPs. Incidentally, such P2P interactions, which facilitate *spatio-temporal* queries among MPs, are generally not freely supported by existing wireless communication infrastructures. Notably, this research will also contribute towards CrowdDB [2], which uses human input via crowdsourcing to process queries that cannot be answered by database systems or search engines.

Our target applications mainly concern slow-moving objects e.g., mobile users in a shopping mall. Notably, our application scenarios consider tolerance to lower data quality depending upon the requirements of the peers. We measure data quality in terms of image resolution or MP3 audio quality. Moreover, observe that the inherently ephemeral nature of M-P2P environments necessitates query deadlines.

* Corresponding author.
E-mail addresses: nileshp@iiitd.ac.in (N. Padhariya), anirban@iiitd.ac.in (A. Mondal), madrias@mst.edu (S.K. Madria), kitsure@tkl.iis.u-tokyo.ac.jp (M. Kitsuregawa).

0140-3664/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comcom.2013.01.012

Data availability in M-P2P networks is typically lower than in fixed networks due to frequent network partitioning arising from peer movement and also due to mobile devices being autonomously switched 'off'. Rampant free-riding further reduces data availability i.e., most peers do not provide any data [3,4]. (Nearly 90% of the peers in Gnutella were free-riders [5].) Incidentally, data availability is less than 20% even in a wired environment [6]. Given the generally limited resources (e.g., bandwidth, energy, memory space) of MPs and the fact that relaying messages requires energy, the relay MPs may not always be willing to forward queries in the absence of any incentives, let alone search *pro-actively* for query results in order to ensure timeliness of data delivery. Thus, providing incentives for relay MPs to pro-actively search for query results becomes a necessity to improve data availability in M-P2P networks. Notably, many schemes such as replication-based schemes, reward-and-punish-based schemes and trust-based schemes can also be used for improving data availability, but the focus in this paper is on incentive-based schemes. Observe that increased MP participation in providing service to the network would likely lead to better data availability, better data quality, higher available bandwidth and multiple paths to answer a given query.

Existing schemes for improving data availability in mobile ad hoc networks (MANETs) [7] focus on replication, but they do not use economic incentives to encourage peer participation. On the other hand, incentive schemes [8–11] for MANETs primarily focus on providing incentives to relay MPs to forward messages, but they do not address the issue of creating pro-active MPs for providing value-added routing services. M-P2P incentive schemes [12,13] also do not incentivize relay MPs to perform value-added routing and to host data.

This work proposes the E-Broker system for improving data availability in M-P2P networks by incentivizing MPs to provide *value-added routing service*. Here, the term "value-added routing service" refers to the broker MPs enabling pro-active search for the query results by maintaining an index of the data items (and replicas) stored at other MPs (as opposed to just forwarding queries). The main contributions of E-Broker are threefold:

1. It proposes the EIB (Economic Incentive-based Brokerage) scheme, which incentivizes relay peers to act as information brokers for performing value-added routing and replication in M-P2P networks, thereby effectively improving data availability.
2. It proposes the EIB+ (enhanced Economic Incentive-based Brokerage) scheme, which extends the EIB scheme by incorporating three different broker scoring strategies for providing additional incentives to brokers towards providing better service. Brokers with higher scores become*preferred brokers* and they earn higher commissions than *common brokers*. EIB+ also facilitates load-sharing among the peers.
3. It experimentally determines the number of brokers, beyond which the mobile peers are better off without a broker-based architecture i.e., they can directly access data from the data-providing peers.

E-Broker also discourages free-riding in M-P2P networks. Both EIB and EIB+ use economic incentives in that every data item is associated with a *price* (in *virtual currency*). Data item price depends upon several factors such as access frequency, data quality and estimated response time of access. The query-issuer pays the price of the queried item to the data-provider, and a commission to the broker and the relay MPs in the successful query path.

Both EIB and EIB+ use a bid-based brokerage approach, in which brokers collect bids from data-providers and then create a summary of recommendations based on the query preferences specified by the query-issuer M_I. Based on the bids and the application, M_I selects a single bid depending upon the price that it wants to pay and its desired data quality. After a bid is accepted, M_I obtains the requested data item directly from the data-provider and pays the commission to the broker. Brokers also replicate frequently queried data items to earn revenues as well as to reduce the traffic.

We have evaluated the performance of EIB and EIB+ w.r.t. the non-economic **E-DCG+**replication scheme [7]. Notably, E-DCG+ is the closest to our schemes since it aims at improving data availability in MANETs. As a baseline, we also do performance comparison w.r.t. a non-incentive and non-broker-based **NIB** (Non-Incentive without Brokerage) scheme to show the performance gain due to brokerage. We experimentally determine that EIB and EIB+ perform best when the percentage of brokers is 20% of the total number of MPs. Moreover, EIB+ performs best when the percentage of preferred brokers is 20% of the total number of brokers. Both EIB+ and EIB outperform E-DCG+ and NIB due to economic incentives and brokerage. EIB+ performs better than EIB due to preferred brokerage and load-sharing. Furthermore, E-DCG+ outperforms NIB due to its superior replication scheme.

Both EIB and EIB+ exhibit good scalability with increasing number of MPs due to increased opportunities for replication. However, their performance degrades with increasing percentage of MP failures essentially due to reduced MP participation. With increasing workload skew, EIB+ shows better performance than the other schemes primarily due to its load-sharing mechanism. A preliminary version of this paper has appeared in [14].

The remainder of this paper is organized as follows. Section 2 presents existing works, while Section 3 discusses economic incentives in the E-Broker system. Section 4 details the EIB brokerage scheme, while Section 5 discusses the enhanced brokerage scheme EIB+. Section 6 reports our performance study. Finally, we conclude in Section 7.

2. Related Work

This section provides an overview of existing works.

2.1. Replication-based schemes

Replication schemes for static P2P networks [15] do not consider peer mobility issues. P2P replication suitable for mobile environments has been incorporated in systems such as ROAM [16], and Clique [17]. However, these systems do not incorporate economic incentives.

The E-DCG+ [7] replication scheme creates groups of MPs that are biconnected components in a MANET, and shares replicas in larger groups of MPs. An RWR (read-write ratio) value in the group of each data item is calculated as a summation of RWR of those items at each MP in that group. In the order of the RWR values of the group, replicas of items are allocated until memory space of all MPs in the group becomes full. However, E-DCG+ does not consider economic incentives and M-P2P architecture.

2.2. Incentive-based economic schemes

Incentive schemes for combating free-riding in static P2P networks involve utility functions to capture peer contributions [3] and EigenTrust scores to capture participation criteria [4]. However, these approaches are too static to be deployed in M-P2P networks since they assume peers' availability and fixed network topology. The works in [18,19] focus on addressing the problem of free-riding in decentralized collaborative environments. In particular, these works propose a taxonomy for classifying and tracking free-riders in multimedia systems based on trustworthiness.

The proposal in [20] addresses free-riding in the popular eMule/eDonkey P2P file-sharing network by evaluating and improving the fairness policy, which rewards contributors. Notably, the works in [20,18,19,3,4] do not address replication and mobile resource constraints.

Schemes for incentivizing peers to forward messages in MANETs involve virtual currency to stimulate node cooperation and counter-based mechanisms at each peer [8]. The auction-based iPass [9] incentive scheme and the works in [10,11] also provide incentives for relaying messages. However, these works do not consider brokerage schemes and M-P2P architecture. Furthermore, they do not use incentives for data hosting and replication.

The work in [13] provides incentives to MPs for participation in the dissemination of reports about spatio-temporal resources in M-P2P networks. The work in [12] considers opportunistic resource information dissemination in transportation application scenarios. These works primarily address data dissemination with the aim of reaching as many peers as possible i.e., they focus on how every peer can get the data. In contrast, we consider on-demand services i.e., the query-issuing peer obtains only the data that it asks for. Thus, we use a query-based approach. Moreover, the proposals in [12,13] do not consider brokerage, replication and incentives for data hosting.

Our work in [1] proposes an economic incentive model for the efficient processing of constraint queries in M-P2P networks, given that M-P2P users may issue queries with varying constraints on query response time, data quality of results and trustworthiness of the data source. The focus in [1] is on how to index the constraints in user queries by using the CR*-tree. Furthermore, our work in [1] provides incentives for peers to form collaborative peer groups for maximizing data availability and revenues by mutually allocating and deallocating data items using royalty-based revenue-sharing. Thus, the focus in [1] is completely different from the focus of this paper in that this paper focuses on brokerage schemes for performing value-added routing and replication (and load-sharing) in M-P2P networks.

The work in [21] discusses resource allocation in distibuted systems by incorporating economy-based optimal file allocation. However, it does not address incentives and M-P2P architecture. Economic schemes for resource allocation in wireless ad hoc networks [22,23] do not consider replication and brokerage schemes. Moreover, unlike our data-centric focus, their emphasis is network-centric.

Payment schemes for mobile environments include coupon-based systems such as adPASS [24] and Coupons [25]. The incentive scheme in [25] is inspired by the eNcentive framework [26], which allows mobile agents to spread digital advertisements with embedded coupons in a mobile-P2P manner. To preserve the integrity of the e-coupon [9], public-key cryptography and digital-watermarking technology are used. MoB [27] is an open market collaborative wide-area wireless data services architecture, which can be used by mobile users for opportunistically trading services with each other. The works in [28–30] discuss how to ensure secure payments using a virtual currency. The payment schemes discussed above can be used in conjunction with our proposal to ensure secure payments.

2.3. Trust-based schemes

The proposal in [31] examines the role of recommenders in P2P systems with the objective of managing trust. In particular, it provides an in-depth treatment of the feedback behavior of the recommenders as well as their role in trust assessment for P2P systems. Non-repudiation systems [32] can also be incorporated to control the deceiving behavior of peers. The work in [33] discusses an experimental model for trust and cooperation for partner selection in social networks.

The work in [34] proposes a human-based model for building a trust relationship between nodes in an ad hoc network. In particular, it proposes the Recommendation Exchange Protocol (REP), which enables nodes to exchange recommendations about their neighbors. Trust is based not only on previous individual experiences, but also on the recommendations of other nodes. Nodes maintain and exchange trust information about nodes within their respective radio ranges.

Notably, the trust-based schemes discussed above can be used in conjunction with our proposal as countermeasures to the selfish and deceiving behaviors of the peers.

3. Economic Incentives in E-Broker

This section discusses the economic incentives in E-Broker. These incentives are used by both the EIB and EIB+ brokerage schemes. We defer the discussion of the brokerage schemes to Sections 4 and 5. Incidentally, each MP maintains recent access statistics of data items (and replicas) hosted at itself for the purpose of computing data item prices. We assume that there could be one original version of any given data item d and multiple replicas of d hosted at different MPs. Memory space of MPs, bandwidth and data item sizes may vary.

3.1. Querying-related incentive issues

Each query is a request for a data item. Queries are of the form $(Q_{id}, DDQ, \epsilon, \tau_S, \tau_H, max_\mu, BType, w_1, w_2, w_3)$, where Q_{id} is the unique identifier of the query, while DDQ represents the desired data quality of the query-issuer M_I. To satisfy query deadlines, M_I stops accepting bids from brokers after ϵ time units have elapsed since the time of query issue. (The significance of ϵ will become clear when we discuss our brokerage scheme in Section 4.) Here, τ_S and τ_H are M_I's specified soft and hard deadlines for answering the query. max_μ is the maximum price that M_I is willing to pay for the query.

BType is M_I's specified broker type for the query and assumes two values i.e., 0 for a common broker and 1 for a preferred broker. As we shall see later, in case of EIB+, *BType* can assume either value, but for EIB, *BType* always equals 0 since EIB does not consider preferred brokers. Here, w_1, w_2 and w_3 are the query-issuer's specified weight coefficients for the query such that $0 \leqslant w_1, w_2, w_3 \leqslant 1$ and $w_1 + w_2 + w_3 = 1$. As we shall see in 4, these weight coefficients pertain to query response time, data quality and data item price respectively, and they are used by the broker for computing the ranking scores for the data items in the query result set (See Eq. 4).

Given that a query Q for a data item d is issued at time t_0, if Q is answered within time $(t_0 + \tau_S)$ (i.e., within the soft deadline), M_I pays the price μ of d to the data-provider 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 d to M_S, thereby penalizing M_S for delayed service. Higher delay implies more reduction in price. Finally, if Q is answered after the hard deadline τ_H, M_I does not pay any currency to M_S. This is consistent with the timeliness requirements of M-P2P environments.

Observe that there is no incentive for a data-provider to answer a query after the deadline. Hence, data-providers estimate (based on past statistics concerning network history) whether their transmitted data item will reach the query-issuer within the deadline. Based on their estimate, they decide whether to send the data. Notably, such estimation requires synchronized clocks among the MPs. For example, if an MP receives a message with a timestamp, clock synchronization among the MPs would become a necessary condition for the MP to calculate the delay. The existing clock synchronization approaches proposed in [35] can be used in conjunction with our proposed approach. However, an MP cannot absolutely know in advance whether its answer will reach the query-issuer in a timely manner because of issues such as network congestion, relay node failures and network partitioning.

Incidentally, if an MP is not able to pay the price of accessing its requested data item, its query fails and it would not be able to access its queried data item. This is in consonance with our overall objective of incentivizing free-riders to provide replica hosting, brokerage and relay services. If our scheme allowed MPs to access data items without having to pay for the access, the free-riders would have little or no incentive to provide service.

3.2. Price of a data item

Each data item d has a *price* μ (in *virtual currency*) that quantitatively reflects its relative importance to the M-P2P network. When an MP issues a query for a data item d, it pays the price of d to the MP serving its request. (A query request could also be satisfied by a replica.).

The price μ 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 μ_{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 μ of d based on the previous N time periods. The moving average price is necessary to take spurious spikes in accesses to d into account to ensure that d's price actually reflects its importance. M_S computes μ_{rec} of d as follows:

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

where $[t_2 - t_1]$ represents a given time period, and δ is the number of hops between the query-issuer M_I and the data-provider M_S during the time of query issue. We assume that the query message maintains a counter that is incremented with each hop. Thus, M_S can know the number of hops between itself and M_I at the time of query issue by examining the query message. Furthermore, we assume that the number of hops between M_I and M_S does not change significantly between the time of query issue and the time of query retrieval. Observe how μ_{rec} decreases as δ increases due to likely increased query response times.

In Eq. 1, η is the access frequency of the given data item d during the most recent time period. τ 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, τ is computed as follows.

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

where τ_S and τ_H are the soft and hard deadlines of a given query respectively. Notably, the data-provider M_S estimates the time when the query results would reach the query-issuer M_I based on the average network conditions and historical information. Hence, in Eq. 1, the data item price is first estimated by M_S using the value of τ based on the estimation of the time when the query results would reach M_I. Payments are done periodically, and information concerning the actual times of query results reaching the respective query-issuers is piggybacked onto the status messages that are sent periodically by the peers to the brokers. Thus, the actual price, which is paid by M_I to M_S, is based on the actual time when the query results reached a given query-issuer.

The term DQ in Eq. 1 reflects the quality of data provided by M_S for queries on d. DQ is essentially application-dependent. For example, data quality could be determined based on MP3 audio quality or image resolution. We compute DQ as follows. Each MP maintains a copy of the table T_{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 Eq. 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, μ_{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 μ_{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. μ_{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 due to longer waiting times of queries.

After computing μ_{rec}, M_S computes the moving average price μ of d. 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 in consonance with the dynamically changing access patterns that are characteristic of M-P2P networks. M_S computes the price μ 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.

3.3. Revenue of an MP

The revenue of an MP M is the difference between the amount of virtual currency that M earns and M spends. M earns virtual currency from accesses to its own data items and replicas that are hosted at itself, and through relay and broker commissions. Conversely, M spends currency when it queries for data items hosted at other MPs.

We incorporate commissions to incentivize relay MPs. Relay commission is a constant k. We use the price μ_{min} of the cheapest data item in the network as a guide to determining a suitable value of k. The value of k is selected to be lower than that of μ_{min} to incentivize data sharing more than relay functions. Observe that the value of μ_{min} could change over time because new items could be introduced into the network. However, based on the application, it is feasible to estimate the value of μ_{min}. Thus, the value of k is essentially application-dependent. We defer the discussion of broker commissions to Section 4.

Notably, every MP joining the system is provided with a small initial amount of currency for bootstrapping the system. Observe that the MPs would soon exhaust this initial amount of currency by issuing queries, and by paying the data item prices and relay commissions. Hence, after that, they would have to earn currency for issuing their own requests, and they can earn currency only by hosting items and relaying messages, thereby effectively combating free-riding. Observe how our economy-based paradigm encourages MPs to increase their revenues, thereby ensuring that they obtain better service.

4. EIB: An economic incentive-based brokerage scheme for M-P2P networks

This section discusses our proposed EIB scheme.

4.1. Role of the brokers in EIB

EIB provides an incentive to the relay MPs to act as brokers by pro-actively searching for the query results as opposed to just forwarding queries. A broker obtains a commission for each query processed successfully through itself. Hence, each MP is incentivized to maintain an index of the data items (and replicas) stored at other MPs. This index is built by each MP on-the-fly in response to queries and data that it relays. Hence, indexes may differ across MPs. Brokers also provide value-added service in EIB by replicating frequently queried data items at themselves.

Notably, the mobile peers participating in the system have software installed in their mobile devices, and this software enables them to use the proposed schemes. Once they use this software, they have to follow our architecture i.e., they have to go through the brokers. Thus, when using the software, a selfish query-issuer cannot contact the data provider directly by bypassing the brokers. In this regard, the rationale behind our architecture (i.e., every query must pass through brokers) is that query-issuing peers would not want to evaluate a large number of replies coming from prospective data-providers. Moreover, such evaluation would drain their limited energy resources. Furthermore, query-issuing peers would want to have more options (e.g., price, quality) about their requested data items, and the broker is in a position to provide such options.

A data-provider may allow a broker to host a replica of some of its 'hot' data items in lieu of a royalty payment. This is possible because we use our proposed royalty-based revenue-sharing scheme [1] in conjunction with EIB. Brokers have an incentive for hosting replicas of 'hot' items because they can earn revenue when those replicas are queried. Data-providers are incentivized to replicate their 'hot' items at brokers because they can earn revenue from accesses to the replicas without necessitating any expenditure of their limited energy resources for answering queries on those items. In this manner, even if a data-provider is disconnected, it can still earn revenues.

To perform replication, every data-provider periodically broadcasts a list of items that it wants to replicate. Brokers intercept this broadcast and decide whether to replicate these items based on their estimate about the future access frequencies and prices of those items. (This estimate is made based on the queries that pass through a broker.) Since brokers have limited memory space for hosting replicas, each broker tries to select only those items, which would maximize its revenue-earning potential. An item's revenue-earning potential is the product of its price and its (estimated) access frequency. Thus, EIB facilitates brokers in replicating frequently queried items, thereby reducing the querying traffic. In essence, EIB effectively converts relay MPs into brokers.

4.2. Illustrative example for the network topology in EIB

The architecture of EIB consists of query-issuers, relay MPs, brokers and data-providers. Fig. 1 depicts an illustrative example of the M-P2P network topology in EIB at a certain point in time. In Fig. 1, M_I is the query-issuer, $R1$ to $R7$ are the relay peers, $D1$ to $D4$ are the data-providers, and $B1$ to $B5$ are the brokers. Using Fig. 1, we shall now make certain key observations. Observe that there can be multiple paths from a query-issuer to a given data-provider and these paths may pass through multiple brokers. As a single instance, a query issued by M_I for a data item hosted by $D4$ could proceed through multiple paths such as $\{M_I, B2, B3, B4, R4, D4\}$ and $\{M_I, B2, B3, B4, R5, D4\}$.

Our scheme stipulates that only one MP can act as the broker in a given query path. This becomes a necessity to avoid conflicts among brokers. Hence, when multiple brokers exist in a given query path, the broker, which occurs first in the traversal starting from the query-issuer, would act as the broker for the query and make the bid to the query-issuer, while the other brokers would simply act as relay MPs. For example, in the query path $\{M_I, B2, B3, B4, R4, D4\}$, $B2$ would act as the broker since it occurs first in the traversal starting from M_I, while $B3$ and $B4$ would act as relay MPs. When an MP decides to act as the broker for a query, it appends a *broker tag* to the query message, thereby enabling other MPs in the same query path to determine that a broker has already been selected in that query path. Notably, even though EIB limits the number of brokers in a given query path to only one, the existence of multiple query paths safeguards against the unavailability of some of the brokers.

The number of relay MPs between a query-issuer and a data-provider may differ. For example, let us consider a query Q issued by M_I for a data item hosted by $D4$. In this case, the query path $\{M_I, B2, B3, B4, R4, D4\}$ has three relay MPs, namely $B3, B4$ and $R4$. On the other hand, the path $\{M_I, D2, B1, R2, B3, R3, R4, D4\}$ has five relay MPs, namely $D2, R2, B3, R3$ and $R4$. Thus, the total cost of relay commissions may vary across query paths since EIB incorporates a constant relay commission per relay MP, as discussed in Section 3.

It is also possible for a given data-provider to be a one-hop neighbour of a query-issuer e.g., M_I and $D2$ are one-hop neighbours. However, our architecture dictates that M_I cannot bypass the brokers for directly obtaining its queried data from $D2$. Recall that the mobile peers are able to use the proposed schemes by installing software in their mobile devices, and this software enforces that each query must follow our architecture by going through the brokers. Thus, the role of the brokers would still be relevant in such cases. For example, some other data-provider 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). In essence, the brokers provide the query-issuer with different paths for accessing its requested data item d or its replica. This allows the query-issuer to choose the copy of d, which best suits its requirements in terms of response time, data quality and price. Furthermore, as discussed earlier, there may be many prospective data-providers replying to a query, and the query-issuer would not want to evaluate a large number of replies since performing such evaluation would drain its limited energy resources.

4.3. Value-added routing by relay MPs in EIB

An MP M_I issues a query Q using a broadcast mechanism[1] and waits until ϵ time units have elapsed (since the time of query issue) to collect the bids from all the brokers. 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 hosts the requested data item or if another broker (in the same query path) has already decided to act as the broker for that query,[2] it simply forwards the query. Otherwise, it determines (from its index) the MPs, which can answer the query, and acts as a broker by issuing a route-finding query to locate these MPs.

Once a given broker obtains the route to one or more MPs that can serve the query, it acquires information about the price and data

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

[2] Recall that only one MP can be the broker in a given query path.

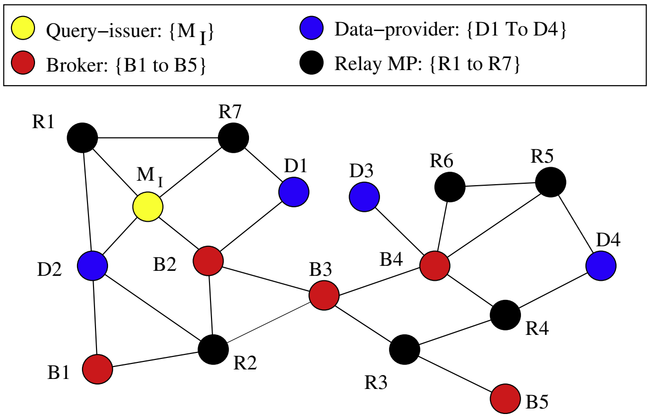


Fig. 1. Illustrative example of an instance of network topology in EIB.

quality of the requested data item at each of these MPs. Thus, the broker summarizes information of the form $(d, MP_{id}, \mu, DQ, Path)$ in a list L_{bid}, where d is the data item being requested, MP_{id} is the unique identifier of the MP that hosts d, DQ is the data quality of d and μ is the price of d. *Path* is a linked list data structure containing the list of MPs, which fall in the path between the broker and the data-provider. In case of multiple paths between the broker and the data-provider, *Path* could be a pointer to a set of linked lists (or a two-dimensional array).

Observe that if the broker were to include in its bid (to the query-issuer) all the data items about which it has acquired information, communication traffic would increase and the query-issuer would have to expend its limited energy resources to evaluate all the query results. On the other hand, if the broker were to include only one data item in its bid, the query-issuer would have limited choices (in terms of query results), which could potentially not satisfy its query requirements in terms of response time, data quality and price. Hence, the broker provides a value-added service by including in its bid only *some* of the data items about which it has acquired information. The broker determines which items it will include in its bid by using the information in the list L_{bid}. For each data item in the list L_{bid}, the broker computes a score γ:

$$\gamma = (w_1/RT) + (w_2 \times DQ) + (w_3/\mu) \tag{4}$$

where *RT* represents the query response time, which is estimated by the broker based on network statistics. *RT* is estimated by the data item size divided by the sum of the bandwidths at the intermediate hops between the query-serving MP and the query-issuer. *DQ* and μ are the data quality and price of the item respectively, and they are evaluated in the same manner as discussed for Eq. 1. In Eq. 4, w_1, w_2 and w_3 are the query-issuer's specified weight coefficients for the query such that $0 \leqslant w_1, w_2, w_3 \leqslant 1$ and $w_1 + w_2 + w_3 = 1$. Thus, EIB takes the requirement of the query-issuer into account.

The value of γ increases with decreasing values of *RT* and μ because the query-issuer would want the results quickly and with lower price. The value of γ increases with increase in *DQ* because higher data quality commands higher bid price. The broker includes in its bid (to the query-issuer) only those items, for which the value of γ exceeds the threshold Th_γ, where Th_γ is the average value of γ for all the items in L_{bid}. Hence, Th_γ equals $(\Sigma_{i=1}^{N}\gamma_i/N)$, where γ_i is the value of γ for the *i*th item and *N* is the total number of items in L_{bid}. The values of *RT* and *DQ* for each item in every bid are also provided by the broker to the query-issuer.

Corresponding to each data item included in the broker's bid, the broker also specifies the total cost of relay commissions and broker commission to inform the query-issuer about the total cost of querying. Since the broker knows the number of relay MPs in the query path, it can compute the total cost of relay commission since the amount of relay commission per MP is a constant, as discussed earlier in Section 3. The amount β of broker commission for a given data item d depends upon the data item price. Given a data item d of price μ, a broker computes β as $(\mu \times \alpha)$, where α is a percentage of the data item price, hence $0 \leqslant \alpha \leqslant 1$. The value of α depends upon the urgency of the query-issuer. Thus, we compute α as $e^{-\tau_S}$, where τ_S is the soft deadline of the query. Increase in τ_S implies decrease in β due to less urgency. Observe that different brokers may bid different amounts of currency for the same data item (or its replica). Incidentally, the broker's commission is significantly higher than that of the relay MPs' commissions, which incentivizes relay MPs to act as brokers by indexing more data items.

Upon receiving bids from possibly multiple brokers, the query-issuer *autonomously* evaluates each item in each of these bids. (Recall that each broker may send multiple items in its bid to provide the query-issuer with more options.) Then the query-issuer selects the item, which best suits its requirements in terms of the weight coefficients w_1, w_2 and w_3 corresponding to (estimated) response time, data quality and price respectively. In particular, EIB does not force a query-issuer to perform bid selection based on any specific algorithm. This is because we believe that query-issuers should be provided the flexibility to choose the item (in the bids) that best satisfies their requirements.

An example to illustrate a possible way in which a query-issuer could select an item from multiple bids is as follows. Suppose $w_1 > w_2 > w_3$. In this case, the query-issuer could first sort the items in all the bids in ascending order of estimated response time into a list L_{Select}. Then from L_{Select}, it could select only those items, whose estimated response time is lower than the average response time of all the items in L_{Select}. Then it could sort the remaining items in L_{Select} in descending order of data quality, and select only those items, whose data quality exceeds the average data quality of all the (remaining) items in L_{Select}. Finally, among the remaining items in L_{Select}, it could select the item with the lowest price.

Upon completion of the bid selection, the query-issuer contacts the broker corresponding to the successful bid, and requests it for the data item. The successful broker contacts the data-provider, which sends the data item to the query-issuer. Finally, upon receiving the query results, the query-issuer pays the commission to the broker and the relay commissions to the MPs in the successful query path.

Algorithm 1 is executed by a query-issuer, while Algorithm 2 is executed by the other MPs, which can either be brokers or relay MPs.

Algorithm 1. EIB algorithm for a query-issuer

begin
Inputs: (a) *Q*: Query (b) *d*: Queried data item
(1) Broadcast its query *Q* for a data item *d*
(2) Receive all bids that arrive within ϵ time units of query issue
(3) Examine each item in every bid and autonomously select the item,which best suits query requirement
(4) Select the broker *Sel* corresponding to the successful bid
(5) Send message to selected broker *Sel* requesting selected item and provide *Sel* with identifier of selected data-provider M_S
(6) Obtain data item from M_S
(7) Pay the price of the item to M_S
(8) Pay the broker commission to the selected broker *Sel*
(9) Pay relay commissions to relay MPs in successful query path
end

Algorithm 2. EIB algorithm for broker and relay MPs

begin
Inputs: (a) Q: Query (b) d: Queried data item
(1) Receive the broadcast query Q for data item d from query-issuer M_I
(2) if broker_tag not attached to Q
/* EIB stipulates one broker per query path */
(3) Check own index to list the identifiers of all MPs hosting d into a set Set_{M_S}
(4) if Set_{M_S} is empty
(5) Forward Q to its one-hop neighbours
(6) else
(7) for each M_S M in Set_{M_S}
(8) Issue a query to find the route (s) to M
(9) List all the routes from itself to M into a set Set_{route}
(10) if Set_{route} is empty
(11) Forward Q to its one-hop neighbours
(12) else
(13) Select the shortest route R from itself to M based on bandwidths at the intermediate hops
(14) Obtain price and data quality of d from M, and add d to a list L_{bid}
(15) Select from L_{bid} only those items, for which the value of γ exceeds Th_γ and include these items in the bid
(16) For each item included in the bid, collate all the price, M_S, response time and data quality information and the bid value β
/* The bid value β for a given data item is a percentage of the data item price. β is the broker commission for a successful bid. */
(17) Send the bid to M_I
(18) Wait for M_I's reply
(19) if M_I accepts bid
(20) Obtain identifier of selected M_S from M_I
(21) Send a message to selected M_S to send the data item to M_I
(22) Receive broker commission from M_I
end

5. EIB+: An enhanced economic incentive-based brokerage scheme for M-P2P networks

This section discusses the EIB+ scheme, which extends the EIB scheme by incorporating three broker scoring strategies for further incentivizing brokers towards providing better service. EIB+ distinguishes two different types of brokers, namely *common brokers* and *preferred brokers*. Brokers with higher scores become preferred brokers and they earn higher commissions than common brokers. Furthermore, only the preferred brokers are allowed to spawn sub-brokers for load-sharing purposes, thereby further incentivizing brokers since they can earn currency from royalty-based revenue-sharing [1] with the sub-brokers.

Notably, in order to become a preferred broker, a broker needs to serve a minimum threshold number of users. Thus, if a broker serves an adequate number of different users, the rating scores from different users average out, thereby implying that a broker cannot become a preferred broker by serving only one peer well because broker scores are based on averages. Even though we understand that it is difficult to synchronize the ratings for different brokers, peers can select in their region their preferred brokers. Furthermore, observe that complete synchronization of broker score ratings across different users is not practically feasible due to subjectivity in human judgment.

5.1. Illustrative example of network topology in EIB+

Fig. 2 depicts an illustrative example of the M-P2P network topology in EIB+ at a certain point in time. M_I is the query-issuer, $D1$ to $D4$ are the data-providers, $R1$ to $R4$ are the relay peers, $CB1$ to $CB3$ are the common brokers, and $PB1$ to $PB2$ are the preferred brokers. $SB1$ is the sub-broker corresponding to the preferred broker $PB1$, while $SB2$ and $SB3$ are the sub-brokers corresponding to the preferred broker $PB2$. Observe that the common brokers such as $CB1$ and $CB2$ do not have any sub-brokers. Consider a query Q issued by M_I for a data item hosted by $D4$. For the query path $\{M_I, PB1, CB2, PB2, SB3, D4\}$, if Q needs to be processed by a common broker, $CB2$ would act as the broker, while $PB1$ and $PB2$ would act as relay MPs. If Q needs to be processed by a preferred broker, $PB1$ would act as the broker.

The broker type (i.e., common or preferred) specified in Q should match with at least one broker in the given query path for it to be processed in that query path. This is in consonance with adhering to the query-issuer's intentions. However, this does not necessarily result in query failures due to the possible existence of multiple brokers (which match the broker type specified in Q) in different query paths. Thus, if M_I issues a query for an item in $D4$ with the condition that it should be processed by a preferred broker, it will not be processed in the path $\{M_I, D2, CB1, R2, CB2, R3, SB3, D4\}$ since this path does not contain any preferred broker. However, it would be processed in other query paths e.g., $\{M_I, PB1, CB2, PB2, SB2, D4\}$ and $\{M_I, PB1, CB2, R3, SB3, D4\}$.

5.2. Strategies for assigning scores to brokers

We propose three strategies for assigning performance-based scores to brokers in EIB+.

5.2.1. Individual Ranking (IR) strategy

In IR, each MP assigns a score λ to each broker, with whom it has interacted within a particular time-period. Each broker returns a bid to the query-issuer M_I, and the bid contains the estimated query response time, data quality (of query result) and the total bid price for processing the query. (Total bid price refers to the sum of data item price, broker commission and relay commissions.) M_I uses this bid information to compute the value of λ for the broker that made the bid. The value of λ is computed for both successful and unsuccessful bids.

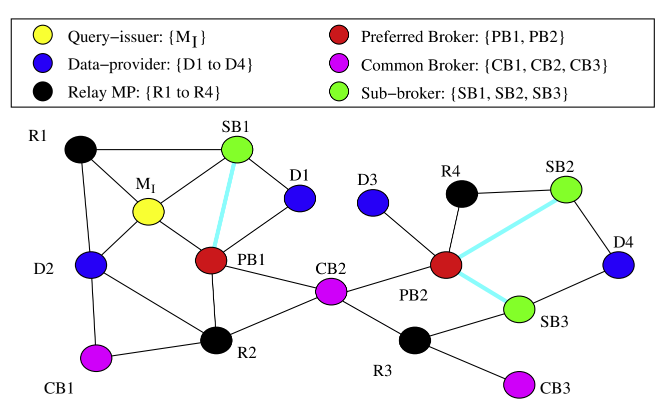


Fig. 2. Illustrative example of an instance of network topology in EIB+.

If a query is answered after the hard deadline τ_H, M_I assigns $\lambda = 0$ for that query to the corresponding broker (s) to penalize broker performance because queries answered after the deadline are not useful to M_I. Furthermore, since a broker has no incentive to bid a total price, which is higher than that of M_I's maximum specified price, the question of the total bid price exceeding the maximum specified price does not arise. λ is computed as follows:

$$\lambda = (w_1 \times \lambda_{RT}) + (w_2 \times (1 - \lambda_{DQ})) + (w_3 \times \lambda_\mu) \tag{5}$$

where $\lambda_{RT}, \lambda_{DQ}$ and λ_μ quantify broker performance w.r.t. broker response time, data quality and total (bid) price respectively, and they are computed in Eqs. (6)–(8) respectively. (Broker response time is the difference between the time of query issue and the time at which the broker's response arrives at M_I.) In Eq. 5, w_1, w_2 and w_3 are weight coefficients such that $w_1 + w_2 + w_3 = 1$. The values of w_1, w_2 and w_3 are decided by M_I for a given query depending upon its requirements. For example, if quick response time is critical to M_I, it will assign a high value to w_1. Observe how EIB+ provides autonomy to the MPs in assigning scores to brokers based on their individual querying requirements. λ_{RT} is computed below:

$$\lambda_{RT} = (\tau_H - RT) \,/\, \tau_H \tag{6}$$

where τ_H and RT are the hard deadline and the broker response time of the query respectively. Observe that the value of λ_{RT} increases as RT decreases. Thus, the objective of Eq. 6 is to reward brokers for providing timely service. The amount of reward is based on the difference between the hard deadline of the query and the broker response time. The computation of λ_{DQ} follows:

$$\lambda_{DQ} = \begin{cases} (DDQ - DQ)/DDQ & \textit{if } DQ < DDQ \\ 1 & \textit{otherwise} \end{cases} \tag{7}$$

where DDQ and DQ are M_I's specified desired data quality and the actual data quality for the query respectively. The objective of Eq. 7 is to penalize brokers, which provide lower quality of data than that of M_I's desired data quality. The amount of penalty is based on the difference between M_I's desired data quality and the actual data quality provided by the broker. The value of λ_{DQ} increases as queries are answered with lower data quality, hence we use the value of $(1 - \lambda_{DQ})$ in Eq. 5 for the computation of λ. However, when $DQ \geqslant DDQ$, we set $\lambda_{DQ} = 1$ to reward brokers, who have performed up to (or better than) M_I's expectations of data quality.

The computation of λ_μ follows:

$$\lambda_\mu = (max_\mu - \mu) \,/\, max_\mu \tag{8}$$

where max_μ and μ are the M_I's specified maximum price and the total price bid by the broker for the query respectively. Observe that the value of λ_μ increases as the total bid price decreases. Thus, the objective of Eq. 8 is to reward brokers, which can serve the queries at lower total price. The amount of such reward is based on the difference between M_I's maximum specified price and the total bid price of the given query. Thus, an MP will have an estimate about the performance of the brokers that it has interacted with. However, IR suffers from the drawback that each MP is likely to be able to interact with and assign scores to only a few brokers that are in its vicinity.

5.2.2. Neighbour-based gossiping (NGS) strategy

To address the drawback of IR in terms of being able to assign scores to only a relatively few brokers, we propose the NGS strategy. In NGS, MPs gossip with their one-hop neighbours to share their respective broker scores (obtained by using IR). Thus, each MP will get to know the performance of brokers, with whom it may not have had any interaction. For example, suppose MP M_1 has interacted with only brokers $B1, B5$ and $B7$, while its one-hop neighbour M_2 has interacted with brokers $B1, B6, B7$ and $B8$. Thus, M_1 will obtain new information from M_2 about the performance of $B6$ and $B8$, while M_2 will obtain information from M_1 about the performance of $B5$. Gossiping facilitates neighboring MPs to refine their information about broker scores. Since MPs are likely to obtain new information, they have an incentive to participate in gossiping.

When a given MP M obtains broker scores from its one-hop neighbours, it computes its score for each broker B_i as follows. If M has not interacted with B_i, it will simply compute its score for B_i as the average Avg of all the scores (for B_i) that it receives from its neighbours, who have interacted with B_i. On the other hand, if M has interacted with B_i, it will compute its score for B_i as the average of the score that it assigned to B_i and Avg.

5.2.3. K-hop neighbour-based gossiping (K-NGS) strategy

The K-NGS strategy extends the NGS strategy by allowing gossiping among K-hop neighbours. (Recall that in NGS, gossiping is limited only to one-hop neighbours.) Thus, K-NGS facilitates MPs in assigning scores to more brokers than NGS and also uses inputs about broker scores from more MPs than NGS, thereby providing a broader and more refined picture of relative broker performance albeit at the cost of increased communication overhead. Note that under the K-NGS strategy, a given MP M computes its score for each broker B_i in the same manner as discussed for the NGS strategy.

5.3. Load-sharing by means of sub-brokers in EIB+

Preferred brokers in EIB+ are allowed to spawn sub-brokers for load-sharing purposes. Now let us examine the concept of sub-brokers. When a preferred broker PB becomes overloaded with too many requests, it replicates its data and index at MPs, which are willing to host its data and index. We designate such MPs as **sub-brokers**. Thus, preferred brokers dynamically create sub-brokers based on load and network performance to effectively convert relay MPs into brokers. This facilitates load-sharing among preferred brokers and sub-brokers, thereby making it likely to improve query response times due to less waiting times at the job queues of these MPs.

The preferred broker is incentivized to share its data and index with the sub-brokers because it can earn currency from such sharing. This is because we use our proposed royalty-based revenue-sharing scheme [1] in conjunction with EIB+. Thus, revenues of preferred brokers are further increased due to the presence of sub-brokers. Observe how EIB+ incentivizes brokers to perform better in order to become preferred brokers.

A preferred broker PB selects its sub-brokers based on three factors, namely remaining energy, bandwidth and current value of currency. PB prefers MPs with higher remaining energy as sub-brokers because such MPs are likely to be able to serve more queries, thereby facilitating them in earning more currency and consequently, also enabling PB to earn more currency because of the royalty-based revenue-sharing scheme [1]. Moreover, PB gives preference to MPs with high bandwidth because such MPs are likely to serve queries relatively quickly, thereby enabling them to earn more currency. (Recall that data item prices depend upon timeliness of query response.) Furthermore, PB prefers MPs with low current value of currency as sub-brokers because such MPs have more incentive to serve queries to earn currency than MPs, whose current values of currency are high. Notably, this also facilitates newly joined MPs (that have low currency) to seamlessly integrate themselves into the system by actively participating in the network as sub-brokers.

Notably, PB selects its sub-brokers from among its one-hop neighbours in order to minimize the communication traffic incurred for allocating replicas at sub-brokers. To select its sub-brokers, *PB* sends a message to its one-hop neighbour MPs requesting them to send their values of remaining energy, bandwidth and currency. Those MPs, which are interested to become sub-brokers of *PB*, reply to *PB* with the requested values. *PB* uses these values to compute a score *S* for each MP as follows.

$$S = (w_1 \times En) + (w_2 \times BA) + (w_3/Curr) \tag{9}$$

where En, BA and $Curr$ are the values of remaining energy, bandwidth and currency of the MP. In Eq. 9, w_1, w_2 and w_3 are weight coefficients such that $(w_1 + w_2 + w_3 = 1)$. The values of these weight coefficients are autonomously selected by a given preferred broker, hence they may differ across preferred brokers. MPs with relatively higher values of *S* are selected by *PB* as its sub-brokers. We leave the determination of the optimal number of sub-brokers per preferred broker to future work.

6. Performance evaluation

This section reports our performance evaluation by means of simulation in OMNeT++ (http://www.omnetpp.org). MPs move according to the *Random Waypoint Model* [36] within a region of area 4×4 km. We believe that the Random Waypoint Model is appropriate for our application scenarios.

Table 1 summarizes our performance study parameters. A total of 8000 data items is uniformly distributed among 1000 MPs i.e., each MP owns 8 data items. For each MP, the available memory space for hosting replicas is its remaining memory space, after memory for storing its 8 data items has been allocated. Query-issuers are selected randomly from among all the MPs. Each query is a request for a single data item. The number of such queries issued in the network per time unit is 10, the query's hard deadline τ_H being varied randomly between 25 to 30 time units. The query's soft deadline τ_S is 90% of τ_H. Query price is chosen randomly in the range of 100 to 500 currency units. Broker commission and relay commission are respectively set to 10% and 1% of the query price. For query routing purposes, we use the AODV protocol until a query is intercepted by a broker. Initial energy of an MP is selected randomly between 90,000 to 10,0000 energy units. Sending and receiving a message require 1.5 and 1 energy units respectively.

In Table 1, *TP* stands for 'replica allocation Time Period'. *Periodically*, every *TP* seconds, MPs broadcast a list of items that they want to replicate. Similar to existing works [7], we assume that network topology does *not* change significantly during replica allocation since it requires only a few seconds. The default communication range of all MPs is a circle of 120 m radius.

For all our experiments, the weight coefficients are set as follows: (a) the values of w_1, w_2 and w_3 for computing γ in Eq. 4 are set to 0.5, 0.25 and 0.25 respectively, (b) the values of w_1, w_2 and w_3 for computing λ in Eq. 5 are set to 0.5, 0.25 and 0.25 respectively, (c) the values of w_1, w_2 and w_3 for computing *S* in Eq. 9 are set to 0.4, 0.3 and 0.3 respectively.

Our performance metrics are **average response time** (**ART**) of queries, **data availability** (**DA**), **query hop-count** (**HC**) and **communication traffic** (**MSG**). $ART = (1/N_Q)\sum_{i=1}^{N_Q}(T_f - T_i)$, where T_i is the time of query issue, T_f is time of the query result reaching the query-issuer, and N_Q is the total number of queries. ART includes the download time, and is computed only for the successful queries. $DA = (N_S/N_Q) \times 100$, where N_S is the number of queries that were answered successfully. Thus, DA measures the percentage of successful queries. Queries may fail due to network partitioning or due to energy-depletion or unavailability of MPs that host the queried data items, or due to queries exceeding the TTL ('hops-to-live'). Preliminary experiments suggested that TTL = 8 is a reasonable value for our application scenarios. Hence, we consider TTL = 8 for our proposed EIB and EIB+ schemes. We define the query hop-count **HC** as the hop-count incurred by the query in the successful query path. Thus, HC is measured only for successful queries. We define MSG as the total number of messages incurred for query processing during the course of the experiment. Thus, $MSG = \sum_{q=1}^{N_Q} M_q$, where M_q is the number of messages incurred for the q^{th} query.

We compare the performance of our proposed broker-based **EIB** and **EIB+**incentive schemes with the non-incentive **E-DCG+**scheme [7]. We adapted the **E-DCG+**scheme [7] to our scenario. As discussed in Section 2, E-DCG+ is a non-incentive and non-economic replication scheme, and it does not provide incentives for replica hosting. E-DCG+ is executed at every replica allocation period. E-DCG+ is the closest to our scheme since it addresses replication in mobile ad hoc networks. Furthermore, we believe that E-DCG+ is among the best approaches for meaningful performance comparison with our proposed schemes because it is the most recent approach and it has already been compared to other non-incentive schemes.

As a baseline, we also do performance comparison w.r.t. a non-incentive and non-broker-based **NIB** (Non-Incentive without Brokerage) scheme to show the performance gain due to brokerage. Notably, querying in NIB is simply AODV-based and broker commissions do not arise. Furthermore, in case of NIB, we set the TTL to be 12 i.e., 50% higher than the TTL for our proposed EIB and EIB+ schemes. NIB does not provide any incentive to a peer to forward messages. In NIB, a peer forwards a message in the multi-hop network with a probability of 0.3.

Recall that EIB+ uses three different strategies for assigning broker scores. Here, we present the performance of EIB+ in conjunction with the K-NGS strategy. We have also performed an experiment to indicate the performance of EIB+ with each of the three broker scoring strategies.

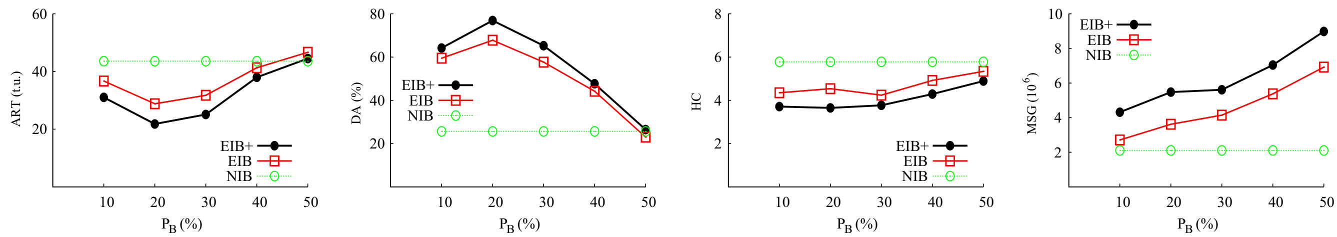
6.1. Determining the percentage of brokers

We performed an experiment to determine the percentage P_B of brokers in the network. Fig. 3 depicts the results. As P_B is increased from 10% to 20%, DA improves (albeit at the cost of higher MSG) for both EIB and EIB+ because the involvement of more brokers increases the probability that a given query is processed by at least one of them. However, as P_B is increased beyond 20%, performance keeps degrading due to reduction in the number of data-providers. This is because the sum total of the number of brokers and the number of data-providers is fixed. Notably, EIB+ exhibits better performance than EIB due to the presence of preferred brokers.

The results in Fig. 3 suggest that there is a trade-off between the performance (in terms of ART, DA and HC) and the communication traffic. Based on our experimental results, we set the percentage of brokers to 20% so that we can obtain good performance of EIB and

Table 1
Parameters of our performance study.

Parameter	Default value	Variations
Number of MPs (N_{MP})	1000	200, 400, 600, 800
% of brokers (P_B)	20%	10%, 30%, 40%, 50%
% of preferred brokers (ψ)	20%	10%, 30%, 40%, 50%
Queries/time unit	10	
Communication Range (CR)	120 m	40 m, 80 m, 160 m, 200 m
Percentage of MP failures (P_F)	20%	10%, 30%, 40%, 50%
Workload skewness (ZF_W)	0.5	0.1, 0.3, 0.7, 0.9
Bandwidth between MPs	1 Mbps to 2 Mbps (Bluetooth)	
Initial energy of an MP	90,000 to 10,0000 energy units	
MP service capacity	1 to 5 units	
Time-to-expire of a data item	3 min to 7 min	
Memory space of each MP	120 MB to 150 MB	
Speed of an MP	1 m/s to 10 m/s	
Size of a data item	0.5 MB to 10 MB	



(a) ART (b) DA (c) HC (d) MSG

Fig. 3. Determining the percentage of brokers.

EIB+ with reasonable communication traffic. Observe that both EIB and EIB+ perform slightly worse than NIB when P_B = 50%. A closer look at the results in Fig. 3 suggests that performance gain of EIB over NIB occurs only when P_B is less than 48%. This is because when P_B exceeds 48%, the benefits from brokerage are offset by the additional overhead of interactions among the relatively larger number of brokers. Hence, when P_B exceeds 48%, the peers are better off without a broker-based architecture i.e., they can directly obtain the data from the data-providers.

6.2. Determining the percentage of preferred brokers in EIB+

We conducted an experiment to determine the percentage ψ of preferred brokers. Here, $\psi = ((N_{Pref}/N_{Total}) * 100)$, where N_{Pref} is the number of preferred brokers, while N_{Total} is the total number of brokers. For example, if N_{Total} = 20 and ψ = 20%, the number of preferred and common brokers would be 4 and 16 respectively. For this experiment, we also varied the number SB of sub-brokers corresponding to each preferred broker. Fig. 4 depicts the results. We use the notations SB-0, SB-2 and SB-4 to represent the scenarios for EIB+ corresponding to 0, 2 and 4 sub-brokers respectively per preferred broker.

The results in Fig. 4 indicate that as ψ is increased from 10% to 20%, the performance of EIB+ improves slightly in the cases of SB-0, SB-2 and SB-4 due to the incentivizing effect of preferred brokerage becoming more pronounced. However, as ψ is increased to 30% and beyond, the performance of EIB+ degrades. This occurs because at higher values of ψ, more brokers are allowed to become preferred brokers, thereby implicitly reducing the level of service required to become a preferred broker. This reduces the incentive for preferred brokerage.

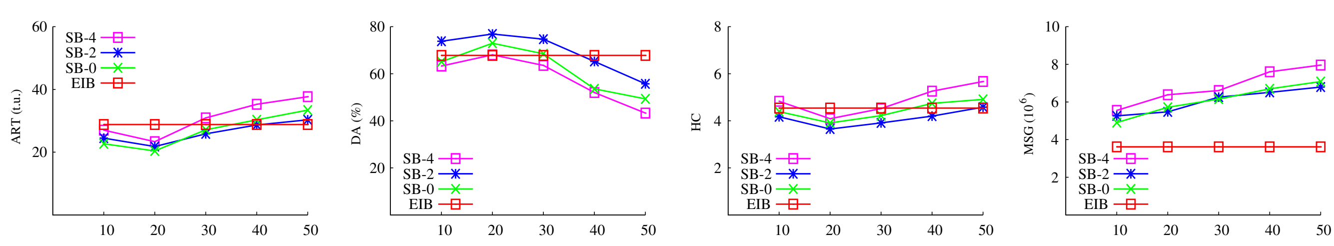
EIB+ performs better in the case of SB-2 (albeit at the cost of higher MSG) as compared to that of SB-0 due to load-sharing among the preferred brokers and their respective sub-brokers. However, in case of SB-4, EIB+ performs worse than for SB-2 because the relatively high overhead of data allocation among a larger number of sub-brokers reduces the performance. The results in Fig. 4 suggest that EIB+ performs best at reasonable communication overhead when ψ = 20% (in case of SB-2). Thus, we experimentally determine ψ to be 20% and SB to be two.

6.3. Performance of EIB and EIB+

Fig. 5 depicts the results using the default values of the parameters in Table 1. The results in Fig. 5aa indicate that after the first 20,000 queries have been processed, EIB, EIB+ and E-DCG+ exhibit comparable performance because the effect of replication is not pronounced at the initial stages. However, over time as more queries are processed, performance improves in terms of ART, DA and HC for all the schemes essentially due to the effect of replication becoming more prominent. Both ART and HC eventually plateau due to reasons such as network partitioning, competition among replicas for memory space and unavailability of some of the MPs.

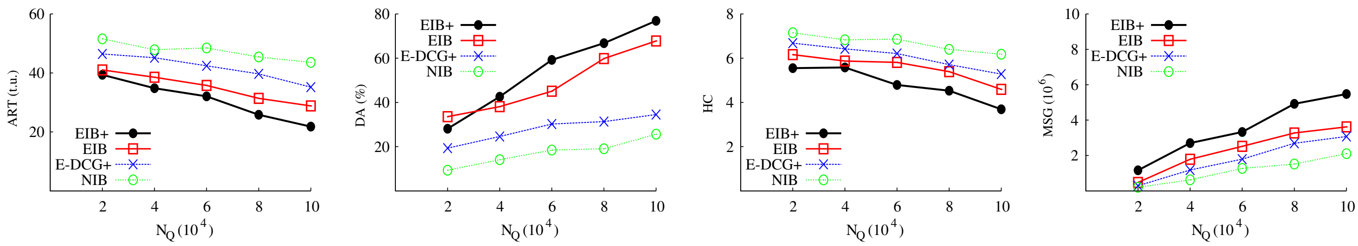
The results in Fig. 5dd indicate that EIB and EIB+ incur higher MSG than E-DCG+ and NIB primarily due to the additional communication overhead introduced by brokers (and sub-brokers in case of EIB+). However, we believe that the additional number of messages incurred by EIB and EIB+ is a small price to pay for the performance benefits of these schemes. EIB+ incurs higher MSG than EIB because it incorporates gossiping among neighboring MPs for computing broker scores. E-DCG+ incurs higher MSG than NIB because in E-DCG+, every MP needs to periodically send messages to other MPs to convey replication-related information.

EIB+ outperforms EIB because it provides additional incentives to brokers for performing value-added routing by incorporating the notion of preferred brokers. Moreover, EIB+ also performs effective load-sharing between preferred brokers and sub-brokers, thereby reducing query waiting times in the job queues of the brokers. EIB performs better than E-DCG+ due to its economic incentives, which encourage MP participation. Increased MP participation implies more opportunities for replication, more memory space for hosting replicas and multiple paths for locating a data item/replica. Furthermore, unlike E-DCG+, EIB maintains indexes at the brokers (which facilitate value-added routing) and it replicates 'hot' data items at the brokers. E-DCG+ exhibits better performance than NIB because of its superior replication mechanism.



(a) ART (b) DA (c) HC (d) MSG

Fig. 4. Determining the percentage of preferred brokers in EIB+.



(a) ART (b) DA (c) HC (d) MSG

Fig. 5. Performance of EIB & EIB+.

6.4. Effect of variations in the number of MPs

We varied the total number N_{MP} of MPs, keeping the number of queries proportional to N_{MP}. Fig. 6 depicts the results. As N_{MP} increases, ART and MSG increase for all the schemes due to increase in network size. However, the rate of increase in ART is lower for EIB and EIB+ than for E-DCG+ and NIB due to their better incentivization of replication by means of economic incentives and brokerage. As N_{MP} increases, DA increases for all the schemes due to increased opportunities for replication. HC follows a pattern similar to that of ART, the slight deviations occurring due to bandwidth differences. Observe that when N_{MP} = 20, EIB+ exhibits slightly worse DA than that of EIB because the benefits provided by preferred brokers are not realized due to the existence of fewer preferred brokers.

6.5. Effect of variations in the communication range

The results in Fig. 7 depict the effect of variations in the communication range CR of the MPs. Increase in CR has the effect of bringing the MPs 'nearer' to each other. Hence, performance improves with increase in CR for all the schemes due to data items becoming 'nearer' and more accessible to query-issuers. However, performance gains occur only until CR = 120 metres. Beyond CR = 120 metres, ART and DA degrade for all the schemes because the MPs become too 'close' to each other, hence a relatively larger number of MPs and brokers become involved in the processing of any given query. This results in a relatively larger number of queries waiting in the job queues of the data-providers, hence some of the query deadlines are missed. Beyond CR = 120 metres, the performance gap between EIB and EIB+ keeps decreasing because the benefits of preferred brokerage become less pronounced when the MPs are already too 'near' to each other. In essence, all the schemes perform best when CR = 120 metres.

As CR increases, MSG increases for all the schemes because the increased reachability causes more MPs to become involved in the processing of a given query. On the other hand, with increase in CR, a lower number of messages are required to reach a given MP. These two opposing effects somewhat offset each other at higher values of CR, thereby explaining the reason why MSG eventually plateaus for all the schemes.

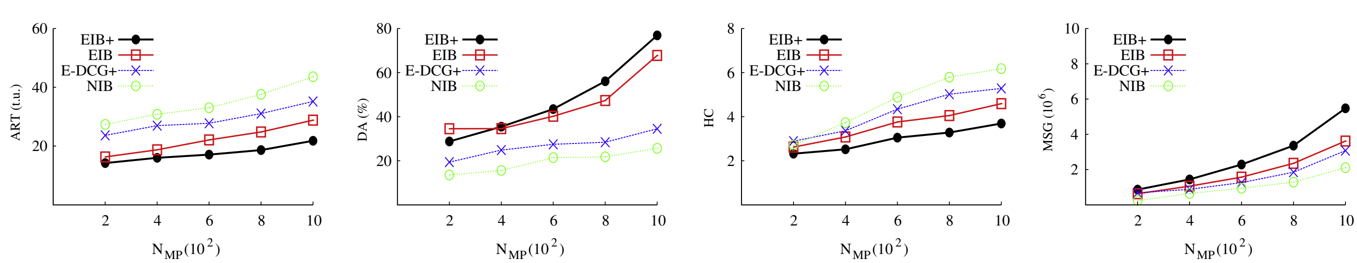
6.6. Effect of MP failures

MPs can fail due to reasons such as depletion of their limited energy resources. Fig. 8 depicts the results of the effect of MP failures. As the percentage P_F of MP failures increases, MP participation decreases, query paths become longer and fewer data-hosting MPs remain available, thereby degrading the performance of all the schemes. Interestingly, at P_F = 50%, all the schemes exhibit comparable ART due to limited MP participation making the effect of economic incentives and brokerage less pronounced. As the results in Fig. 8d indicate, MSG decreases with increase in P_F for all the schemes due to reduced communication overhead among a lower number of available MPs. Interestingly, at $P_F = 50\%$, EIB incurs lower MSG than E-DCG+ due to scarcity of brokers when the total number of available MPs become relatively low. However, EIB+ still incurs higher MSG than E-DCG+ due to gossiping-related communication overheads.

6.7. Effect of different strategies for assigning performance-based scores to brokers in EIB+

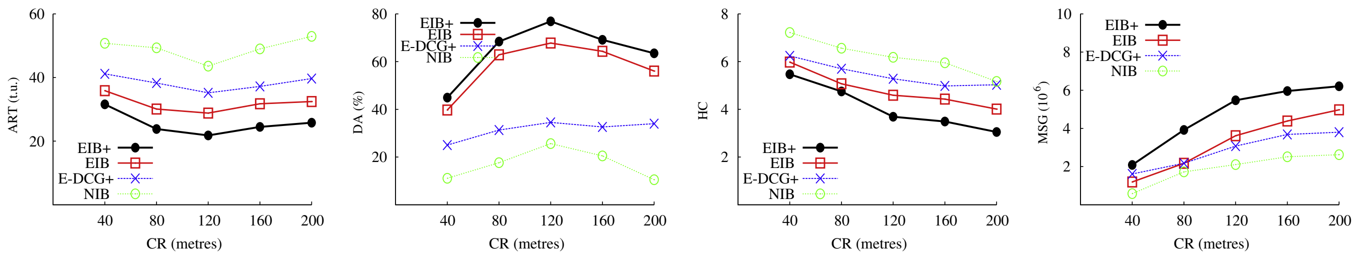
We conducted an experiment to investigate the relative performance of EIB+ with the different strategies, namely IR, NGS and K-NGS, for assigning performance-based scores to brokers. Fig. 9 depicts the results. K-NGS outperforms NGS because its gossiping among k-hop neighbours better incentivizes preferred brokerage by incorporating broker scores from a larger number of MPs albeit at the cost of higher MSG. Similarly, NGS performs better than IR since its gossiping among one-hop neighbours provides better incentives for preferred brokerage than IR. The performance of all the three strategies improve over time as more queries are processed due to the reasons discussed for the results in Fig. 5.

Fig. 9-f depicts the snapshots of broker scores at the time-points of 40000 and 100000 queries respectively under IR, NGS and



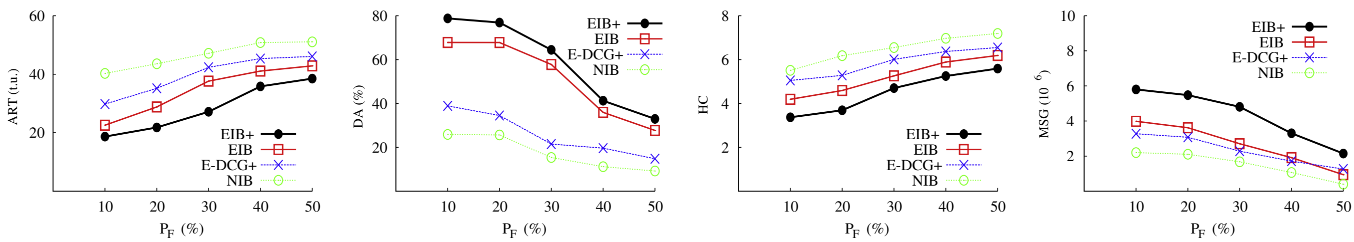
(a) ART (b) DA (c) HC (d) MSG

Fig. 6. Effect of variations in the number of MPs.



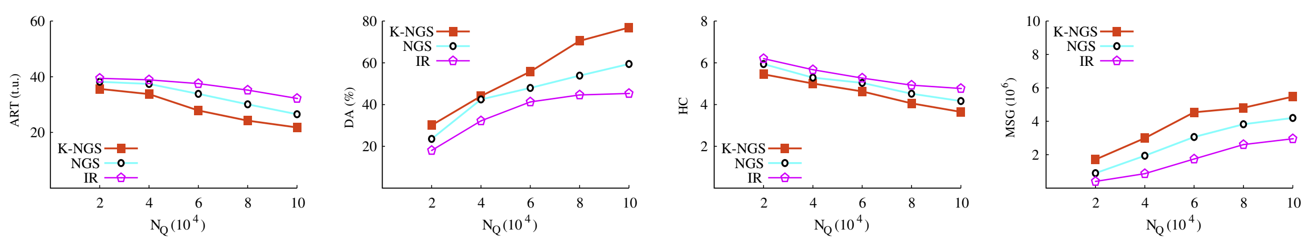
(a) ART (b) DA (c) HC (d) MSG

Fig. 7. Effect of variations in the communication range.

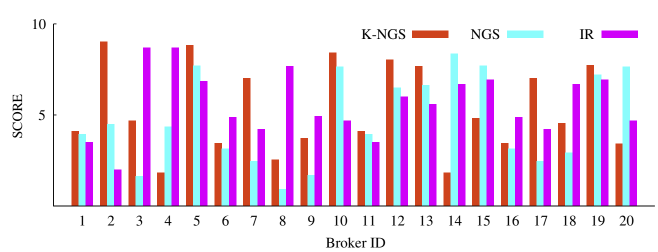


(a) ART (b) DA (c) HC (d) MSG

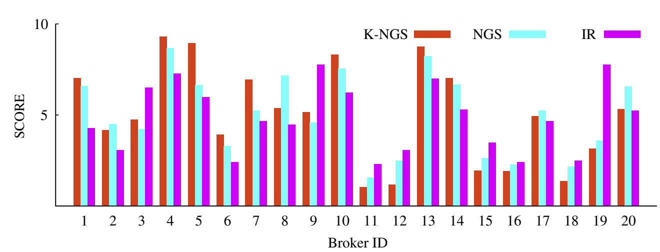
Fig. 8. Effect of MP failures.



(a) ART (b) DA (c) HC (d) MSG



(e) Snapshot of the broker scores at the 40000-query time-period



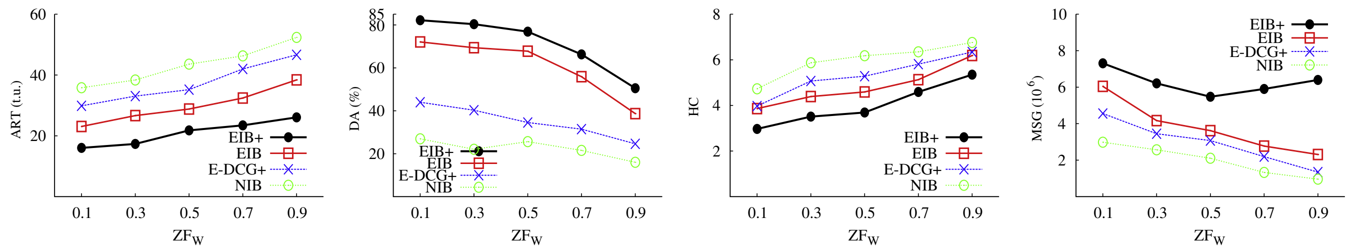
(f) Snapshot of the broker scores at the 100000-query time-period

Fig. 9. Effect of different strategies for assigning performance-based scores to brokers in EIB+.

K-NGS. The X-axis represents the unique identifiers of the brokers, while the Y-axis depicts the score of each broker. Periodically, after every 20,000 queries, the scores of brokers are recorded. The scores are on a scale of 1 to 10, where a higher score indicates better performance. Each MP assigns an initial score of 5 to all the brokers at the start of every 20,000-query time-period. (This periodic resetting of scores is necessary to reflect current performance of brokers.) Factors such as a broker's location, mobility pattern and current network conditions result in variation of scores across brokers.

Now let us examine the results in Fig. 9e. We will denote the broker with ID of i as Bi. Observe that there is no clear pattern regarding any specific scoring strategy assigning higher or lower scores than the others. For example, K-NGS assigned the lowest score to $B4$, but it assigned the highest score to $B2$. Observe that $B8$ is assigned a much higher score by IR than by NGS and K-NGS. Broker scores vary across scoring strategies because they consider varying amounts of interaction with other MPs. These strategies may also assign comparable scores to any given broker e.g., $B1$ and $B11$ in the results in Fig. 9e. This occurs when a broker's performance remains comparable in providing services to MPs at different locations. A broker's score may fall below 5 (e.g., $B1$ in Fig. 9(e)) due to reasons such as connectivity to limited resources in its mobility path and limited energy.

Even though broker scores may vary across scoring strategies, the results in Fig. 9-f serve as a guide for evaluating broker performance, thereby facilitating in distinguishing between common and preferred brokers. For example, in Fig. 9e, $B5, B12, B13$ and $B19$ and in Fig. 9f, $B4, B5, B10$ and $B13$ would be the preferred brokers, while the other brokers would be common brokers.



(a) ART (b) DA (c) HC (d) MSG

Fig. 10. Effect of variations in the workload skew.

6.8. Effect of variations in the workload skew

Fig. 10 depicts the results when the zipf factor ZF_W is varied. Notably, among all the schemes, only EIB+ supports load-sharing, which occurs between preferred brokers and sub-brokers. As ZF_W increases (i.e., increasing skew in the workload), performance degrades for all the schemes. This occurs due to increased waiting times at the job queues of overloaded data-providers, thereby causing some of the queries to miss the deadlines. Observe how EIB+'s load-sharing mechanism facilitates it in outperforming the other schemes. However, the performance gap between EIB+ and the reference schemes decreases with decreasing skew due to the effect of load-sharing becoming less pronounced. As ZF_W increases, the number of query failures increase (due to queries missing their deadlines), thereby reducing MSG. However, for EIB+, MSG increases beyond $ZF_W = 0.5$ due to the interactions between the preferred brokers and their sub-brokers.

7. Conclusion

In M-P2P networks, data availability is typically low due to rampant free-riding, frequent network partitioning and mobile resource constraints. We have proposed the E-Broker system for improving data availability in M-P2P networks. E-Broker incorporates two economic incentive-based brokerage schemes, namely EIB and EIB+. EIB incentivizes relay peers to act as information brokers for performing value-added routing and replication in M-P2P networks, thereby effectively improving data availability. The EIB+ scheme extends the EIB scheme by incorporating three different broker scoring strategies for providing additional incentives to brokers. EIB+ also facilitates load-sharing among the peers.

We have also evaluated the number of brokers, beyond which the peers are better off without a broker-based architecture. Our performance study indicates that the proposed schemes are indeed effective in improving query response times, data availability and query hop-counts at reasonable communication traffic cost in M-P2P networks. In the future, we plan to extend this work by using game-theoretic approaches for data item pricing.

References

[1] A. Mondal, S.K. Madria, M. Kitsuregawa, An economic incentive model for encouraging peer collaboration in mobile-P2P networks with support for constraint queries, Journal of Peer-to-Peer Networking and Applications 2 (3) (2009) 230–251.

[2] M. Franklin, D. Kossmann, T. Kraska, S. Ramesh, R. Xin, CrowdDB: Answering queries with crowdsourcing, in: Proceedings of the 2011 ACM SIGMOD International Conference on Management of data (SIGMOD'11), Athens, Greece, 2011, pp. 61–72.

[3] M. Ham, G. Agha, ARA: A Robust Audit to prevent free-riding in P2P networks, in: Proceedings of the Fifth IEEE International Conference on Peer-to-Peer Computing (P2P'05), Konstanz, Germany, 2005, pp. 125–132.

[4] S. Kamvar, M. Schlosser, H. Garcia-Molina, Incentives for combatting free-riding on P2P networks, Journal of Euro-Par 2003 Parallel Processing 2790 (2003) 1273–1279.

[5] E. Adar, B.A. Huberman, Free riding on Gnutella, Journal of First Monday 5 (10) (2000).

[6] S. Saroiu, P. Gummadi, S. Gribbler, A measurement study of P2P file sharing systems, in: Proceedings of Multimedia Computing and Networking 2002 (MMCN'02), San Jose, CA, January 2002.

[7] T. Hara, S.K. Madria, Data replication for improving data accessibility in ad hoc networks, Journal of IEEE Transactions on Mobile Computing 5 (11) (2006) 1515–1532.

[8] L. Buttyan, J. Hubaux, Stimulating cooperation in selforganizing mobile ad hoc networks, Journal of ACM/Kluwer Mobile Networks and Applications 8 (5) (2003) 579–592.

[9] K. Chen, K. Nahrstedt, iPass: An incentive compatible auction scheme to enable packet forwarding service in MANET, in: Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04), Hachioji, Tokyo, Japan, 2004, pp. 534–542.

[10] J. Crowcroft, R. Gibbens, F. Kelly, S. Ostring, Modelling incentives for collaboration in mobile ad hoc networks, Journal of Performance Evaluation - Selected papers from the first workshop on modeling and optimization in mobile, ad hoc and wireless networks (WiOpt'2003) 57 (4) (2003) 427–439.

[11] V. Srinivasan, P. Nuggehalli, C. Chiasserini, R.R. Rao, Cooperation in wireless ad hoc networks, in: INFOCOM 2003 Twenty-Second Annual Joint Conference of the IEEE Computer and Communications IEEE Societies, Sun Francisco, 2003, pp. 808–817.

[12] O. Wolfson, B. Xu, A. Sistla, An Economic model for resource exchange in Mobile Peer to Peer networks, in: Proceedings of the 16th International Conference on Scientific and Statistical Database Management (SSDBM'04), Santorini Island, Greece, 2004, pp. 235.

[13] B. Xu, O. Wolfson, N. Rishe, Benefit and pricing of spatio-temporal information in Mobile Peer-to-Peer networks, in: Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS '06), Poipu, Kauai, Hawaii, 2006, pp. 223b.

[14] A. Mondal, S.K. Madria, M. Kitsuregawa, ABIDE: a bid-based economic incentive model for enticing non-cooperative peers in mobile-P2P networks, in: Proceedings of the 12th international conference on Database systems for advanced applications (DASFAA'07), Berlin, Heidelberg, 2007, pp. 703–714.

[15] A. Datta, M. Hauswirth, K. Aberer, Updates in highly unreliable, replicated peer-to-peer systems, in: Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS'03), Providence, RI, USA, 2003, pp. 76.

[16] D. Ratner, P. Reiher, G. Popek, G. Kuenning, Replication requirements in mobile environments, Journal of Mobile Networks and Applications 6 (6) (2001) 525–533.

[17] B. Richard, D. Nioclais, D. Chalon, Clique: A transparent, peer-to-peer collaborative file sharing system, in: Proceedings of the 4th international conference on Mobile Data Management (MDM'03), Melbourne, Australia, 2003, pp. 21–24.

[18] F. Azzedin, Trust-based taxonomy for free riders in distributed multimedia systems, in: Proceedings of the International Conference on High Performance Computing and Simulation (HPCS), Caen, France, 2010, pp. 362–369.

[19] F.Azzedin, Classifying and tracking free riders in multimedia-based systems, Journal of Universal Computer Science 16 (10) (2010) 1368–1387.

[20] Y. Li, D. Gruenbacher, C.M. Scoglio, Reward only is not enough: Evaluating and improving the fairness policy of the P2P file sharing network eMule/eDonkey, Journal of Peer-to-Peer Networking and Applications 5 (1) (2012) 40–57.

[21] J.F. Kurose, R. Simha, A microeconomic approach to optimal resource allocation in distributed computer systems, Journal of IEEE Transactions on Computers 38 (5) (1989) 705–717.

[22] J. Liu, V. Issarny, Service allocation in selfish mobile ad hoc networks using vickrey auction, in: Proceedings of the 2004 international conference on Current Trends in Database Technology (EDBT'04), Heraklion, Greece, pp. 385–394.

[23] Y. Xue, B. Li, K. Nahrstedt, Optimal resource allocation in wireless ad hoc networks: A price-based approach, Journal of IEEE Transactions on Mobile Computing 5 (4) (2006) 347–364.

[24] T. Straub, A. Heinemann, An anonymous bonus point system for mobile commerce based on word-of-mouth recommendation, in: Proceedings of the 2004 ACM symposium on Applied computing, Nicosia, Cyprus, 2004, pp. 766–733.

[25] A. Garyfalos, K. Almeroth, Coupon based incentive systems and the implications of equilibrium theory, in: Proceedings of the IEEE International Conference on E-Commerce Technology (CEC '04), San Diego, California, USA, 2004, pp. 213–220.
[26] O. Ratsimor, T. Finin, A. Joshi, Y. Yesha, eNcentive: A framework for intelligent marketing in Mobile Peer-to-Peer environments, in: Proceedings of the 5th international conference on Electronic commerce (ICEC '03), Pittsburgh, Pennsylvania, USA, 2003, pp. 87–94.
[27] R. Chakravorty, S. Agarwal, S. Banerjee, I. Pratt, MoB: A mobile bazaar for wide-area wireless services, in: Proceedings of the 11th annual international conference on Mobile computing and networking (MobiCom'05), Cologne, Germany, 2005, pp. 228–242.
[28] P. Daras, D. Palaka, V. Giagourta, D. Bechtsis, A novel peer-to-peer payment protocol, in: Proceedings of IEEE EUROCON 2003 Computer as a Tool, Ljubljana, Slovenia, 2003, pp. 2–6.
[29] E. Elrufaie, D. Turner, Bidding in P2P content distribution networks using the lightweight currency paradigm, Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04), Las Vegas, NV, USA, 2004, pp. 129 .
[30] S. Zhong, J. Chen, Y. Yang, Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks, in: INFOCOM 2003 Twenty-Second Annual Joint Conference of the IEEE Computer and Communications IEEE Societies, Sun Francisco, 2003, pp. 1987–1997.
[31] F. Azzedin, A. Ridha, Feedback behavior and its role in trust assessment for peer-to-peer systems, Journal of Telecommunication Systems 44 (3-4) (2010) 253–266.
[32] J. Sabater, C. Sierra, Review on computational trust and reputation models, Artificial Intelligence Review 24 (1) (2005) 33–60.
[33] R. Boero, G. Bravo, F. Squazzoni, Trust and partner selection in social networks: An experimentally grounded model, Journal of CoRR (2010).
[34] P. Velloso, R. Laufer, D. de O Cunha, O. Duarte, G. Pujolle, Trust management in mobile ad hoc networks using a scalable maturity-based model, Journal of IEEE Transactions on Network and Service Management 7 (3) (2010) 172–185.
[35] J. Sheu, C. Chao, W. Hu, C. Sun, A Clock synchronization algorithm for multi-hop wireless ad hoc networks, in: Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS), Hachioji, Tokyo, Japan, 2004, pp. 574–581.
[36] J. Broch, D. Maltz, D. Johnson, Y. Hu, J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocol, in: Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile computing and networking (MobiCom'98), Dallas, Texas, USA, 1998, pp. 85–97.