



An analytical framework for self-organizing peer-to-peer anti-entropy algorithms

Öznur Özkasap^a, Mine Çağlar^{b,*}, Emine Şule Yazıcı^b, Selda Küçükçiççi^b

^a Department of Computer Engineering, Koç University, Istanbul, Turkey

^b Department of Mathematics, Koç University, Istanbul, Turkey

ARTICLE INFO

Article history:

Received 16 June 2008
Received in revised form 18 September 2009
Accepted 30 September 2009
Available online 9 October 2009

Keywords:

Peer-to-peer
Epidemic
Anti-entropy
Self-organizing
Counting
Overhead
Delay
Markov chain

ABSTRACT

An analytical framework is developed for establishing exact performance measures for peer-to-peer (P2P) anti-entropy paradigms used in biologically inspired epidemic data dissemination. Major benefits of these paradigms are that they are fully distributed, self-organizing, utilize local data only via pair-wise interactions, and provide eventual consistency, reliability and scalability. We derive exact expressions for infection probabilities through elaborated counting techniques on a digraph. Considering the first passage times of a Markov chain based on these probabilities, we find the expected message delay experienced by each peer and its overall mean as a function of initial number of infectious peers. Further delay and overhead analysis is given through simulations and the analytical framework. The number of contacted peers at each round of the anti-entropy approach is an important parameter for both delay and overhead. These exact performance measures and theoretical results would be beneficial when utilizing the models in several P2P distributed system and network services such as replicated servers, multicast protocols, loss recovery, failure detection and group membership management.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Self-organization has emerged as a promising paradigm when designing services and applications for distributed systems. A self-organizing system consists of a large number of components that function autonomously and interact via basic and local rules. This paradigm has the potential of being scalable, robust and fault-tolerant since the individual components do not depend on centralized mechanisms and they are capable of tolerating the failures of the other components. The global behavior becomes apparent from the local interactions and such systems are often inspired by biological phenomena.

In this endeavor, biologically inspired approaches have become an appealing alternative for building self-organizing P2P applications in distributed settings as opposed to traditional centralized network mechanisms. An efficient approach for data dissemination in distributed systems is to utilize biologically inspired epidemic algorithms. We investigate variations of the epidemic algorithms used in the context of distributed data dissemination and derive exact performance measures for them. Epidemic algorithms are fully distributed and randomized approaches such that every peer in a data diffusion session picks a (subset of the other) peer(s) randomly for efficient propagation of data, which happens through periodic rounds. The underlying epidemics theory for the biological systems study the spreading of infectious diseases through a population [1,2]. When applied to a data diffusion application, such protocols have beneficial features such as scalability, robustness against failures and provision of eventual consistency.

* Corresponding author. Tel.: +90 212 338 1315; fax: +90 212 338 1559.
E-mail address: mcaglar@ku.edu.tr (M. Çağlar).

There are different classes of epidemic processes, one of which is referred to as anti-entropy. The term *anti-entropy* [3] refers to protocols that detect and correct inconsistencies in a distributed system by means of continuous epidemic rounds (or gossiping). The length of each round is larger than the maximum round-trip time between peers in the system. The round-trip time corresponds to the duration of a remote procedure call over the links used by the protocol. In each round, every peer picks another peer at random, and sends its state information. We study three approaches for data dissemination, namely pull, push, and push&pull cases, as particular models of anti-entropy. Algorithms and details of these approaches are described in Section 3.

In this article, we propose an analytical framework for P2P epidemic anti-entropy mechanisms. The main contribution of our study is the derivation of exact expressions for infection probabilities as the data diffusion progresses. We model the diffusion through a Markov chain which represents the number of peers informed at each round. The transition probabilities on the chain are calculated through elaborated counting techniques on a digraph, with no resort to approximate probability distributions that rely on several independence assumptions. Once the transition probability matrix is known exactly, then the mean delay until a particular peer gets the data, and the expected time until all peers receive the data, can be computed numerically.

Our preliminary results have been reported in [4] for the exact infection probability distributions and the associated mean delays. In the present study, we find the distribution of the dissemination time to the whole system in addition to its mean. Then, the exact results are compared in depth with the approximate probability distributions as well as asymptotical delay results. We further analyze the push approach in the case when each peer contacts with multiple peers at each round. This speeds up the epidemic diffusion while producing overhead in the form of duplicate message transmission. A key contribution of the present study is the extensive evaluation of the degree of redundancy in data dissemination such as the overhead. Simulations are performed to verify the latter analytical results qualitatively.

The article is organized as follows. Related work is discussed in Section 2. In Section 3, our models for the pull, push and push&pull anti-entropy are explained. The exact diffusion probabilities are derived in Section 4. In Section 5, the Markov chain formulation and delay computations are given. Section 6 describes the overhead analysis for the push model, and discusses the analytical and simulation results. Finally, Section 7 states the conclusions.

2. Related work

One of the first studies that applied epidemic methods to computer systems used the idea for spreading updates in a replicated database [3]. Several succeeding work utilized epidemic (or sometimes so-called gossip-style, to reflect rumor propagation in a social network) communication in a variety of contexts such as large-scale direct mail systems [5], group membership tracking [6], support for replicated services [7], message garbage collection [8], failure detection [9], loss recovery in reliable multicast [10], and distributed information management [11]. An overview of epidemic data dissemination is given in [12], where the focus is on four design constraints, namely, membership, network awareness, buffer management, and message filtering.

In [13], flat and hierarchical gossip-based protocols are evaluated considering the relationship between reliability of dissemination and system parameters such as population size, link/node failure rates and fan-out. For the gossip protocol, a push approach is considered as described in [3]. The way probabilistic gossip algorithms can be modeled via random graphs are described. Organizing peers into a hierarchy based on their network proximity is shown to reduce the network load considerably in comparison to flat gossip.

Another study [14] is based on the pull approach in the context of a gossip-based broadcast protocol named *lpbcast* (lightweight probabilistic broadcast). The protocol focuses on scalable buffer management and membership management. In addition to dissemination of event notifications, gossip messages are also used for propagating digests of notifications and membership information. As a key result of this study, it has been shown that there is little dependency between the reliability of dissemination and the size of views each member has.

Likewise, [15] proposes a general purpose gossip-based framework for a peer-sampling service which provides each node a random subset of peers to gossip with. Each peer has a partial view and this view is updated periodically using gossiping. It has been shown that, in gossip-based membership protocols a push&pull model for communication should be preferred since pull only or push only models can cause partitioning of peers.

Exact as well as asymptotical distributions have been studied for different epidemic models from anti-entropy. In our prior work [16], pull and push anti-entropy approaches have been compared through a binomial probability distribution for information flow where the push approach is shown to be superior in terms of message delays. In [17,18], the epidemic process is defined on a random graph. In [19], the infection is spread through random contact in a manner less structured than a random graph and simpler than anti-entropy.

In the file-sharing context, [20] provides analytical bounds for the dissemination time using the epidemic nature of the file-sharing applications. Although the dissemination process is optimized in many ways in such applications, the true diffusion time is found to be close to the analytical bounds. The lower bound for the dissemination time is calculated by the so-called occupancy distribution [21], which is exact for the diffusion mechanism described in [20]. This mechanism is very similar to the push approach described in the present article with one distinction that brings their diffusion probability formula closer to that of our pull approach. Since the latter formula is computationally harder to evaluate, [20] carry on simulations for larger group sizes.

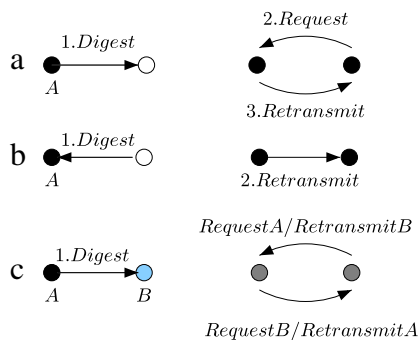


Fig. 1. Model descriptions.

Our study differs from earlier work in the following aspects. First of all, the anti-entropy algorithms we use for data dissemination have some differences from the ones described in [3] proposed for update exchange in a replicated database. In [3], when a site makes a call to another site in an epidemic round, in order to resolve differences on an update in the database, these two sites need to exchange their updates and they both execute the actions for resolving differences. However, in our algorithms an update, or data, is not sent in a digest message. Instead, identifiers (such as sequence numbers) of the local data at a peer are put into the digest. Also, actions for resolving differences are only performed at a peer upon receiving a digest message. If differences in some data are found, then the receiving peer either pulls data, pushes data or performs both push&pull depending on the type of anti-entropy. Sending digest messages helps to reduce redundant data transmissions. The action of push, pull or push&pull is performed by the calling site in [3], whereas it is performed by the gossip receiving peer in our case. Consequently, the push and pull terms swap in [3] and the present work.

Our pull anti-entropy approach is similar to the one used in Bimodal Multicast [10] for message loss recovery. However, the approach in this study is used for peer-to-peer data dissemination in contrast to the multicast loss recovery. What is more, we derive exact diffusion probabilities while previous studies [10,14,16] provide only approximate expressions. Therefore, the delay computations of the present study are also exact. We evaluate the mean delay per peer in addition to the dissemination time for which asymptotic results are provided in [22,23] to compare the push and pull approaches. In [23], push and pull mechanisms are investigated asymptotically in the context of spreading multiple messages as in file sharing applications. Our results are consistent with these asymptotic studies.

3. Model descriptions

In the anti-entropy process [1], non-faulty peers are always either susceptible or infectious. According to the terminology of epidemiology, a peer holding data or an update it is ready to share is called *infectious*. If a peer has not yet received data, it is called *susceptible*. Diffusion of data progresses periodically via rounds of epidemics. At each round, every peer randomly picks a *fan-out* number of peers, and sends its digest message. A digest message contains the identifiers (such as sequence numbers) of the data received by the peer. We study the pull, push and push&pull approaches for data dissemination that are described next. These approaches execute in a fully distributed manner at each peer.

Pull approach: In this approach, spreading data is triggered by susceptible peers (by *pulling* data) when they are picked as digest destinations by infectious peers. The steps involved in the dissemination between two such peers are depicted in Fig. 1(a) for a single data message. The infectious peer (on the left) has data labeled A. The infectious peer sends a digest message including its state information. On receiving the digest and comparing it with its local data, the susceptible peer finds out it lacks A and sends a request for A back to the infectious peer. Upon getting the request, the infectious peer sends a retransmission of data A which causes the susceptible peer to be infectious for A.

The actions performed at each peer for the pull anti-entropy data dissemination are given in Algorithm 1. At each epidemic round, every peer picks a randomly *fan-out* number of peers and sends its digest (containing the identifiers of the data messages it has received). In fact, each peer in the system performs state exchange periodically and concurrently with the others. We define three events that can occur in a round at a peer, namely *digest receipt*, *request receipt*, and *retransmission receipt*. When a digest message is received by a peer, it compares the data IDs in the digest with the IDs of its local data. If the peer determines some data messages that it lacks, then it can request the data from the digest sender. When a peer gets a data request message, it retransmits the data requested from its buffer. We call this operation retransmission to distinguish it from the original transmission by the data source. Thus, a retransmission is done as a result of a digest and request transmission in epidemic rounds, and it is used for reliable data dissemination. When a retransmission of data is received by a peer, it updates its local buffer with the new data.

Push approach: When a susceptible peer picks an infectious peer and sends its digest, this would trigger data dissemination (by *pushing* data) from the infectious peer to the susceptible peer. Hence, spreading data is triggered by infectious peers when they are selected as digest targets by susceptible peers. The steps involved in the dissemination between two such

Algorithm 1 Pull Anti-entropy Algorithm

Algorithm executed periodically once per epidemic round at each peer p :

for fan-out number of randomly selected peers q **do**

 Send *Digest* (containing list of p 's data ids) to q

end for

Event(Digest Receipt) from peer r :

 Compare *Digest* of r with p 's local data

if r has data d that p is missing **then**

 Request d from r

end if

Event(Request Receipt) for data d from peer q :

 Retransmit d to q

Event(Retransmission Receipt) for data d :

 Update p 's local data with d

peers are depicted in Fig. 1(b) where the infectious peer (on the left) has data labeled A. The infectious peer on receiving digest and comparing it with its local data finds out that the digest owner lacks A and directly retransmits, or pushes, data A, which causes the susceptible peer to become infectious for A.

Push anti-entropy data dissemination is described in Algorithm 2. Like in the pull algorithm, at each epidemic round, digest transmission is performed. We define two events that can occur in a round at a peer, namely *digest receipt* and *retransmission receipt*. Note that in contrast to the pull approach, in the push approach, no request messages are used. When a peer receives a digest message, it compares the data IDs in the digest with the IDs of its local data. If there are data messages that the digest sender lacks, then it retransmits these data to the digest sender. When a peer receives retransmission of data, it updates its local buffer with the new data.

Algorithm 2 Push Anti-entropy Algorithm

Algorithm executed periodically once per epidemic round at each peer p :

for fan-out number of randomly selected peers q **do**

 Send *Digest* (containing list of p 's data ids) to q

end for

Event(Digest Receipt) from peer r :

 Compare *Digest* of r with p 's local data

if p has data d that r is missing **then**

 Retransmit d to r

end if

Event(Retransmission Receipt) for data d :

 Update p 's local data with d

Push&pull approach: This is a hybrid of the pull and push approaches described above. When a peer sends its digest to a randomly selected peer in the population, this may trigger data dissemination at both peers. Consider the case where a peer has data A and the other has data B, as illustrated in Fig. 1(c). When the former selects the latter as the digest target in a given round, data A and B would be disseminated to the peer that lacks it using pull and push approaches together. In particular, push&pull would be useful for delay sensitive applications since it decreases the overall delay during data dissemination at the cost of possible duplicate data transmissions.

The actions performed at each peer for the push&pull anti-entropy data dissemination are given in Algorithm 3. We define three events that can occur in a round at a peer, namely *digest receipt*, *request receipt*, and *retransmission receipt*. On receipt of a digest message, a peer compares the data IDs in the digest with the IDs of its local data. If the peer determines some data messages that it lacks, then it can request the data from the digest sender. This corresponds to the pull process. In addition, if the peer determines some data messages that the digest sender lacks, then it retransmits these data to the digest sender. This corresponds to the push process. On receipt of a data request message, a peer retransmits the data requested from its buffer. On receipt of a retransmission of data, a peer updates its local buffer with the new data.

For the anti-entropy approaches described above, when the fan-out is 1, there are no duplicate data messages sent to an already infectious node. On the other hand, when the fan-out is greater than 1, duplicate retransmissions of data are possible for the push and push&pull approaches. Duplicates could happen when multiple infectious peers receive a digest message from a susceptible peer, and they all retransmit the data that the susceptible peer lacks. In this study, we also analyze the overhead formed by the duplicate messages. Note that for the pull approach, there is no overhead in the case of fan-out greater than 1, since a susceptible peer would request data only from one of the infectious peers it becomes aware of.

For the system failures, we consider a fail-stop model for peers and receive-omissions for message losses. Such failures are assumed to be transient. For instance, if a data transmission to a peer fails due to a receive omission, continuous epidemic

Algorithm 3 Push&pull Anti-entropy Algorithm

Algorithm executed periodically once per epidemic round at each peer p :
for fan-out number of randomly selected peers q **do**
 Send Digest (containing list of p 's data ids) to q
end for
Event(Digest Receipt) from any peer r :
Compare Digest of r with p 's local data
if r has a data d that p is missing **then**
 Request d from r
end if
if p has a data d that r is missing **then**
 Retransmit d to r
end if
Event(Request Receipt) for data d from any peer q :
Retransmit d to q
Event(Retransmission Receipt) for any data d :
Update p 's local data with d

rounds will help to disseminate data. Note that in order to obtain exact expressions, our simplified model of the analytical framework does not assume network delays and packet losses. However, our network simulations used for comparison with analytical findings consider network delays between peers and also realistic topologies.

4. Exact diffusion probabilities

In this section, we will restrict our attention to the processes of distributing a single data message with the assumption that the fan-out is 1. Therefore, a peer with a copy of the data message is referred to as infectious; otherwise, it would be susceptible. A digraph D is a directed graph consisting of a node set $V(D)$ and an arc set $E(D)$, where each arc is an ordered pair of nodes. If $(u, v) \in E(D)$ then we call (u, v) an arc, with u being the tail and v being the head. As the anti-entropy protocol proceeds with a request phase before the actual data transmission, each step of the diffusion process can be represented by a digraph D where a node corresponds to a peer in the population and the arcs correspond to the selection of a peer by another. If node u chooses to communicate with node v , then there will be an arc with tail u and head v in D . Since the fan-out is taken to be 1, the out degree of each node will be 1 in D as it chooses exactly one node at each step. The digraph, which describes a realized request phase, also characterizes the actual data transmission that immediately follows. Exactly how many and which susceptible peers get infected can be read from this digraph according to the type of anti-entropy protocol.

4.1. Derivation of probability distributions

Assume there are n peers in the population. It follows that the number of all possible digraphs with n nodes is $(n - 1)^n$. All of these digraphs are equally likely for each step of the dissemination process. Therefore, we will count the number of digraphs that infect i more nodes and take the ratio of this number with the number of all possible digraphs to find the probability of infecting i more nodes at each step. Note that, if there are k infectious nodes present, after one step there will be $k + i$ infectious nodes.

Let S be the set of all susceptible nodes and I be the set of all infectious nodes with $|I| = k$ and $|S| = n - k$. For simplicity, we will denote arcs with susceptible heads and infectious tails as IS -arcs; similarly, arcs with infectious heads and susceptible tails, infectious heads and infectious tails, and susceptible heads and susceptible tails will be represented as SI -arcs, II -arcs, and SS -arcs, respectively. Note that D is the disjoint union of four subgraphs formed by IS -arcs, SI -arcs, II -arcs, and SS -arcs.

We will use Stirling numbers of the second kind for our calculations. The Stirling number of the second kind, denoted by $S(n, k)$, is defined as the number of all partitions of a n -element set into k nonempty subsets. The Stirling numbers of the second kind satisfy the relation

$$S(n, k) = kS(n - 1, k) + S(n - 1, k - 1).$$

Also $S(n, k)$ can be written in closed form as

$$S(n, k) = \frac{1}{k!} \sum_{i=0}^k (-1)^{k-i} \binom{k}{i} i^n.$$

For further information on these numbers see [24].

Pull case: We form the digraph D as above. In the pull case, a susceptible node s will be infected if and only if there exists an IS -arc in D with the head s . Therefore, SI -arcs, II -arcs, and SS -arcs will not contribute to the number of new infectious

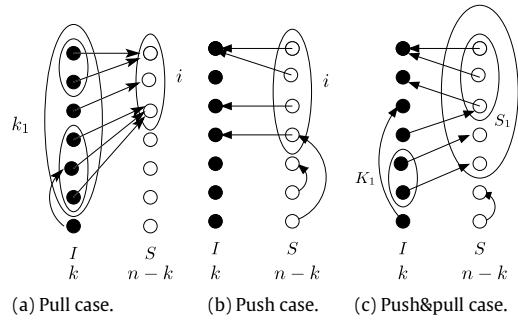


Fig. 2. Illustration of the anti-entropy approaches.

nodes. Fig. 2(a) illustrates the pull case. We will determine the number of digraphs representing a step that results in i more infectious nodes.

The number of different possible subgraphs formed by SI -arcs and SS -arcs is $(n - 1)^{n-k}$, since each of the $n - k$ nodes in S can be adjacent to $n - 1$ different nodes.

Now we need to count the number of different possible digraphs that can be formed by IS -arcs and II -arcs. Let k_1 be the number of IS -arcs. Note that k_1 has to be at least i since each IS -arc infects at most one new node in S . Also there are $\binom{k}{k_1}$ such k_1 -subsets of I . We have $k - k_1$ II -arcs and the number of different possible subgraphs formed by these arcs is $(k - 1)^{k-k_1}$. Finally, we will count the number of different subgraphs that can be formed by IS -arcs. Among $n - k$ susceptible nodes there are $\binom{n-k}{i}$ different i -subsets of S that may be infected. There are $S(k_1, i)!$ different ways for k_1 nodes to infect exactly i new nodes since we partition k_1 nodes into i subsets where each subset represents the set of nodes in I that chooses to communicate with a specific node in S . Therefore, the number of different subgraphs that can be formed by IS -arcs and II -arcs is $\sum_{k_1=i}^k \binom{k}{k_1} (k - 1)^{k-k_1} \binom{n-k}{i} S(k_1, i)!$. Hence, the probability of infecting i more nodes in the next step is

$$p(i|k) = \frac{(n - 1)^{n-k} \binom{n-k}{i} i! \sum_{k_1=i}^k \binom{k}{k_1} (k - 1)^{k-k_1} S(k_1, i)}{(n - 1)^n}$$

$$= \frac{\binom{n-k}{i} i! \sum_{k_1=i}^k \binom{k}{k_1} (k - 1)^{k-k_1} S(k_1, i)}{(n - 1)^k}$$

where $k = 2, 3, \dots, n - 1$ and $i = 0, 1, \dots, n - k$.

When $k = 1$, we can easily see that $p(0|1) = 0$ and $p(1|1) = 1$.

Push case: In the push case, a susceptible node s will be infected if and only if there exists an SI -arc with the tail s . Therefore, IS -arcs, II -arcs, and SS -arcs will not contribute to the number of new infectious nodes.

Fig. 2(b) illustrates the push case. The number of different possible subgraphs formed by IS -arcs and II -arcs is $(n - 1)^k$. Since i new nodes will be infected there are iS I -arcs and $\binom{n-k}{i}$ different i -subsets of S . For each SI -arc there are k different choices for the head of the arc; therefore there are $\binom{n-k}{i} k^i$ different possible subgraphs formed by these arcs. Finally, as there are $n - k - i$ SS -arcs, the number of different possible subgraphs formed by SS -arcs is $(n - k - 1)^{n-k-i}$. Hence, the probability of infecting i more nodes after this step is

$$p(i|k) = \frac{(n - 1)^k \binom{n-k}{i} k^i (n - k - 1)^{n-k-i}}{(n - 1)^n} = \frac{\binom{n-k}{i} k^i (n - k - 1)^{n-k-i}}{(n - 1)^{n-k}}$$

where $k = 1, 2, \dots, n - 2$ and $i = 0, 1, \dots, n - k$.

Clearly, when $k = n - 1$, we get $p(0|k) = 0$ and $p(1|k) = 1$. The probability distribution above can be rewritten as

$$p(i|k) = \binom{n-k}{i} \left(\frac{k}{n-1}\right)^i \left(\frac{n-k-1}{n-1}\right)^{n-k-i} \tag{1}$$

which can now be recognized as a binomial distribution with parameters $n - k$ and success probability $k/(n - 1)$.

Push&pull case: In the hybrid case of push and pull, a susceptible node s will be infected if and only if there exists either an SI -arc with the tail s or an IS -arc with the head s . Therefore, II -arcs and SS -arcs will not contribute to the number of new infectious nodes.

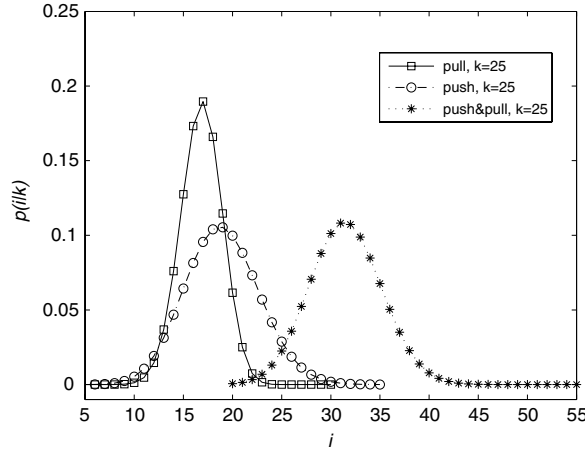


Fig. 3. The probability distributions of i more nodes getting infected, for $k = 25$ starting number of infectious peers and $n = 100$.

Fig. 2(c) illustrates the push&pull case. There are i new infectious nodes and $\binom{n-k}{i}$ different i -subsets of S . Let S_1 be the set of the tails of SI -arcs with $|S_1| = i_1$. There are $\binom{i}{i_1}$ i_1 -subsets of each i -set. The number of different possible subgraphs formed by SI -arcs and SS -arcs is $\binom{n-k}{i} \binom{i}{i_1} k^{i_1} (n-k-i_1)^{n-k-i_1}$.

Let K_1 be the set of nodes that are the tails of the IS -arcs whose heads are in $S \setminus S_1$, with $|K_1| = k_1$. There are $\binom{k}{k_1}$ different ways to choose K_1 . These k_1 arcs will infect $i - i_1$ new nodes and there are $S(k_1, i - i_1)(i - i_1)!$ different ways to do this. Finally, the remaining $k - k_1$ arcs can be chosen in $(k - 1 + i_1)^{k-k_1}$ different ways. Therefore, the number of different possible subgraphs formed by IS -arcs and II -arcs can be calculated as

$$\Theta_{k,i}(i_1) = \sum_{k_1=i-i_1}^k \binom{k}{k_1} (k - 1 + i_1)^{k-k_1} S(k_1, i - i_1)(i - i_1)!$$

Hence, the probability of infecting i more nodes in the next step is

$$p(i|k) = \frac{\binom{n-k}{i}}{(n-1)^n} \sum_{i_1=0}^i \binom{i}{i_1} k^{i_1} (n-k-i_1)^{n-k-i_1} \Theta_{k,i}(i_1)$$

where $k = 2, 3, \dots, n - 2$ and $i = 0, 1, \dots, n - k$.

Now, we will consider the end points. If $k = n - 1$, then $p(0|k) = 0$ and $p(1|k) = 1$. If $k = 1$, then $p(0|1) = 0$ and $p(i|1) = \frac{\binom{n-1}{i}(n-2)^{n-i-1}(n-1)}{(n-1)^n}$ for all $i \geq 1$. To see this, there can be $\binom{n-1}{i}$ different i -subsets of S and i different possibilities for the head of an SI -arc; call this node u . There are $n - 1$ possibilities for the arc with the tail u . The arcs coming out of the rest of the $i - 1$ nodes will have heads in I and there is a unique way to do this. Finally, the remaining $n - i - 1$ arcs can be chosen in $(n - 2)^{n-i-1}$ different ways.

For an illustrative comparison, we plot the diffusion probabilities in Figs. 3 and 4 for $k = 25$ and $k = 75$, respectively. Although the mean and variances vary, all distributions are close to a normal distribution for $n = 100$. A normal distribution is expected for fairly large group sizes since it is a good approximation for the binomial distribution in the push case. In pull and push&pull cases, the distributions are close to an occupancy distribution which has a Poisson approximation and hence is close to a normal distribution for large n [21].

4.2. Comparison with approximate results

For the pull approach, we note that binomial probability distributions suggested in [16,14] are based on approximate arguments and in fact do not represent the true distribution. In [14], probabilistic broadcasting is considered through gossiping, as in [3]. In the pull case, the epidemic spreads by the call of an infectious peer to a susceptible one. It is adequately argued in [16,14] that the probability of a fixed susceptible peer's getting infected is $1 - q^k$ with $q = 1/(n - 1)$ when there are k infectious peers. Clearly, q is the probability that an arbitrary infectious peer does not select the mentioned susceptible peer as the gossip target and q^k is the probability that none of the infectious peers select it. Furthermore, q can be written more elaborately, as in [10,14], to take into account process and link failures. Then, a binomial distribution follows if such an infection for the fixed susceptible peer happens independently from the other susceptible peers. Under

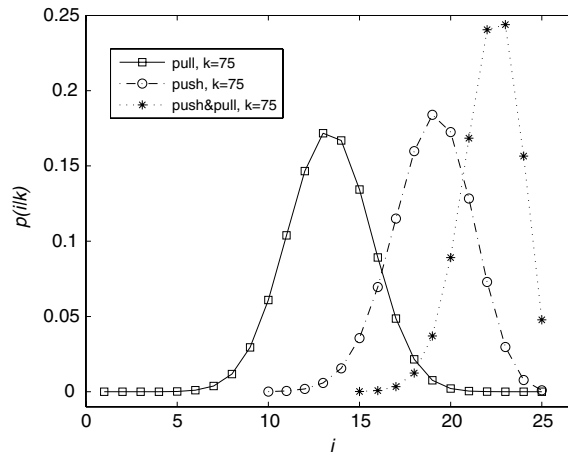


Fig. 4. The probability distributions of i more nodes getting infected, for $k = 75$ starting number of infectious peers and $n = 100$.

the implicit assumption of such independence, [16,14] use a binomial distribution. However, if a particular susceptible peer is chosen as the gossip target by an infectious peer, this implies that any other susceptible peer has not been chosen by the same infectious peer. Therefore, the mutual independence of infection event of each susceptible from another is violated and the binomial distribution serves only as an approximation.

For the push approach, the probabilistic reasoning given in our prior work [16] turns out to be exact and is essentially the same binomial distribution as that derived above. The only difference is that the number of possible nodes among which an infectious node chooses to communicate has been rounded as n . We restate the arguments which yields the exact distribution. Assume that there are k infectious peers and $n - k$ susceptible peers at a given moment. In the push case, recall that a susceptible member's selecting an infectious peer is sufficient for infection. Fix one of the susceptible nodes arbitrarily. The probability that this susceptible member selects an infectious peer is given by

$$p = \frac{k}{n-1} \quad (2)$$

since the susceptible member gossips to one of the $n - 1$ members due to $f = 1$ and only k of them are infectious. Since all susceptible members send gossip messages independently from each other and in an identical manner, the number of new infections follows a binomial distribution with parameters $n - k$ and p by definition. Using the probability of success in (2), we get

$$P_{kj} = \binom{n-k}{j-k} p^{j-k} (1-p)^{n-j} \quad (3)$$

as given in Eq. (1) with $i = j - k$ new infections.

For the pull approach, the approximate distributions used in [10,14] are essentially the same as the binomial distribution suggested in [16], since they are all constructed with similar arguments. The only difference is that the latter does not take link failures into account for the sake of simplicity. Using these approximate distributions rather than the exact distribution derived above has the advantage of simplicity and tractability of computation for large n . However, they are not accurate and the difference from the exact distribution cannot be quantified analytically. In Fig. 5, we compare the approximate distribution given in [16] with the exact distribution for the intermediate starting value $k = 50$ and $n = 100$. Both distributions are close to a normal distribution with almost the same mean, but the standard deviation of the approximate distribution is larger. It is more important to quantify the impact of the error of approximation on performance measures such as mean delay and dissemination time. This will be discussed in the next section. As for the push&pull approach, no approximate distribution has been suggested previously. It would be good to have at least asymptotical approximations for the pull and push&pull approaches for large n , and hence express the error of approximation analytically.

Note that our exact expressions supersede the earlier approximate results with no further simplifying assumptions. Features such as network delays, packet losses due to congestion and user heterogeneity are essentially omitted in order to obtain closed-form expressions in the other analytical studies of anti-entropy mechanisms as well [14,22,20,23]. On the other hand, it is possible to insert a packet drop probability which is identically the same in the network as in [10,14] without taking heterogeneity into account. As another aspect of real systems, the peers could have partial views rather than complete membership knowledge. However, a thorough incorporation of this feature into a Markov chain model would make it intractable [14]. In view of these, we construct the underlying Markov chain for epidemic diffusion to obtain the relevant performance measures in the following sections.

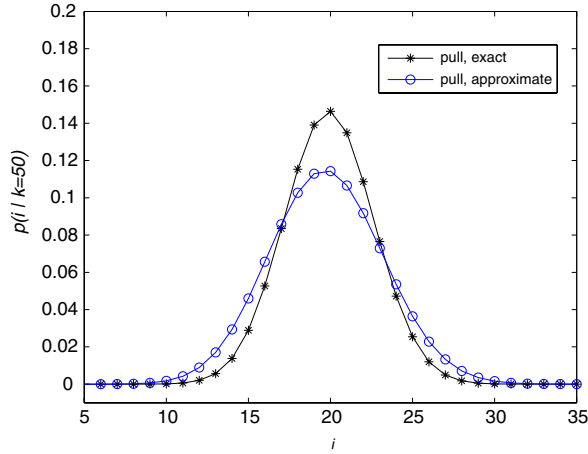


Fig. 5. The exact and approximate probability distributions of i more nodes getting infected, for $k = 50$ and $n = 100$.

5. Delay analysis

Many stochastic models of epidemic processes are based on the fact that the number of infectious peers, or equivalently the number of susceptibles when the population size n is fixed, forms a Markov chain [1]. In existing models, the transition probabilities are modeled according to a probability distribution or left as rates to be estimated from the network due to the complexity of the problem. What is accomplished in Section 3 is that we have analyzed the true dynamics taking place at each transition of the Markov chain with no assumptions on any parameters. Using the analytical expressions derived for the transition probabilities, we find the message delay in this section. An important performance measure is the mean delay per user from the user perspective. On the other hand, the total latency for dissemination to all group members gives an overall measure for the system. In this section, we study both quantities and compare the results with the previous asymptotic results.

5.1. Mean delay

The Markov chain under consideration is $\{I_t : t = 0, 1, 2, \dots\}$, where I_t denotes the number of infectious processes at time t . The transition probabilities $P_{kj} = P\{I_{t+1} = j | I_t = k\}$ can be obtained from $p(i|k)$ given in the previous section by

$$P_{kj} = p(j - k | k) \quad j = k, k + 1, \dots, n$$

where $j - k$ is the number of newly infected peers. Clearly, $P_{kj} = 0$ if $j < k$.

The delay experienced by each peer can be found by considering the first passage time of I_t to a specific set of states [16]. Let d_{ij} denote the expected value of the first passage time from state i to the set of states $\bar{j} = \{j, j + 1, \dots, n\}$, for $i = 1, 2, \dots, j - 1$. If the Markov chain enters the set \bar{j} by taking a value m which is strictly greater than j , there will be at least j infectious peers in the system and the j th infection will occur only at the time of transition to m . Therefore, we can interpret d_{ij} as the expected time for the j th infection to occur. In other words, it is the mean delay that a member, who is in the j th position to receive the message, experiences. At any instant, there is a positive probability that the realization of the delay may be the same as the m th member experiences for some $m > j$, in view of the argument above and due to the discreteness of time in the epidemic model. However, the delays of the j th and m th members will be different on average.

For each j , we form a set of equations to solve for d_{ij} using a one-step analysis of the Markov chain. Recall that the transition matrix P is upper triangular. For $j = 2$,

$$d_{1\bar{2}} = 1 + P_{11}d_{1\bar{2}}$$

as the chain has to make at least one transition, equivalent to one gossip round to enter the states $\{2, 3, \dots\}$. If it remains in state 1, which occurs with probability P_{11} , then the process restarts itself and has to wait $d_{1\bar{2}}$ amount of time again on average. As a result, $d_{1\bar{2}} = 1/(1 - P_{11})$, the mean of a geometric random variable, as expected. Similarly,

$$\begin{aligned} d_{1\bar{3}} &= 1 + P_{11}d_{1\bar{3}} + P_{12}d_{2\bar{3}} \\ d_{2\bar{3}} &= 1 + P_{22}d_{2\bar{3}}. \end{aligned}$$

Solving these equations yields both $d_{1\bar{3}}$ and $d_{2\bar{3}}$. In general, for $j \geq 2$,

$$d_{i\bar{j}} = 1 + \sum_{k=i}^{j-1} P_{ik}d_{k\bar{j}} \quad i = 1, 2, \dots, j - 1$$

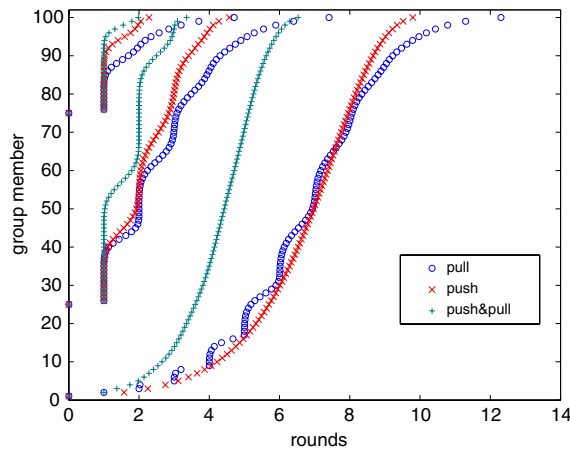


Fig. 6. Peers ordered according to their expected delays given in rounds, for the starting number of infectious peers being 1, 25 and 75, and $n = 100$.

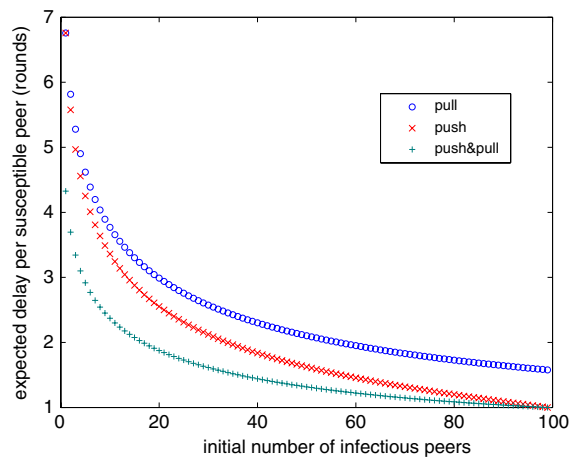


Fig. 7. Mean delay per susceptible versus initial number of infectious peers for $n = 100$.

which is equivalent to the system

$$(I - P_j)D_j = \mathbf{1} \quad (4)$$

where P_j is the upper left $(j-1) \times (j-1)$ portion of matrix P , I is the $(j-1) \times (j-1)$ identity matrix, $D_j = [d_{1j}, d_{2j}, \dots, d_{j-1,j}]^T$ and $\mathbf{1}$ is a vector of 1's. The k th row of the solution matrix D provides information on the mean delay experienced by the peers when the initial number of infectious peers I_0 is k . In order to find d_{kj} even for a single k value, one needs to solve the complete set of Eq. (4). Since P is upper triangular, the system can be solved very efficiently.

The delay experienced by each peer is an important performance measure from the user perspective. In Fig. 6, the peers in the order they receive the message are plotted against the expected number of rounds for different starting number k of infectious peers, for $k = 1, 25, 75$ and $n = 100$. That is, d_{kj} appears in the x -coordinate for $j = k, k+1, \dots, n$ and $I_0 = k$. The push&pull approach performs significantly better than the others. Although the push approach is only slightly better than the pull case in terms of mean delay when $k = 1$, its total time to disseminate to the whole population is much lower. The delay is clearly lower for the push case when $k > 1$. On the other hand, some peers have lower expected delay in the course of the information diffusion process when $k = 1$ such as the 10th to 15th peers in the order of receiving the message. In the pull approach, some infections are expected in bursts although the curves are very close for both the pull and the push cases.

The mean delay experienced per susceptible peer is depicted in Fig. 7. In terms of this performance measure, the pull and push approaches behave similarly for small k , and the push&pull approach behaves like the push case as k increases to n .

The advantage of epidemic dissemination, the anti-entropy paradigm in particular, is its scalability with respect to the population size. In order to demonstrate scalability, we have tabulated the expected time of dissemination to the whole population and the mean delay (in number of epidemic rounds) per peer in Table 1. These values are consistent with the prediction of asymptotical results for epidemic processes that the delay values increase only logarithmically as n increases [19]. The computation of formulas for the pull and push&pull approaches poses a precision problem. For instance,

Table 1
Expected time to dissemination and mean delay (in rounds) when $I_0 = 1$.

	Time to dissemination		Mean delay per peer	
	$n = 100$	$n = 200$	$n = 100$	$n = 200$
Pull	12.30	14.05	6.76	7.75
Push	9.79	11.03	6.75	7.75
Push&pull	6.53	7.40	4.33	4.96

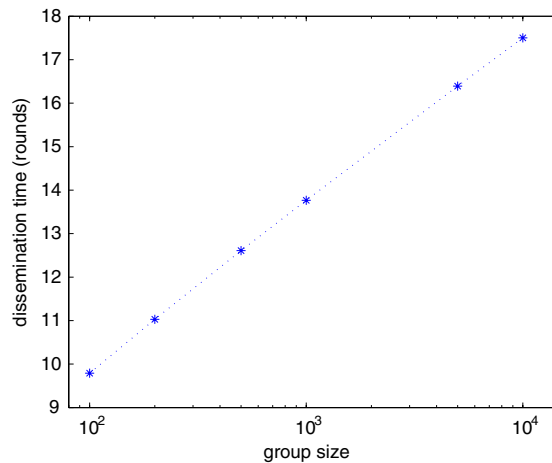


Fig. 8. Dissemination time versus group size starting with $I_0 = 1$ in the push approach.

the transition probabilities for $n = 200$ in the pull case have been computed using arbitrary precision arithmetic. A similar solution is necessary for the push&pull approach with larger network sizes. As a result, the computation time is prolonged. This problem is also reported in [20] for a (specified) occupancy distribution that is very similar to our pull distribution due to the similarity of the respective data diffusion mechanisms. In contrast, the push approach is tractable computationally as well as being more efficient than the pull approach. The MATLAB statistics toolbox can efficiently compute binomial probabilities for such large group sizes as 10,000. It turns out that the binomial probabilities emerging for the push case are a useful model of epidemic dissemination. See [1,2] for binomial models. We demonstrate the scalability of dissemination time through the push model numerically in Fig. 8, where the dissemination time indeed increases as the logarithm of the group size.

Although the binomial distribution is only approximate for the pull approach, it is computationally tractable for large n . Therefore, it is important to analyze the impact of the approximations of [10,16,14] on delay. The delays are illustrated through the expected number of infected processes for each round in [10,14], in accordance with the Markov chain analysis of [16] and the present work. The approximate results can be compared with those in Fig. 6 for the pull case. For $f = 1$, the results given in [16] for $n = 100$ differs from the exact result of Fig. 6. The total dissemination time is found as 14 rounds in [16], only 2 rounds more than the exact value, which is about 12. The approximate delay can serve as a conservative estimate in this case. On the other hand, the delay result of [14] is based on fan-out 3 for the pull case. There are no exact results to compare with fan-out values greater than 1. Since the analytical delay curve in [14] matches well with the simulations imitating the analytical model in the same study, it appears to be a good approximation.

5.2. Distribution of dissemination time

In addition to the mean delay analysis, we can find the distribution of the dissemination time exactly. The dissemination time T is the time to absorption to 0 of the Markov chain I_t , that is when the number of infectious processes becomes n , or equivalently the number of susceptibles become 0. Considering the transition of the Markov chain, one can easily verify the following expressions about the distribution of T [25, pg. 46]. Denoting $P\{T = t\}$ by $f_T(t)$, the cumulative distribution function by $F(t)$ and the expectation by E , we get

$$\begin{aligned}
 f_T(t) &= \alpha M^{t-1} v \quad t = 1, 2, \dots \\
 F(t) &= 1 - \alpha M^t \mathbf{1} \quad t = 0, 1, \dots \\
 E[T] &= \alpha (I - M)^{-1} \mathbf{1}
 \end{aligned}$$

where α is the row vector of the probability that the chain starts at transient states $1, \dots, n - 1$, again $\mathbf{1}$ is a vector of 1's, M is the transient portion of the transition matrix P and v is the column vector of one-step transition probabilities from the

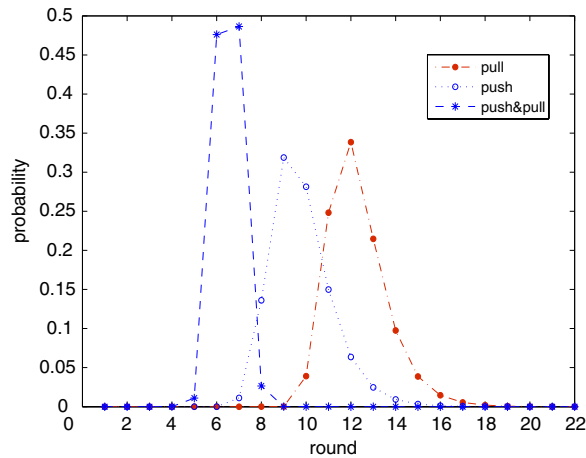


Fig. 9. The distribution of dissemination time for $n = 100$.

transient states to the absorbing state n as depicted in

$$P = \begin{bmatrix} M & v \\ 0 & 1 \end{bmatrix}.$$

The probability mass functions f_T for the three approaches with $n = 100$ and starting with $I_0 = 1$ are given in Fig. 9. The distributions are clearly discrete as the unit of time is a round, but the probability points are connected for the aim of distinguishing the distribution of various approaches visually. Clearly, the expected dissemination times coincide with the number of rounds given in Fig. 6 corresponding to the last group member, namely 100.

Finding the distribution of the dissemination time T is especially useful for computing the reliability of dissemination. For instance, in real-time applications, where there is a deadline of delivery, the probability of meeting this deadline can be considered as a factor affecting reliability. From Fig. 9, we see that the dissemination occurs in at most 7 rounds with more than 90% probability with the push&pull approach, and similar conclusions can be drawn for the other approaches. As another example, the probability of the dissemination time being less than the time of discarding a message from all buffers of the peers can be interpreted as the reliability of the system when each peer has a limited buffer space for keeping previously received messages for retransmission.

5.3. Comparison with asymptotic results

The delay of push and pull algorithms have been investigated asymptotically in the context of transmission of updates to distributed copies of a database in [22] where the terminology of push and pull follows the descriptions in [3]. Recall that the differences from the present paper are stated at the end of Section 2. However, the asymptotic analysis is also valid for our anti-entropy approaches, as the data dissemination occurs in the same fashion once the terms are correctly matched. Although network latency is not modeled explicitly, both asymptotic and exact results are useful for comparing the different cases intrinsically for the same network structure.

As explained in [22], the number of infectious peers grows exponentially until about $n/2$ peers are informed in the pull case (push case of [22]). At about this time, the exponential growth stops and the remaining diffusion takes an additional $\Theta(\ln n)$ rounds, during which the number of susceptible peers decreases by a constant factor approximately equal to $1 - 1/e$. On the other hand, in the push scheme (pull case of [22]) when the rumor starts with a single peer, that is when $k = 1$, it may take some time before this peer is called for the first time by a susceptible peer. That is why the initial phase of the push case is expected to proceed slowly for $O(\ln n)$ rounds with high probability until about $n/2$ peers are infected. After this point, the push algorithm takes over the pull algorithm as the fraction of susceptible peers roughly squares from round to round. The so-called shrinking phase is expected to take an additional $\Theta(\ln \ln n)$ rounds.

Our delay results given in Fig. 6 confirm these predictions as $n = 100$ is numerically large enough for evaluation of the asymptotic expressions. After the data is disseminated to about half of the peers, in this case 60 or so, the remaining time for diffusion to the whole population is indeed about $\log 100$ rounds for the pull case and $\log \log 100$ for the push case. Before the data reaches about half of the population, the pull approach is expected to disseminate faster for most of the peers as predicted when the initial state I_0 is 1. In [22], it is furthermore proved for the push&pull approach that the data is disseminated to all peers in $\log_3 n + O(\log \log n)$ rounds for large n . This bound is not precisely distinguished from the bound $O(\log n) + \Theta(\log \log n)$, which is discussed above for the push approach. However, our analysis shows that the push&pull case performs significantly better than the mere push case, by about $2 \log \log n$ rounds in the exact results of Fig. 6 and Table 1. What is more, the present analysis allows more detailed performance evaluation. We observe that the

push approach is more efficient than the pull approach throughout the dissemination when it starts with more than a single peer holding the data, that is, $k \gg 1$. We have been able to compute the mean delay per peer for various starting states to measure this efficiency, whereas the asymptotic analysis only suggests that the delays before $n/2$ and after $n/2$ may compensate each other for $k = 1$.

Spreading of multiple messages is investigated in [23] as in file sharing applications. The total number of pieces to be disseminated explicitly appears in the analytical expressions derived asymptotically in n . The pull and push convention is as in the classical random gossip literature [3,22]. The main focus of this study is on the advantage of file splitting with different approaches if any, while consistent results with [22] are obtained for no splitting. The fact that the pull approach (push in [23]) is slow in the final stages of dissemination is confirmed through the probability bounds derived. A hybrid algorithm is devised to do pull and push alternating in even and odd rounds slightly different from our push&pull approach, which allows bidirectional communication in the same round. Better performance with file splitting is guaranteed in this case with complete dissemination in at most a multiple of “log n +number of pieces” time.

6. Overhead analysis for the push approach

When the fan-out is 1 as in the previous sections, there are no duplicate data messages sent to an infectious node. In the push&pull case, we neglect the fact that a member can request the data message through the pull approach before another member sends the same message through the push mechanism. Although there may be duplicates in this case, we may assume that in such a tie situation the member gets the message only by the push approach and does not request it from others. A streamlined push&pull type protocol has been devised in [23]. However, the duplicates are inevitable when the fan-out is greater than 1 for the push and push&pull approaches. Such extra messages or duplicates are called the overhead of an approach. For the pull approach, no extra data messages would be sent as the susceptible node will request from only one of the infectious peers it becomes aware of.

In this section, we study the overhead messages produced by the push approach for fan-out greater than 1. This is possible due to the simple form of the distribution function for the push case, namely a binomial distribution. The exact diffusion probability analysis becomes quite complicated for the push&pull approach for fan-out greater than 1. The results for the push case will serve as an upper bound for the push&pull case since not all infections occur due to the push mechanism; some are due to the pull approach with no duplicates.

6.1. Larger fan-out values

In the previous sections, we have analyzed the three anti-entropy approaches for the fan-out parameter f equal to 1. However, we can find the exact dissemination probability distribution for the push case even for larger fan-out values because it also turns out to be a binomial distribution, as follows. For fan-out $f = 2, 3, \dots$, a fixed susceptible peer gossips to f of the $n - 1$ members, where only k of them are infectious. There is a chance that the peer sends gossip messages only to the other $n - k - 1$ susceptible members, which may happen with probability

$$\frac{\binom{n-k-1}{f}}{\binom{n-1}{f}}$$

if $f < n - k$, that is $k < n - f$, in which case the peer does not get infected at this round. Therefore, the probability that a fixed susceptible member selects at least one infectious peer and hence gets infected can be found as

$$p_f = 1 - \frac{\binom{n-k-1}{f}}{\binom{n-1}{f}} \tag{5}$$

when $k < n - f$. Then, the probability that the Markov chain I_t makes a transition from k infectious peers to j infectious peers is again given by Eq. (3) with the probability of success of (5) when $k < n - f$ as all infectious peers get infected identically and independently. That is, the transition probabilities are

$$P_{kj} = \binom{n-k}{j-k} p_f^{j-k} (1 - p_f)^{n-j} \quad k = 1, \dots, n - f - 1.$$

On the other hand, if $f \geq n - k$, that is $k \geq n - f$, then the peer gets infected for sure. Therefore, the transition probabilities are $P_{kn} = 1$ and $P_{kj} = 0$ for $j = 1, \dots, n - 1$ if $k = n - f, \dots, n$.

We show the expected delays for different fan-out values in Fig. 10 as found from system (4). It implies qualitatively similar conclusions as those given in [14,16] based on an approximate analysis. That is, the delay decreases for all members as the fan-out increases. For clarity, we have shown results only up to $f = 7$; the larger fan-out values 8 to 10 show a similar behavior. Here, we use $n = 500$ for comparison with the simulations given below for the same group size. The decrease in delay practically stops at larger values of f , as is also evident from Fig. 11, which shows the dissemination time to the

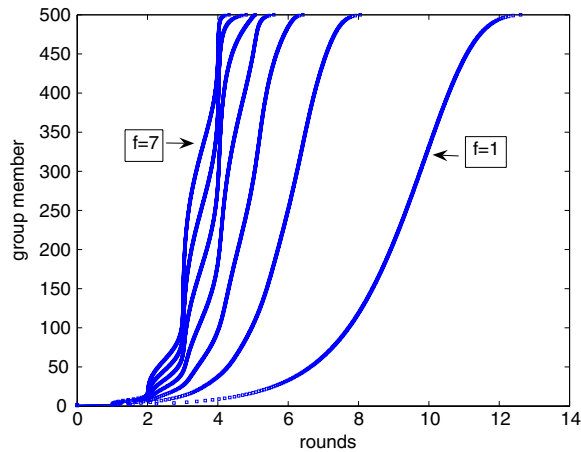


Fig. 10. Expected delays versus fan-out $f = 1, \dots, 7$ from right to left, starting with 1 infectious peer for $n = 500$.

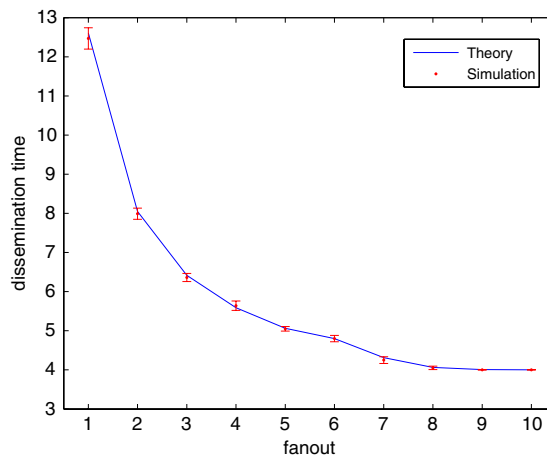


Fig. 11. Expected dissemination time in rounds versus fan-out starting with 1 infectious peer for $n = 500$. Error bars denote 95% confidence intervals for the simulation.

whole group. In this figure, simulation results with 100 independent replications are also depicted with approximate 95% confidence intervals, which are found by $\bar{x} \pm 2s/\sqrt{100}$, where \bar{x} and s denote the mean and the standard deviation of the dissemination times, respectively. In each simulation run, f peers are selected randomly. They are classified as susceptible and infectious dynamically, and the number of rounds until all peers become infectious is recorded. On the other hand, the overhead which we study in the next subsection is also expected to increase as the fan-out increases. In real information dissemination, this may cause the dissemination time to increase for larger values of fan-out. An optimal fan-out value could exist in this case which would minimize the delay.

6.2. Expected overhead

For the push algorithm in the case of fan-out greater than 1, a fixed susceptible peer may receive the same data message from multiple infectious peers due to the fact that these infectious peers have been selected as gossip targets by the susceptible one in the same epidemic round. Such extra messages cause an overhead associated with the push approach. We first find the expected value of the extra data messages, which will simply be discarded by the receiving susceptible peer. Then, we can find the expected overhead messages for all susceptible peers due to their identical behavior. Note that the overhead of communication investigated in [22] refers to a different quantity. It is the total number of replicas of the data message generated until complete dissemination in comparison to a deterministic scheme which would require only $n - 1$ transmissions from the sender to all other $n - 1$ peers.

Consider a fixed susceptible peer which gets infected through the push mechanism. We will study the cases $k < n - f$ and $k \geq n - f$ separately as the infection probability differs. Given that the susceptible peer got infected, this could have been achieved by a single data message sent to it or multiple messages including the duplicates sent by several other infectious peers, at most f in number. The conditional probability of the fixed susceptible peer getting infected by receiving exactly m

messages (given that it is one of the susceptible peers which gets infected at this round) can be found for $k < n - f$ and $m = 1, \dots, \min(f, k)$ as

$$p_m = \binom{k}{m} \binom{n-k-1}{f-m} / \sum_{l=1}^{\min(f,k)} \binom{k}{l} \binom{n-k-1}{f-l} \quad (6)$$

where the numerator of (6) denotes the number of possible ways to select exactly m from the infectious members and $f - m$ from the susceptible processes, and the denominator contains all possible combinations for getting infected. The probability (6) is also that of having $m - 1$ duplicates. Therefore, the expected number of duplicates for our fixed susceptible is

$$\mu_k = \sum_{m=1}^{\min(f,k)} (m-1)p_m$$

which is identically the same for all other susceptible peers. As a result, if the Markov chain makes a transition from k to j infectious members, the expected overhead at this transition is given by

$$S_{kj} = (j - k)\mu_k \quad \text{if } k = 1, \dots, n - f - 1$$

for $j = k, k + 1, \dots, n$. On the other hand, when the fan-out is larger than the number of susceptible peers, namely $k \geq n - f$, the susceptible peer gets infected for sure, but with an overhead of

$$\alpha_k = \sum_{m=f-(n-k-1)}^{\min(f,k)} (m-1) \left[\binom{k}{m} \binom{n-k-1}{f-m} / \sum_{l=f-(n-k-1)}^{\min(f,k)} \binom{k}{l} \binom{n-k-1}{f-l} \right]$$

where we have used the fact that the number of infectious members that receive push messages must be at least $f - (n - k - 1)$ as the number of susceptible peers which gossip is $n - k - 1$, and the limits of the summation are consistent because $k \leq n - 1$ and $f \leq n - 1$. Then, the expected overhead when $k \geq n - f$ is given by

$$Q_{kn} = (n - k)\alpha_k \quad \text{if } k = n - f + 1, \dots, n$$

since all susceptible members get infected.

The total number of extra messages during the whole dissemination period will be computed through the first-step analysis of the Markov chain. Let v_k denote the total overhead incurring until the end of dissemination, namely the absorption instant of the Markov chain I_t to state n , when the initial number of infectious members is k . For $k = 1, \dots, n - f - 1$, we have

$$v_k = \sum_{j=k}^n P_{kj}(S_{kj} + v_j) \quad (7)$$

since v_j is the overhead incurring from state j to n , and S_{kj} is the incremental overhead when a transition occurs from j to k with probability P_{kj} . On the other hand, the overhead for $k = n - f, \dots, n$ is simply $v_k = Q_{kn}$ which also satisfies Eq. (7) with S_{kj} replaced with Q_{kj} since $v_n = 0$ and $P_{kn} = 1$ in this case. Therefore, system (7) can be solved to get all v_1, \dots, v_n . It is an upper triangular system given by

$$(I - P)v = M$$

where M is the vector defined by $M_k = \sum_{j=k}^n P_{kj}S_{kj}$ if $k = 1, \dots, n - f - 1$ and $M_k = P_{kn}Q_{kn}$ if $k = n - f, \dots, n$.

Fig. 12 shows the results for fan-out values 1 to 7 with $n = 500$. The computations can be accomplished easily for larger group sizes as well. However, the combinatorial expression in (6) requires a slight increase of precision during calculations for fan-out 7 and larger. The simulation results verify these calculations, as shown in Fig. 12 [26]. The error bars denote approximate 95% confidence intervals, which are found by $\bar{x} \pm 2s/\sqrt{100}$, where \bar{x} and s denote the mean and the standard deviation of the number of duplicates found in 100 independent simulation runs, respectively. In each run, the random selection of peers is simulated and the total number of duplicates is counted until all peers become infectious. The curves get closer for fan-out values 8 to 10, which are not shown here for better visualization purposes. The dependence of the overhead on the fan-out can be also inferred from Fig. 12. The total overhead increases with fan-out, as expected. The case of $k = 1$ is scrutinized in Fig. 13 up to $f = 10$. The increase is quite steady until $f = 8$ and becomes slower after that, which is clearly a threshold depending on the network size $n = 500$.

The oscillating behavior of the curves, which is emphasized more for larger fan-out values in Fig. 12, implies an intriguing relationship of the starting number of infectious peers with the overhead. Starting the dissemination with higher number of infectious peers helps the dissemination time to be lower, as shown in the previous section when delay due to overhead is neglected. However, Fig. 12 indicates that there exist certain states of the number of infectious peers that cause a high number of duplicates when started with. In real networks, this could cause a non-trivial dependence of the dissemination time on the initial number of infectious peers. Although unexpected, the oscillations in Fig. 12 can be explained as follows. Consider the case of only one infectious peer to start with. Even if all other susceptible peers do manage to gossip to the infectious one in the next round, no duplicates will be generated as only one data message can be sent to each susceptible

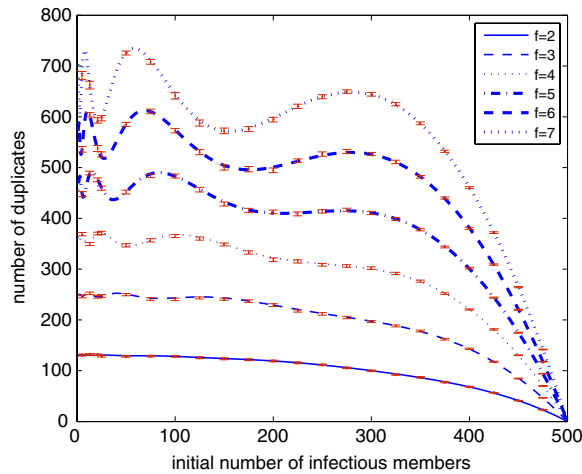


Fig. 12. Overhead versus $k = 1, \dots, 500$ initial number of infectious members for $n = 500$. Line graphs denote theoretical computations and error bars denote 95% confidence intervals from the simulation for various values of k .

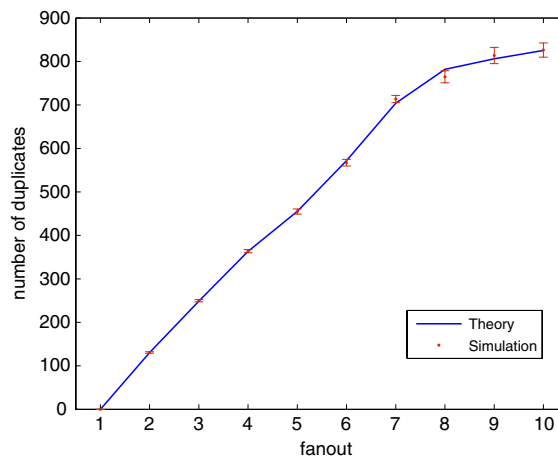


Fig. 13. Overhead versus fan-out starting with 1 infectious peer for $n = 500$. Error bars correspond to 95% confidence intervals for the simulation.

by the single source. When the number of infectious peers increases in the system, the probability of duplicates received by each fixed susceptible peer increases as well. However, the same cannot be said for the total number of duplicates since the number of susceptible peers which will receive these duplicates decreases at the same time. Therefore, the expected overhead even just at the next round is a non-trivial function of the current state i of the Markov chain illustrated in Fig. 14. It could be low in one state, and high in the other depending on n and the fan-out. What is more, the oscillating behavior can now be expected from the dynamics. The total number of duplicates accumulate as expressed in (7) in the expected sense, when the chain moves from one state to another. It is conceivable that the chain is likely to visit less overhead incurring states until absorption to n when it starts from some initial states and more overhead when it starts from the others. This is also due to the non-monotonicity of the transition probability distribution, which is binomial with parameters depending on each visited state i at each step. On the other hand, it may be argued that, starting with a state smaller than a high overhead incurring state such as about 300 in Fig. 12 for $f = 6$, the chain might eventually visit this high overhead incurring state. But this may be with such a low probability that the total number of duplicates turns out to be low, such as for those states about 200. Note that the nonlinearity observed is intrinsic to the epidemics and not the Markov model because it is a good abstraction of the true dynamics.

6.3. Comparison with ns-2 simulations

We have developed a simulation model for the push-based dissemination on the ns-2 network simulator [27], and performed an overhead analysis on large-scale hierarchical topologies. Our aim in conducting these simulations is to observe the overhead behavior through realistic network settings. Thus, the network simulation study serves as a qualitative comparison with the analytical results.

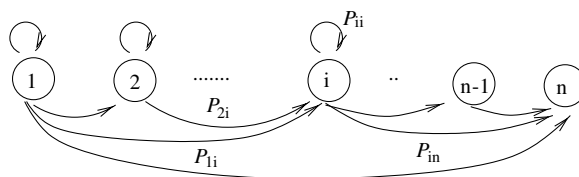


Fig. 14. Illustration of the Markov chain.

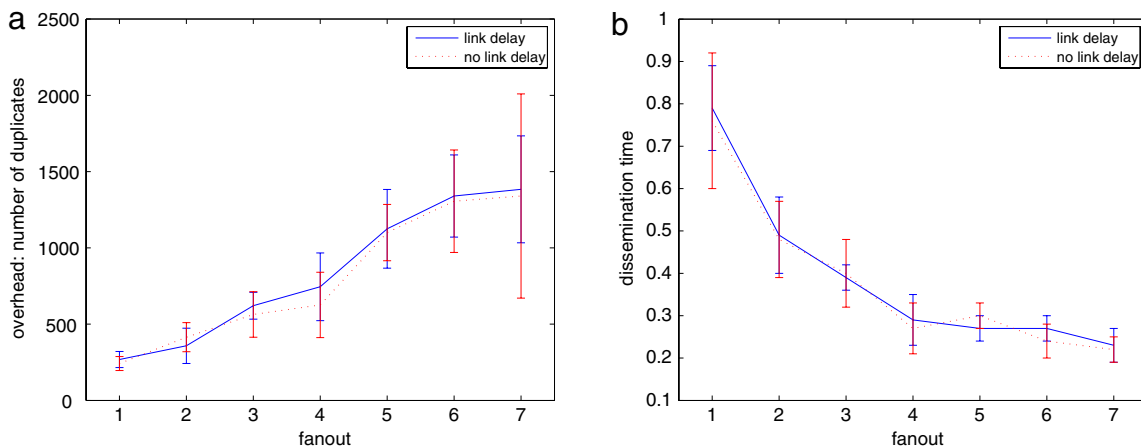


Fig. 15. (a) Overhead versus fan-out; error bars denote 68% confidence intervals. (b) Dissemination time (seconds) versus fan-out; error bars denote 95% confidence intervals. 1 round = 100 ms, $n = 500$ on 1500-node topology, starting with 1 infectious peer.

The simulations consider realistic network features such as link delays between peers and topologies. For the hierarchical topologies, we have used several randomly generated transit-stub graphs produced by the gt-itm generator [28]. These topologies consist of interconnected routing domains where each domain can be classified as either a stub or a transit domain. The network consists of 1500 nodes and the link delays are 2 ms. The number of peers is set to 500 and the node positions are chosen randomly according to uniform distribution on the network topology. Initially, there exists a single infectious peer in the system. The duration of the epidemic round is set to 100 ms. The gossip rounds are synchronous across the peers. A data message is disseminated to all peers in the system by means of the push approach. We have also repeated the simulations by setting the link delays to zero in order to approach the simplified settings of the analytical results. For each setting, average values and confidence intervals are reported over several independent runs with different seeds.

We examine the impact of increasing fan-out on the overall number of duplicate messages and dissemination time. Fig. 15(a) gives the results for the system-wide overhead in the form of duplicate messages transmitted, as a function of fan-out. The overhead increases as the fan-out increases, verifying our analytical findings reported in Fig. 13. Fig. 15(b) shows the dissemination time to the entire peer population as a function of fan-out. In our simulations, fully reliable data dissemination is achieved. Hence, the dissemination times in the graphs indicate the values when full data dissemination to all peers is completed. Consistent with the analytical results reported in Fig. 11, the dissemination time decreases as the fan-out increases. In Fig. 15, simulations with both link delays/topology effects and no link delays are reported. The results turn out to be close in both cases. Although we set the link delays to zero, the node processing delays cannot be removed in the underlying ns-2 simulation. Due to such delays, a susceptible peer can gossip more than once before the “first gossiped” infectious peer pushes the data message. By the same token, we have nonzero overhead for $f = 1$. The standard deviation is high among the simulation runs due to the variance introduced by the shifts from the delays. We have performed 10 replications and we report the 68% confidence intervals in Fig. 15(a) for better visualization. Since the standard deviation is lower in dissemination times, 95% confidence intervals are shown in Fig. 15(b).

Fig. 16 shows the results for overhead and dissemination time as the initial number of infectious peers varies between 1 and 450 when $f = 2$ and $n = 500$. As depicted in Fig. 16(a), there is an oscillatory behavior for the overhead for smaller values of the initial number of infectious peers. This phenomenon is confirmed by the analytical model results of Fig. 12. Likewise, as the number of initially infectious peers increases, lower dissemination times on average are observed, as given in Fig. 16(b).

7. Conclusions and future work

An analytical framework is developed to establish the exact probability distributions for the pull, push, and push&pull data dissemination models of anti-entropy. This study is the first one deriving exact distributions. In contrast, previous

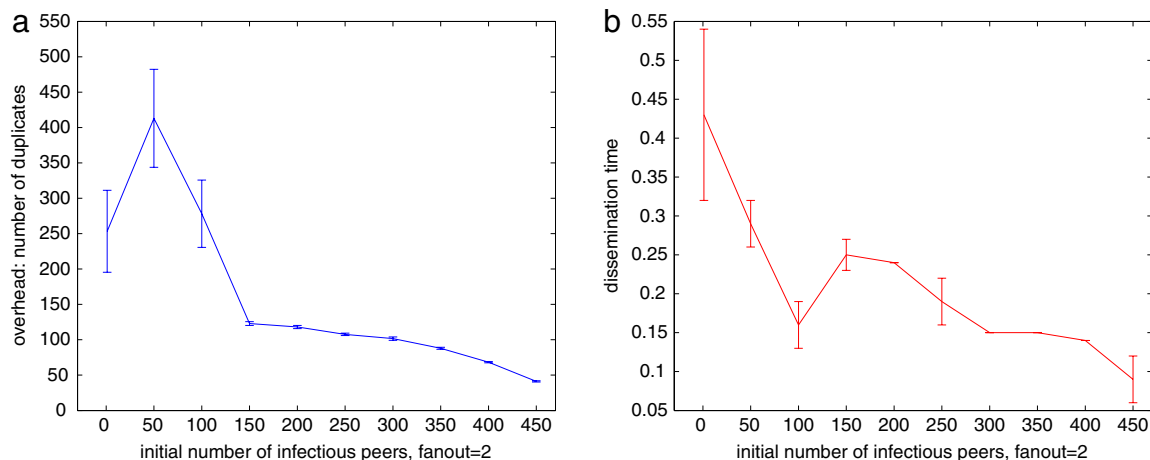


Fig. 16. (a) Overhead versus initial number of infectious peers; error bars denote 68% confidence intervals. (b) Dissemination time (seconds) versus initial number of infectious peers; error bars denote 95% confidence intervals. 1 round = 100 ms, for $n = 500$ on 1500-node topology.

studies rely on simplified models of epidemics usually requiring the estimation of several parameters. Our findings show that the binomial model used previously for the pull case is not accurate, whereas the model for the push case is exact. There exists no previous probability model for the push&pull case, the exact distribution of which is derived in this article.

The exact distributions of the anti-entropy models found at the first stage of the study are put into use through further performance analysis and comparison in terms of delay and overhead. The results are useful when integrating the diffusion models in distributed scenarios such as replicated servers, loss recovery, failure detection and group membership management. Therefore, it is important to differentiate the performance of the different paradigms. We have computed the expected delay of each peer as well as per arbitrary peer exactly, depending on the initial number of infectious members in the population. The push&pull approach outperforms both the pull and the push paradigms, and the push case is better than the pull case in terms of delay.

Analyses of larger fan-out values are accomplished for the push approach. Analytical expressions are derived for the transition probability distributions and the expected delays are computed. What is more, a thorough overhead analysis is given. We have found the expected numbers of duplicates received until the end of dissemination starting with an arbitrary number of infectious peers, analytically. These are verified with simulations which represent the random selection of peers and the push mechanism through MATLAB computations. The expected number of duplicates has a decreasing trend as the initial number of infectious members k increases, but this is not monotone for smaller k , especially in the case of larger of fan-out values. However, the overhead increases with fan-out, as expected, for a fixed number of initially infectious peers. The dissemination time decreases as the fan-out increases, also as expected. For comparison with more realistic network scenarios, we have run ns-2 simulations as well. The average behavior of overhead with fan-out and different starting number of infectious peers is confirmed.

We have considered dissemination of a single data message. As future work, characteristics with a bigger volume of content such as in file sharing applications could also be analyzed. We also plan to consider the partial membership knowledge among peers and information exchange based on proximity. The effect of network topology is another aspect to be included in our framework [29].

Acknowledgements

The authors would like to thank the editors and anonymous reviewers for their constructive comments that helped to improve the quality and the accuracy of this work. The first author's research was supported by TUBITAK (The Scientific and Technical Research Council of Turkey) under CAREER Award Grant 104E064. The fourth author's research was partly supported by a TUBA (Turkish Academy of Sciences) GEBIP Award Grant.

References

- [1] N.T.J. Bailey, *The Mathematical Theory of Infectious Diseases and its Applications*, Charles Griffin and Compan, London, 1975.
- [2] D.J. Daley, J. Gani, *Epidemic Modelling: An Introduction*, Cambridge University Press, UK, 2005.
- [3] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, D. Terry, Epidemic algorithms for replicated database maintenance, in: Proc. of the Sixth ACM Symp. on Principles of Distributed Computing, Vancouver, British Columbia, Canada, 1987, pp. 1–12.
- [4] Ö. Özkasap, E.Ş. Yazıcı, S. Küçükçifçi, M. Çağlar, Exact Performance Measures for Peer-to-Peer Epidemic Information Diffusion, in: Proc. ISCS 2006, Lecture Notes in Computer Science, vol. 4263, Springer, Berlin, 2006, pp. 866–876.
- [5] A. Birrell, R. Levin, R. Needham, M. Schroeder, Grapevine: Providing availability using lazy replication, *ACM Transactions on Computer Systems* 10 (1992) 360–391.

- [6] R. Golding, K. Taylor, Group Membership in the Epidemic Style, Technical Report, UCSC-CRL-92-13, University of California at Santa Cruz, 1992.
- [7] R. Ladin, B. Lishov, L. Shrira, S. Ghemawat, An exercise in distributed computing, Communications of the ACM 25 (1982) 260–274.
- [8] K. Guo, Scalable message stability detection protocols, Ph.D. dissertation, Cornell University, Dept. of Computer Science, 1998.
- [9] R. van Renesse, Y. Minsky, M. Hayden, A gossip-style failure detection service, in: Proceedings of Middleware 98, The Lake District, England, 1998, pp. 55–70.
- [10] K. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, Y. Minsky, Bimodal multicast, ACM Transactions on Computer Systems 17 (1999) 41–88.
- [11] R. van Renesse, K. Birman, W. Vogels, Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining, ACM Transactions on Computer Systems 21 (2003) 164–206.
- [12] P.T. Eugster, R. Guerraoui, A.-M. Kermarrec, L. Massoulie, Epidemic information dissemination in distributed systems, IEEE Computer (May) (2004) 60–67.
- [13] A.-M. Kermarrec, L. Massoulie, A.J. Ganesh, Probabilistic reliable dissemination in large-scale systems, IEEE Transactions on Parallel and Distributed Systems 14 (2003) 248–258.
- [14] P.T. Eugster, R. Guerraoui, S.B. Handurukande, P. Kouznetsov, A.-M. Kermarrec, Lightweight probabilistic broadcast, ACM Transactions on Computer Systems 21 (2003) 341–374.
- [15] M. Jelasity, S. Voulgaris, R. Guerraoui, A.-M. Kermarrec, M. van Steen, Gossip-based peer sampling, ACM Transactions on Computer Systems 25 (2007) 8.
- [16] M. Çağlar, Ö. Özkasap, A chain-binomial model for pull and push-based information diffusion, in: Proc. of IEEE ICC, Istanbul, Turkey, 2006.
- [17] I.B. Gertsbakh, Epidemic process on a random graph: Some preliminary results, Journal of Applied Probability 14 (1977) 427–438.
- [18] J. Jaworski, Epidemic processes on digraphs of random mappings, Journal of Applied Probability 36 (1999) 780–798.
- [19] Boris Pittel, On spreading a rumor, SIAM Journal on Applied Mathematics 47 (1987) 213–223.
- [20] J. Mundinger, R. Weber, G. Weiss, Optimal scheduling of peer-to-peer file dissemination, Journal of Scheduling 11 (2008) 105–120.
- [21] N.L. Johnson, S. Kotz, A.W. Kemp, Univariate Discrete Distributions, Wiley, New York, 1992.
- [22] R. Karp, C. Schindelhauer, S. Shenker, B. Vocking, Randomized rumor spreading, in: Proc. Foundations of Computer Science, 2000.
- [23] S. Sanghavi, B. Hajek, L. Massoulie, Gossiping with multiple messages, IEEE Transactions on Information Theory 53 (2007) 4640–4654.
- [24] J.H. van Lint, R.M. Wilson, A Course in Combinatorics, Cambridge University Press, 1992.
- [25] M.F. Neuts, Matrix-Geometric Solutions in Stochastic Models, Dover, New York, 1994.
- [26] MATLAB codes for simulations can be found under <http://home.ku.edu.tr/~mcaglar>.
- [27] S. Bajaj, L. Breslau, D. Estrin, et al., Improving simulation for network research, USC Computer Science Dept. Technical Report, 1999, pp. 99–702.
- [28] GT-ITM topology generator, <http://www.isi.edu/nsnam/ns/ns-topogen.html>.
- [29] L. Massoulie, A. Ganesh, D. Towsley, The effect of network topology on the spread of epidemics, in: Proc. of IEEE INFOCOM, Miami, 2005.



Öznur Özkasap 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 Koç University, which she joined in 2000. Her research interests include distributed computing systems, multicast protocols, peer-to-peer systems, bio-inspired distributed algorithms and computer networks.



Mine Çağlar received her B.S. and M.S. degrees in Industrial Engineering from Middle East Technical University and Bilkent University, respectively. She received a Ph.D. degree in Statistics and Operations Research from Princeton University in 1997. She worked as a post-doctoral research scientist at Bellcore in Morristown, in the Network Design and Traffic Research Group, during 1997–1998. She is an associate professor in the Department of Mathematics at Koç University, which she joined in 1999. Her current research interests include stochastic modeling in telecommunication networks.



Emine Şule Yazıcı received her B.S. degree from Boğaziçi University in 2000 and her Ph.D. degree from Auburn University in 2003 in Mathematics. She worked as a post-doctoral research scientist during 2003–2004 at Auburn University and 2004–2005 at the University of Queensland in Brisbane, Australia. She joined Koç University in 2005, where she now works as an assistant professor of mathematics. Her area of research is Combinatorics, with primary emphasis on Combinatorial Design Theory and Combinatorial Computing.



Selda Küçükçifçi received her B.S. and M.S. degrees in Mathematics from Boğaziçi University in 1995 and 1997, respectively. She received her Ph.D. degree in Mathematics from Auburn University in 2000. She worked as a post-doctoral research scientist during 2000–2001 at Auburn University and Università degli Studi di Catania. She joined Koç University in 2001, where she now works as an associate professor of mathematics. Her area of research is Combinatorics, with primary emphasis on Combinatorial Design Theory and Graph Theory.