Future Generation Computer Systems 36 (2014) 257–266

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Flat and hierarchical epidemics in P2P systems: Energy cost models and analysis☆

Oznur Ozkasap[a,*], Emrah Cem[a,b], Sena Efsun Cebeci[a], Tugba Koc[a]

[a] *Department of Computer Engineering, Koc University, Istanbul, Turkey*

[b] *Department of Computer Science, University of Texas at Dallas, USA*

HIGHLIGHTS

- Power awareness of flat and hierarchical epidemics in P2P systems is addressed.
- We developed energy cost model formulations for flat and hierarchical epidemics.
- We proposed a dominating-set based and power-aware hierarchical epidemic approach.
- Through large scale simulations on PeerSim, we compared the epidemic approaches.
- We analyzed the effect of epidemic protocol parameters on energy consumption.

ARTICLE INFO

Article history:
Received 2 November 2012
Received in revised form
12 July 2013
Accepted 5 September 2013
Available online 1 October 2013

Keywords:
Energy cost model
Energy efficiency
Epidemic
Gossip-based
Peer-to-peer
Churn

ABSTRACT

In large scale distributed systems, epidemic or gossip-based communication mechanisms are preferred for their ease of deployment, simplicity, robustness against failures, load-balancing and limited resource usage. Although they have extensive applicability, there is no prior work on developing energy cost models for epidemic distributed mechanisms. In this study, we address power awareness features of two main groups of epidemics, namely flat and hierarchical. We propose a dominating-set based and power-aware hierarchical epidemic approach that eliminates a significant number of peers from gossiping. To the best of our knowledge, using a dominating set to build a hierarchy for epidemic communication and provide energy efficiency in P2P systems is a novel approach. We develop energy cost model formulations for flat and hierarchical epidemics. In contrast to the prior works, our study is the first one that proposes energy cost models for generic peers using epidemic communication, and examines the effect of protocol parameters to characterize energy consumption. As a case study protocol, we use our epidemic protocol ProFID for frequent items discovery in P2P systems. By means of extensive large scale simulations on PeerSim, we analyze the effect of protocol parameters on energy consumption, compare flat and hierarchical epidemic approaches for efficiency, scalability, and applicability as well as investigate their resilience under realistic churn.

1. Introduction

Scalability, simplicity, efficiency and robustness characteristics of epidemic mechanisms make them attractive for several distributed services such as reliable multicasting [1–3], global information computation, frequent items discovery [4], overlay topology construction, failure detection, P2P streaming and data dissemination [5]. Epidemic algorithms use a round based approach in which nodes periodically communicate with each other. In each round, each peer contacts at least one node chosen at random, called neighbor, to exchange its state. Completion of the algorithm takes place in multiple rounds, and data is disseminated to the network like an epidemic disease. One advantage of epidemics is its robustness against peer failures. Removal or failure of a peer does not affect the dissemination speed significantly. Epidemic algorithms are mostly preferred for their simplicity, ease of deployment and robustness [6]. On the other hand, the disadvantage is communication overhead since any two neighbors may communicate multiple times during the execution, hence resulting in redundant information exchange when compared to hierarchical approaches.

There exist two main groups of epidemic algorithms, namely flat and hierarchical. Flat algorithms involve basic and neighborhood epidemics. The basic epidemic requires global knowledge of the peer population and performs uniform gossiping, therefore it

☆ This work was partially supported by the COST (European Cooperation in Science and Technology) framework, under Action IC0804, and by TUBITAK (The Scientific and Technical Research Council of Turkey) under Grant 109M761.

* Corresponding author. Tel.: +90 212 3381584; fax: +90 212 338 1548.
E-mail addresses: oozkasap@ku.edu.tr (O. Ozkasap), Emrah.Cem@utdallas.edu (E. Cem), senacebeci@ku.edu.tr (S. Efsun Cebeci), tugbakoc@ku.edu.tr (T. Koc).

0167-739X/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.future.2013.09.009

is not practical. On the other hand, neighborhood epidemic uses local knowledge, which is more practical and performs gossiping with neighbors. Although neighborhood epidemic is better when compared to basic, it still has the problem of redundant communication. However, hierarchical epidemic makes use of structure among peers and aims to reduce communication overhead. In addition, it provides the possibility of active/passive peers to save energy.

Designing energy efficient epidemic protocols and services has become significant due to their wide usage in large scale distributed systems. However, there is a lack of studies on the power usage and energy efficiency of epidemic approaches. In terms of their power usage, efficiency of three models of epidemic protocols, namely basic, neighborhood and hierarchical epidemics, has been examined in [7]. Basic epidemic that requires full membership knowledge of peers were found to be inefficient in their power usage. It has been shown that; in neighborhood epidemics, a peer's power consumption amount is independent of the population size. On the other hand, for hierarchical epidemics, power usage increases with population size. In fact, [7] is the only study that considers power awareness features of epidemic protocols. However, it evaluates different epidemics through simulations only and provides results on latency and power (proportional to the gossip rate). Moreover, the effects of gossip parameters such as fan-out and maximum gossip message size were not investigated. In terms of energy efficiency, an epidemic protocol's reliability has been discussed in mobile ad hoc networks [8] and a gossip-based protocol was proposed for wireless sensor networks [9]. In [8], based on packet delivery ratio, nodes with high delivery ratio are classified as active, and the energy consumption is therefore affected significantly with less packet drops. In [9], low energy consumption and fault tolerance are achieved by early detection of the aggregation convergence independent of changes in the network topology.

In this study, we address power awareness features of two main groups of epidemics, namely flat and hierarchical. We propose a dominating-set based and power-aware hierarchical epidemic approach that eliminates significant number of peers from gossiping. In contrast to the prior works on hierarchical epidemics, we use dominating-set to construct a hierarchy, and to choose peers performing gossip operation for energy efficiency. In this adaptive approach, only a subset of the peer population is active in gossiping by forming an overlay consisting of dominating set peers, so that the other peers can switch to an idle state. It also allows data aggregation that can be utilized to reduce gossip message size. There exist prior studies using a dominating set in P2P networks. For instance, a dominating-set-based P2P protocol is proposed in [10,11] that aims to use a minimum number of forwarding nodes for data delivery. Another study [12] uses a dominating set to solve a searching problem in P2P networks. As a case study protocol, we use our epidemic-based approach ProFID for frequent items discovery [4] and its simulation model on PeerSim. ProFID uses a novel atomic pairwise averaging in order to compute global frequencies of items.

Contributions of our study are as follows.

- We address power awareness features and develop energy cost model formulations for flat and hierarchical epidemics. In contrast to the prior works, our study is the first one that proposes energy cost models for generic peers using epidemic communication, and examines the effect of protocol parameters to characterize energy consumption.
- We propose a dominating-set based and power-aware hierarchical epidemic approach that eliminates a significant number of peers from gossiping. To the best of our knowledge, using a dominating set to build a hierarchy for epidemic communication and provide energy efficiency in P2P systems is a novel approach.
- Through extensive large scale simulations on PeerSim, we analyze the effect of protocol parameters on energy consumption, compare flat and hierarchical epidemic approaches for efficiency, scalability, and applicability as well as investigate their resilience under realistic churn.

Table 1
Different operations that consume energy.

Value	Description
E_{send}	Energy required to send the item tuple
E_{recv}	Energy required to receive the item tuple
$E_{compStarter}$	Energy required to choose tuple to send and update the state
$E_{compTarget}$	Energy required to compute the average and prepare the tuple to send

This paper is organized as follows. In Section 2, we describe flat epidemics and develop an energy cost model for a peer using flat epidemic based protocol. In Section 3, we propose our dominating-set based and power-aware hierarchical epidemic approach, and develop its energy cost model. Section 4 discusses the churn effect in epidemics and describes our adaptive flat epidemic approach that aims to cope with churn behavior. Simulation environment and the performance criteria are described in Section 5. In Section 6, extensive analysis results of flat and hierarchical epidemics are presented. Finally, Section 7 states conclusions and future directions.

2. Flat neighborhood epidemics

In flat neighborhood epidemics, all peers periodically exchange local state with neighbors until convergence. A peer i is the neighbor of peer j if they are directly connected in the overlay network. The main advantage of this approach over basic epidemics is that scalability problem is solved since no peer needs global knowledge. On the other hand, all peers may communicate more than once with their neighbors, which is necessary for convergence. Thus, high communication overhead is a drawback of this approach.

Energy cost model. We propose an energy cost model for a generic peer using gossip-based communication in our epidemic case study protocol ProFID [4]. ProFID protocol depends on three main components of operations performed by each peer: energy consumed while (1) computing new state, (2) sending messages and (3) receiving messages. In the formulation of the energy cost model, we are inspired by studies [13,14]. In [13], energy cost models for client–server and publish–subscribe styles were developed. Then, application and platform specific model parameters were also taken into consideration and an energy prediction model was developed. Work of [14] introduces a quorum-based model to compute energy costs of read and write operations in replication protocols, and proposes an approach to reduce the energy cost of the tree replication protocol. In contrast to these prior works, we develop an energy cost model for a peer using gossip-based communication and consider the effects of gossip parameters on the cost representation.

We start with an analysis of the energy consumption during an atomic pairwise averaging operation between peers P_i and P_j. As depicted in Fig. 1, a peer that initializes averaging chooses a target peer among its neighbors and sends a *push* gossip message to this peer. The target peer computes the average and sends the result in a *pull* gossip message. In order to compute the correct global averages, an averaging operation must be atomic meaning that when a peer is in contact with another peer, no other peer would interfere. Different operations consuming energy are explained in Table 1.

During an atomic pairwise averaging, the energy cost of a peer that initiates a gossip (*gossip starter*) is represented by:

$$E_{gossipStarter} = E_{send} + E_{receive} + E_{compStarter}. \quad (1)$$

On the other hand, the energy cost of the gossip target can be formulated as follows:

$$E_{gossipTarget} = E_{receive} + E_{send} + E_{compTarget}. \quad (2)$$

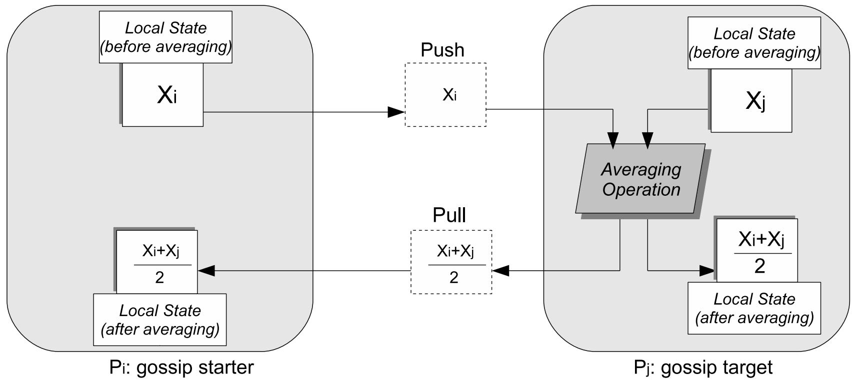

Fig. 1. Illustration of an atomic pairwise averaging operation between P_i and P_j.

Note that $E_{compTarget}$ and $E_{compStarter}$ are both proportional to the gossip message size, and they can simply be represented as E_{comp}. Hence, $E_{i,j}$ (the energy consumption of a peer P_i during an atomic pairwise averaging with P_j) can be written as:

$$E_{i,j} = E_{send,j} + E_{receive,j} + E_{comp} + C \tag{3}$$

where $E_{send,j}$ is the energy consumed while sending a gossip message to P_j, $E_{receive,j}$ is the energy consumed while receiving a gossip message from P_j, and E_{comp} is the local computation of the peer. Note that this is the energy cost of a peer that performs an atomic pairwise averaging operation. In real network scenarios, energy consumption may include extra factors such as CPU's energy consumption during I/O. Hence, a constant C is added to the equation.

To represent the energy cost of a gossip-based peer during an atomic pairwise averaging operation, the formula was given with respect to the basic conditions (gossip to one neighbor, one round, one item). Step by step, we now extend this cost model of a peer for the ProFID protocol. A peer may initiate multiple gossip operations during a single round depending on the *fanout* value as well as it may become a gossip target multiple times. Fanout is the number of peers to whom to send the gossip message at each round. The energy cost of P_i that gossips a single item tuple in a round can be formulated as:

$$E_{P_i}(\text{single round, single item}) = \sum_{j \in V \cup W} E_{i,j} \tag{4}$$

where V is the set of neighbors chosen by P_i as gossip targets, and W is the set of neighbors that initiates an atomic pairwise averaging with P_i. Note that the number of elements in V corresponds to the fanout value.

In general, a gossip message comprises multiple item tuples whose number is upper-bounded by *maximum message size* (*mms*) parameter. Since $E_{send,j}$ and $E_{receive,j}$ are the energies consumed while sending and receiving a single tuple respectively, the total energy consumed during a gossip round would linearly increase with the *mms*. Hence, the energy cost of P_i in a round can be expressed as:

$$E_{P_i}(\text{single round}) \le mms \cdot \sum_{j \in V \cup W} E_{i,j}. \tag{5}$$

Since a peer repeats those operations in every round, the number of rounds R would increase the energy cost of a peer proportionally. Hence, the overall energy cost of P_i can be written as:

$$E_{P_i} \le R_F \cdot mms \cdot \sum_{j \in V \cup W} E_{i,j}. \tag{6}$$

3. Hierarchical epidemics

In our hierarchical model, we use a dominating set idea to build a high-level overlay as illustrated in Fig. 2. Dominating Set (DS) can be defined as a subset B of a graph $G = (V, E)$ such that every vertex in G is either in B or adjacent to a vertex in B. In our approach, a DS corresponds to a subset of peers such that a peer in the system is either in DS or a neighbor of a DS peer. There are two types of peers in the system, namely the dominating set (DS) peer and non-dominating set (NDS) peer that can be chosen through distributed algorithms for DS construction [11,15].

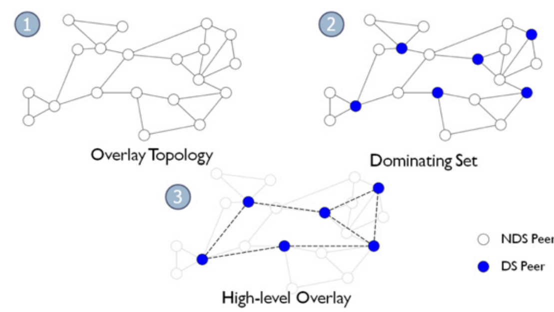

Fig. 2. Construction of high level overlay using dominating set.

In our hierarchical epidemics model, we aim to save energy by reducing the number of peers performing the gossip operation. NDS peers are excluded from the gossip operation. DS peers collect the local states of NDS neighbors. Consequently, DS peers form a high level overlay topology on which gossiping is performed. There are two main advantages of this approach. (1) NDS peers send their local state to one or multiple DS neighbors and then they switch to passive mode in which they just wait for the result to be announced by a DS neighbor. During this time period, they contribute to reducing the energy consumption. (2) Only DS peers participate in gossiping and hence the convergence time and message complexity of the epidemics are reduced.

Algorithm 1 shows the hierarchical epidemic algorithm used for frequent item discovery. Descriptions of the parameters used in Algorithm 1 are as follows:

Fanout: Number of peers to whom to send gossip message at each round.

ui: Unique item used in system size estimation.

convLimit: Number of successive rounds a peer needs to satisfy epsilon condition in order to converge.

ε: Parameter used to determine epsilon condition. A peer satisfies epsilon condition if current frequency of item ui at that peer changes at most ε (epsilon) percentage after the previous gossip. Definition of epsilon condition and algorithm for convergence check (ISCONVERGED) are given in [4].

T: Threshold value used to distinguish frequent items from infrequent items.

S: Set of local items.

Algorithm 1: Hierarchical Epidemics Algorithm

```
Require: fanout, ui, convLimit, ε, T, S
Ensure: F (Set of frequent items)
  DS Peer
  x= number of NDS neighbors;
  for i=1 to x do
    send(req(LSI,DS)) to neighbor i;
  end for
  do for NDS neighbor state collection
  msg = accept();
  DS.update(msg);
  S.update(msg);
  if completed then
    do periodically
    if !converged then
      targets = getNeighbors(fanout);
      for i=1 to fanout do
        send(push, S, targets(i));
      end for
    end if
  end if
  do until convergence
  msg=accept();
  if msg.Type == push then
    avg = AVG(S, msg); S.update(avg); send(pull, avg,
    sender);
  else if msg.Type == pull then
    S.update(msg);
  end if
  currSizeEstim=msg.getVal(ui);
  if ISCONVERGED(convLimit, ε, currSizeEstim) then
    converged=true;
  end if
  Query
  F={item | item ∈ S},
  S.getAvgFrequency(item)* 1/currSizeEstim ≥ T
  NDS Peer
  msg = accept();
  if msg.Type == request(LSI, DS) then
    send(LSI, DS, sender);
  else if msg.Type == request(DS) then
    send(DS, sender);
  else if msg.Type == result then
    S.update(msg);
  end if
```

Distributed Greedy Algorithm for DS construction. We use an approximation algorithm for DS construction, namely Distributed Greedy Algorithm (DGA) [11]. In DGA, a node coloring method is used to represent the current state of the nodes as black, white and gray. Nodes in the DS are black, nodes covered by the nodes in DS are gray, and the uncovered nodes (that is, non-neighbor nodes of black nodes) are white. The span of a node v (i.e., Span(v)) is defined as the number of white nodes among the direct neighbors of node v including v. Every node v that has at least one white neighbor executes Algorithm 2.

Energy cost model for DS peer. There are three main operations for DS peers causing energy consumption. Firstly, DS peers send request messages to their NDS neighbors to get their local state information and the id set of their DS neighbors. The energy cost of a DS peer P_i during a request operation from a NDS peer P_j is:

$$E_{i,j,1} = E_{send,j} + (s + d)(E_{receive,j} + E_{comp,i}) + C \tag{7}$$

where $E_{send,j}$ is the energy consumed while sending request message to a NDS peer P_j, s is the number of item tuples in local state

Algorithm 2: DGA for DS construction

```
y= number of white neighbors of v;
for i=1 to y do
  send(Span(v)) to neighbor i;
  if Span(v) is the largest within 2-hop distance then
    joinDS(v); // v becomes a black node
  end if
end for
```

information, d is the number of DS neighbors, $E_{receive,j}$ is the energy consumed while receiving the local state information and the id set of DS neighbors of a NDS peer P_j, and $E_{comp,i}$ is the energy consumed to aggregate local state information of NDS neighbor. As mentioned before, constant C is included to indicate additional factors such as CPU's energy consumption during I/O. A DS peer sends request messages to all its NDS neighbors and the formula can be extended as:

$$E_{P_{i,1}} = \sum_{j \in X} E_{i,j,1} \tag{8}$$

where X is the set of NDS neighbors of peer P_i. Secondly, after getting local state information and set of other DS neighbors, DS peers start gossiping with other DS peers. This operation can be considered as a flat gossip among DS peers in the high level of the hierarchy. Therefore, the energy consumption of a DS peer P_i during an atomic pairwise averaging with P_j can be written as:

$$E_{i,j,2} = E_{send,j} + E_{receive,j} + E_{comp,i} + C \tag{9}$$

where $E_{send,j}$ is the energy consumed while sending a gossip message to P_j; $E_{receive,j}$ is the energy consumed while receiving a gossip message from P_j; and $E_{comp,i}$ is the local computation cost of P_i while choosing and preparing the tuple, computing the average and updating the state. To represent the energy cost of a DS peer during an atomic pairwise averaging operation, the formula was given with respect to the basic conditions (gossip to one neighbor, one round, one item). So, if we extend this cost model, the formula can be written as:

$$E_{P_{i,2}} = R_H \cdot mms \cdot \sum_{j \in Y \cup Z} E_{i,j,2} \tag{10}$$

where Y is the set of DS neighbors chosen by P_i as gossip targets and Z is the set of DS neighbors that initiate an atomic pairwise averaging with P_i. In general, a gossip message comprises multiple item tuples whose number is upper-bounded by maximum message size (mms) parameter. Since a peer repeats those operations in every round, the number of rounds to converge in hierarchical gossip (R_H) would increase the energy cost of a peer proportionally. Finally, after gossip and convergence in the high level of the hierarchy finish, DS peers should send the result to their NDS neighbors. During this send operation, the energy cost is:

$$E_{i,j,3} = s(E_{send,j} + E_{comp,i}) + C \tag{11}$$

where $E_{send,j}$ is the energy consumed while sending final state information to a NDS peer P_j, s is the number of item tuples in local state information, $E_{comp,i}$ is the energy consumed while preparing the tuple to send. A DS peer sends a result message to all its NDS neighbors. Hence, the formula can be extended as:

$$E_{P_{i,3}} = \sum_{j \in X} E_{i,j,3}. \tag{12}$$

The overall energy cost of P_i would be:

$$E_{P_i} = \sum_{j \in X} E_{i,j,1} + R_H \cdot mms \cdot \sum_{j \in Y \cup Z} E_{i,j,2} + \sum_{j \in X} E_{i,j,3}. \tag{13}$$

In this formula, $E_{P_i,1}$ and $E_{P_i,3}$ do not have a significant effect on the overall energy consumption. The important effect is due to $E_{P_i,2}$ which is related to the atomic pairwise averaging operation.

However, in hierarchical epidemics the number of peers performing the atomic pairwise averaging operation (i.e., DS peers) is less than those in the flat epidemics. Due to this reason, R_H (the number of rounds to converge in hierarchical gossip) is smaller than R_F (the number of rounds to converge in flat gossip). The effect of epidemic protocol parameters on the number of rounds to converge (R_H) in the hierarchical model can be represented as follows. R_H is proportional to the log N_H and convLimit, and it is inversely proportional to the log ε, fanout and *mms*:

$$R_H \approx (1/\log \varepsilon) \cdot \log N_H \cdot convLimit \cdot (1/fanout) \cdot (1/mms). \quad (14)$$

Energy cost model for NDS peer. The energy consumption of NDS peers in the system is relatively very less compared to DS peers, since NDS peers do not take role in the gossip operations. Their responsibility in the system is to receive request messages, send their local state information and the id set of their DS neighbors. At the end, they receive final state information from their DS neighbors and update their state. Energy consumption of a NDS peer can be analyzed in two parts. The first part involves sending response messages to the DS neighbors.

$$E_{i,j,1} = E_{receive,j} + (s + d)(E_{send,j} + E_{comp,i}) + C \quad (15)$$

where $E_{receive,j}$ is the energy consumed while receiving request message from a DS neighbor P_j, $E_{send,j}$ is the energy consumed while sending local state information and the id set of the other DS neighbors, s is the number of item tuples, d is the number of DS neighbors, and $E_{comp,i}$ is the energy consumed while preparing the tuple and DS neighbor set. A NDS peer gives a response to all its DS neighbors, so the formula can be extended as:

$$E_{P_{i,1}} = \sum_{j \in K} E_{i,j,1} \quad (16)$$

where K is the set of DS neighbors. In the second part, NDS peers consume energy when receiving result messages from their DS neighbors. So, the energy cost can be written as:

$$E_{i,j,2} = s(E_{receive,j} + E_{comp,i}) + C \quad (17)$$

where $E_{receive,j}$ is the energy consumed while receiving final state information from a DS neighbor, s is the number of item tuples, and $E_{comp,i}$ is the energy consumed while updating local state information. The extended formula can be represented as:

$$E_{P_{i,2}} = \sum_{j \in K} E_{i,j,2}. \quad (18)$$

Therefore, the overall energy consumption of a NDS peer P_i can be written as:

$$E_{P_i} = \sum_{j \in K} (E_{i,j,1} + E_{i,j,2}). \quad (19)$$

4. Considering churn: dynamic peer arrival and departures in epidemics

Understanding the dynamics of P2P networks is a key issue because it affects the performance, accuracy and robustness of algorithms such as searching, distributing, and aggregating. Without a correct model of churn characteristics, it is not possible to evaluate P2P algorithms with realistic scenarios. One of the first churn models described in [16] assumes that peers' arrival and departure follow a Poisson distribution and the lifetime of each peer is exponential. However, this model does not consider OFF duration of a peer. Each peer enters the network as a new peer, and neighbor replacements occur as a response to neighbor failures. However, only peers that are in a centralized cache can be used in replacement. An important outcome is that under churn the graph remains connected and its diameter stays logarithmic to the network size. There are also other studies [17,18] that perform neighbor replacements independent of failures. Periodic consistency check is performed, but this results in frequent neighbor change which makes system stabilization harder. In Leonard et al. [19], neighbor replacements are initiated whenever a link failure is detected and link detection is provided by using keep-alive messages (i.e. periodic ping messages). OFF durations are not considered in this model. Whenever a neighbor fails, another peer is replaced randomly among alive peers. A churn model which considers node arrivals, lifetime distribution, and OFF time is proposed in [20]. When a peer first enters the network, it changes its state to connected. Then, it is assigned a random lifetime based on a distribution function. When the lifetime is over, the peer leaves the network either permanently or temporarily based on a parameter called abandon ratio. If it is a temporary departure, the peer enters the network after its OFF time is over. Furthermore, new peers arrive based on a Poisson distribution.

In flat epidemics, peers do not consider neighbors' ON/OFF state information before communicating. Therefore, when a peer starts an averaging operation with an OFF neighbor, it waits for a reply and a timeout occurs. In this case, the peer just waits and cannot perform any operation. In order to handle this issue, we propose an adaptive approach (adaptive flat) that overcomes problems due to peer dynamics and improves convergence speed under churn. In the adaptive approach, a peer simply checks the neighbor's state before it initiates a communication. If the neighbor state is ON, then it can choose it as the gossip target, otherwise it chooses another peer in ON state. If the states of all neighbors are OFF, then the peer fails to send a gossip message in that round.

5. Simulation environment

We have performed various experiments in PeerSim [21], a scalable P2P simulation environment, to compare energy efficiency of flat neighborhood and hierarchical epidemic approaches in terms of message overhead and convergence time. For hierarchical epidemics, we conducted experiments to observe the characteristics of high-level overlay consisting of DS peers. As a case study protocol, we use our epidemic protocol ProFID for frequent items discovery in P2P systems via data aggregation. By means of extensive large scale simulations up to 30,000 peers, we analyze the effect of protocol parameters on energy consumption, compare flat and hierarchical epidemic approaches for efficiency, scalability, and applicability as well as investigate their resilience under realistic churn. We used the Barabasi–Albert (BA) model [22] with average degree 10 and the Erdos–Renyi random graph model [23] for constructing the overlay topologies. All data points presented in the graphs are average of 50 experiments.

We also conducted experiments to analyze the impact of churn on epidemic data aggregation in terms of convergence time, message cost, and accuracy. We used Affluenza [24] to generate realistic churn scenarios based on the Yao model. The Yao churn model is one of the well known models that represents the heterogeneity of peers. In this model, peers leave/enter the system independent of each other, and they have uncorrelated lifetime characteristics [25]. When a peer logs in to the system its state is ON, otherwise its state is OFF as shown in Fig. 3(a). ON duration of a peer p_i is represented as L_i and has a joint distribution $F_i(x)$ with mean l_i. Similarly, OFF duration is represented as D_i and has a joint distribution $G_i(x)$ with mean d_i. What makes this model preferable is that it considers the heterogeneous characteristics of peers since each peer may have a different distribution for ON/OFF durations. It also considers that a single user behaves similarly in a session in comparison to its previous sessions. As an example, Yao churn model behavior observed on a 30,000-peer network during 90 s is depicted in Fig. 3(b) which shows that roughly 38% of the peers are active during the churn.

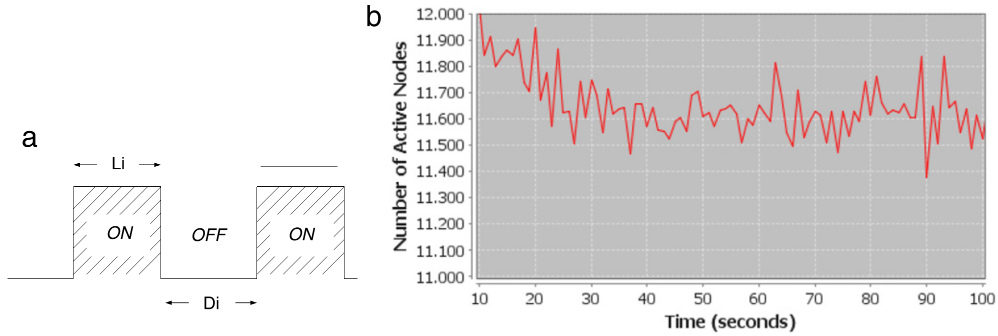

Fig. 3. Yao churn model: (a) ON/OFF states of peer i (b) churn behavior of 30,000 peers.

In order to include the churn model in PeerSim, we developed a module that takes the availability trace (AVT) file as an input. We used shifted Pareto distribution for ON/OFF durations to produce heterogeneous users. Distributions for ON and OFF durations have default values of parameters given in Table 2. Note that while setting parameters, we considered the mean of the shifted Pareto, which is $\beta_i/(\alpha_i-1)$. The mean ON duration is set to 1 s which is the length of a round [26]. Mean OFF distribution is set to 2 to comply with the model [25] in which mean OFF duration is twice of mean ON duration.

The following performance metrics have been used in our analysis.

- *Number of rounds* (*to converge*): Time intervals that peers periodically communicate with each other are called rounds. A peer contacts one or more peers to exchange its state at each round. This metric measures how fast the algorithm converges in terms of rounds. The effects of convergence parameters, average degree of peers, and number of peers on convergence speed have been analyzed.
- *Number of messages sent per peer:* This metric measures the message cost and energy efficiency of the epidemics. The effects of convergence parameters and average degree of peers have been analyzed.
- *Energy savings:* This metric indicates the energy savings of hierarchical epidemics in comparison to flat epidemics. It is defined in two forms, one based on the convergence time in rounds and the other based on the number of messages sent per peer. The effect of scaling up the number of peers is investigated in terms of energy savings metric.

  E_C = Energy savings percentage in convergence time.
  $H_{Convtime}$ = Convergence time of hierarchical epidemics in rounds.
  $F_{Convtime}$ = Convergence time of flat epidemics in rounds.

$$E_C = 100 - \frac{H_{Convtime} \times 100}{F_{Convtime}} \tag{20}$$

  E_M = Energy savings percentage in number of messages sent.
  $H_{Msgsent}$ = Number of messages sent in hierarchical epidemics.
  $F_{Msgsent}$ = Number of messages sent in flat epidemics.

$$E_M = 100 - \frac{H_{Msgsent} \times 100}{F_{Msgsent}}. \tag{21}$$

6. Analysis and results

In this section, we first analyze the characteristics of high-level overlay in hierarchical epidemics. We consider peer average degree and the proportion of peers that are in DS for different network sizes. We analyze the effects of epidemic protocol parameters on the energy consumption. We compare flat and hierarchical epidemic approaches for efficiency, scalability, and applicability. Furthermore, churn effect and energy saving percentages are examined for scalability and protocol comparisons.

Table 2
Churn parameters.

State	Distribution	Parameters' default values
ON	$F(x) = 1 - (1 + x/\beta_i)^{\alpha_i}$	$\alpha = 3, \beta = 2$
OFF	$G(x) = 1 - (1 + x/\beta_i)^{\alpha_i}$	$\alpha = 3, \beta = 4$

6.1. Characteristics of high-level overlay in hierarchical epidemics

High-level overlay characteristics directly affect the convergence time and the scalability of hierarchical epidemics. If the average degree at the high-level is too low then the hierarchical epidemics algorithm will converge slowly. If the average degree at the high-level is too high, then the approach will behave like basic epidemics at the high-level. The number of peers in DS is also important because, the less the number of peers in DS, the faster the convergence. On the other hand, the higher the number of peers in DS, the more robust the algorithm is. Therefore, there is a trade off between the speed and the robustness of the algorithm.

While constructing the topology, we use the Barabasi–Albert model [22] which produces power-law distribution with exponent $\gamma = 3$. Based on this model, the network is initialized with m_0 peers, where m_0 is much less than the network size. Then, at each step a new peer is added to the existing network and the probability that the newly added peer is connected to peer i is defined as:

$$\Pi(k_i) = \frac{k_i}{\sum_j k_j} \tag{22}$$

where k_i is the degree of peer i. The above equation states that a peer with a higher degree on the topology has more probability to connect to the newly added peer. This phenomena is also known as preferential attachment. In order to modify the average degree of the topology, we simply change the number of peers to which a newly added peer connect.

Based on our DGA implementation used for DS construction, our evaluation considers topologies with different average peer degrees and size up to 30,000 peers. As depicted in Fig. 4(a), the percentage of peers that are in DS decreases when the average degree of peers increases, which is expected because a lower number of peers can cover all the peers in the network as the average number of neighbors per peer increases. Another deduction is that more than half of the peers in the network are eliminated from gossiping, which means more than half of the peers will not consume energy during gossiping. Moreover, the convergence time of the algorithm is expected to be reduced since the number of peers participating in gossiping decreases.

Fig. 4(b) shows that the average degree of peers in the high-level overlay logarithmically increases as the network size scales up. This property would have a positive effect on both the scalability and robustness of the algorithm. The more the number of peers in DS, the more robust the algorithm is because computation

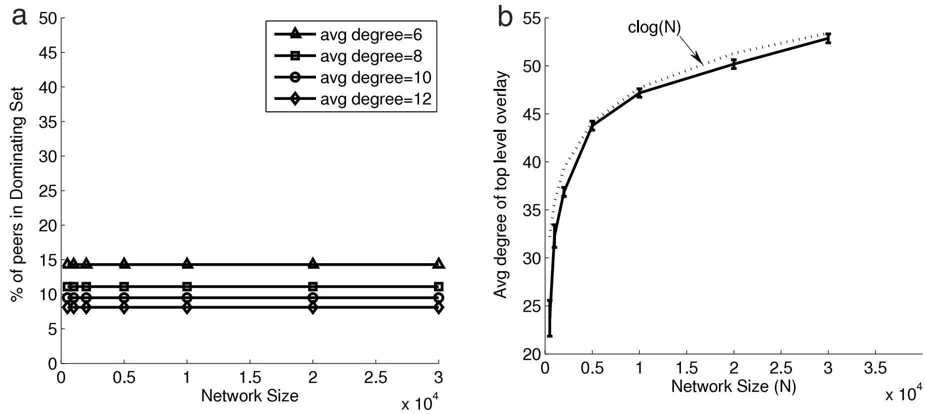

Fig. 4. (a) Percentage of peers in DS (Barabasi–Albert). (b) Average degree of DS peers.

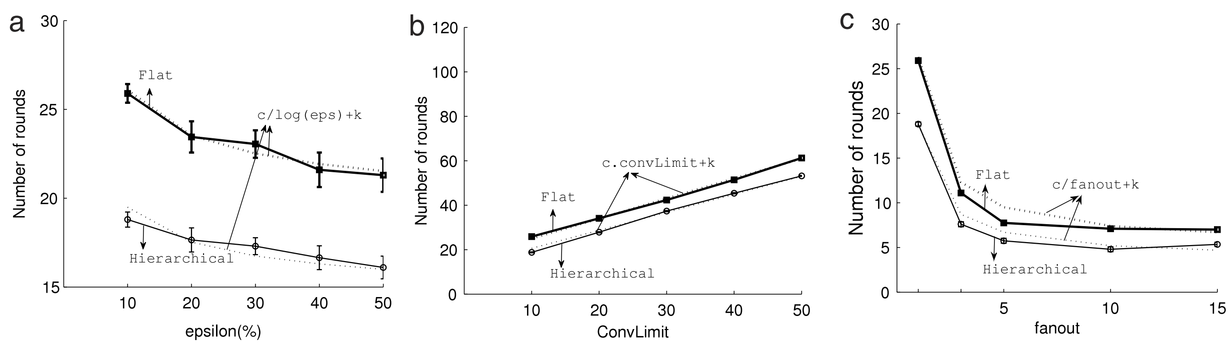

Fig. 5. Effects of (a) ε on R (b) *convLimit* on R (c) *fanout* on R in flat and hierarchical epidemics.

Table 3
Default parameter values.

Parameter	Value	Parameter	Value	Parameter	Value
N	1000	M(number of items)	100	*convLimit*	10
ε	10	*mms*	100	*fanout*	1

involves more peers, and the failure of some peers causes less information loss. In order to see the effect of topology on the percentage of peers in DS, we have also tested the Erdos–Renyi (ER) model which is well known for generating random network topologies. In this model, an edge is added between each pair of peers with a probability p independently of other edges. We selected p such that average degree is 10 for fair comparison with power-law topology. We observed that the percentage of peers in DS is around 10% of the total number of peers on the overlay.

6.2. Effect of epidemic protocol parameters

In this section, we analyze the effects of epidemic protocol parameters on the energy consumption. Epidemic protocol parameters and their default values used in the experiments are given in Table 3, where N is the number of peers and M is the number of data items aggregated through epidemic protocol.

Convergence Parameters (*convLimit*, ε): Convergence parameters are used for self-termination of peers and they have direct effects on R. Fig. 5(a) shows that R is inversely proportional to $\log \varepsilon$. This is because *convCounter* will be incremented with less chance and it will take longer time to reach *convLimit*. However, R is directly proportional to *convLimit* as depicted in Fig. 5(b), and this is because *convCounter* needs to be incremented more to take a convergence decision.

Fanout: Intuitively, increasing *fanout* will increase energy consumption in a single round. On the other hand, the algorithm will converge faster since a peer exchanges its state with more peers in a single round. Fig. 5(c) depicts that *fanout* has an inverse proportion with R. Note also that *fanout* has a direct proportion with the upper bound given in Eq. (6) since *fanout* is the cardinality of set V.

Gossip message size: Parameter *mms* is the upper bound for a gossip message size in terms of number of $\langle item, frequency\rangle$ tuples. Large *mms* means more state information is sent in a single gossip message. On one hand, this causes faster convergence, but on the other hand, the energy consumption of sending a single gossip message increases. *mms* is inversely proportional to R. Note also that *mms* is directly related to the energy cost of a peer in a single round, and these cancel each other out in our cost formulation. Recall that ProFID assumes each peer knows about its neighboring peers only and gossips with them, and hence it is based on neighborhood epidemics. In this respect, our results are also consistent with [7] that reports the efficiency of neighborhood epidemics in its power usage.

6.3. Comparison of flat and hierarchical epidemics

Fig. 6(a) and (b) depict the energy consumption characteristics of flat and hierarchical epidemics on two different topology models (BA and ER) in terms of convergence time and message cost as the system size scales up to 30,000 peers. In Fig. 6(a), when the effect of the algorithm is considered, the hierarchical epidemic approach outperforms the flat case in both topology types. Reasons for faster convergence are pre-aggregation before gossiping and the lower number of peers participating in gossiping. Faster convergence also means lower number of messages sent per peer since gossip rate is the same.

Fig. 6(b) shows that the number of messages sent per peer is much less in hierarchical epidemics when compared to flat epidemics. Since the number of gossiping peers is affected by the topology, the number of messages sent per peer is also directly affected. In ER topologies, peer degrees are uniformly distributed

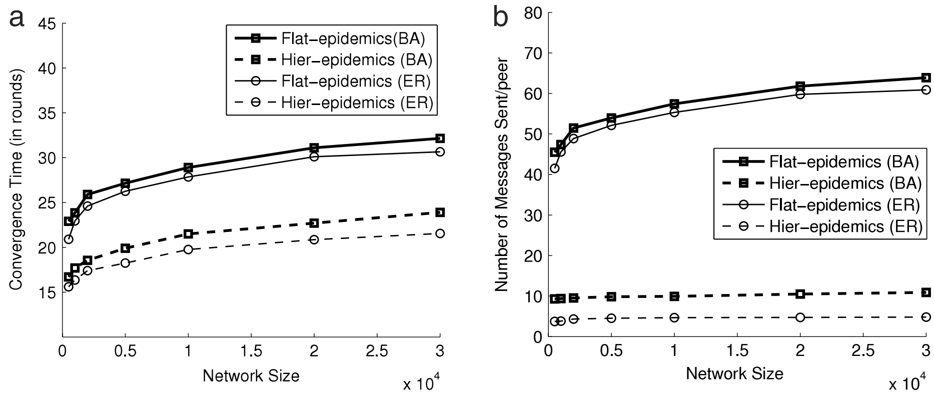

Fig. 6. Flat and Hierarchical epidemics on BA and ER topologies. (a) Convergence time. (b) Average number of gossip messages sent per peer.

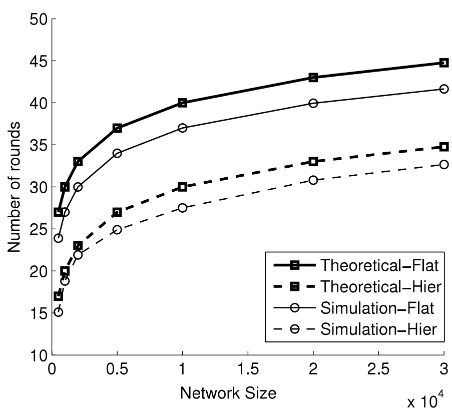

Fig. 7. Theoretical model and simulation comparison: the number of rounds to converge as a function of network size for flat and hierarchical epidemics.

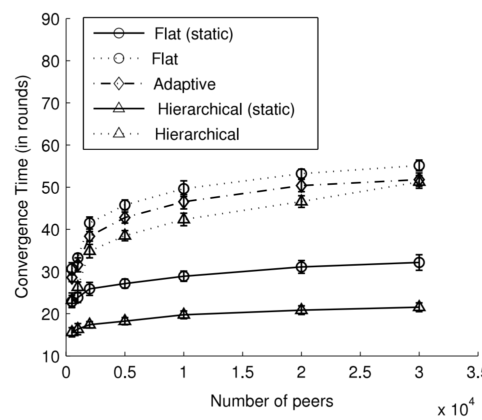

Fig. 8. Scalability in terms of convergence time.

as opposed to BA topologies where most of the peers have a low degree. Results show that slightly more peers participate in gossiping in BA when compared to ER topologies.

We also compared results obtained through our theoretical cost model formulations with the simulations. In theoretical results, the number of rounds needed to converge is calculated using Eq. (14) for hierarchical and flat epidemics. Network size (N) is scaled up to 30,000 peers. The default parameter values that are used as $\varepsilon = 10$, $convLimit = 10$, and $fanout = 1$. For both flat and hierarchical epidemics, the number of rounds (R) calculated using the theoretical cost model and those obtained with the simulations as a function of network size are shown in Fig. 7. Our findings confirm that the theoretical formulations set an upper bound and are found to be consistent with the simulation results.

6.4. Effect of churn on convergence time

Fig. 8 depicts how churn behavior affects the convergence time in flat, adaptive flat and hierarchical epidemics. In order to make the effect of churn clear, static topology (i.e. without churn) results are also included which can be considered as the optimal result because peers are always active and exchanging data until convergence. Analysis results show that churn increases the convergence time of flat and hierarchical epidemics, since peers in OFF state cannot contribute to the averaging operation and churn behavior causes a delay in convergence. In the adaptive flat epidemics, a peer searches for an ON neighbor for initiating an atomic pairwise averaging operation, and this behavior helps to reduce the convergence time. On the other hand, the flat epidemics choose a random neighbor and attempt to communicate with it. If the neighbor is in OFF state, it cannot initiate an averaging operation in that round and the convergence speed slows down.

6.5. Comparison of protocols

In this section, a well-known gossip based aggregation technique named Push-Sum is compared with the adaptive flat and hierarchical epidemics in terms of message cost and accuracy under a realistic churn model. As opposed to the flat epidemics, Push-Sum uses push-based gossiping, which means peers choose neighbors just to send their local state. They do not wait for a pull message as in the case of push–pull based gossiping. An adaptation is proposed for Push-Sum algorithm in [27] that also copes with churn by considering the neighbors' state. If the target neighbor is in OFF state, then the peer sends the gossip message to itself. On the contrary, in the adaptive flat, another gossip target is chosen.

In order to have a fair comparison, we apply the same convergence rule to the Push-Sum algorithm. Fig. 9(a) demonstrates that Push-Sum needs more time to converge in comparison to hierarchical and adaptive epidemics. This can be explained by the difference in gossiping types. Adaptive flat uses push–pull based gossip, so both ends of pairwise communication update their local states. On the other hand, Push-Sum uses push based gossip in which only one end updates its local state. Therefore, the adaptive flat performs twice as much local state update as Push-Sum in a single round. Message costs of adaptive flat and Push-Sum are close, as depicted in Fig. 9(b). This can be explained by considering Fig. 9(a) and (b) together. As shown in Fig. 9(a), adaptive flat converges faster than the Push-Sum. On the other hand, adaptive flat sends twice as many messages than Push-Sum in each round. The reason is again the difference in gossiping type. Push-Sum uses one way communication, whereas adaptive flat uses two way commu-

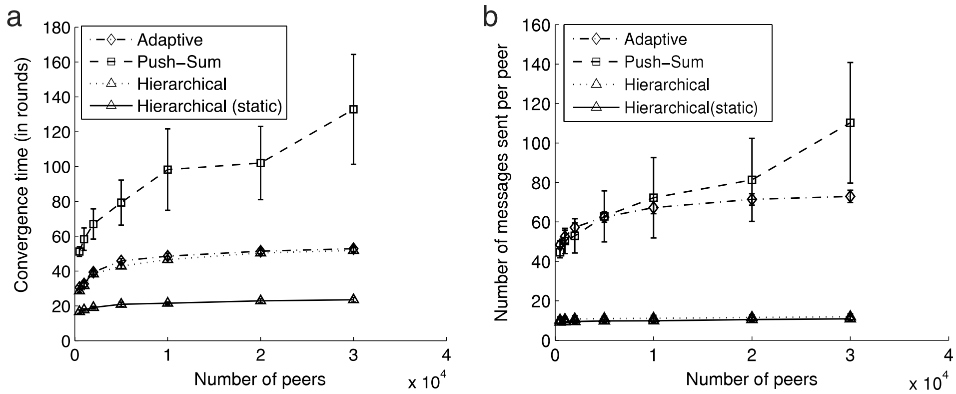

Fig. 9. Push-Sum vs. adaptive flat and hierarchical: (a) convergence time (b) average gossip messages sent per peer.

Table 4
Theoretical and simulation energy saving percentages (in convergence time).

Network size	Theoretical	Simulation
500	37.06%	36.83%
1 000	33.34%	30.38%
2 000	30.30%	27.00%
5 000	27.04%	26.75%
10 000	25.00%	25.68%
20 000	23.26%	22.91%
30 000	22.34%	21.59%

nication during a single pairwise aggregate operation. Hence, it can be concluded that aggregation can be performed with the adaptive flat in less number of rounds and with a similar message cost when compared to Push-Sum.

6.6. Energy savings comparison

Table 4 shows the energy saving percentage measurements for simulations and the theoretical model for varying network size (from 500 peers to 30,000 peers). In general, the theoretical energy savings results (based on the energy cost model formulation) are very close to the values through simulations for all network sizes.

We also analyzed the effect of different overlay topologies used in our simulations on the energy savings. Barabasi–Albert and Erdos–Renyi topologies for different network sizes are considered. Tables 5 and 6 show the energy saving percentages in convergence time and in number of messages sent per peer, respectively. Energy savings in convergence time for both topologies as the network size scales up are similar. When energy savings in number of messages sent are considered, a difference between two topology types is observed. The energy savings for the case of Erdos–Renyi topologies are observed on average 10% higher than that of Barabasi–Albert topologies for all network sizes. This can be explained by the fact that Barabasi–Albert forms a scale-free network and the number of nodes in the overlay increases over time, whereas Erdos–Renyi generates a random graph.

7. Conclusions and future work

Designing energy efficient protocols has become as important as considering performance criteria such as scalability, reliability and fault-tolerance. We studied the energy efficiency aspect of epidemic protocols used in large scale distributed systems, and proposed a novel hierarchical epidemic approach that uses the dominating set while constructing the hierarchy. The proposed approach utilizes the benefits of both epidemic and hierarchical approaches. It uses only local knowledge and provides the possibility of active/passive peers to save energy. Robustness against peer failures is improved and the message overhead is significantly reduced thanks to the hierarchy. We developed energy cost model formulations for flat and hierarchical epidemics. In contrast to the prior works, our study is the first one that proposes energy cost models for generic peers using epidemic communication, and examines the effect of protocol parameters to characterize energy consumption. As a case study protocol, we use our epidemic protocol ProFID for frequent items discovery in P2P systems. Through large scale simulations on PeerSim, we analyzed the effect of protocol parameters on energy consumption, compared flat and hierarchical epidemic approaches as well as investigated their resilience under realistic churn. As future work, we aim to extend our analysis for accuracy performance metric and investigate its effect on flat and hierarchical epidemic protocols.

Table 5
Energy saving percentages (in convergence time).

Network size	Barabasi–Albert	Erdos–Renyi
500	27.08%	25.37%
1 000	25.79%	28.76%
2 000	28.38%	29.27%
5 000	26.71%	30.48%
10 000	25.61%	29.09%
20 000	27.02%	30.74%
30 000	25.67%	29.73%

Table 6
Energy saving percentages (in number of messages sent per peer).

Network size	Barabasi–Albert	Erdos–Renyi
500	79.56%	91.02%
1 000	80.23%	91.67%
2 000	81.52%	91.18%
5 000	81.78%	91.35%
10 000	82.71%	91.6%
20 000	83.03%	92.11%
30 000	82.96%	92.1%

References

[1] K. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, Y. Minsky, Bimodal multicast, ACM Transactions on Computer Systems 17 (2) (1999) 41–88.
[2] O. Ozkasap, Z. Genc, E. Atsan, Epidemic-based reliable and adaptive multicast for mobile ad hoc networks, Computer Networks 53 (2009) 1409–1430.
[3] P. Eugster, R. Guerraoui, S.B. Handurukande, P. Kouznetsov, A.M. Kermarrec, Lightweight probabilistic broadcast, ACM Transactions on Computer Systems 21 (4) (2003) 341–374.
[4] E. Cem, O. Ozkasap, ProFID: practical frequent items discovery in peer-to-peer networks, Future Generation Computer Systems 29 (6) (2013) 1544–1560. http://dx.doi.org/10.1016/j.future.2012.10.002.
[5] O. Ozkasap, M. Caglar, A. Alagoz, Principles and performance analysis of SeCond: a system for epidemic peer-to-peer content distribution, Journal of Network and Computer Applications 32 (3) (2009) 666–683.
[6] P. Eugster, R. Guerraoui, A.M. Kermarrec, L. Massoulie, Epidemic information dissemination in distributed systems, Computer 37 (5) (2004) 60–67.

[7] R. van Renesse, Power-aware epidemics, in: Proc. of IEEE Symposium on Reliable Distributed Systems, 2002.
[8] S. Rajeswari, D.Y. Venkataramani, An adaptive energy efficient and reliable gossip routing protocol for mobile adhoc networks, International Journal of Computer Theory and Engineering 2 (5) (2010) 740–745.
[9] S. Fauji, K. Kalpakis, A gossip-based energy efficient protocol for robust in-network aggregation in wireless sensor networks, in: 2011 IEEE International Conference on Pervasive Computing and Communications Workshops, PERCOM Workshops, 2011, pp. 166–171.
[10] S.D.T. Ahmed, A. El-Saddik, A dominating set based peer-to-peer protocol for real-time multi-source collaboration, in: WETICE, 2007.
[11] L. Jia, R. Rajaraman, T. Suel, An efficient distributed algorithm for constructing small dominating sets, Distributed Computing (ISSN: 0178-2770) 15 (2002) 193–205.
[12] C. Yang, X. Li, Dominating-set-based searching in peer-to-peer networks, International Journal of High Performance Computing and Networking 3 (2005) 205–210.
[13] C. Seo, G. Edwards, et al. A framework for estimating the energy consumption induced by a distributed system's architectural style, in: Proc. of SAVCBS, 2009.
[14] R. Basmadjian, H. de Meer, An approach to reduce the energy cost of the arbitrary tree replication protocol, in: Proc. of e-Energy, 2010, pp. 151–158.
[15] F. Kuhn, R. Wattenhofer, Constant-time distributed dominating set approximation, in: Proceedings of the Twenty-Second Annual Symposium on Principles of Distributed Computing, PODC'03, 2003, pp. 25–32.
[16] G. Pandurangan, P. Raghavan, E. Upfal, Building low-diameter P2P networks, in: FOCS, 2001, pp. 492–499.
[17] S. Krishnamurthy, S. El-Ansary, E. Aurell, S. Haridi, A statistical theory of chord under churn, CoRR abs/cs/0501069.
[18] D. Liben-Nowell, H. Balakrishnan, D.R. Karger, Analysis of the evolution of peer-to-peer systems, in: PODC, 2002, pp. 233–242.
[19] D. Leonard, Z. Yao, V. Rai, D. Loguinov, On lifetime-based node failure and stochastic resilience of decentralized peer-to-peer networks, IEEE/ACM Transactions on Networking 15 (3) (2007) 644–656.
[20] A. Duminuco, E. Biersack, T. En-Najjary, Proactive replication in distributed storage systems using machine availability estimation, in: CoNEXT, vol. 27, 2007.
[21] The PeerSim simulator, 2012. http://peersim.sf.net.
[22] A.L. Barabasi, R. Albert, Emergence of scaling in random networks, Science (New York, NY) (ISSN: 1095-9203) 286 (5439) (1999) 509–512.
[23] P. Erdos, A. Renyi, On the strength of connectedness of a random graph, Acta Mathematica Academiae Scientiarum Hungaricae (ISSN: 0001-5954) 12 (1–2) (1961) 261–267.
[24] E. Fernández-Casado, M.S. Artigas, P.G. López, Affluenza: towards universal churn generation, in: Peer-to-Peer Computing, 2010, pp. 1–2.
[25] Z. Yao, D. Leonard, X. Wang, D. Loguinov, Modeling heterogeneous user churn and local resilience of unstructured P2P networks, in: ICNP, 2006, pp. 32–41.
[26] M. Jelasity, Ö. Babaoglu, T-Man: gossip-based overlay topology management, in: Engineering Self-Organising Systems, 2005, pp. 1–15.
[27] D. Kempe, A. Dobra, J. Gehrke, Gossip-based computation of aggregate information, in: Proc. of FOCS, 2003, pp. 482–491.

Oznur Ozkasap received her M.S. and Ph.D. degrees in Computer Engineering, both from Ege University, in 1994 and 2000, respectively. From 1997 to 1999 she was a graduate research assistant at Cornell University, Department of Computer Science, where she completed her Ph.D. dissertation. She is currently an associate professor in the Department of Computer Engineering at Koc University, which she joined in 2000. Her research interests include distributed systems, multicast protocols, peer-to-peer systems, bio-inspired distributed algorithms, mobile ad hoc networks, energy efficiency, cloud computing and computer networks. Prof. Ozkasap is an area editor of Computer Networks (*COMNET*) journal, Elsevier Science, and a management committee member of the European COST Action IC0804: Energy efficiency in large scale distributed systems.

Emrah Cem received his B.S. degree in Computer Engineering and M.S. degree in Electrical and Computer Engineering from Koc University, in 2008 and 2010, respectively. He is currently pursuing his Ph.D. degree at the University of Texas, Dallas. His research interests include distributed systems, peer-to-peer protocols, and statistical analysis of large networks.

Sena Efsun Cebeci received her B.S. degrees in Computer Engineering and Computer Science from Bahcesehir University, in 2008. She earned her M.S. degree in Computer Science from Oakland University, in 2010. She has started her Ph.D. degree at Oakland University in 2011, in Computer Science and Informatics. She currently continues her Ph.D. degree at Koc University. Her research interests include energy efficiency in peer-to-peer networks, distributed systems and web services in e-Healthcare.

Tugba Koc received her B.S. degree in Computer Engineering from Kadir Has University in 2010. Then, she worked as a graduate research and teaching assistant at Koc University. Her research interests include distributed systems, computer networks and energy efficiency in peer-to-peer networks.