Magnet: Practical Subscription Clustering for Internet-Scale Publish/Subscribe

Sarunas Girdzijauskas
Swedish Institute of Computer Science (SICS), Stockholm, Sweden
sarunas@sics.se

Gregory Chockler, Ymir Vigfusson, Yoav Tock, Roie Melamed
IBM Haifa Research Laboratory, Israel
{chockler,ymirv,tock,roiem}@il.ibm.com

ABSTRACT

An effective means for building Internet-scale distributed applications, and in particular those involving group-based information sharing, is to deploy peer-to-peer overlay networks. The key pre-requisite for supporting these types of applications on top of the overlays is efficient distribution of messages to multiple subscribers dispersed across numerous multicast groups.

In this paper, we introduce MAGNET: a peer-to-peer publish/subscribe system which achieves efficient message distribution by dynamically organizing peers with similar subscriptions into dissemination structures which preserve locality in the subscription space. MAGNET is able to significantly reduce the message propagation costs by taking advantage of subscription correlations present in many large-scale group-based applications.

We evaluate MAGNET by comparing its performance against a strawman pub/sub system which does not cluster similar subscriptions by simulation. We find that MAGNET outperforms the strawman by a substantial margin on clustered subscription workloads produced using both generative models and real application traces.

1. INTRODUCTION

The Internet of tomorrow must be poised to support applications that involve large collections of users engaged in group-based interactions and information sharing, including Internet TV (IPTV) [7, 23], collaborative editing [32], and massive multi-player games [18, 30]. These applications require a group communication substrate capable of dealing with immense numbers of users and multicast groups in a scalable fashion. DHT-based peer-to-peer substrates offer almost unlimited growth capacity and efficient routing functionality while incurring only a modest maintenance overhead at each participant [26, 24]. They are an attractive design choice to serve as a basis for a scalable multicast solution. However, for reasons of connectivity and load balancing, most existing DHTs support *name-independent* routing topologies in which the node placement is entirely determined by a uniform hash of its name, and hence independent of its geographical location, interest preferences, and other node-specific attributes.

To provide efficient overlay-based multicast routing, a prerequisite is that peers who share the same (or similar) interests are *well-clustered*, i.e., separated from each other by a small number of peers with different interests. Exploiting well-clustered interests may be accomplished by using the techniques underlying locality-aware DHTs [24, 35, 1] or metric embeddings [33]. These approaches, however, rely on various assumptions about the distribution of node subscriptions, and are insufficient for supporting a general purpose multicast system wherein the participant subscriptions are a priori unknown and may change over time.

In this paper, we introduce MAGNET, an efficient peer-to-peer multicast system that supports the publish/subscribe (pub/sub) API and exploits well-clustered topic interests. MAGNET requires an underlying DHT which allows node specific attributes (and their ordering) to be directly incorporated into the routing structure [5, 15, 14]. We used the OSCAR DHT [12, 13, 14] to dynamically cluster the nodes in the MAGNET overlay based on their subscription preferences. Our choice of OSCAR was motivated by its ability to construct topologies which are both provably small-world [16, 11], and have a low maintenance overhead. Nonetheless, we believe that our techniques are general enough to also produce good results on top of the other name-dependent DHT substrates, such as Mercury [5] and GosSkip [15].

At the core of MAGNET is a clustering algorithm that takes as an input the subscription of a node, and outputs the node location (or equivalently, the node identifier) on a logical ring, which is a part of the underlying OSCAR DHT. The goal is to ensure that the identifier values reflect similarity in the subscription space, that is, the nodes with similar subscriptions are assigned numerically closer identifiers than those with dissimilar ones. Specifically, we define a similarity metric sim over the set of all possible subscriptions S (that is, S contains all subsets of T, the set of all topics) such that for any two subscriptions s_1 and s_2 in S, $\text{sim}(s_1, s_2) = |s_1 \cap s_2|/|s_1 \cup s_2|$.

DEBS '10, July 12-15, 2010, Cambridge, UK.

Dynamic clustering. Note that since the subscription space can be arbitrarily large, and the input distribution of the node preferences is a priori unknown, the mapping from subscriptions to identifiers cannot be fixed in advance, but should instead be computed dynamically based on the preferences of the nodes already in the overlay. In MAGNET, this is accomplished through a distributed membership service, which, for each topic t, maintains a random sample of the current subscribers to t along with their interests. Subsequently, a peer p who subscribes to t will first query this membership service to determine which of t's current subscribers (as known to the membership service) has most similar interests to p using the distance metric defined above; p will then join the ring next to that subscriber. Our experiments showed that effective clustering is possible even if the size of the subscribers' sample maintained by the membership service is very small, thus can be maintained in a lazy fashion using a low-bandwidth background gossip protocol which has low impact on the overall system throughput.

Routing topology. Once a peer joins the logical ring, it is connected into a small-world routing topology maintained by the underlying OSCAR overlay. The set of the peer's outgoing connections is augmented with a few additional long-range pointers (or *fingers*) chosen so that the probability of connecting to a node is inversely proportional to the ring-hop distance to the node[1]. To estimate the locations of the long-range neighbors, OSCAR maintains a digest of the identifier distribution on the ring. This digest is maintained by periodically sampling the node population using a series of random walks. Note that the overhead of maintaining this digest is small since, as it was shown in [14], a logarithmic number of random walkers would suffice to reliably estimate the identifier distribution.

Message dissemination. In the final step of our construction, the underlying small-world routing structure is leveraged to create locality preserving distribution trees. As in [6], MAGNET maintains a *home location* for each topic determined by uniformly hashing the topic name. The home location for topic t serves as the root of the multicast tree used to distribute the messages posted on t (and also as a root of the spanning tree used to maintain the samples of the t's subscriber interests; see above). Unlike [6], the trees in MAGNET are created in a top-down fashion so that the paths from the root to each of the subscribers coincide with the point-to-point greedy routing paths from the root to those subscribers in the overlay. The actual tree construction algorithm does not necessitate contacting the topic's root on each subscribe request. Instead, each new subscriber joins the tree by following the routing path towards the topic's home location until a grafting point lying on the top-down routing path from the root to that node is found (or the topic's home location is reached). We will argue that the routing trees constructed in this way preserve locality in the overlay, and therefore, maintain the desired subscription clustering.

Although the techniques behind MAGNET were devised for topic-based pub/sub, they may be generalized to content-based pub/sub by extending the notion of similarity to a multi-dimensional attribute space. The method, however, is out of the scope of this paper.

Results on synthetic models. The improvement in propagation costs achieved by the MAGNET's clustering depends on the degree of similarity in the input node subscriptions. As we show in Section 4, the cost savings are most significant in subscription workloads that exhibit well-defined structural dependency among the individual subscriber interests. For example, subscriptions to IPTV channels have been shown to embody substantial correlation between users [23], which is intuitive considering that news and other content on local channels are of primary interest to users located within that geographical or administrative region.

Following the approach of Wong et al. [34], we generate structurally correlated workloads by grouping the topics into several categories (or modes) in either one or two dimensions, and then select subscriptions from one or several of those categories either deterministically or by using a power-law popularity distribution. Our findings show that on these workloads, MAGNET saves between 20% to 80% of message propagation costs to uninterested relays over a strawman peer-to-peer pub/sub implementation which does not cluster subscriptions.

We present a new HIERARCHICAL-TOPICS model to synthetically generate subscriptions that follow a hierarchical classification scheme. The idea is to first assign users with a home topic, then repeatedly pick some home topic with preference for higher popularity and select another topic with preference for "similar" topics according to a binary classification hierarchy. We then make the subscriber of the first topic join the second one. We find that MAGNET saves between 20% and 60% of the costs incurred by the strawman under this model.

Results on real-world subscription patterns. We also evaluate our system on subscription patterns that arise in large-scale collaborative applications. In particular, we used a trace of all edits of Wikipedia articles by registered users over a 6 year period to both directly evaluate MAGNET and to generate a model of real-world subscription patterns. Our experiments determined that the strawman implementation involved 77% more uninterested peers in message distribution than MAGNET.

In addition, all our experiments indicate that the MAGNET performance is *adaptive* to the degree of correlation in the input subscription, and is, in particular, never worse than that exhibited by the strawman.

2. PRELIMINARIES

We with definitions and notation that will be used throughout the paper. We then briefly describe OSCAR, our underlying DHT substrate, focusing on the properties that are relevant in context of MAGNET.

2.1 Definitions and Notation

We let $T = \{t_1, \ldots, t_m\}$ denote the set of all topics. We define a *similarity* metric, sim, to be the function mapping a pair of node subscriptions $s_1, s_2 \subseteq T$ to the normalized size of their intersection, with the range $[0, 1]$. Formally,

$$\mathrm{sim}(s_1, s_2) = \frac{|s_1 \cap s_2|}{|s_1 \cup s_2|}.$$

In some of our experiments, we will also consider a similar-

[1]Strictly speaking, the probability of creating a link from node u to node v is inversely proportional to the integral of the probability density function of peer identifier distribution between the identifiers of u and v in the identifier space. For more detailed analysis please refer to our prior work [11].

ity metric weighted by the topic transmission rate. Specifically, for λ_i being the transmission rate of topic t_i, $\mathrm{sim}(s_1, s_2)$ is defined to be

$$\mathrm{sim}(s_1, s_2) = \frac{\sum_{i:t_i \in s_1 \cap s_2} \lambda_i}{\sum_{i:t_i \in s_1 \cup s_2} \lambda_i}.$$

Identifiers. Each MAGNET node p is connected into two independent ring-based DHT structures: one, called the *control* DHT, for supporting the interest-based membership service and the topic's home location, and the other one, called the *skewed* DHT for clustering peers according to their interests. We will also use the terms "control" and "skewed" to refer to the underlying ring structures maintained by those DHTs. Consequently, p is assigned two identifiers, denoted $id(p)_c$ and $id(p)_s$, one for the control and the other one for the skewed DHTs respectively. The routing table of p, $RT(p)$, is the union of the control and skewed DHT routing tables $RT(p)_c$ and $RT(p)_s$. We write $succ(p)_c$ and $succ(p)_s$ to denote the p's successor on the control and skewed rings respectively. The set of topics p is subscribed to is referred to as the p's *subscription* (or *interest*), and denoted $p.sub$.

The skewed DHT connectivity is maintained by the OSCAR protocol described below; and the control DHT can be supported by either OSCAR itself or any of the existing name-independent DHTs, such as Chord and Pastry [26, 24].

2.2 The Underlying Small-world DHT

The nodes in OSCAR are organized into a logical ring structure augmented with additional long-range pointers, or fingers. As discussed in Section 1, the fingers are created based on the actual distribution of the node identifiers in the input, which can be arbitrarily skewed. To this end, OSCAR performs periodic sampling of the node population in order to estimate the current distribution, and re-wires the network accordingly.

Specifically, each OSCAR peer P_u executes the following protocol (see Figure 1).

- **a)** First, we simultaneously start a constant number of random walks (5 in Figure 1) to sample the node population. The median of the sample set p_1 is then used to estimate the median for the entire population from P_u's perspective.
- **b)** P_u then proceeds in the same fashion by sampling the sub-population occupying the range (P_u, p_1) to estimate its median p_2.
- **c)** Next, the range (P_u, p_2) is sampled to estimate the median p_3, and so on. Continuing in this fashion, P_u will eventually learn the approximate locations of $k = O(\log n)$ medians $p_1, p_2, \ldots, p_k$, which define k partitions $X_1, X_2, \ldots, X_k$ of exponentially decreasing size.
- **d)** P_u then selects between 1 and k fingers so that the i^{th} finger, $1 \leq i \leq k$, is selected by first choosing one of the partitions X_j, and then picking a peer within this partition.

Both selections are done uniformly at random. As we show elsewhere (see [14]), this protocol produces a small-world topology, which implies that each node is reachable from any other one in at most $O(\log^2(n)/k)$ hops by following a greedy routing procedure on the node identifiers.

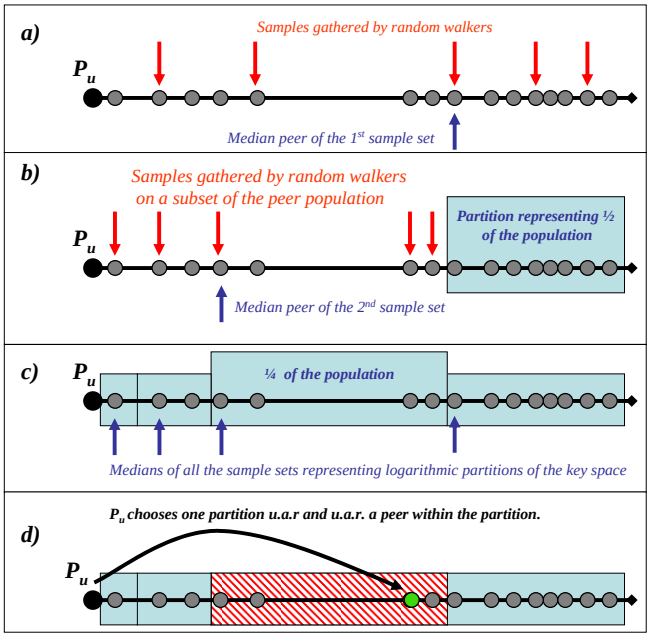


Figure 1: The Finger Selection Protocol in OSCAR.

Number of long-range links. The number of the fingers selected by a node is a parameter of the protocol, and can vary from node to node (but must be at least 1 at each node to maintain the small-world properties). Moreover, since the long-range neighbors are selected from a range of possibilities, there is an additional flexibility to incorporate other criteria into the selection process, such as e.g., the "power of two choices" [19]. In MAGNET, we utilize this property to bias the long-range neighbor selection towards the nodes with closer interests. Also, as we show in Section 4, the higher node degree is instrumental in improving connectivity among the subscribers who are interested in the same topics, and yet due to the imperfection of the clustering algorithm ended up residing in disjoint ring regions.

3. IMPLEMENTATION

The crux of MAGNET's implementation is the node join protocol which is executed every time a node subscribes to a new topic, or drops one of the existing topics from its subscription (provided, this is not the last topic it is subscribed to). The join protocol consists of three main steps:

- **a)** First, the node acquires an identifier on the skewed DHT (OSCAR) based on its subscription, and joins the skewed ring based on that identifier.
- **b)** The node then connects to additional long-range neighbors as prescribed by the underlying OSCAR DHT (see Section 2).
- **c)** Finally, the node joins the distribution tree for each topic to which it subscribes.

The core mechanisms in the MAGNET implementation are steps (**a**) and (**c**) which we describe in details in Sections 3.1–3.3.

Both of the identifier acquisition and distribution tree protocols rely on the new *interest-aware* membership service which is responsible for maintaining (possibly partial) views

of the interests of the nodes in the system. The interest-aware membership is a core part of MAGNET, and its implementation is described in Section 3.4. For the description of the clustering and tree construction protocols (which are presented first), we assume that each node p maintains a local state variable $view(p)$, populated by the membership service, which maintains the current mapping from the set of the node identifiers on the skewed ring $ID_s = \{id(q)_s\}$ to the set node subscriptions.

MAGNET's membership service implementation guarantees that for each topic t in the p's subscription, $view(p)$ includes the interest of at least one other node q which is also interested in t (unless p is the first to subscribe to t). This property is instrumental for improving both the clustering quality (since each node is guaranteed to see at least one node with a common subscription), and the performance of the tree join protocol (since the node can join t's tree through another node already in t's tree, instead of always going through the root). The details of the distributed maintenance protocol for $view(p)$ are given in Section 3.4.

3.1 Topic Home Locations

As in Scribe [6], each topic t's is associated with a home location, $home(t)$, which is determined by uniformly hashing the t's name, and looked up using a ring-based control DHT. The t's home location serves as the root of two spanning trees: one built over the skewed overlay, and used for disseminating the messages posted on t (see Section 3.3); and the other one built over the control overlay and used for maintaining partial subscription views of the nodes interested in t (see Section 3.4).

As we mentioned in Section 2, any of the popular ring-based DHT implementations (such as e.g., [26, 24, 35]) can serve as the MAGNET's control DHT, provided that it guarantees logarithmic routing latency under the assumption of the uniformly distributed node and object identifiers. Accordingly, the implementation details of the control DHT are omitted in the remainder of this paper.

3.2 Identifier Acquisition Protocol

The identifier acquisition protocol is depicted in Algorithm 1, and is the core part of the MAGNET clustering implementation. It is executed whenever a node p first joins MAGNET, and every time it changes its interest (that is, subscribes or unsubscribes to a topic[2]). Its goal is to ensure that the nodes with the close subscriptions (as indicated by the subscription similarity metric in Section 2) will be assigned identifiers which are numerically as close to each other as possible. Note that rejoining when interests change is only necessary to maintain the clustering and does not affect correctness of our system. Our system is designed to be flexible and adaptive, allowing each MAGNET node to decide locally on how often the change of its identifier is permitted depending on the node's load, available resources, and so forth. Thus, even when subscriptions are changing frequently, the nodes are free to remain stable and retain their existing locations in the identifier space. The connectivity of MAGNET guarantees that the system remains fully operational and assures that the performance is never worse than that of a pub/sub system built on a name-independent DHT. Consequently, the rejoin mechanism may be used sparingly in practice, for instance by requiring a minimum number of interest changes between rejoins.

[2]The worst case latency of rejoining the network upon subscription change is $O(\log N)$, and its communication complexity is at most $O(t \log N)$ where t is the node's subscription size, and N is the total number of the nodes.

The algorithm starts by inspecting $view(p)$ to discover a node q such that $q = \arg\max\{\text{sim}(p, q') : q' \in view(p)\}$, breaking ties randomly. Node p then joins the ring between q and the q's ring successor. If p fails to make contact with q (e.g., due to a failure), then q is excluded from $view(p)$ and the entire join algorithm is re-executed. If $view(p)$ is empty at the time p joined,then p will join the ring at a location determined by uniformly hashing its identifier.

Whenever $view(p)$ changes, p may attempt to improve its location by re-executing the join protocol. Note though that this is not strictly necessary since the other nodes will take into account the p's present interest (and location) when they join the ring, or change their subscriptions.

Algorithm 1 The identifier acquisition protocol for peer p: $id(p)_s = getLocation(p, view(p))$

```
1: if view(p) ≠ ∅ then
2:    find q, such that sim(p.sub, q.sub) =
      max{sim(p.sub, q'.sub) : (id(q')_s, q'.sub) ∈ view(p)}
3:    id(p)_s := mean(id(q)_s, id(succ(q)_s)_s)
4: else
5:    id(p)_s := hash(p.name)
6: end if
7: return id(p)_s
```

Observe that since the identifiers are chosen from a one-dimensional space, it is impossible to guarantee that the identifiers of the nodes with close subscriptions will always be sufficiently close numerically to be well-clustered. However, as we show in our experiments, one-dimensional clustering turns out to work quite well for a wide range of realistic subscription patterns. Extending the MAGNET techniques to better support multi-dimensional subscription correlation is the subject of future work.

3.3 The Tree Join Protocol

Once the node p's identifier on the skewed ring is fixed, and p is connected into the OSCAR overlay, the node will proceed to join the multicast tree for each topic to which it subscribes. In the following, we describe the steps taken by p to join the multicast tree $T(t)$ for one such topic t.

The tree join protocol consists of the three main phases (see Algorithm 2). At the first phase (lines 2.1–6), p consults $view(p)$ to find another node p_s interested in t which already belongs to the t's multicast tree. If no such node is found, then the tree's root, $home(t)$, will be used in its stead. During the next phase (2.7–11), p will traverse the t's tree upwards starting at p_s, until reaching a node p_g such that p_g is either $home(t)$, or has a finger q in its skewed DHT routing table such that q is the next hop on the greedy routing path from p_g to p, and (p_g, q) is already an edge of $T(t)$. The tree join protocol will then enter the final phase (2.12–14) at which the greedy routing path towards p will be followed until encountering a node r that has p in its finger table. At this point, p will join $T(t)$ as a child of r.

By induction over the node join events, it is easy to see that for each subscriber p of t, the path from $home(t)$ to p in the resulting tree $T(t)$ will coincide with a greedy routing path from $home(t)$ to p on the skewed DHT. Since each

Algorithm 2 Multicast tree join algorithm $joinTree(p, t)$

```
    ; PHASE 1:
 1: S := {q : (q, q.sub) ∈ view(p) ∧ p ∈ T(t)}
 2: if S ≠ ∅ then
 3:    p_s := q ∈ S, chosen uniformly at random
 4: else
 5:    p_s := home(t)
 6: end if
    ; PHASE 2:
 7: Traverse T(t) from p_s upwards until reaching p_g such
    that:
 8:    (1) p_g = home(t), ∨
 9:    (2) ∃id(q)_s ∈ RT(p_g)_s such that:
10:       (a) id(q)_s is the closest to p (from below) ∧
11:       (b) (p_g, q) ∈ T(t).edges
    ; PHASE 3:
12: Greedily route from p_g to p until reaching r such that:
13:    id(p)_s ∈ RT(r)_s
14: Join T(t) as a child of r
```

consecutive step of the greedy routing procedure exponentially decreases the distance to destination in the identifier space, the nodes with close identifiers will also be close to each other on the greedy routing path. We conclude that $T(t)$ preserves locality in the identifier space, and therefore also in the subscription space (to the extent it is maintained by the identifier acquisition protocol in Section 3.2).

Multicast tree maintenance. The tree structure is maintained by having each node to periodically ping its parent in the tree using a heartbeat message. The node's parent is declared to be disconnected if it fails to respond to a preconfigured number of heartbeats. At this point, the node issues a new $joinTree$ request which would reconnect the node to the closest available branch of the tree in terms of overlay hops. Such DHT-based tree maintenance is known to be robust (see e.g., Scribe [6], Bayeux [36].) When the network is stable, the messages are delivered to all their subscribers deterministically; whenever a failure occurs, the underlying small-world DHT is robust enough to allow the involved nodes to recover quickly, restore the overlay connectivity and heal the affected trees by repeating the tree join procedure and bypassing the failed node(s) by reconnecting through alternative paths.

3.4 Interest-Aware Membership

The MAGNET's membership service implementation maintains partial views of the node subscriptions, and is based on the randomized sampling over the interests of the entire node population in the system. Our currently implemented sampling strategy maintains a separate sample for the interests of the subscribers of each particular topic t, and propagates this sample to all the current subscribers of that topic. This ensures that each subscriber p of t knows of the interest of at least one other subscriber of t (unless p is the first node to subscribe to t), as required by the tree join protocol above.

Sampling protocol. For each topic t, the interest sampling protocol is implemented as follows. The subscribers of t are maintained in a spanning tree built over the edges of the control DHT, and rooted at $home(t)$. The tree is maintained dynamically, driven by the arrival of the new t's subscribers as well as the departure of the existing ones (due to either an explicit unsubscribe request, or a failure). The tree maintenance protocol is based on the same techniques as those of [6], and will not be discussed further here.

The sampling protocol executes in rounds, each of which is triggered by either a passage of time, or an explicit "round start" message multicast by $home(t)$. At each round, the node subscriptions are propagated layer-by-layer in the bottom-up fashion starting from the t's tree leaves, and ending up at $home(t)$. Upon receiving the subscription sample from its direct descendants, each inner node q will combine them with its own interest, and possibly, truncate the sample if it includes more than a configured number of the node interests k. The sample truncation is done by choosing k interests uniformly at random, and discarding all the rest. The sampling round terminates once $home(t)$ is reached, at which point, $home(t)$ will propagate the resulting view downstream to the t's subscribers.

Practical considerations. The scheme above guarantees that at steady state, every subscriber of t will have a consistent view of the interests of t's other subscribers. Also, as we show in Section 4, effective clustering is possible even if the number of the node interests in the sample is very small. The sampling collection can therefore be implemented efficiently, even under relatively high churn, provided the average size of the node interest is not too large. One approach to deal with large interests is to replace topic names in the propagated interests with their hashes. Another is to use hybrid sampling strategies combining Bloom filters and gossip-based sampling for large topics and tree-based sampling for small and medium ones. Further comparison of different approaches for maintaining partial interest views is the subject of our ongoing study.

4. PERFORMANCE EVALUATION

We implemented MAGNET in a simulated setting to evaluate the effect of clustering on the message propagation cost. Recall that the level of clustering is an artifact of the correlation between user interests in the input. We used several synthetic models of real-world user interest correlation, including one of our own, as well as a real-world trace to drive our experiments. We compare the cost of propagating messages over the MAGNET trees against that exhibited by a strawman implementation, in which the propagation trees are constructed directly on top of a name-independent DHT. In effect, our strawman implementation is expected to exhibit the behavior similar that of the Scribe [6] system. We measure the propagation cost as the number of uninterested relay nodes on the distribution trees. In this way, we also indirectly evaluate the reduction of bandwidth consumption in the system, which is a direct function of the number of relays. In what follows, we will explain each model and data set separately and evaluate MAGNET and the strawman implementation on each of them.

4.1 Overview of Models

Multi-Modal and Spatial. We first consider synthetic generative models for user interests inspired by Wong et al. [34]. They attempt to capture structural dependency among the subscriber interests as found in applications such as network games or news dissemination. We considered two models of this type, MULTI-MODAL and SPATIAL, which are described in details in Sections 4.2 and 4.3 respectively.

Hierarchical-Topics. In addition to the structurally correlated workloads, we also considered the subscription

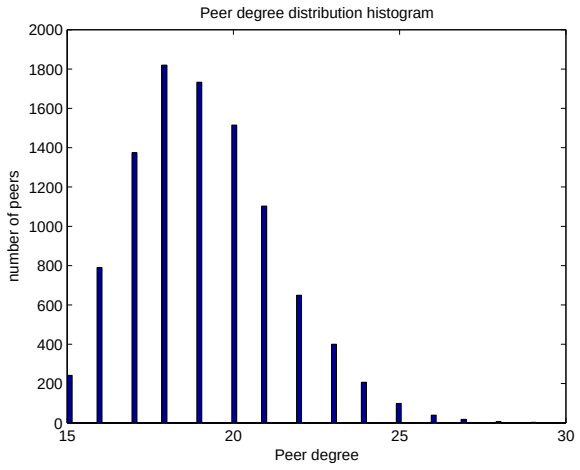


Figure 2: MULTI-MODAL model: Distribution of the number of categories (modes) chosen by peers (the average is 19).

patterns arising in large-scale collaborative applications, such as Wikipedia or Yahoo! Groups. Although the statistical structure of the user preferences in these applications is not yet well understood, empirical evidence suggests that the topic popularity distribution in these applications follows the power-law distribution[3] with the α parameter ranging between 2 and 3 [28, 20]. Unfortunately, the simple technique of populating topic subscriptions by iteratively selecting a random subset of users whose size is drawn from a power-law distribution fails to capture the more complex dynamics for group overlaps: human users tend to favor topics popular among other users with similar roles or interests.

In our HIERARCHICAL-TOPICS model, we instead make use of the *preferential attachment* model which is known to generate a random graph with a power-law degree distribution [20]. We augmented the basic preferential attachment model by embedding the topic space within a tree structure which models the hierarchical refinement of the interests. For example, the topics such as "Hardware Companies" and "Software and Services" are both refining a broader category, called "Technology Stock". The resulting model, which we call the HIERARCHICAL-TOPICS model, is described in more detail in Section 4.4.

Wikipedia: We obtained a trace of a real-world large-scale collaborative system, namely a trace of all edits of Wikipedia articles by registered users over a 6 year period [10]. We ran one experiment in which the MAGNET simulation was fed the subscription patterns extracted from the Wikipedia trace. In this experiment, we modeled topics as articles and user subscriptions as the set of the articles edited by that user. We describe the results of the WIKIPEDIA trace in Section 4.5.

4.2 Multi-Modal Model

In the MULTI-MODAL model [34], the topic space is partitioned into a fixed number b_n of categories (or modes). The peer subscription is generated by first choosing b_p categories out of b_n uniformly at random, and then selecting a topic from those categories following a power-law popularity distribution with parameter α. The subscription generation proceeds until the average peer has subscribed to a desired number of topics, which is the parameter of the model. The MULTI-MODAL model is a good match for applications such as news dissemination where user preferences are determined by their geographical or administrative location or both.

The degree of correlation among the peer interests can be adjusted by changing the b_p and b_n parameters while keeping the b_p/b_n ratio intact. In other words, the resulting topic frequencies will be the same, although the correlation between the peer subscriptions will be the highest when $b_p \to 1$ and the lowest (uncorrelated) when $b_p \to b_n$.

Default values. Unless stated otherwise, the workloads produced by the MULTI-MODAL model were used for the network, consisting of $10,000$ nodes, where each node had on average 19 links (overlay edges). The resulting peer degree distribution is shown in Figure 2. Each node subscribes to a random subset of 50 out of 1000 distinct topics. The power-law topic popularity parameter within each category is set to $\alpha = 1$ by default. Our algorithms used 10 samples for every topic, as described in Section 3.4.

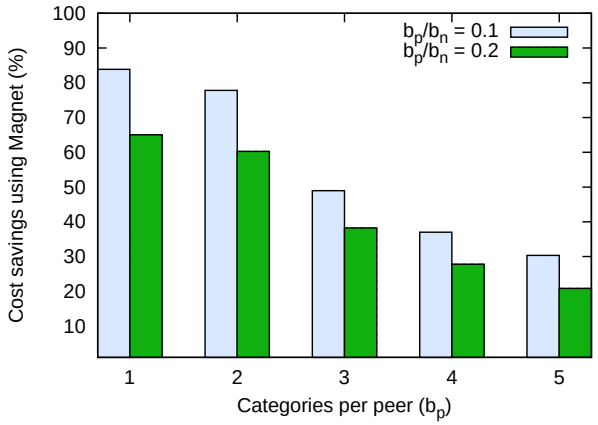
Publication rates. We also evaluated the MAGNET performance under various publication rates for each topic. In one experiment every topic was assigned a different publication rate, which was drawn from a power-law distribution with $\alpha = 0.75$, while in the other the publication rate was uniform. Figure 3 shows that MAGNET outperforms the strawman implementation under both publication rate scenarios even with low subscription correlations. As expected, MAGNET performed better for workloads with non-uniform publishing rate, because the rate is always taken into account by MAGNET's peer placement algorithms upon calculating similarity distance among the peers.

Varying correlations. We performed extensive simulations to verify whether MAGNET can exploit the subscription correlation generated by MULTI-MODAL model. We fixed the ratio of b_p/b_n to 0.1 and 0.2, and varied b_p and b_n from mostly correlated (every peer chooses one mode out of 10 and 5 modes respectively) to the least correlated (5 modes out of 50 and 25 respectively). In the reported experiments, every peer has been assigned 50 unique topics on average.

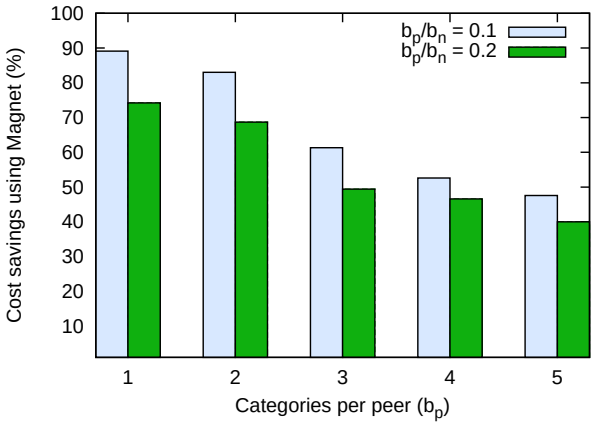
Figure 5 shows the fraction of messages sent via uninterested peers in MAGNET as compared to the strawman for the most correlated (Figure 5(a), $b_p = 1$, $b_n = 10$) and the least correlated case (Figure 5(b), $b_p = 5$, $b_n = 25$) with the power-law topic publication rate ($\alpha = 0.75$). Publishers send messages to topics in decreasing order of popularity with number of messages sent to each in accordance to the topic publication rate. The plots show the fraction of messages that were sent to uninterested peers for the first x topics over all messages sent to these topics. The graphs reveal almost no overhead for the most popular topics (the first 100 seconds), thus confirming that a good quality of clustering is achieved under those workloads.

Effects of topic popularity. We have also measured MAGNET's performance under different subscription sizes at each peer and the impact of different topic distribution scenarios. We vary the α parameter of the power-law topic popularity distribution, which directly influences the number of most popular topics in the system.

[3]In the power-law distribution (also called Pareto or Zipf), the fraction of topics of popularity x is roughly $\frac{1}{x^\alpha}$ for a constant value of α.

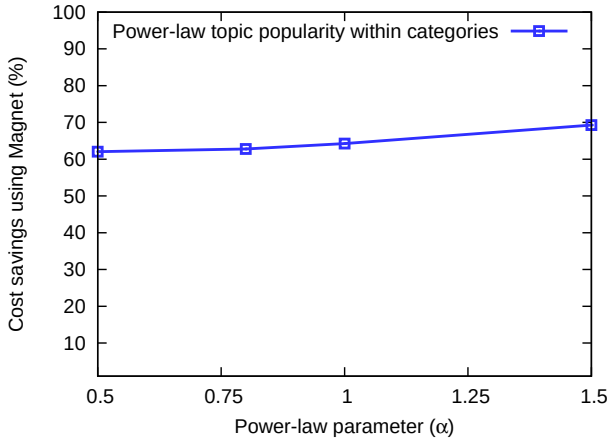


(a) Uniform topic publication rate.

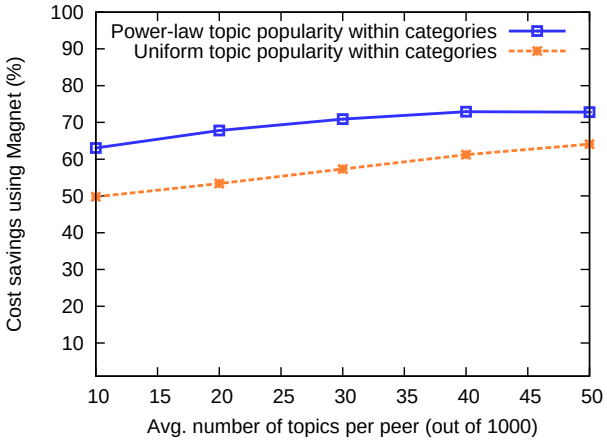


(b) Power-law topic publication rate.

Figure 3: MULTI-MODAL **model: Cost is measured by the number of uninterested relays who receive a message. The remaining model parameters are specified in Section 4.2. The ratio b_p/b_n for the number of categories joined by peer is fixed; higher values of b_p imply less correlation in the model.** MAGNET **adapts well to subscription correlations (left) and non-uniform publishing rates (right). In the former case,** MAGNET **benefits by clustering groups with similar membership; in the latter, it benefits from ensuring that the subscribers of the relatively few high-rate topics are close to the source.**



(a) Varying the power-law popularity parameter.



(b) Varying the per-peer subscription size.

Figure 4: MULTI-MODAL **model: System performance while varying topic popularity distributions (left) and the subscription size (right). The number of peers is 5000 and the average number of subscriptions per peer is 18. On the right, the power-law topic popularity parameter within each category is set to $\alpha = 0.75$.**

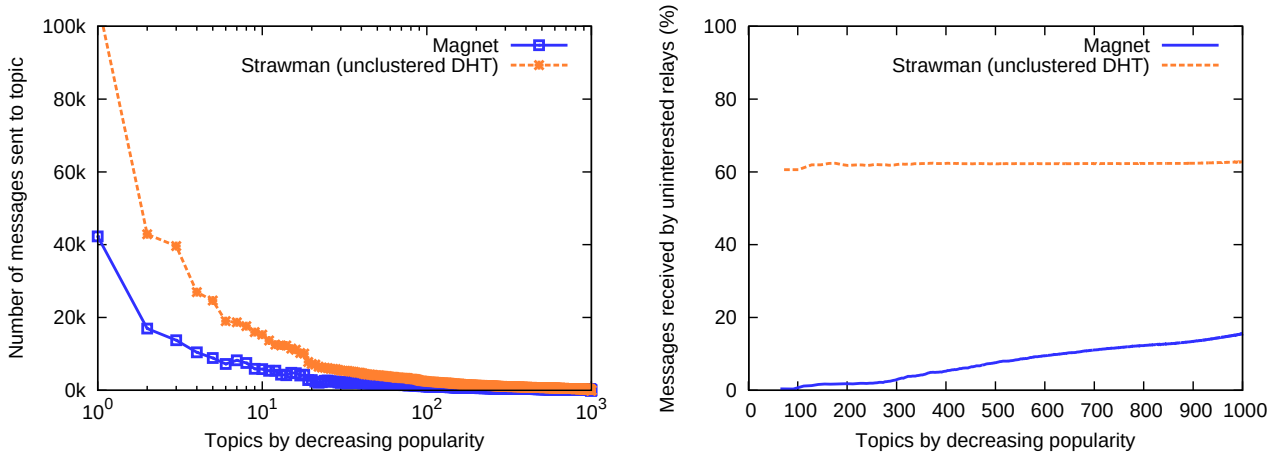
Figure 4(a) shows the performance of MAGNET given different α values with the MULTI-MODAL model ($b_p = 1$ and $b_n = 5$) and uniform publication rate. We see that MAGNET performs better with the higher α values since the steep power-law function pushes more peers to subscribe to the same few popular topics, thus increasing subscription correlation.

Varying subscription sizes. Figure 4(b) shows the performance of MAGNET with uniform publication rate as compared to the non-uniform one (power-law with $\alpha = 0.75$) while varying the peer subscription sizes. The results show that MAGNET's algorithms consistently outperform the unclustered DHT-based pub/sub system on all subscription sizes.

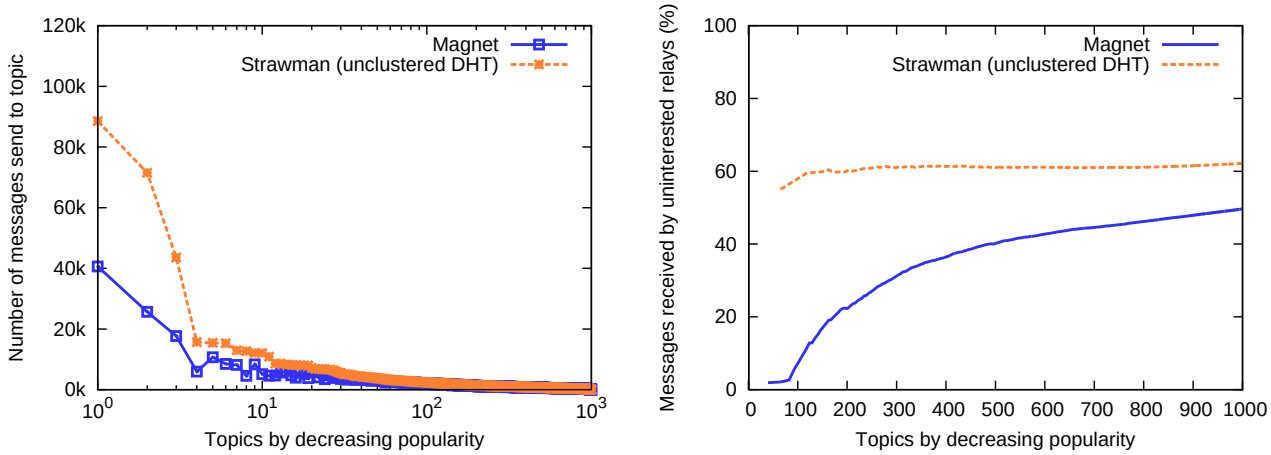
4.3 Spatial Model

In the SPATIAL model [34], the users are distributed uniformly at random on a unit square (1×1) and each user is associated with a single topic which is unique to that user. The subscriptions are generated so that each user is interested in the topics associated with the users located within radius r from its own location on the unit square. The spatial model is might predict subscriptions patterns that are typical in network games where the players would be most likely interacting with those located close to them on the virtual game space [30, 18].

Results. In our experiments, we varied the value of r so that on average every participant is interested in t topics, and experimented with the values of t being 8, 16 and 32.



(a) High correlation between peers ($b_p = 1, b_n = 10$).



(b) Low correlation between peers ($b_p = 5, b_n = 25$).

Figure 5: MULTI-MODAL model: Publishers send messages to every topic in decreasing order of popularity. The rate of traffic on the topics follows a power-law distribution, as shown on the left. The plots on the right show the CDF of cost (messages received by uninterested relays) normalized by the total number of messages sent to the first x topics. MAGNET is able to reduce cost by exploiting the correlation between user interests in the model compared to the strawman.

We measured the performance of MAGNET by publishing on all the topics in the system.

Figure 6 shows the relative decrease in message cost of MAGNET as compared to the strawman implementation with varying t values for different sizes of the network. We can observe that MAGNET is highly efficient and saves almost 80% of unwanted messages over the strawman implementation for very large networks (10,000 nodes) with large peer subscriptions (32 on average). The results are not surprising since the SPATIAL model produces highly correlated peer subscription patterns. The correlation increases as the number of topics each peer subscribes to grows, making spatially driven applications with many users (e.g., online network games) some of the most favorable environments for MAGNET's deployment.

4.4 Hierarchical Topics Model

The main ingredient in our HIERARCHICAL-TOPICS model is to embed the topics as leaves of a hierarchy such that nodes that are close together in the tree (have short tree distance) are more similar and should thus share more common users. The technique to populate the hierarchy is similar to Kleinberg's tree model for decentralized search [17].

As mentioned earlier, crafting a generative model for group subscription which displays the power-laws that have been observed in real-life social data sets is an open problem [28]. We will not attempt to solve the challenge here — instead we devise a model that leverages one of the most prominent models to generate a power-law degree distribution and is used to model the web hyperlink graph: the *preferential attachment* model [20]. In this model, nodes join the network one at a time and construct an edge to another node with probability proportional to that node's current degree.

Our model works as follows. The idea is to first bootstrap topics to be non-empty and follow a rough power-law distribution, giving users at least one "home" topic. We let parameter λ represent *homophily*, the tendency for peers to subscribe to topics which are similar to their existing interests. After the initialization, the peers in the popular topics then join other topics iteratively with a preference for those

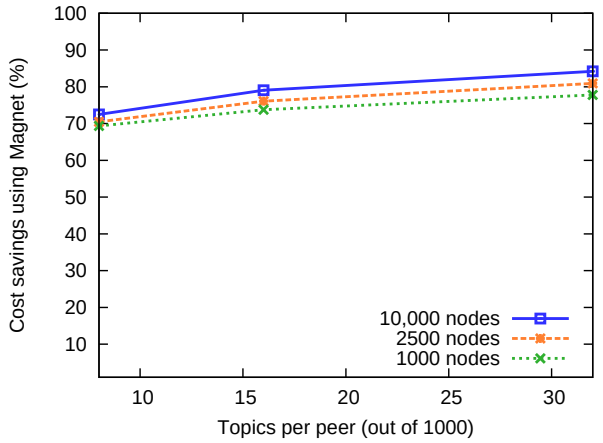


Figure 6: SPATIAL model: As the average number of subscription per peer increases, MAGNET's performance improves due to correlations stemming from locality.

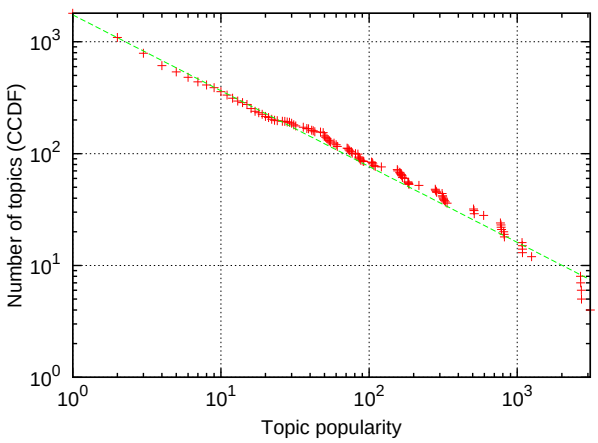


Figure 7: HIERARCHICAL-TOPICS model: Complementary CDF (CCDF) of topic popularity with 2^{14} nodes, 2^{14} topics, 16 topics/node on average (initFrac = 10%, $\alpha_0 = 2$, $\lambda = 2$). The green fit line is the CCDF of a power-law distribution with $\alpha = 1.68$.

close by in the hierarchy according to the λ parameter.

a) We start by populating all topics with random peers such that the topic popularity follows a power-law distribution with exponent α_0. We add subscriptions iteratively until the target average degree of $Z \cdot$ initFrac is reached, where Z is the model parameter representing an average number of subscriptions by each peer. The initFrac parameter effectively characterizes what fraction of the total number of links should be picked at random during the initialization.

b) The topics are then organized as the leaves of a binary tree. We then repeat the following steps until the target average peers subscription size Z is reached.

  1) Topic t is picked with probability proportional to t's popularity. This step is a variation of the preferential-attachment model [20].

  2) Next, peer p is picked uniformly at random from the list of all the t's subscribers.

  3) Let ℓ be a random variable representing tree distance, such that $\Pr[\ell = x] = Ce^{-\lambda x}$ where x can be at the most the height of the hierarchy and C is a normalizing constant. Peer p now subscribes to topic t', which is picked uniformly at random among topics at distance ℓ from t.

Figure 7 shows a workload produced by running the model with 2^{14} nodes, 2^{14} topics, $Z = 16$ topics per node on average, initFrac = 10%, $\alpha_0 = 2$ and $\lambda = 2$.

Results. Like in our previous experiments, we studied how well MAGNET can exploit the correlation among the peer subscriptions generated by the HIERARCHICAL-TOPICS model. We fixed the number of both topics and peers to 2^{14} and the exponent of the initial power-law distribution to $\alpha_0 = 2$. Since the main parameters affecting the correlation rate among the peer subscriptions are λ and initFrac, for the first set of experiments we have investigated MAGNET's behavior as we vary those parameters.

Figure 8(a) shows MAGNET's performance with the average number of topics per peer set to $Z = 16$. It is evident that MAGNET performs better than the strawman implementation with high values of λ and low values of $initFrac$ since these produce the most correlated subscription patterns. In the second set of experiments we have fixed the value of initFrac to 10% and studied the MAGNET's performance under different values of the average subscription size Z. We also see that since the smaller values of Z imply higher correlation among the peer subscriptions in model, MAGNET performed best with $Z = 8$ (see Figure 8(b)).

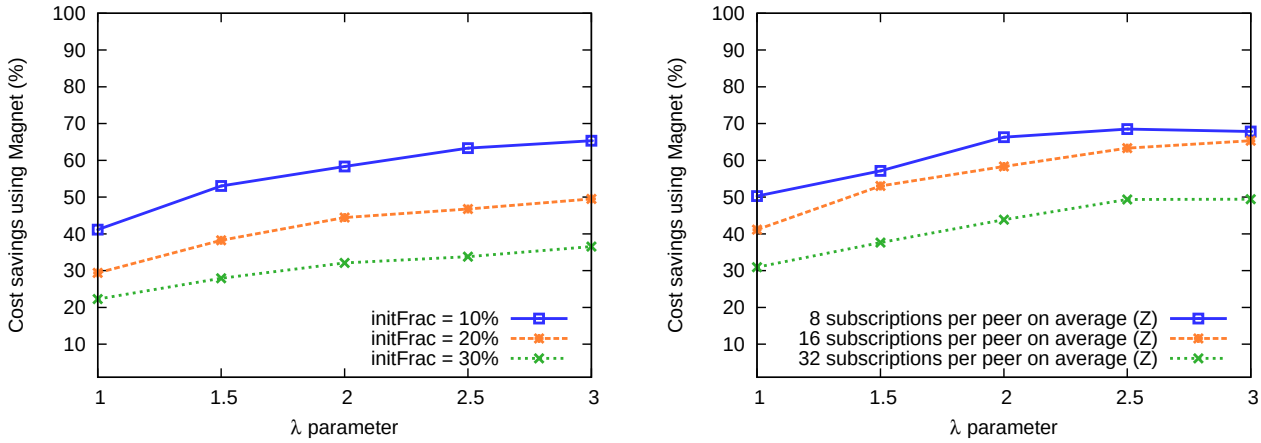
4.5 Wikipedia Subscription Patterns

We have also analyzed the performance of MAGNET using the subscription workload extracted from the trace of all edits of Wikipedia entries by registered users over a 6 year period. In this experiment, each entry of the encyclopedia was treated as a topic and each unique editor as a MAGNET peer. The entries edited by a specific users were interpreted as the the interest of the corresponding peer.

For our experiments we have selected 3000 random topics from the entire entry set, which were edited by nearly 10,000 unique users. The topic popularity varies from 1 to 348 subscribers per topic, and on average every topic has 5.4 subscribers. These subscription data were fed to MAGNET and to the strawman implementation. Average node degree for both P2P networks was set to 12. We measured the cost of publishing messages for each of the topics in the network. The experiments showed that the distribution trees constructed by the strawman implementation included on average 77% more uninterested peers than those constructed by MAGNET.

5. RELATED WORK

Several publish/subscribe systems based on structured overlays have been proposed in the past, notably Scribe [6] and Bayeux [36]. Generally speaking, none of these systems attempt to cluster peers based on their subscription similarity. An exception is TERA [4], which creates an overlay for each topic to accelerate dissemination. The scalability of TERA, however, is limited when the number of topics subscribed to by nodes is large.



(a) Varying the initFrac parameter.

(b) Varying the avg. subscriptions size Z per peer.

Figure 8: HIERARCHICAL-TOPICS model: MAGNET's performance improves with increased subscription correlation (higher value of λ). Higher values of initFrac imply fewer exploitable correlations, supporting the trend seen on the curve on the left. Increasing the average degree Z while keeping initFrac constant incorporates randomness from the initialization stage as suggested by the decline on the right. We see that MAGNET brings significant cost savings over the strawman approach on the HIERARCHICAL-TOPICS model.

The benefits of clustering peer subscriptions have been investigated in the context of unstructured overlays [31, 22, 9, 3]. In particular, Sub-2-Sub [31] achieved clustering by organizing the subscribers to each topic into a separate ring structure. This approach, however, results in overlays whose average degree grows linearly with the average subscription size which limits scalability. Rappel [22] provides a feed-based pub/sub service using gossip-like mechanisms and exploits interest similarity to avoid messages being received by uninterested nodes. Due to the assumption that each topic has only a single publisher, as is the case in feed-based systems, Rappel differs fundamentally from our work.

The idea of exploiting subscription similarity to reduce the space per node requirements of the clustering was explored in Spidercast [9] and Data-aware multicast [3]. The trade-off between the node degree and the clustering quality has been addressed in a theoretical study [8].

Topic clustering [21, 27, 2, 29] looks into amortizing overheads associated with message dissemination in large pub/sub systems by aggregating multiple topics into larger groups (or channels). It was first introduced in [2] in the context of optimal assignment of multicast groups to multicast addresses, and subsequently extended to general purpose pub/sub systems in [27, 29]. The existing solutions to topic clustering rely on approximation techniques (such as k-means [27]) whose convergence depends on the accurate common knowledge of the current assignment of topics to channels. They are not easy to implement in a decentralized fashion [25].

6. CONCLUSION

In this paper, we have presented MAGNET, an overlay-based infrastructure for scalable topic-based pub/sub which takes advantage of the existing subscription correlation patterns among the subscribers. The technique we proposed allows the formation of clusters of peers with similar interests in the underlying topology, which enables the construction of efficient dissemination structures (specifically spanning trees) that are known to be robust (e.g., Scribe, Bayeux). However, the clustering process leads to the non-uniform peer identifier distributions which renders all name-independent DHT solutions (e.g., Chord, Pastry) unusable. Therefore, MAGNET employs the OSCAR overlay as the underlying topology which is provably small-world and can efficiently operate with arbitrary distribution scenarios. Because of its inherent small-world design, MAGNET scales well with the number of nodes, and ensures fixed network degree regardless the number of topics or the size of subscriptions.

We simulate MAGNET on a variety of subscription models, including a novel one, as well as on real-life subscription patterns from Wikipedia. Our experiments show that MAGNET is able to achieve significant savings — sometimes up to 80% — of the message dissemination costs over a strawman running a typical peer-to-peer publish/subscribe system based on name-independent DHTs. Furthermore, we demonstrate that this cost reduction is adaptive to both the extent to which the individual node subscriptions correlate, and the amount of information about the other node subscriptions available to each node. In particular, in the worst case scenario when subscriptions are completely uncorrelated or unknown or both, the message dissemination costs are no worse than those of name-independent DHT systems.

Our findings suggest that subscription clustering techniques can detect and exploit correlation at low cost, and may improve the performance of large-scale publish/subscribe systems for a variety of settings.

7. ACKNOWLEDGMENTS

We thank our shepherd and the anonymous reviewers for helpful feedback. This work is partially supported by EU IST Project CoMiFin FP7-ICT-225407/2008 and partially carried out within the SICS Center for Networked Systems funded by VINNOVA, SSF, KKS, ABB, Ericsson, Saab Systems, TeliaSonera, T2Data, Vendolocus and Peerialism.

8. REFERENCES

[1] I. Abraham, D. Malkhi, and O. Dobzinski. D.: LAND: Stretch $(1 + \varepsilon)$ locality-aware networks for DHTs. In *Proc. 15th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 550–559, 2004.

[2] M. Adler, Z. Ge, J. F. Kurose, D. F. Towsley, and S. Zabele. Channelization problem in large scale data dissemination. In *ICNP 2001: Proceedings of the Ninth International Conference on Network Protocols*, page 100, Washington, DC, USA, 2001. IEEE Computer Society.

[3] S. Baehni, P. T. Eugster, and R. Guerraoui. Data-aware multicast. In *DSN '04: Proceedings of the 2004 International Conference on Dependable Systems and Networks*, page 233, Washington, DC, USA, 2004. IEEE Computer Society.

[4] R. Baldoni, R. Beraldi, V. Quema, L. Querzoni, and S. Tucci-Piergiovanni. Tera: topic-based event routing for peer-to-peer architectures. In *DEBS '07: Proceedings of the 2007 inaugural international conference on Distributed event-based systems*, pages 2–13, New York, NY, USA, 2007. ACM.

[5] A. Bharambe, M. Agrawal, and S. Seshan. Mercury: Supporting scalable multi-attribute range queries. In *ACM SIGCOMM, Portland, USA*, 2004.

[6] M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron. SCRIBE: a large-scale and decentralized application-level multicast infrastructure. *IEEE J. Selected Areas in Comm. (JSAC)*, 20(8):1489–1499, 2002.

[7] M. Cha, P. Rodriguez, J. Crowcroft, S. Moon, and X. Amatriain. Watching television over an IP network. In *IMC '08: Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, pages 71–84, New York, NY, USA, 2008. ACM.

[8] G. Chockler, R. Melamed, Y. Tock, and R. Vitenberg. Constructing scalable overlays for pub-sub with many topics. In *PODC '07: Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing*, pages 109–118, New York, NY, USA, 2007. ACM.

[9] G. Chockler, R. Melamed, Y. Tock, and R. Vitenberg. SpiderCast: A Scalable Interest-Aware Overlay for Topic-Based Pub/Sub Communication. In *1th International Conference on Distributed Event-Based Systems (DEBS)*. ACM, 6 2007.

[10] D. J. Crandall, D. Cosley, D. P. Huttenlocher, J. M. Kleinberg, and S. Suri. Feedback effects between similarity and social influence in online communities. In *KDD*, pages 160–168. ACM, 2008.

[11] S. Girdzijauskas, A. Datta, and K. Aberer. On small world graphs in non-uniformly distributed key spaces. In *NetDB2005, Tokyo, Japan*, 2005.

[12] S. Girdzijauskas, A. Datta, and K. Aberer. Oscar: Small-world overlay for realistic key distributions. In *DBISP2P 2006, Seoul, Korea*, 2006.

[13] S. Girdzijauskas, A. Datta, and K. Aberer. Oscar: A Data-Oriented Overlay For Heterogeneous Environments. In *ICDE 2007, Istanbul, Turkey*, 2007.

[14] S. Girdzijauskas, A. Datta, and K. Aberer. Structured overlay for heterogeneous environments: Design and evaluation of oscar. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, Volume 5, February 2010.

[15] R. Guerraoui, S. B. Handurukande, K. Huguenin, A.-M. Kermarrec, F. L. Fessant, and E. Riviere. GosSkip, an efficient, fault-tolerant and self organizing overlay using gossip-based construction and skip-lists principles. *IEEE International Conference on Peer-to-Peer Computing*, 0:12–22, 2006.

[16] J. Kleinberg. The Small-World Phenomenon: An Algorithmic Perspective. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, 2000.

[17] J. Kleinberg. Complex networks and decentralized search algorithms. *Proceedings of the International Congress of Mathematicians (ICM)*, 2006.

[18] B. Knutsson, H. Lu, W. Xu, and B. Hopkins. Peer-to-peer support for massively multiplayer games. *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, 1, 2004.

[19] M. Mitzenmacher. The power of two choices in randomized load balancing. *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1094–1104, 2001.

[20] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[21] K. Ostrowski, K. Birman, and D. Dolev. Live distributed objects: Enabling the active web. In *IEEE Internet Computing, 11(6), p. 72*, 2007.

[22] J. A. Patel, E. Riviere, I. Gupta, and A.-M. Kermarrec. Rappel: Exploiting interest and network locality to improve fairness in publish-subscribe systems. *Computer Networks*, 53(13):2304 – 2320, 2009. Gossiping in Distributed Systems.

[23] T. Qiu, Z. Ge, S. Lee, J. Wang, Q. Zhao, and J. Xu. Modeling channel popularity dynamics in a large IPTV system. In *SIGMETRICS '09: Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems*, pages 275–286, New York, NY, USA, 2009. ACM.

[24] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany*, 2001.

[25] A. Shraer, G. Chockler, I. Keidar, R. Melamed, Y. Tock, and R. Vitenberg. Local on-line maintenance of scalable pub/sub infrastructure. In *DSN 2007: in the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, page 100, Edinburgh, UK, 2007.

[26] I. Stoica, R. Morris, D. R. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM*, pages 149–160, 2001.

[27] Y. Tock, N. Naaman, A. Harpaz, and G. Gershinsky. Hierarchical clustering of message flows in a multicast data dissemination system. In *17th IASTED International Conference Parallel and Distributed Computing and Systems*, pages 320–327, 2005.

[28] Y. Vigfusson. *Affinity in Distributed Systems*. PhD thesis, Cornell University, 2009.

[29] Y. Vigfusson, H. Abu-Libdeh, M. Balakrishnan,

K. Birman, R. Burgess, G. Chockler, H. Li, and Y. Tock. Dr. Multicast: Rx for Data Center Communication Scalability. In *EuroSys '10: Proceedings of the ACM SIGOPS/EuroSys European Conference on Computer Systems*, April 2010.

[30] K.-H. Vik, C. Griwodz, and P. Halvorsen. Applicability of group communication for increased scalability in MMOGs. In *NetGames '06: Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games*, page 2, New York, NY, USA, 2006. ACM.

[31] S. Voulgaris, E. Riviere, A.-M. Kermarrec, and M. van Steen. Sub-2-sub: Self-organizing content-based publish subscribe for dynamic large scale collaborative networks. In *IPTPS*, 2006.

[32] S. Weiss, P. Urso, and P. Molli. Wooki: A P2P wiki-based collaborative writing tool. In *WISE*, volume 4831 of *Lecture Notes in Computer Science*, pages 503–512. Springer, 2007.

[33] B. Wong, Y. Vigfússon, and E. G. Sirer. Hyperspaces for object clustering and approximate matching in peer-to-peer overlays. In *HotOS'07: Proceedings of the 11th USENIX Workshop on Hot Topics in Operating Systems*, pages 1–6, Berkeley, CA, USA, 2007. USENIX Association.

[34] T. Wong, R. Katz, and S. Mccanne. An evaluation of preference clustering in large-scale multicast applications. In *Proceedings of IEEE INFOCOM*, pages 451–460, 2000.

[35] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph. Tapestry: An infrastructure for fault-tolerant wide-are location and routing. Technical Report UCB/CSD-01-1141, UC Berkeley, 2001.

[36] S. Q. Zhuang, B. Y. Zhao, A. D. Joseph, Y. H. Katz, and J. D. Kubiatowicz. Bayeux: An architecture for scalable and fault-tolerant wide-area data dissemination. pages 11–20, 2001.