Distrib Parallel Databases (2010) 28: 1–31
DOI 10.1007/s10619-010-7063-6

E-ARL: An Economic incentive scheme for Adaptive Revenue-Load-based dynamic replication of data in Mobile-P2P networks

Anirban Mondal · Sanjay Kumar Madria · Masaru Kitsuregawa

Published online: 27 April 2010

Abstract In mobile ad hoc peer-to-peer (M-P2P) networks, frequent network partitioning occurs due to peer movement or owing to peers switching 'off' their mobile devices. This leads to typically low data availability in M-P2P networks, thereby necessitating data replication. This work proposes E-ARL, which is a novel **Economic scheme** for **Adaptive Revenue-Load-based** dynamic replication of data in **dedicated** M-P2P networks with the aim of improving data availability. Thus, E-ARL considers a mobile cooperative environment, where the MPs are working towards the same goal, and the network performance is facilitated by the economic scheme. E-ARL essentially allocates replicas based on its economic scheme. Each data item has a *price* in *virtual currency*. E-ARL requires a query issuing peer to pay the *price* of its queried data item to the query-serving peer and a commission to relay peers in the successful query path. The main contributions of E-ARL follow. First, it uses an economic scheme for efficiently managing M-P2P resources in a context-aware manner by facilitating effective replica hosting and message relaying by peers. Second, it *collaboratively* performs *bid-based* replica allocation to facilitate better quality of service. Third, it incorporates both revenue-balancing and load-balancing to improve peer participation and performance. Fourth, it conserves the energy of low-energy

Communicated by Ashfaq Khokhar.

Part of this work was done when Anirban Mondal was at the University of Tokyo, Japan.

A. Mondal (✉)
IIIT Delhi, Delhi, India
e-mail: anirban@iiitd.ac.in

S. Kumar Madria
University of Missouri-Rolla, Rolla, USA
e-mail: madrias@umr.edu

M. Kitsuregawa
University of Tokyo, Tokyo, Japan
e-mail: kitsure@tkl.iis.u-tokyo.ac.jp

MPs to facilitate network connectivity. Our performance evaluation shows that E-ARL is indeed effective in improving peer participation in M-P2P networks, thereby improving query response times, query success rates, query hop-counts and replica allocation traffic.

Keywords Mobile P2P · Replication · Availability · Economic 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. 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. Mobile devices with support for wireless device-to-device P2P communication are beginning to be deployed such as Microsoft's Zune [23].

This work focuses on improving the performance (i.e., query response time and data availability) of *dedicated* and *cooperative* M-P2P networks by means of an economic model, which facilitates data replication. Some M-P2P application scenarios for dedicated networks follow. Suppose a group of geologists are performing studies of soil in a remote Amazonian rainforest, where communication infrastructures (e.g., base stations) do not exist. They need to share data concerning soil types and soil contents (e.g., salt and sodium contents in the soil, and amount of nitrogen in the soil). In the same vein, a group of archaeologists, who are performing excavations in a remote area of Egypt, need to share information (e.g., artefacts found, historical clues and ancient maps found) with each other by means of mobile devices. Similarly, moving salespersons, who are involved in the sales of products such as tickets and insurance in a given neighborhood, need to share sales data (e.g., total units sold, sales profits) with each other by means of mobile devices because they are working towards the same collaborative goal.

Our target applications mainly concern slow-moving objects e.g., geologists and archaeologists moving in a remote area or salesmen walking in a neighborhood. MPs move within the spatial region, which is divided into a rectangular grid structure, the size of the rectangles being application-dependent. Notably, M-P2P ephemerality emphasizes the need for queries to be answered in a fast and timely manner, thereby necessitating query *deadlines*. In this work, the notion of replica consistency is based on the time of the latest update e.g., a copy of sales data, which was updated an hour ago, is considered to be more consistent than one that had been updated five hours ago. Our application scenarios do not require absolute replica consistency [11, 33], hence we consider tolerance to weaker replica consistency. For simplicity, this work considers only numerical data.

Data availability in M-P2P networks is typically lower than in fixed networks due to frequent network partitioning arising from user movement and mobile devices switching 'off' when their generally limited energy is drained. Incidentally, data availability is less than 20% even in a wired environment [39]. Furthermore, MPs generally have limited resources (e.g., bandwidth, energy, memory space). Consequently, MPs, which host important data, may quickly run out of energy, thereby making the data unavailable. This further reduces data availability and connectivity of M-P2P networks. To maintain accessibility to data items, the limited energy resources of the data-hosting MPs as well as the paths to these MPs need to be effectively preserved. Additionally, preservation of the energy of relay MPs also becomes a necessity to maintain shorter query paths for satisfying timeliness requirements of queries.

We propose E-ARL, which is a novel **Economic scheme** for **Adaptive Revenue-Load-based** dynamic replication of data in **dedicated** M-P2P networks with the aim of improving data availability. Thus, E-ARL considers a mobile cooperative environment, where the MPs are working towards the same goal, and the network performance is facilitated by the economic scheme. E-ARL essentially allocates replicas based on its economic scheme. Each data item has a *price* in *virtual currency*. E-ARL requires a query issuing MP to pay the *price* of its requested data item to the MP serving its request and a *commission* to each relay MP in the successful query path from which it eventually downloads the data, thereby enticing them to forward queries quickly.

The price of a data item depends on its access frequency, the number of users who accessed it, its expiry time, the number of its existing replicas, its (replica) consistency, the average response time required for accessing it, and load and energy of its host MP. Relay commission increases with increasing size of relayed messages and decreasing energy of relay MPs. This facilitates network connectivity since query-issuing MPs would likely prefer lower cost query paths for obtaining their requested data items. An MP can earn currency by providing service e.g., sharing its own data items, hosting replicas and relaying messages.

We define the **revenue** of an MP as the difference between the amount of virtual currency that it earns by providing service and the amount that it spends by issuing queries. Thus, E-ARL ensures that an MP has to provide service to earn enough revenue to be able to issue its own queries, thereby facilitating better network performance due to proactive participation and collaboration of the MPs. The main contributions of E-ARL follow:

1. It uses an economic scheme for efficiently managing M-P2P resources in a context-aware manner by facilitating effective replica hosting and message relaying by peers.
2. It *collaboratively* performs *bid-based* replica allocation to facilitate better quality of service.
3. It incorporates both revenue-balancing and load-balancing. Revenue-balancing prevents gross imbalance of revenues across MPs, which could result in some of the MPs not having adequate revenue for issuing queries, thereby decreasing MP participation. Load-balancing prevents queries from incurring long waiting times in the job queues of overloaded MPs.
4. It conserves the energy of low-energy MPs to facilitate network connectivity.

E-ARL improves MP participation, thereby leading to better data availability, higher available energy and bandwidth, and multiple paths to answer a given query. Unlike [21], it allocates replicas *fairly* by considering the origin of queries for data items to determine their relative importance to the network as a whole. It provides economic incentives to peers to perform replication based on data item characteristics (e.g., price, popularity), MP characteristics (e.g., revenue, load, energy, bandwidth, memory space) and network connectivity considerations (i.e., MP energy conservation). It considers both read-only and updatable data items, but prefers to replicate items with relatively low update frequencies for facilitating the maintenance of replica consistency.

E-ARL considers two types of items, namely *priority items* and *normal items*. Queries on priority items and normal items are designated as *priority queries* and *normal queries* respectively. Priority items are those for which queries need to be answered *urgently* i.e., within the shortest possible response time. For example, an archaeologist could become lost in a maze of an archaeological site. In such cases, he would urgently need to issue a priority query to ask information from his colleagues concerning possible roadmaps of the maze. Similarly, a geologist, who loses his way while looking for soil samples in a rainforest, could issue a priority query for obtaining roadmaps from his colleagues. As such, it is not considered unusual for geologists and archaeologists, who work in remote areas, to lose their way since such remote areas generally do not have specific landmarks. In such scenarios, priority queries become necessary to ensure that the geologists or the archaeologists do not fall behind on their (research) work schedule, or they do not fail to satisfy the schedules of the buses, which transported them to these sites.

In contrast, for normal items, queries need to be answered *non-urgently* from the query path with the lowest cost, albeit within the query deadline. A normal query could be issued by an archaeologist asking his colleague about the number of artefacts he found. Similarly, a geologist could issue a normal query to ask his colleague about the average nitrogen content of soil samples found by his colleague. A salesman could issue a normal query to his colleagues for finding out the number of units sold by them. Observe that normal queries are not associated with any immediate need or urgency. Understandably, E-ARL tries to replicate the priority items first before replicating the normal items due to urgency reasons.

To manage replication efficiently, E-ARL deploys a super-peer architecture [47]. The super-peer (SP) is an MP, which generally moves within the region and which has maximum energy and processing capacity at a given time. For our application scenarios, the chief geologist or the chief archaeologist or the most senior salesman would act as SP. SP facilitates replica allocation and avoids broadcast storm during replica allocation. Each MP *periodically* sends a message to SP with information such as its current location, revenue, access statistics, load, available memory space and energy status, thereby facilitating SP to better manage replication. In contrast, for an architecture without any SP (e.g., the E-DCG+ approach [21]), each MP needs to broadcast its status to all other MPs to make each other aware of the regional status, thereby creating an undesirable broadcast storm during replica allocation. Our architecture does not require queries to pass via SP, thereby preserving P2P autonomy. This is possible because every MP periodically sends the list of data items/replicas hosted at itself to SP, and SP broadcasts this information to all MPs.

Observe that there is some communication overhead associated with MPs requiring to report information to SP. However, we note that status messages are generally exchanged periodically in any network (e.g., to check whether nodes are alive or not), and the information from an MP to the SP could effectively be piggybacked onto these status messages, thereby essentially leading to amortization of the communication overhead. Furthermore, the communication overhead associated with the MPs reporting information to the SP is negligible w.r.t. the overall communication overhead associated with sending and receiving data among the MPs in the network. This is partly because MPs report information to the SP only periodically and partly due to the fact that the size of the information sent by an MP to the SP is only in terms of a few Kilobytes (small amounts of text data). On the other hand, actual data transfers between the mobile peers occur much more frequently as compared to the communication between an MP and the SP, and the communication overhead associated with the actual data transfers are in the range of around 50 to 350 Kilobytes.

We have evaluated E-ARL's performance w.r.t. three other schemes, namely the non-economic **E-DCG+** [21] scheme, the **Revenue-based Incentive (RI)** scheme, and the **Load-based Non-incentive (LN)** scheme. RI considers revenue without considering load, while LN takes load into account without considering revenue. Our performance study shows that E-ARL's incentives entice better MP participation, thus it significantly outperforms the reference approaches in terms of query response times, query success rates, query hop-counts and replica allocation traffic. At higher workload skew, the performance gap between E-ARL and the reference approaches increases due to E-ARL's better load-balancing capability. When the revenues of more MPs exceed the average revenue in the system, performance of E-ARL improves due to more MPs providing service.

Our experiments also show that data item prices increase with decreasing number of replicas due to higher demand. However, when too few replicas exist, the price decreases in an anomalous manner due to decrease in the quality of service (i.e., query response times and data availability) for queries on that item. This is due to E-ARL's economic scheme, which considers not only the demand and supply for data item pricing, but also the quality of service associated with the given data items. As the duration of the replica allocation period decreases, all the approaches perform better albeit at the cost of higher replica allocation traffic. E-DCG+ incurs the highest replica allocation traffic as it requires every MP to periodically broadcast its status to all MPs in the network. As the percentage of priority queries increases, performance of all the approaches degrades due to less replica allocation for normal items, which penalizes the performance for normal items. However, when majority of the queries are priority queries, all the approaches show some performance improvement since adequate replicas for the priority items have been allocated. E-ARL exhibits good scalability and is able to effectively maintain network connectivity due to preserving the energy of low-energy MPs.

In E-ARL, each MP uses the same formulae for data pricing and relay commissions (these formulae are part of the M-P2P software, which every MP has to use). This is possible since E-ARL is aimed at dedicated M-P2P network applications in which the MPs are working collaboratively towards a common goal. Furthermore, we do not claim that our formulae for data item pricing and relay commissions are the only ways of pricing. However, our economic M-P2P scheme performs better than traditional replication schemes using the proposed formulae. Thus, we provide the framework for economic collaboration among MPs, and our formulae can be viewed as guidelines for developing new formulae, which can be deployed within this framework. In essence, the focus of E-ARL is effective allocation of resources in a mobile cooperative environment by means of an economic model for facilitating better performance. We are using the economic model to decide upon issues such as who will have access to resources, who will answer queries and so on.

The remainder of this paper is organized as follows. Section 2 discusses existing works, while Sect. 3 describes the E-ARL economic scheme. Section 4 discusses the query model of E-ARL. Section 5 discusses the adaptive revenue-load-based replica allocation scheme of E-ARL. Section 6 discusses the *bid-based* economic replica allocation algorithms in E-ARL. Section 7 reports our performance evaluation. Finally, we conclude in Sect. 8 with directions for future work.

2 Related work

This section provides an overview of existing works.

2.1 Non-incentive-based replication in Mobile ad hoc networks (MANETs)

The proposals in [20, 21] discuss replication in MANETs. **E-DCG+** [21] creates groups of MPs that are biconnected components in a MANET, and shares replicas in larger groups of MPs to provide high stability. An RWR (read-write ratio) value in the group of each data item is calculated as a summation of RWR of those data 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. Each replica is allocated at an MP, whose RWR value to the item is the highest among MPs that have free memory space to create it. The work in [20] aims at classifying different replica consistency levels in a MANET based on application requirements, and proposes protocols to realize them. Consistency maintenance is performed via quorums and it is based on local conditions such as location and time. The proposals in [20, 21] do not consider any economic scheme and M-P2P architecture. P2P replication suitable for mobile environments has been incorporated in systems such as ROAM [35], Clique [37] and Rumor [18]. However, these systems do not incorporate economic schemes.

2.2 Incentive schemes for combating free-riding in MANETs

The proposals in [3, 4, 7, 8, 41] combat free-riding in MANETs. The work in [3] introduces a virtual currency to stimulate node cooperation. The work in [4] stimulates the nodes to forward messages by means of a simple counter-based mechanism at each node. The auction-based iPass [7] incentive scheme and the works in [8, 41] also provide incentives for relaying messages. However, these works do not consider M-P2P architecture and data item prices. Furthermore, they do not use price/incentives for data replication.

2.3 Non-incentive-based replication in M-P2P networks

The work in [32] has proposed a context and location-based approach for replica allocation in M-P2P networks. It exploits user mobility patterns, and considers load and different levels of replica consistency. The proposal in [31] has discussed both collaborative replica allocation and deallocation in tandem to facilitate optimal replication and to avoid 'thrashing' conditions. However, these proposals do not consider economic schemes.

2.4 Incentive schemes for combating free-riding in M-P2P networks

The proposals in [43, 44] discuss incentive schemes for combating free-riding in M-P2P networks. The work in [44] provides incentives to MPs for participation in the dissemination of reports about resources in M-P2P networks. Each disseminated report contains information concerning a spatial-temporal resource e.g., availability of a parking slot at a given time and location. The work in [44] has been extended in [43], which considers opportunistic resource information dissemination in transportation application scenarios. An MP transmits its resources to the MPs that it encounters, and obtains resources from them in exchange. The works in [43, 44] 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, our work considers on-demand services i.e., the query-issuing peer obtains only the data that it asks for (query-based approach). Furthermore, replication and incentives for encouraging relay peers to forward queries are not considered in [43, 44].

2.5 Payment schemes

A small study [30], which was conducted on users' motivation and decision to share resources in P2P networks, revealed that 50% of the questioned users would share more, if some materialistic incentives (e.g., money) are dispensed by the application. Herein lies the motivation for coupon-based systems like adPASS [42]. The works in [9, 12, 48] discuss how to ensure secure payments using a virtual currency. Another way proposed in [16] describes Coupons, an incentive scheme that is inspired by the eNcentive framework [36], which allows mobile agents to spread digital advertisements with embedded coupons among mobile users in a P2P manner.

Several non-repudiation [26, 38] systems, which can be incorporated to control the deceiving behavior of peers, have been developed. In many applications such as content distribution, the price can also be controlled by the service-providers [15]. MoB [6] 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. MoB also handles incentive management, user reputation management and accounting services. A bootstrap kind of mechanism can also be used in many applications [10]. Symella is a Gnutella file-sharing client for Symbian smartphones. It expects that illegal acts occur, such as interpolation or destruction of the distribution history to get incentives. Therefore, the distribution history attached to the e-coupon [7] is enciphered with a public-key cryptographic system so that users cannot peruse the distribution history. Furthermore, a message digest (MD) of the distribution history is embedded by digital-watermarking technology to check the validity of the history. Tribler [22] is a first attempt towards turning bandwidth into a global currency. Notably, the secure payment schemes discussed above are complementary to our proposal, but they can be used in conjunction with our proposal.

2.6 Economic schemes for resource allocation

The proposals in [29, 45, 46] discuss economic schemes for resource allocation in wireless ad hoc networks. However, they do not consider replication. Moreover, their focus is network-centric, while our focus is data-centric. Economic schemes have also been discussed for resource allocation in distributed systems [13, 14, 27]. However, they do not address M-P2P issues such as node mobility, frequent network partitioning and mobile resource constraints.

2.7 Schemes for static P2P networks

Replication schemes for static P2P networks [1, 11] and traditional replication strategies [25] for distributed systems do not consider peer mobility issues. Schemes for encouraging more peer participation in static P2P networks involve formal game-theoretic model for incentive-based P2P file-sharing systems [17], utility functions to capture peer contributions [19, 34], EigenTrust scores to capture participation criteria [24] and asymmetric incentives based on disparities between upload and download bandwidths [28]. However, these approaches are too static to be deployed in M-P2P networks since they assume peers' availability and fixed topology. Furthermore, they do not address replication and mobile resource constraints such as energy.

3 The E-ARL economic scheme for M-P2P data replication

This section discusses the economic incentive scheme of E-ARL for data replication in M-P2P networks.

Each data item has a *price* ρ (in *virtual currency*) that quantitatively reflects its relative importance to the M-P2P network as a whole. When an MP M_I accesses a data item (or replica) d hosted by an MP M_S, it pays the *price* ρ of d to M_S and a *commission* to each relay MP in the successful query path. Thus, M_I spends $(\rho + \sum_{i=1}^{r} R_i)$, where r is the number of relay MPs in the successful query path and R_i is the ith relay MP's commission. M_S and the ith relay MP earn ρ and R_i respectively.

Incidentally, if an MP is not able to pay the accessing cost, its query fails and it would not be able to access its queried data item. This is in consonance with the overall objective of E-ARL i.e., incentivizing free-riders to provide services (e.g., replica hosting and relay services). Thus, E-ARL ensures that an MP has to provide service to earn adequate currency to be able to issue its own queries, thereby facilitating better network performance due to proactive participation and collaboration of the MPs in hosting data and in relaying messages. Observe that on the other hand, if our scheme allowed peers to access data items without paying for the access, the free-riders would have little or no incentive to provide service to the network in terms of replica hosting as well as relay services.

If a query is answered *after* the query deadline, the query-issuing MP does not pay the data item price and the relay commissions. This is consistent with the timeliness requirements of M-P2P environments. Observe that there is no incentive for a data-providing MP to answer a query after the deadline. Hence, data-providing MPs estimate (based on past statistics concerning network history) whether their transmitted data item will reach the query-issuing MP within the deadline time, and based on their estimation, they decide whether or not to send the data. However, an MP cannot absolutely know in advance whether its answer will reach the query-issuing MP in a timely manner because of issues such as network congestion, relay node failures and network partitioning. Incidentally, even if a data-providing MP's query result reaches the query-issuing MP after the deadline, the data-providing MP does not lose any currency that it had previously earned. This is because E-ARL does not financially penalize data-providing MPs, whose answers arrive late, by subtracting currency from them for answering late. The implicit penalty is that the data-providing MP receives no currency for answering late. Furthermore, an additional implicit penalty arises because the MP expends its limited resources such as energy and bandwidth without being able to earn any currency to show for its expenditures. Furthermore, E-ARL precludes the possibility of refunds by enforcing that queries, which miss deadlines, entail no payments from the query-issuing peer.

We define the **revenue** of an MP as the difference between the amount of virtual currency that it earns (by hosting data items/replicas and relaying messages) and the amount that it spends (by issuing queries).

Load $L_{i,j}$ of an MP M_i at time t_j equals $(J_{i,t_j}/\sigma_i)$, where J_{i,t_j} is the job queue length of M_i at time t_j. σ_i is the normalized value of the service capacity of M_i. $\sigma_i = (\sigma_{M_i}/\sigma_{\min})$, where σ_{M_i} is the service capacity of M_i and $\sigma_{\min}$ is a low service capacity. We have used the minimum service capacity among all the MPs as $\sigma_{\min}$. Since σ_{M_i} is hardware-dependent, σ_i is fixed for a given MP. Notably, our definition of load takes into account the fact that service capacities of the MPs may differ. Incidentally, memory space and available bandwidth of MPs, and data item sizes may vary.

Notably, we define the service capacity in a normalized way because such normalization would yield values of service capacity within a closed range. As we shall see later in Equation 1, the price of a data item d also depends on the service capacity of the MP, which serves the query related to d. Consequently, normalization of service capacities ensures that the prices of data items also would fall within a closed range. If an MP's service capacity is unusually high or unusually low, and if we do not normalize the service capacity of an MP, the discrepancies among the prices of data items may become too high exclusively due to the significant variations in the service capacities of the MPs. Thus, the normalized values of service capacities are more meaningful than absolute values primarily from the perspective of keeping the data item prices within a closed range.

Observe that the normalization does not have any influence on relative prices of data items because it is performed in the same manner for the purpose of computing the price of every data item. As a single instance, given any two data items d_1 and d_2, the ratio of their prices would remain the same, regardless of whether normalization is performed.

Each MP maintains recent read-write logs (including timestamps) of the data items and replicas hosted at itself. Each MP uses this information for computing the prices of the data items and replicas stored at itself. Each data item d is owned by only *one* MP, which can update d *autonomously* anytime; other MPs cannot update d. It is the data item owner's responsibility to inform replica holders about the consistent values using mechanisms such as lazy updates (which can run in the background). There are many ways in which replicas can be made consistent in an M-P2P environment, and such discussion has appeared in [21]. Therefore, the consistency management discussion is out of scope w.r.t. the focus of this paper. Each MP guarantees the latest version as it exists at that time, and later a query-issuing MP can compare the versions to determine the latest version and the corresponding price. If an MP is only looking for the consistent version at that instance, it can issue the query to only the data item owner. Replica holders can also pull the latest version from the owner in case an application is looking for only the consistent version.

3.1 Price of a data item

Now we examine the factors influencing the price ρ of a data item (or replica) d. Recent **access frequency of** d relates to its popularity, hence d's price increases with increasing access frequency. This also prevents the energy of d's host MP M_S from being quickly drained by too many requests due to real-time needs of several users. This is because lesser number of MPs would be likely to request d from M_S due to the higher price. The larger the **number of MPs served by** d, the greater is d's importance to the network as a whole. Hence, to ensure *fairness*, d's price increases as it serves requests originating from more MPs. Notably, this is in contrast with existing works [21], which consider only the total access frequencies of data items without taking into account the number of MPs served by a given data item.

Observe that the price of a data item d increases with the number of queries for d, even though not all queries are processed successfully. This is because we have used the economics of market demand for determining data item prices. Under the economics of market demand, the number of queries for a given data item can be a reasonable indicator of its market demand, regardless of whether the query was eventually processed successfully. For the economics of market demand to apply appropriately in our case, we have also normalized the number of queries by considering the origin of requests for a given data item, as mentioned earlier. This ensures that the same peer issuing multiple queries for a given data item d cannot artificially 'inflate' the price of d. Notably, there could also be other approaches for computing data item prices.

Replica consistency and **query response time** relate to the *quality of service*. Hence, higher replica consistency implies higher price. Faster response times for queries on d command higher price, given the timeliness requirements of M-P2P applications. Given a query Q pertaining to d, which is hosted by MP M_S, response time τ equals $(T_W + T_D + T_{\mathrm{delay}})$, where T_W is the waiting time spent by Q in M_S's job queue. T_D is the download time for d, and T_{delay} is the path delay. T_W depends on

M_S's current **load**. T_D depends upon the **bandwidth** allocated by M_S for d's download, which is related to M_S's total bandwidth and the number of concurrent access requests to M_S. $T_{\text{delay}} = (\sum_{i=1}^{n_{\text{hop}}} (R_{\text{Size}}/B_i))$, where n_{hop} is the number of 'hops' between M_S and the query-issuing MP M_I, R_{Size} is the size of the query result and B_i is the bandwidth between the MPs at the ith hop. Thus, T_{delay} considers the connectivity of the query-serving host MP M_S.

Notably, our proposed approach 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 [5, 40] can be used in conjunction with our proposed approach.

As the remaining **energy** of an MP decreases, the price of accessing items at that MP increases. Thus, users would be less likely to access items at low-energy MPs if they are able to obtain them at higher-energy MPs, where item prices would be lower. This conserves the energy of low-energy host MPs and prevents them from dying out, thereby facilitating network connectivity. Given the ephemerality of M-P2P environments, data items often *expire* after a given time-frame. Data items with higher **time-to-expiry** command higher price due to their better revenue-earning potential over a longer period of time.

Data item prices increase with decreasing **number of replicas** due to lower supply w.r.t. demand. However, when the number of replicas of d falls below a certain threshold, d's price decreases. This anomaly occurs because quality of service (i.e., response time and data availability) is one of the factors influencing data item prices. Hence, when too few replicas exist, quality of service for queries on d decreases significantly, thereby decreasing the price.

Based on the factors discussed above, an MP M_S, which hosts a data item d, computes d's price in two steps. First, M_S computes ρ_{rec}, which is the price of d based on the accesses to d at M_S during the most recent replica allocation period. Second, M_S uses moving averages of ρ_{rec} over a fixed number of replica allocation periods to compute the price ρ of d. This is necessary because ρ_{rec} may not always be able to reflect the true importance of d to the network (e.g., when spurious 'spikes' in d's access frequency occur). Table 1 summarizes the notations used in this paper.

Computation of ρ_{rec}: M_S sorts the MPs in *descending* order of their access frequencies for d during the most recent replica allocation period i.e., the first MP in this order made the most accesses to d. Given this order and using the notations in Table 1, M_S computes ρ_{rec} of d.

$$\rho_{\text{rec}} = w \sum_{i=1}^{N_{MP}} (n_i \times C_i \times BA_{M_S} \times Ex) \,/\, (E_{M_S} \times (N_R + 1) \times (J_{M_S,t_j}/\sigma_{M_S})) \quad (1)$$

where the weight coefficient w equals (N/N_{MP}), where N is the number of different MPs which queried the data item d e.g., $N = 5$ means that 5 different MPs queried d. N_{MP} is the total number of MPs. Thus, ρ_{rec} increases with increase in the number of MPs served by d. In essence, the weight coefficient w ensures fairness in serving multiple MPs.

The replica consistency factor $C_i = 1$ for queries answered by M_S's own data items, which are always consistent. For queries answered by replicas hosted at M_S,

Table 1 Summary of notations

Notation	Significance
d	A given data item
M_I	MP which issues a query for a given data item d
M_S	MP that hosts a given data item d or d's replica and serves requests for d or d's replica
ρ_{rec}	Price of d during most recent allocation period
ρ	Moving Average Price of d across multiple allocation periods
N_{MP}	Number of MPs which accessed d
n_i	Number of access requests for d originating from a given MP i
w	Weight coefficient for fairness in serving multiple MPs
C_i	Average (replica) consistency with which queries (on d) originating from MP i were answered
BA_{M_S}	Average bandwidth allocated by MP M_S for d's download
Ex	Time-to-expiry of d
E_{M_S}	Remaining energy of host MP M_S
N_R	Number of existing replicas of d
J_{M_S,t_j}	Job queue length at query serving MP M_S during time t_j
σ_{M_S}	Service capacity of query serving MP M_S

we consider three different levels of replica consistency, namely *high*, *medium* and *low* [32]. C_i is assigned values of 1, 0.5 and 0.25 for high, medium and low consistency respectively. Each MP maintains a table $T_{\epsilon,C}$, 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. Thus, C_i is computed using $T_{\epsilon,C}$, which is replicated at each MP and is same for each MP.

Bandwidth BA_{M_S} equals (T_B/N_a), where T_B is the sum of all the bandwidths that M_S allocated to MP i over each of the times when MP i accessed d at M_S. N_a is the total number of access requests that MP i made for d. Observe how (1) considers item expiry times Ex, host MP energy E_{M_S} and host MP load i.e., job-queue length J_{M_S,t_j} (at time t_j) normalized by MP service capacity σ_{M_S}. Furthermore, the total number of copies of d in the M-P2P network equals the number N_R of replicas in addition to the original data item, which explains the term $(N_R + 1)$ in (1).

Examples of computing ρ_{rec} from (1) follow. Suppose $N_{MP} = 50$; for all i, $C_i = 1$; $BA_{M_S} = 50$ units, $E_{M_S} = 50$ units, $Ex = 10$ minutes, $N_R = 1$ and $J_{M_S,t_j}/\sigma_{M_S} = 1$. If a single MP makes 100 accesses to d, $n_i = 100$ when $i = 1$ (and 0 otherwise), hence $\rho_{\text{rec}} = 10$. However, if 4 MPs make 25 accesses each to d, $n_i = 25$ for $i = 1$ to 4 (and 0 otherwise), hence $\rho_{\text{rec}} = 40$. On the other hand, if a single MP makes 100 accesses to d, and E_{M_S}'s value is changed to 10 units, $\rho_{\text{rec}} = 50$. Observe across the above examples, how ρ_{rec} increases as d serves more MPs (even though d's total access frequency remains same) and how it increases with decreasing energy of the query-serving MP. Similarly, we can understand how ρ_{rec} would be affected for the other terms in (1). Notably, priority items are generally higher-priced than normal items due to higher bandwidth consumption (to ensure fast download of data) and the possible need to quickly download priority items from a low-energy host MP (to satisfy timeliness requirements).

Computation of the moving average price ρ: To account for sudden fluctuations in accesses to any given data item d, its host MP M_S computes the Exponential Moving Average (EMA) price ρ of d. EMA gives higher weights to recent access patterns, hence it is appropriate for dynamically changing M-P2P access patterns. We do not use simple moving averages since they are not able to react quickly to changing access patterns. The computation of ρ according to the exponential moving average formula follows.[1]

$$\rho = ((\rho_{\mathrm{rec}} - EMA_{\mathrm{prev}}) \times 2/(N+1)) + EMA_{\mathrm{prev}} \tag{2}$$

where EMA_{prev} is the EMA that was computed for the previous replica allocation period, and N is the number of replica allocation periods over which the moving average is computed. Since M_S only needs to store the value of EMA_{prev} (as opposed to the values of price data for the entire period being averaged), its memory space usage is optimized. Results of our preliminary experiments indicate that $N = 5$ is a reasonably good value for our application scenarios.

3.2 Commission for relay service

Now let us examine the factors influencing the commission of relay MPs. As the remaining **energy** E of a relay MP R decreases, R charges higher relay commission to conserve its energy. Thus, to minimize relay costs, query issuers would be less likely to choose relay paths comprising low-energy MPs, thereby facilitating network connectivity. As the **size** of the relayed message increases, relay commission also increases due to higher energy and bandwidth consumption of the relay MP.

We define the **connectivity** of an MP as the number of its one-hop neighbors. **Spatial density** of an MP is defined as the number of MPs currently moving within the grid in which the MP is located. (Recall that space is divided into a rectangular grid structure, the size of the rectangles being application-dependent.) MPs in regions of higher spatial density with higher connectivity are likely to have greater proximity to data sources and can provide better service due to multiple path options for relaying queries/data. Hence, the relay commission increases with increase in the connectivity and spatial density of a relay MP. This also ensures that critically located MPs (e.g., biconnected components), the availability of which influence network connectivity significantly, are less frequently used for relaying messages since they would charge higher relay commission.

Based on the factors discussed above, a relay MP R computes its commission ρ_{Relay} as follows.

$$\rho_{\mathrm{Relay}} = (size \times \lambda \times \mu)/E \tag{3}$$

where $size$ is the size of the relayed message, λ is the spatial density of the (current) region of movement of R, μ is R's connectivity and E is R's remaining energy.

[1] Although the formula does not have any exponential term, this is the standard and widely used formula for computing exponential moving averages.

In contrast with existing works, relay commission for the same message may differ across hops. Incidentally, priority queries generally incur higher relay commission than normal queries due to the possible involvement of low-energy relay MPs with higher connectivity in the query path since priority queries require query paths with short response times.

Notably, there is at least some correlation between the values of λ and μ. However, we chose to use both λ and μ in (2) because we believe that each of these parameters has reasonable significance in its own right so as to warrant individual treatment. As a single instance, MPs may be moving within the same spatial region, but they may not have connectivity to each other due to their respective transmission ranges being less than their respective distances from each other. In such scenarios, an MP's spatial density λ could be high, but its connectivity μ would be low.

Observe that this is in consonance with our application scenarios in which the respective transmission ranges of MPs are likely to be significantly less than that of the area of the spatial region under consideration. For example, in our application scenario for a group of archaeologists moving in a remote area of Egypt or a group of geologists moving in a remote Amazonian forest, the transmission range of existing mobile devices is likely to be significantly lower than that of the area of the spatial region under consideration. Other causes for separately treating λ and μ include communication failures among peers and bandwidth issues. For example, in the case of MPs using radio technology, the bandwidth gets divided among the number of users.

Although the spatial density information does not directly reflect MP connectivity, we also consider the spatial density information separately because MPs moving in a region of higher spatial density can be reasonably expected to have a higher likelihood of moving into the communication range of more MPs than MPs, which move in regions of low spatial density.

Notably, one might correctly argue that the relay commissions should not vary because the communication cost does not change in our architecture. However, in our model, we have variations in the relay commissions to take the issue of delay into consideration. For example, if a relay MP is in a congested path, its commission should decrease because it might potentially cause delays in relaying a given message. Although it is clearly not the relay MP's fault for being in a congested network path, we allow relay commissions to be variable in order to provide better incentives to relay MPs, which transmit messages with lower amounts of delays.

3.3 Revenue of an MP

An MP M earns virtual currency from accesses to data items and replicas that it hosts and by relaying messages. M spends its virtual currency by accessing items hosted by other MPs, and by paying commissions to the relay MPs corresponding to its queries.

Suppose M hosts p data items of its own and q replicas. Let the price of the ith data item and the ith replica be $\rho_{s_{d_i}}$ and $\rho_{s_{r_i}}$ respectively. Let $n_{s_{d_i}}$ and $n_{s_{r_i}}$ be the access frequencies of the ith data item and the ith replica respectively. Moreover, suppose M accesses v original data items and w replicas. Let the price of the ith data item and the ith replica be $\rho_{r_{d_i}}$ and $\rho_{r_{r_i}}$ respectively. Let $n_{r_{d_i}}$ and $n_{r_{r_i}}$ be the access frequencies of M for the ith original item and the ith replica respectively. Suppose M

relays m messages, the relay cost of the ith message being Rel_i. On the other hand, suppose M requires to pay commissions to n relay peers in the course of issuing different queries, the relay cost for the ith such relay MP being $RelReq_i$. The revenue ω of M is computed below:

$$\omega = \left(\sum_{i=1}^{p}(\rho_{s_{d_i}} \times n_{s_{d_i}}) + \sum_{i=1}^{q}(C_i \times \rho_{s_{r_i}} \times n_{s_{r_i}}) + \sum_{i=1}^{m} Rel_i\right) - \left(\sum_{i=1}^{v}(\rho_{r_{d_i}} \times n_{r_{d_i}}) + \sum_{i=1}^{w}(C_i \times \rho_{r_{r_i}} \times n_{r_{r_i}}) + \sum_{i=1}^{n} RelReq_i\right) \tag{4}$$

The first three terms in (4) represent M's earnings, while the other three terms depict M's spending. Thus, **revenue** of an MP is the difference between the amount that it earns and the amount that it spends. In the second and fifth terms of (4), C_i indicates the average consistency with which queries on the replicas were answered. C_i does not occur in the first and fourth terms since these terms concern an MP's own data items, which are always absolutely consistent.

When an MP joins the M-P2P network, the super-peer SP provides it with an initial small amount of revenue so that it can issue a few queries. However, once this initial revenue is exhausted, the MP will eventually have to provide service to the network, otherwise it will not be able to issue any further queries.

4 Query model of E-ARL

This section discusses the query model of E-ARL.

User queries Q are of the form $\{Q_{\text{id}}, (k_1, k_2, \ldots, k_n), \tau_{\max}, \mathit{Priority}, \mathit{DC}, \rho_{\max}\}$, where Q_{id} is the unique identifier of a query, and k_i are user-specified keywords e.g., if an M-P2P user requests the song 'Save the dance' by Ricky Martin, $k_1 =$ 'Save the dance' and $k_2 =$ 'Ricky Martin'. $\tau_{\max}$ is the *deadline* for query response from the user's perspective. M-P2P ephemerality necessitates query deadlines. Priority queries generally have shorter deadlines than normal queries. The values of *Priority* for priority queries and normal queries are 1 and 0 respectively. Recall that in Sect. 3, we discussed three levels of replica consistency, namely *high*, *medium* and *low*. *DC* represents the desired (replica) consistency level with which the user wants his query to be answered. Since the table concerning the replica consistency values is replicated across all the MPs, the user can easily specify the value of *DC* in his query. $\rho_{\max}$ is the *maximum price* that the user is willing to pay for obtaining the query result. $\rho_{\max}$ is higher for priority queries since such queries are costlier, as discussed in Sect. 3. Based on the queries which the user relays, he can generally gain some knowledge about these querying parameters, which guide him in specifying them.

Querying mechanism in E-ARL: A query-issuing MP M_I broadcasts its query Q for a data item d. Each query has a TTL (time-to-live) of 8 hops. If an MP M_S receiving Q contains d (or its replica), it puts its MP_{id} (unique identifier of an MP) into the query message and informs M_I about d's price and (replica) consistency. Otherwise, it just puts its MP_{id} into the query message, increments the number of hops in the query message, and forwards Q to its one-hop neighbors. Incidentally, M_S returns the price and consistency information of d to M_I only if it estimates that it can satisfy the $\tau_{\max}$ and $\rho_{\max}$ constraints. M_S estimates the time to answer Q based on the size of d, the bandwidth that it can make available for d and its knowledge of the previous history of the network. M_S gains such knowledge by examining queries which pass through itself as well as by periodically exchanging messages with its neighbors. M_S incorporates a safety margin of ϵ time units into its estimated response time to ensure query responses within the deadline. ϵ is essentially application-dependent.

When M_I receives messages from possibly multiple MPs, which host d or its replica, it lists the query paths associated with each of these messages. (Recall that each MP appends its MP_{id} to the query messages.) From this list of query paths, M_I selects one query path depending upon the total query path cost, the estimated query response time, the query deadline, the data consistency level and the query type (i.e., priority query or normal query). We define the **total query path cost TQPC** as $(\sum_{i=1}^{r} \rho_{\mathrm{Relay}_i} + \rho_d)$ where r is the number of relay MPs in the given query path, ρ_{Relay_i} is the relay commission of the ith relay MP, and ρ_d is d's price at the target MP, which hosts d or its replica. The values of ρ_{Relay_i} and ρ_d are computed as discussed in Sect. 3.

Let L_{QP} denote the list of possible query paths from M_I to the queried data item. Query paths, for which *TQPC* exceeds the user-specified maximum price $\rho_{\max}$, are deleted from L_{QP}. Query paths not satisfying $\tau_{\max}$ and desired replica consistency constraints are also deleted from L_{QP}. Observe that all the remaining query paths in L_{QP} satisfy the query constraints. For priority queries, M_I selects the query path with the shortest estimated response time from L_{QP}. Priority queries need to be answered *urgently*, hence M_I does not try to optimize *TQPC*. For normal queries, M_I sorts the query paths in L_{QP} in ascending order of *TQPC*, and selects the query path with lowest *TQPC* value. Observe the trade-off between query response time and *TQPC*.

After selecting the query path, M_I sends a message to the target host MP M_S of d in the selected query path to indicate its interest to download d from M_S. Then M_S transfers d to M_I through the relay MPs in the selected query path. Finally, M_I pays the price of d to M_S. M_I also pays the commissions to the relay MPs in the selected query path.

5 Adaptive Revenue-Load-based replica allocation in E-ARL

This section discusses the adaptive revenue-load-based replica allocation scheme in E-ARL using the economic scheme discussed in Sect. 3.

Each MP M maintains access statistics and price information concerning its own data items and the replicas that it hosts. These statistics guide M in selecting its own data items that need to be replicated and in deleting infrequently accessed and time-expired replicas. M also separately maintains statistics for priority items and normal items so that it can give preference to priority items for replica allocation. Access statistics are *periodically* refreshed to reflect *recent* accesses.

Now let us examine the interaction between the revenue and the load of an MP. An underloaded MP could have high revenue by serving only a few requests for high-priced data items, while not issuing any access requests of its own. An overloaded

MP could have low revenue due to serving a large number of requests for low-priced data items, while issuing several access requests for high-priced items. Thus, there is no direct correlation between MP revenue and load. E-ARL uses a parameter λ that can be tweaked to adjust the relative importance of revenue and load during replica allocation so that it can *adapt* to the needs of different types of applications.

Computation of λ uses normalized values of revenue and load to correctly reflect their relative weights. We define the normalized revenue R of an MP M as $M_{\text{Rev}}/\mathit{Total}_{\text{Rev}}$, where M_{Rev} is M's revenue and $\mathit{Total}_{\text{Rev}}$ is the sum of revenues of all MPs in the network. Similarly, normalized load L of M is defined as $M_{\text{Load}}/\mathit{Total}_{\text{Load}}$, where M_{Load} is M's load and $\mathit{Total}_{\text{Load}}$ is the sum of loads of all MPs. For every MP, we normalize further to make $(R+L)=1$, by multiplying the value of $(R+L)$ of every MP by a real number k, whose value may differ across MPs. We shall henceforth use $R+L=1$ to reflect the above normalization. Computation of λ for different cases follows.

Case 1: Revenue and load are both assigned equal weight: E-ARL computes a function $f=R\times L=R\times(1-R)$. Differentiating f w.r.t. R, we obtain $df/dR=R(-1)+1-R=1-2R$. To find f's maximum value, the derivative (df/dR) is set to zero. Hence, $1-2R=0\Rightarrow R=1/2$. Since $R+L=1$, $L=1/2$. Thus, f's maximum value occurs when $R=L=1/2$. Hence, $\lambda=(R+L)$.

Case 2: Revenue is assigned higher weight than load: E-ARL computes the function $f=R^2\times L=R^2\times(1-R)$. Hence, $df/dR=R^2(-1)+2R(1-R)=R(-3R+2)$. To find f's maximum value, we set the derivative (dp/dR) to zero. Since $R\neq 0$, $-3R+2=0\Rightarrow R=(2/3)$. Hence, $L=1/3$. Thus, the maximum value of f occurs when $R=2L$, hence $\lambda=2R+L$.

Case 3: Revenue is assigned lower weight than load: In this case, $f=R\times L^2=R\times(1-R)^2$ Hence, $df/dR=R(2)(1-R)(-1)+(1)(1-R)^2=(1-R)(1-3R)$. To find f's maximum value, we set the derivative (df/dR) to zero. Since $L\neq 0$, $1-R\neq 0$, hence $1-3R=0\Rightarrow R=(1/3)$. Hence, $L=2/3$. Thus, the maximum value of f occurs when $L=2R$. Thus, $\lambda=R+2L$.

5.1 Factors for effective replica allocation in E-ARL

The super-peer SP makes the replica allocation decisions. This is possible because *periodically*, each MP sends its current (x,y) coordinates, its revenue value, the prices and access frequencies of items hosted at itself, its load, energy, bandwidth and available memory space status to SP. SP collates the (x,y) coordinate information of MPs to estimate the network topology for facilitating replica allocation.

E-ARL prefers priority items for replication due to urgency reasons. Given the higher prices of priority items, MPs have greater incentive to host replicas of priority items as they can earn more revenue. E-ARL avoids replicating priority and normal items with high update frequencies due to the high communication overhead required for maintaining their replica consistency. E-ARL avoids replication at low-energy MPs to facilitate network connectivity by prolonging the lifetime of low-energy MPs since answering queries on replicas would have quickly drained their limited energy. Moreover, low-energy MPs cannot effectively maintain replica availability as they will die out once their limited energy is depleted. E-ARL avoids replication at overloaded MPs, and MPs with low bandwidth to ensure acceptable query response times. MPs with inadequate memory space are also avoided for replication.

E-ARL aims at allocating replicas of relatively higher-priced data items (e.g., priority items) to MPs with low values of λ. This facilitates both revenue-balance and load-balance since low value of λ implies relatively lower MP revenue and lower MP load. Revenue-balancing becomes a necessity because gross imbalance of revenues across the MPs may result in undesirably low revenues for some of the MPs. This could prevent these MPs from obtaining their desired services (i.e., issuing access requests) from the network, thereby decreasing overall network participation. On the other hand, load-balancing becomes a necessity to reduce query waiting times, thereby optimizing query response times.

5.2 Selection of candidate data items for replication

Each MP M has priority items and normal items. All priority items (except those with high update frequencies) are candidates for replication due to their urgent response time requirements. Among the low update-frequency normal items, M selects those items, whose access frequencies exceed the average access frequency ψ for normal items, as candidates for replication. ψ equals $((1/N_d)\sum_{k=1}^{N_d}\eta_k)$, where N_d is the number of M's normal items and η_k is the kth item's access frequency. Note that E-ARL replicates only items with relatively low update frequencies. M separately sorts the set of priority items and the set of normal items in descending order of their access frequencies since more popular items are better candidates for replication. M sends these two sets to SP as its candidate items for replication.

Upon receiving these two sets from different MPs, SP combines all sets of priority items and normal items into two different large sets, namely *Priority* and *Normal* respectively. SP separately sorts *Priority* and *Normal* in descending order of their item access frequencies. Finally, SP appends the sorted set *Normal* at the end of the sorted set *Priority*, thereby creating a new set L_{Rep}, which constitutes the set of candidate data items for replica allocation. As all priority items are placed higher in L_{Rep} than normal items, replicas for priority items are allocated before normal items. Notably, replicas may not necessarily be allocated for every item in L_{Rep} due to memory space constraints. Now let us see how SP determines the number of replicas for each item in L_{Rep}.

5.3 Determining the number of replicas for a data item

E-ARL aims at maintaining acceptable query response times so that MPs do not lose revenues due to failure of queries with response time constraints. Periodically, SP receives access statistics information from the MPs, and this information includes access failures and the average response times for accessing each data item at each MP. SP collates this information to estimate the access failures and the average response time $Resp_{\text{avg}}$ required to access a data item. In case all queries on a data item d had been successful in satisfying the response time constraint, SP takes no action. Otherwise, SP checks the response time constraints on the failed queries on d

to determine the failed query (on data item d), which had the lowest response time constraint $Resp_{\text{min}}$. SP computes the optimal number K'_d of replicas of d as follows. $K'_d = K_d(1 + \lceil(Resp_{\text{avg}} - Resp_{\text{min}})/Resp_{\text{avg}}\rceil)$, where K_d is the existing number of replicas of d. Hence, $(K'_d - K_d)$ additional replicas of d need to be created. Thus, SP estimates the number of replicas of data items based on response time constraints posed by queries on them. SP does not guide the MPs in performing replica deallocation since MPs autonomously delete infrequently accessed and time-expired replicas.

6 Algorithms for bid-based economic replication in E-ARL

This section discusses *bid-based* economic replica allocation in E-ARL. An MP, which provides its own data item for replication, is designated as a **provide-MP**, while an MP, which hosts a replica, shall be referred to as a **host-MP**. We present the replication algorithms executed by SP and the prospective host-MPs. MPs send their bids to SP for replicas, and SP allocates replicas of higher-priced items to higher-bidding MPs with relatively low revenue and load (as characterized by the parameter λ) to facilitate both revenue-balance and load-balance.

Each host-MP needs to make a *one-time payment* to the provide-MP for the corresponding replica to *buy* the replica. This provides an economic incentive to the provide-MPs to allocate replicas for their data items, thereby improving data availability. Since the revenue-earning potential of a replica depends on the load (imposed by queries on the replica), and the available bandwidth and current load of the host-MP, the higher-bidding MPs are more likely to be the ones with more available bandwidth and lower load. Thus, higher-bidding MPs would be likely to provide better service for replica hosting in order to recoup their initial investment in buying the replica.

After SP makes its replica allocation decision, it sends a message to the provide-MP of data item d to replicate d at the selected host-MPs, whose bids succeeded. The provide-MP sends d to these host-MPs, which are allowed to make the one-time replica buying payment to the provide-MP, after they have earned some revenue by hosting the replica. Observe that if prospective host-MPs had to make the replica buying payment *before* hosting the replica, network participation would likely decrease since some of the MPs may not have adequate revenue to make the payment. Thus, E-ARL facilitates increased MP participation and collaboration.

If an MP's bid succeeds, it is *required* to host the items that it has bid for. This facilitates in avoiding extra communication overheads associated with reallocations of replicas by SP. Hence, each MP has a **total bid potential capacity** γ, which is the maximum amount of currency that it can bid for the purpose of hosting replicas at itself. γ equals $((E_M \times BA_M)/(Rev_M \times Load_M))$, where E_M, BA_M, Rev_M and $Load_M$ are the normalized values of the energy, bandwidth, revenue and load of MP M respectively. MPs with high energy and bandwidth, and low load can provide better service, hence they can earn more revenue by hosting replicas. Thus, their bid potential capacity γ is also higher. γ decreases with increase in M's revenue to ensure that MPs with higher revenue have lower bid capacity potential. This facilitates revenue-balancing by giving preference to lower-revenue MPs for hosting replicas, thereby improving MP participation and collaboration.

Algorithm *E-ARL_Replica_Allocation_SP*

L_{Rep}: List of data items that are candidates for replication
(1) Send a broadcast message to the MPs containing the prices, sizes and expiry times of each data item of L_{Rep}
(2) for each data item d in L_{Rep}
(3) Receive bids from the prospective host-MPs and add them to Bid_List
(4) From Bid_List_d, eliminate the bids from MPs, whose energy is below the threshold TH_E
(5) From Bid_List_d, eliminate the bids from MPs, whose load is above the threshold TH_L
(6) From Bid_List_d, eliminate the bids from MPs, whose bandwidth is below the threshold TH_B
(7) if Bid_List_d is empty
(8) Do not allocate any replica for d
(9) else
/* N_R is the number of required replicas for d, N_B is the number of remaining bids for d */
(10) if $N_B <= N_R$
(11) Select all the MPs corresponding to the bids in Bid_List_d into a list *Selected*
(12) else
(13) Select from Bid_List_d the top-k bids into a list L_d
(14) From L_d, select N_R MPs with the lowest values of λ for storing the replica of d into a list *Selected*
(15) for each MP M in *Selected*
(16) Send message to the provide-MP of d to replicate d at M
(17) Send message to M to make the one-time payment for buying the replica to the provide-MP of d
(18) Recompute ρ of d /* ρ depends on number of replicas */
end

Fig. 1 E-ARL replica allocation algorithm executed by SP

Replica allocation algorithm of SP: SP *periodically* allocates replicas for the candidate set of items in L_{Rep} using a *bid-based* scheme. (The creation of L_{Rep} has been discussed in Sect. 5). Figure 1 depicts the replica allocation algorithm executed by SP. In Line 1 of Fig. 1, SP's broadcast message facilitates prospective host-MPs in computing their respective bids for the data items. SP sends only *one* broadcast message for all the items in L_{Rep} to optimize energy and bandwidth usage. In Lines 4–6, observe how SP considers energy, load and bandwidth constraints. TH_E and TH_L are the average energy and average load respectively among the MPs in the network. TH_B is an application-dependent bandwidth threshold, which essentially depends upon data item sizes in the application. As Lines 9–14 suggest, given the number N_B of bids for d and the number N_R of required replicas for d, the following two cases arise:

1. $N_B \leq N_R$: SP allocates replicas to all the MPs corresponding to the remaining bids.
2. $N_B > N_R$: SP first selects the top-k bids from the remaining bids based on the bid values. Here, k equals $(2 \times N_R)$. Then, among these selected bids, SP allocates replicas to the N_R MPs (corresponding to the bids) with the lowest values of λ.

Algorithm ***E-ARL_Replica_Allocation_HostMP***

L_{Rep}: List of data items that are candidates for replication

γ: Total bid capacity potential of the host-MP

Mem: Remaining memory space of the MP

(1) Receive the broadcast message (from SP) containing the prices, sizes and expiry times of all data items in L_{Rep}

(2) Select all data items of L_{Rep} into a list To_Bid

(3) From To_Bid, delete data items whose size exceeds its own remaining memory space

(4) for each item d in To_Bid

(5)   Estimate the future access frequency of d's replica at itself

(6)   Compute the revenue-earning potential ω of d at itself

(7)   Compute the bid value β for d

(8) Sort the data items in To_Bid in descending order of the bid value β

(9) for each data item d in To_Bid

(10)   if $(Mem > 0)$

(11)   if $\beta > \gamma$

(12)   **break**

(13)   else

(14)   if $(Mem > size_d)$ /* $size_d$ is the size of d */

(15)   Add d to Bid_Select

(16) $\gamma = \gamma - \beta$

(17) $Mem = Mem - size_d$

(18) if Bid_Select is non-empty

(19)   for each data item d in Bid_Select

(20)   Send bid to SP with β_d as the bid value /* β_d is the bid value for d */

end

Fig. 2 E-ARL replica allocation algorithm executed by a prospective host-MP

In Line 17, the price ρ of d is recomputed since ρ depends upon the number of existing replicas.

Replication algorithm of prospective host-MPs: Figure 2 depicts the replica allocation algorithm executed by a prospective host-MP M. In Fig. 2, γ is the total bid potential capacity of M, which is computed as discussed earlier in this section. In Line 3, observe how M considers memory space constraints while bidding. In Lines 5–6, M estimates the future access frequency of data item d's replica at itself by examining the access statistics for the queries that it relays as well as the failed queries for d at itself. M computes the revenue-earning potential ω of d at itself as $(\rho_d \times \eta_d)$, where ρ_d is the estimated price of d at M (computed according to (1)) and η_d is the estimated future access frequency of d at M.

In Line 7, M computes its bid value β for a given data item d as $(0.5 \times \omega)$. Observe that M's bid value is 50% of d's estimated revenue-earning potential at itself. This is because M needs to make a *one-time payment* of $(0.5 \times \omega)$ to *buy* the replica from the corresponding provide-MP. This acts as an incentive for provide-MPs to allocate

Table 2 Parameters of our performance study

Parameter	Default value	Variations
No. of MPs (N_{MP})	100	20, 40, 60, 80
Percentage of free-riders	70%	
Zipf factor (ZF)	0.9	0.1, 0.3, 0.5, 0.7
Allocation period TP (10^2 s)	5	10, 15, 20
Queries/second	10	
Percentage of priority queries	20%	40%, 60%, 80%, 100%
Bandwidth between MPs	28 to 100 Kbps	
Probability of MP availability	50% to 85%	
Initial energy of an MP	90 000 to 100 000 energy units	
MP service capacity	1 to 5 service capacity units	
Size of a data item	50 to 350 Kb	
Time-to-expiry of a data item	3 to 7 minutes	
Memory space of each MP	4 to 6 MB	
Speed of an MP	1 to 10 meters/s	
Size of message headers	220 bytes	

replicas, while providing the other MPs an opportunity to earn revenue by hosting replicas, which they do not own. Furthermore, this improves the data availability due to increased participation and collaboration of MPs.

In Lines 8–17, observe how M gives preference to items with higher bid values to maximize its revenue since such items also have higher revenue-earning potential. M sorts the items in descending order of their bid value β, and traverses this sorted list to select items to bid for, until either its bid capacity potential is exhausted or its remaining memory space becomes inadequate. In Lines 18–20, M sends its list of selected items to bid for, and their respective bid values, to SP.

7 Performance evaluation

Our experiments consider 100 MPs and 1 SP moving according to the *Random waypoint model* (RWP) [2] in a region of area 1000 metres × 1000 meters. RWP is appropriate for our application scenarios, which consider random movement of users e.g., salesmen moving in a neighborhood (or in a shopping mall) generally move randomly without following any specific mobility pattern.

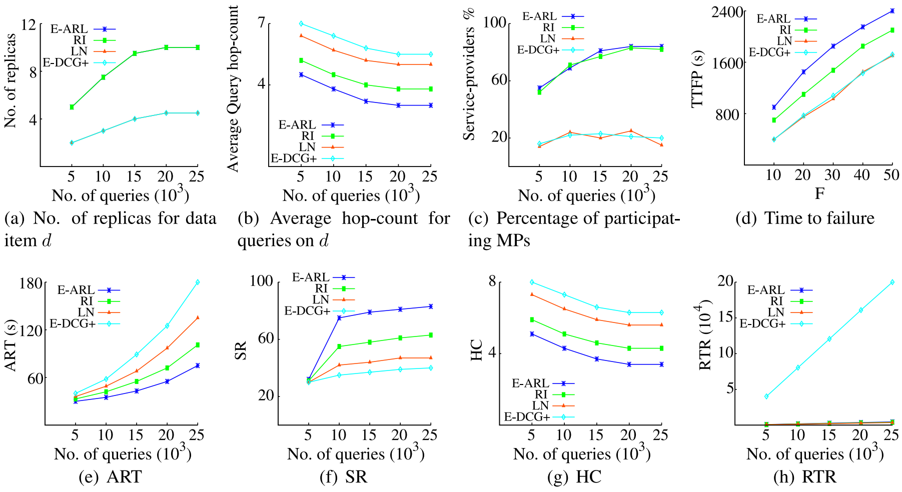
Table 2 summarizes our performance evaluation parameters. As Table 2 indicates, 70% of the MPs are free-riders i.e., they will not host replicas in the absence of incentives, while the other 30% will host replicas even without any incentives. Notably, in real-world P2P networks, the percentage of free-riders generally exceeds 70%. Each MP owns and stores 8 data items, of which 4 are *priority items* and 4 are *normal* items. Each query is a request for a single data item. For query routing purposes, we have used the AODV protocol. 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. Thus, the available memory space for hosting replicas may differ across MPs due to variations in MP memory space and data item sizes.

TP stands for 'replica allocation Time Period'. *Periodically*, every TP seconds, SP decides whether to perform replica allocation. Similar to existing works [21], we assume that network topology does *not* change significantly during replica allocation since it requires only a few seconds. In all our experiments, 10 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 (except SP) is a circle of 100 meter radius.

Performance metrics are **average response time** (**ART**) of queries, **query success rate** (**SR**), **query hop-count** (**HC**) and **traffic for replica allocation** (**RTR**). $\mathrm{ART} = (1/N_Q)\sum_{i=1}^{N_Q}(T_f - T_i)$, 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. ART includes the download time, and is computed only for the successful queries. $SR = (N_S/N_Q) \times 100$, where N_S is the number of queries that were answered successfully and N_Q is the total number of queries. 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. 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. Replica allocation traffic **RTR** is defined as the total hop-count for replica allocation during the course of the experiment.

We compared our proposed incentive-based **E-ARL** scheme with three other schemes, namely the non-incentive **E-DCG+** [21], **RI** (**Revenue-based Incentive scheme**) and **LN** (**Load-based Non-incentive scheme**). We adapted the **E-DCG+** approach [21] to our scenario. As discussed in Sect. 2, E-DCG+ is a non-incentive and non-economic approach, 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 *dynamic* replica allocation in mobile ad-hoc networks. None of the existing proposals on economic issues addresses replication in M-P2P networks.

Recall that λ adjusts the relative importance between revenue R and load L during replica allocation. **RI** is an adaptation of E-ARL in which $\lambda = R$. Thus, RI is an economic incentive scheme in which replica allocation is performed based on MP revenue, without considering MP load. **LN** is a non-incentive adaptation of E-ARL in which $\lambda = L$. Replica allocation in LN is performed based on MP load, without considering MP revenue. Unlike E-ARL and RI, LN does not provide any incentives to MPs for hosting replicas since it is non-economic. Observe how comparing E-ARL with RI and LN enables us to gain more insight into E-ARL's adaptive nature by better exploration of the interaction between MP revenue and load. E-ARL showed comparable performance for different values of λ, hence we present here the results of E-ARL corresponding to equal weight for both revenue and load (i.e., $\lambda = R + L$, as discussed in Sect. 5). In essence, E-ARL performs comparably as long as revenue and load are both considered.



(a) No. of replicas for data item *d* (b) Average hop-count for queries on *d* (c) Percentage of participating MPs (d) Time to failure

(e) ART (f) SR (g) HC (h) RTR

Fig. 3 Performance of the E-ARL economic scheme for replica allocation

7.1 Performance of the E-ARL economic scheme for replica allocation

Figure 3 depicts the results of our experiments using default values of the parameters in Table 2.

The results in Figs. 3a and 3b concern an experiment in which we observed the number of replicas and the corresponding querying hop-counts for a *single* 'hot' data item *d* over a period of time, *d* being selected randomly from the top 10% hottest data items. The incentive-based E-ARL and RI schemes create significantly more replicas of *d* than the non-incentive LN and E-DCG+ schemes because incentives encourage more MPs to host replicas, thereby implying more available memory space for replica allocation. E-ARL and RI create comparable number of replicas because they provide similar incentives to MPs for hosting replicas. LN and E-DCG+ create comparable number of replicas as both of them essentially allocate replicas based on access frequencies. For all four approaches, the number of replicas initially increases over time due to replica allocations at MPs with adequate available memory space for hosting replicas. However, the number of replicas eventually plateaus due to competition among replicas for MP memory space.

Figure 3b indicates the average number of hop-counts required for querying the same data item *d* during different periods of time. E-ARL and RI incur lower querying hop-counts than LN and E-DCG+ since they create more replicas for *d*, as discussed for Fig. 3a. More replicas generally decrease the querying hop-count. Interestingly, even though E-ARL and RI create comparable number of replicas for *d*, RI incurs more hop-counts than E-ARL. This occurs because RI is oblivious to MP load, hence it may allocate replicas to overloaded MPs, thus failing to satisfy query deadlines due to longer waiting times at their job queues. Thus, RI requires more hops to find *d* at an underloaded MP so that the query deadline can be satisfied. In contrast, E-ARL does not allocate replicas to overloaded MPs since it considers MP load. Incidentally, E-DCG+ incurs more hop-counts than LN essentially due to the same reason i.e., unlike LN, E-DCG+ does not consider MP load. For all four approaches, the querying hop-count initially decreases and then eventually plateaus. The initial decrease is in response to creation of more replicas, while the plateau occurs due to the eventual plateau in the number of replicas.

The results in Figs. 3c–3h correspond to an experiment using default values of the parameters in Table 2. Unlike the experiments for Figs. 3a and 3b, which considered only a single data item, this experiment concerns all data items in the network. As replica allocation time period TP is 500 seconds and query interarrival rate is 10 queries/s, there are 5000 queries for every time period.

The results in Fig. 3c depict the percentage of service providers in the M-P2P network for all four approaches during different time periods. An MP is regarded as a service provider during a time period TP if it hosts a data item/replica that is accessed at least once during TP. Interestingly, the participation levels in E-ARL and RI vary irregularly to a certain extent. This is because even though an MP may host data items/replicas to provide service, it may not receive any queries for them during a given time period. Due to the same reason, the percentage of service-providers also varies irregularly for LN and E-DCG+. Overall, participation levels in E-ARL and RI are comparable since both schemes offer similar incentives for MP participation. For E-ARL and RI, the MPs initially have little revenue, but as more queries are issued, MP revenues increase, thereby increasing MP participation levels upto the point where the majority of the MPs are providing service to the network essentially due to their economic replication scheme. Observe that participation levels in LN and E-DCG+ are comparable essentially due to their non-incentive nature.

Figure 3d shows the effect of network connectivity maintenance by E-ARL. We define time-to-failure of a percentage of nodes (TTFP) as the time duration at which $F\%$ of the nodes fail due to depletion of energy. As F increases, TTFP also increases for all the approaches because larger number of nodes take more time to fail. E-ARL's economic scheme facilitates preservation of the energy of low-energy MPs, thereby facilitating 'energy-balancing' and consequently, improved network connectivity. Hence, E-ARL provides better TTFP than the other approaches. RI performs worse than E-ARL since it does not consider load, thereby causing some of the MPs to become overloaded by access requests and consequently, running out of energy. RI performs better than E-DCG+ and LN due to more MPs participating because of incentives. E-DCG+ and LN perform comparably since both are non-incentive-based and their TTFP varies irregularly due to node failures occurring randomly in the absence of any 'energy-balancing'.

Figures 3e–3h indicate that ART increases over time for all the approaches due to the highly skewed query distribution (i.e., zipf factor = 0.9), while SR initially increases due to replication and then plateaus due to competition among replicas for memory space. HC initially decreases for all the approaches due to replication, and eventually plateaus due to the plateau in the number of replicas. E-ARL outperforms RI due to its better load-balancing capability. Moreover, since RI allocates replicas solely based on revenue, most MPs try to *greedily* host replicas of higher-priced data items, thereby increasing the ART and decreasing the SR for lower-priced data items, which may have high access frequency. However, RI outperforms both E-DCG+ and LN because of more MP participation due to incentives. LN performs better than E-DCG+ due to load-balancing.

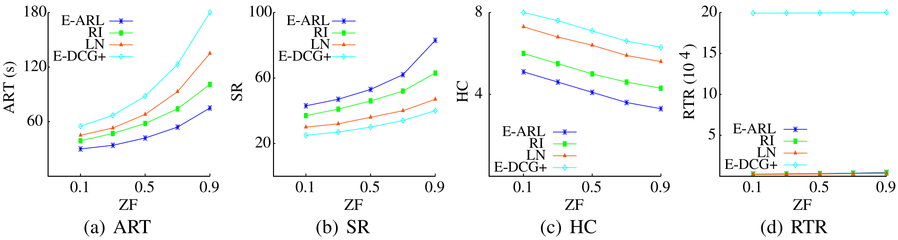


Fig. 4 Effect of variations in the workload skew

E-ARL outperforms E-DCG+ due to more MP participation and better load-balancing. Additionally, E-ARL creates larger number of replicas for many different data items depending upon data item prices. E-ARL would create a replica for a data item d, which is accessed by a large number of MPs, even if d's total access frequency is low, in which case E-DCG+ would not create any replica. Interestingly, even though Figs. 3a and 3c suggest that the number of replicas and percentage of service-provider MPs plateau over time, Fig. 3e indicates that the ART performance gap between E-ARL and E-DCG+ keeps increasing. This is because E-ARL facilitates better network connectivity maintenance. The explanation for Fig. 3g follows that of Fig. 3b.

Let N_{MP} be the number of MPs. During replica allocation, E-ARL, RI and LN require each MP to send only one message to SP, and SP to send a message to each MP, thus incurring O(N_{MP}) messages, hence their RTR is comparable. However, E-DCG+ requires every MP to broadcast its RWR values to every MP, thereby incurring O(N_{MP}^2) messages, which explains Fig. 3h.

7.2 Effect of variations in the workload skew

Figure 4 depicts the results when the zipf factor (ZF) is varied. These results can be mostly explained by the discussion for Fig. 3. As ZF increases (i.e., higher skew), ART increases for all approaches due to overloading, which results in longer waiting times in the MPs' job queues. As ZF increases, SR increases and HC decreases due to more replica allocations in response to load-imbalance conditions. As ZF increases (i.e., at highly skewed workloads), the ART and SR performance gap between E-ARL and E-DCG+ increases due to E-ARL's better load-balancing capability. At low ZF values, the performance gap decreases since the need for replication decreases at lowly skewed workloads. E-ARL outperforms the other approaches due to the reasons explained for Fig. 3. Explanations for Figs. 4c and 4d follow that of Figs. 3g and 3h.

7.3 Effect of revenue threshold

We define revenue threshold TH_R as the average revenue in the system i.e., the ratio of the total revenue in the system to the total number of MPs. Figure 5 depicts the effect of variations in the number N_{TH_R} of MPs, whose revenue exceeds TH_R. When the revenue of more MPs exceeds TH_R, ART, SR and HC improve for both E-ARL and RI due to more MPs providing service as MP revenues increase, thereby

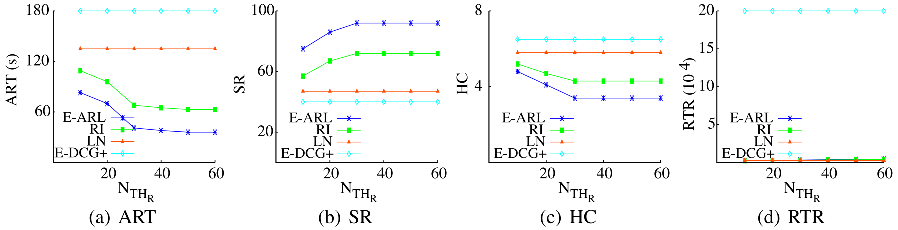
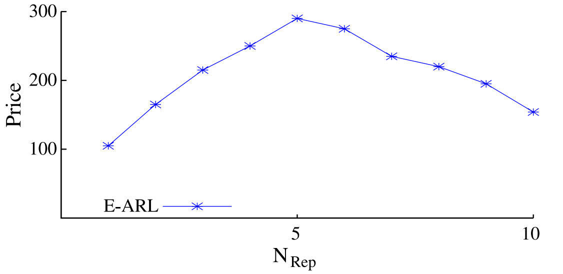


Fig. 5 Effect of revenue threshold

Fig. 6 Effect of variations in the number of replicas on data item price



implying more opportunities for replication, more memory space and multiple paths for locating a data item/replica. Interestingly, beyond $N_{TH_R} = 30$, performance of both E-ARL and RI plateau since it is upper-limited by competition among replicas for memory space and availability of MPs. E-ARL outperforms RI due to the reasons explained for Fig. 3 i.e., load-balancing and RI's preference for greedily hosting higher-priced data items, which decreases query performance for lower-priced items. E-DCG+ and LN show relatively constant performance as they are independent of revenue. The explanation for Fig. 5d follows that of Fig. 3h.

7.4 Effect of variations in the number of replicas on data item price

Figure 6 shows how the price of a data item varies when the number N_{Rep} of its replicas is varied. We randomly selected a data item and observed its price over time as its number of replicas varied due to changes in access frequency. We repeated this experiment 400 times with a different randomly selected item each time and averaged the results. Beyond $N_{\text{Rep}} = 5$, item price decreases with increase in N_{Rep} due to decrease in demand. However, when too few replicas exist, quality of service for queries on the item decreases, thereby decreasing its price, as explained in Sect. 3. Thus, below $N_{\text{Rep}} = 5$, price increases with increasing N_{Rep} due to better quality of service.

7.5 Effect of variations in the replica allocation period

Figure 7 depicts the results of varying the replica allocation period TP. This experiment was done by issuing 25 000 queries at query interarrival rate of 10 queries/s. Thus, for TP values of 500 s, 1000 s, 1500 s and 2000 s, there were 5, 2, 1 and 1

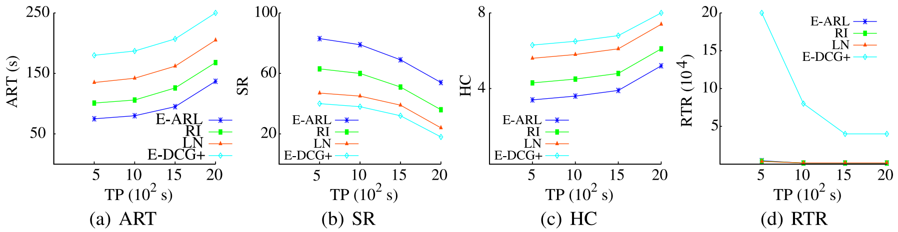


Fig. 7 Effect of variations in the replica allocation period TP

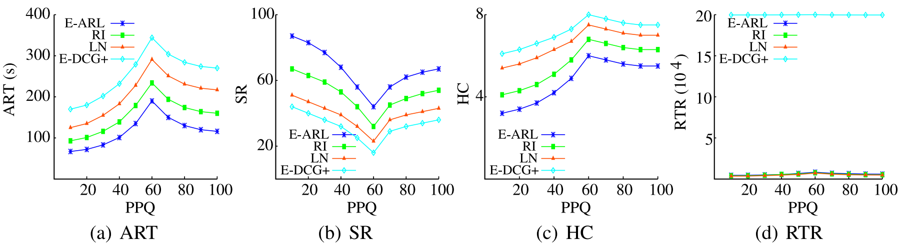


Fig. 8 Effect of variations in the percentage of priority queries (PPQ)

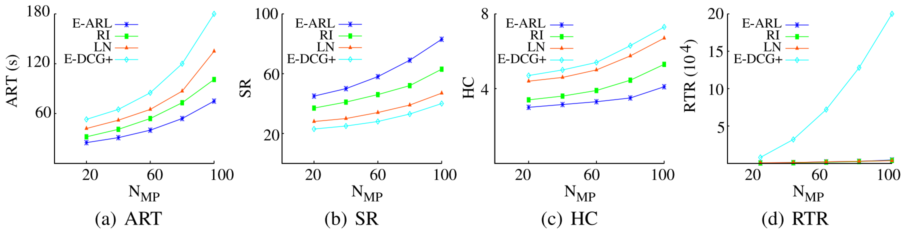
replica allocation periods respectively. RTR is comparable for TP values of 1500 s and 2000 s (due to equal number of allocation periods), although ART, SR and HC performance at these two time points vary. This is because performance is affected not only by the *number* of allocation periods, but also by *when* such periods occur.

When TP is low, more number of replica allocation periods occur, hence replica allocations are able to react quickly to changing access patterns, hence all the approaches perform better albeit at the cost of higher RTR. As TP increases, replica allocations are performed less frequently, hence performance degrades for all the approaches, even though RTR improves. Thus, there is a trade-off between RTR and ART. This trade-off is evident in Fig. 7d, which indicates that RTR decreases dramatically for E-DCG+ with increasing TP due to decreased number of allocation periods.

7.6 Effect of variations in the percentage of priority queries

Figure 8 depicts the effect of variations in the percentage of priority queries (PPQ). Interestingly, as PPQ increases, ART and HC increase, while SR decreases for all the approaches, even though priority queries are answered within shorter deadlines. This anomaly occurs because for answering larger number of priority queries, the performance of normal queries becomes compromised because more replicas for priority items are created in response to the higher demand for priority items, so less memory space is available for hosting normal items. Furthermore, priority queries decrease network connectivity due to the possible involvement of MPs with relatively lower energy in answering priority queries as well as in relaying priority items.

However, when PPQ exceeds 60%, the performance of E-ARL improves since majority of the queries are priority queries, and replicas for the items corresponding



(a) ART (b) SR (c) HC (d) RTR

Fig. 9 Effect of variations in the number of MPs

to the priority queries have been adequately allocated due to E-ARL's preference for priority items. Even though the other approaches do not prefer priority items for replication, their performance also improves since they also try to allocate replicas for the priority items (in response to more priority queries) to satisfy the shorter query deadline requirements. Observe that the performance of all the approaches beyond a PPQ of 60% is considerably worse than their performance at lower values of PPQ such as 20%. This is due to decreased network connectivity induced by priority queries, as explained earlier. The explanation for the results in Fig. 8d is essentially the same as that of Fig. 3h.

7.7 Effect of variations in the number of MPs

To test E-ARL's scalability, we varied the number N_{MP} of MPs, keeping the number of queries proportional to N_{MP}. Figure 9 depicts the results. The results in Fig. 9 can be mostly explained by the reasons explained for Fig. 3. All four approaches exhibit better performance as N_{MP} increases due to increased opportunities for replication. As N_{MP} decreases, the performance gap between the approaches decreases due to limited replication opportunities. Replica allocation traffic for E-DCG+ dramatically decreases with decreasing N_{MP} due to reduced broadcast traffic.

8 Conclusion

We have proposed E-ARL, which is a novel **Economic scheme** for **Adaptive Revenue-Load-based** dynamic replication of data in **dedicated** M-P2P networks with the aim of improving data availability. E-ARL considers a mobile cooperative environment, where the MPs are working towards the same goal, and the network performance is facilitated by the economic scheme. E-ARL essentially allocates replicas based on its economic scheme. E-ARL uses an economic scheme for efficiently managing M-P2P resources in a context-aware manner by facilitating effective replica hosting and message relaying by peers. E-ARL *collaboratively* performs *bid-based* replica allocation to facilitate better quality of service. It incorporates both revenue-balancing and load-balancing to improve peer participation and performance. It also effectively maintains network connectivity. Extensive performance evaluation demonstrates that E-ARL is indeed effective in improving query response times, query success rates, query hop-counts and replica allocation traffic.

In the near future, we will implement a real prototype of E-ARL. We also plan to use game-theoretic approaches for data item pricing and compare E-ARL's performance for different economic models.

References

1. Bhagwan, R., Moore, D., Savage, S., Voelker, G.M.: Replication strategies for highly available P2P storage. In: Proc. Future Directions in Distributed Computing (2003)
2. Broch, J., Maltz, D.A., Johnson, D.B., Hu, Y.C., Jetcheva, J.: A performance comparison of multi-hop wireless ad hoc network routing protocol. In: Proc. MOBICOM (1998)
3. Buttyan, L., Hubaux, J.: 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. Buttyan, L., Hubaux, J.P.: Stimulating cooperation in self-organizing mobile ad hoc networks. Mob. Netw. Appl. **8**(5), 579–592 (2003)
5. Cao, G., Welch, J.L.: Accurate multihop clock synchronization in mobile ad hoc networks. In: Proc. ICPPW (2004)
6. Chakravorty, R., Agarwal, S., Banerjee, S., Pratt, I.: MoB: a mobile bazaar for wide-area wireless services. In: Proc. MobiCom (2005)
7. Chen, K., Nahrstedt, K.: iPass: an incentive compatible auction scheme to enable packet forwarding service in MANET. In: Proc. ICDCS (2004)
8. Crowcroft, J., Gibbens, R., Kelly, F., Ostring, S.: Modelling incentives for collaboration in mobile ad hoc networks. In: Proc. WiOpt (2003)
9. Daras, P., Palaka, D., Giagourta, V., Bechtsis, D.: A novel peer-to-peer payment protocol. In: Proc. IEEE EUROCON, vol. 1 (2003)
10. Datta, A., Hauswirth, M., Aberer, K.: Beyond web of trust: enabling P2P e-commerce. In: Proc. ICEC (2003)
11. Datta, A., Hauswirth, M., Aberer, K.: Updates in highly unreliable replicated peer-to-peer systems. In: Proc. ICDCS (2003)
12. Elrufaie, E., Turner, D.: Bidding in P2P content distribution networks using the lightweight currency paradigm. In: Proc. ITCC (2004)
13. Ferguson, D.F., Nikolaou, C., Yemini, Y.: An economy for managing replicated data in autonomous decentralized systems. In: Proc. International Symposium in Autonomous Decentralized Systems, pp. 367–375 (1993)
14. Ferguson, D.F., Yemini, Y., Nikolaou, C.: Microeconomic algorithms for load balancing in distributed computer systems. In: Proc. ICDCS, pp. 491–499 (1988)
15. Figueiredo, D.R., Shapiro, J., Towsley, D.: Payment-based incentives for anonymous peer-to-peer systems. UMass CMPSCI Technical Report 04-62 (2004)
16. Garyfalos, A., Almeroth, K.C.: Coupon based incentive systems and the implications of equilibrium theory. In: Proc. IEEE International Conference on E-Commerce Technology Proceedings (2004)
17. Golle, P., Brown, K.L., Mironov, I.: Incentives for sharing in peer-to-peer networks. In: Proc. Electronic Commerce (2001)
18. Guy, R., Reiher, P., Ratner, D., Gunter, M., Ma, W., Popek, G.: Rumor: mobile data access through optimistic peer-to-peer replication. In: Proc. ER Workshops (1998)
19. Ham, M., Agha, G.: ARA: a robust audit to prevent free-riding in P2P networks. In: Proc. P2P, pp. 125–132 (2005)
20. Hara, T., Madria, S.K.: Consistency management among replicas in peer-to-peer mobile ad hoc networks. In: Proc. IEEE SRDS (2005)
21. Hara, T., Madria, S.K.: Data replication for improving data accessibility in ad hoc networks. IEEE Trans. Mob. Comput. **5**(11), 1515–1532 (2006)
22. http://tv.seas.harvard.edu/research.php. Accessed 17 April 2010
23. http://www.microsoft.com/presspass/presskits/zune/default.mspx. Accessed 17 April 2010
24. Kamvar, S., Schlosser, M., Garcia-Molina, H.: Incentives for combatting free-riding on P2P networks. In: Proc. Euro-Par (2003)
25. Kemme, B., Alonso, G.: A new approach to developing and implementing eager database replication protocols. ACM Trans. Database Syst. **25**(3), 333–379 (2000)

26. Kremer, S., Markowitch, O., Zhou, J.: An intensive survey of non-repudiation protocols. Technical Report 473, ULB (2002)
27. Kurose, J.F., Simha, R.: A microeconomic approach to optimal resource allocation in distributed computer systems. IEEE Trans. Comput. **38**(5), 705–717 (1989)
28. Liebau, N., Darlagiannis, V., Heckmann, O., Steinmetz, R.: Asymmetric incentives in peer-to-peer systems. In: Proc. AMCIS (2005)
29. Liu, J., Issarny, V.: Service allocation in selfish mobile ad hoc networks using Vickrey auction. In: Proc. Current Trends in Database Technology—EDBT Workshops Revised Papers. Lecture Notes in Computer Science, vol. 3268, pp. 443–447. Springer, Berlin (2004)
30. Mannak, R., de Ridder, H., Keyson, D.V.: The human side of sharing in peer-to-peer networks. In: Proc. European Union Symposium on Ambient Intelligence (2004)
31. Mondal, A., Madria, S.K., Kitsuregawa, M.: CADRE: a collaborative replica allocation and deallocation approach for Mobile-P2P networks. In: Proc. IDEAS (2006)
32. Mondal, A., Madria, S.K., Kitsuregawa, M.: CLEAR: an efficient context and location-based dynamic replication scheme for Mobile-P2P networks. In: Proc. DEXA (2006)
33. Pitoura, E., Bhargava, B.: Maintaining consistency of data in mobile distributed environments. In: Proc. ICDCS (1995)
34. Ramaswamy, L., Liu, L.: Free riding: a new challenge to P2P file sharing systems. In: Proc. HICSS, p. 220 (2003)
35. Ratner, D., Reiher, P.L., Popek, G.J., Kuenning, G.H.: Replication requirements in mobile environments. Mobile Netw. Appl. **6**(6), 525–533 (2001)
36. Ratsimor, O., Finin, T., Joshi, A., Yesha, Y.: eNcentive: a framework for intelligent marketing in mobile peer-to-peer environments. In: Proc. ICEC (2003)
37. Richard, B., Nioclais, D., Chalon, D.: Clique: a transparent, peer-to-peer replicated file system. In: Proc. MDM (2003)
38. Sabater, J., Sierra, C.: Review on computational trust and reputation models. Artif. Intell. Rev. **24**(1), 33–60 (2005)
39. Saroiu, S., Gummadi, P.K., Gribbler, S.D.: A measurement study of P2P file sharing systems. In: Proc. MMCN (2002)
40. Sheu, J.P., Chao, C.M., Hu, W.K., Sun, C.W.: A clock synchronization algorithm for multihop wireless ad hoc networks. J. Wirel. Pers. Commun. (2006)
41. Srinivasan, V., Nuggehalli, P., Chiasserini, C.F., Rao, R.R.: Cooperation in wireless ad hoc networks. In: Proc. INFOCOM (2003)
42. Straub, T., Heinemann, A.: An anonymous bonus point system for mobile commerce based on word-of-mouth recommendation. In: Proc. ACM SAC (2004)
43. Wolfson, O., Xu, B., Sistla, A.P.: An economic model for resource exchange in mobile peer-to-peer networks. In: Proc. SSDBM (2004)
44. Xu, B., Wolfson, O., Rishe, N.: Benefit and pricing of spatio-temporal information in mobile peer-to-peer networks. In: Proc. HICSS-39 (2006)
45. Xue, Y., Li, B., Nahrstedt, K.: Channel-relay price pair: towards arbitrating incentives in wireless ad hoc networks. J. Wirel. Commun. Mob. Comput., Special Issue on Ad Hoc Networks (2005)
46. Xue, Y., Li, B., Nahrstedt, K.: Optimal resource allocation in wireless ad hoc networks: a price-based approach. In: IEEE Trans. Mob. Comput. (2005)
47. Yang, B., Garcia-Molina, H.: Designing a super-peer network. In: Proc. ICDE (2003)
48. Zhong, S., Chen, J., Yang, Y.R.: Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks. In: Proc. IEEE INFOCOM (2003)