Computer Networks 100 (2016) 124–140

Contents lists available at ScienceDirect

Computer Networks

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

Popularity-based scalable peer-to-peer topology growth[☆]

Gurhan Gunduz[a,1,*], Murat Yuksel[b]

[a] Computer Engineering Department, Pamukkale University, Turkey
[b] Computer Science and Engineering, University of Nevada - Reno, Reno, NV 89557, USA

ARTICLE INFO

Article history:
Received 22 May 2015
Revised 3 February 2016
Accepted 14 February 2016
Available online 23 February 2016

Keywords:
Peer-to-Peer networks
Scalable
Popularity-based
Overlay networks

ABSTRACT

Peer-to-peer (P2P) networks have gained importance and spread significantly during the last decades resulting in raised research interest in this area. The basic premise of P2P design is higher scalability and many existing large-scale applications, such as Twitter and Skype, use a form of P2P design. Although *structured* P2P designs enable one to guarantee finding of every item (rare or popular), they do not scale beyond a point and support from servers are needed. This breaks the decentralized design of the P2P system and results in a hybrid scheme. *Unstructured* P2P networks, on the other hand, can scale to much larger nodes but yet cannot give time guarantee for finding a rare item.

Topological characteristics of unstructured P2P networks have impact on the efficiency of a search for items. Earlier studies have shown that popularity of the nodes has significant impact in the self-organization of the overlay topology. It is well known that unstructured P2P designs are very good when searching for a popular item. But, when a new trend is emerging, it is a rare item until it becomes highly popular. During that transitional period, unstructured designs suffer from inefficiency in finding *to-be-popular* (TBP) items. Such TBP items could be stuck at nodes with small degrees and thus become hard to find. We hypothesize that if the overlay topology is established and grown with a more proactive consideration of the items' popularity, the delay in finding TBP items could be reduced significantly. Further, such topology growth will reduce the search time for popular items. Thus, the overall search performance of the P2P system will significantly improve as well, since most of the searches are for popular items. In this paper, we investigate incorporating item popularity into the overlay topology in a scalable way.

1. Introduction

Peer-to-peer (P2P) networks have gained importance and spread significantly during the last decades resulting in raised research interest in this area. The basic premise of P2P design is higher scalability and many existing large-scale applications, such as Twitter and Skype, use a form of P2P design. In P2P networks, a peer can search and download resources from other peers while contributing own resources to others in the same network. The approach of peering distributes the load and functions across the participant peers/nodes, and hence, attains a much better scalability in comparison to centralized designs. The peers typically use an overlay topology to search and find each other to share or exchange their resources. Although overlaying has excellent advantages in providing fast deployment of protocols and flexibility in function placement, it can cause significant holdbacks in terms of performance if not carefully handled. Particularly in unstructured P2P networks [1], the characteristics of the overlay topology

[☆] This work was supported in part by United States NSF awards AST-1444077 and CNS-1321069, and Republic of Turkey's TUBITAK.

[*] Corresponding author. Tel.: +905055827396; fax: +1 775 7841877.
E-mail addresses: ggunduz@pau.edu.tr (G. Gunduz), yuksem@cse.unr.edu (M. Yuksel).

[1] Dr. Gunduz was a visiting researcher at the University of Nevada - Reno during most of this work.

http://dx.doi.org/10.1016/j.comnet.2016.02.017
1389-1286/© 2016 Elsevier B.V. All rights reserved.

have profound impact on the efficiency of flooding-based P2P search and applications.

Several algorithms have been designed for constructing the topology of unstructured P2P networks [2]. These algorithms typically devise a distributed mechanism by defining how a new node joins or an existing node leaves the network. With the assumption that every node is a peer and the same, most of the recent work on this topic looked at using node degree for organizing the topology. In fact, every node or peer in these networks may be different than others in terms of computation power, bandwidth, capacity and the resources that it is sharing. One implicit way of determining which node has more resources is to measure how popular the nodes are. This imbalance of resources naturally gives rise to a skew in the popularity of the nodes, as some nodes become more popular due to their larger resources and/or the popularity of the items it is maintaining. Intuitively, if a node has more and especially popular items, then more peers will be interested in interacting with that node. This intuition also can be used as the foundation for relating popularity of the nodes to their degrees, since peers will be interested in establishing and maintaining a link to a resourceful node only. Further, some nodes may become more popular if they have items that everybody is looking for. Other peers will want to create a connection with these popular nodes. Therefore, the term “popular” in this context means having relatively higher number of links compared to other nodes in the system.

Another key issue the existing P2P topology construction algorithms need to address is the efficient and effective treatment of trends and memes in the recent social networks [3,4]. Handling such trends and memes in a scalable way alongside with the growing usage of device-based networking and sharing applications (such as Twitter) is a challenge [5]. It is well known that unstructured P2P designs are very good when searching for a popular item. But, when a new trend is emerging, it is a rare item until it becomes highly popular. During that transitional period unstructured designs suffer from inefficiency in finding to-be-popular (TBP) items. Such TBP items could be stuck at nodes with small degrees and thus become hard to reach. We hypothesize that if the overlay topology is established and grown with a more proactive consideration of the items’ popularity, the delay in finding TBP items could be reduced significantly. Further, such topology growth will reduce the search time for popular items. Thus, the overall search performance of the P2P system will significantly improve as well, since most of the searches are for popular items.

The primary focus of this research is to find scalable ways of *embedding the item popularity* into the overlay topology so that flooding-based search on overlay topologies is efficient. We introduce a popularity parameter for each node, which is based on the popularity of the items that the node has. This information is used when a new node wants to join the network. Topologies are generated based on this factor along with degree. In order to give our method a practical perspective, we limit the number of neighbors of a node to a predefined value indicated by hard cutoff [6]. The ad-hoc nature of peers is exploited by rewiring the network upon failure or departure of a peer. We also investigate the effect of these popularities and rewiring of ad-hoc nodes on the P2P search efficiency.

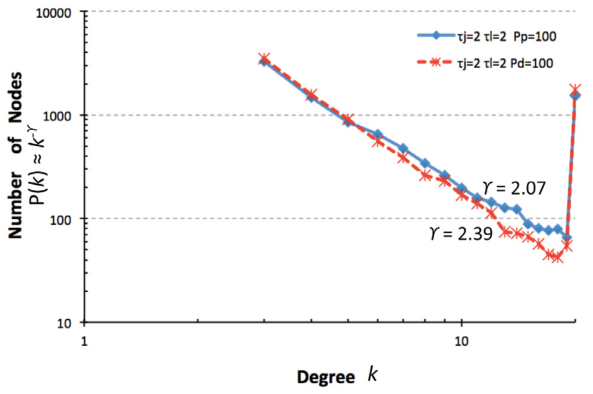

Fig. 1. Comparison of degree distribution of popularity ($P_p = 100$) and degree ($P_d = 100$) based P2P overlay topologies with a degree cutoff $k_c = 20$: Popularity-based overlays have fatter tail, indicated by the smaller power exponent, $\gamma = 2.07$.

Our test results show the pattern in Fig. 1 for a high-level comparison of degree-based topology and our popularity-based topology. Detailed analyses of the results show us the following: The number of minimum and close-to-minimum degree nodes is higher in the degree-based topology. This is not desirable since having only few neighbors will affect the search performance of the topology. As the node degree increases, our popularity-based topology has more nodes than the degree-based topology. Having high degree nodes results in more connected topology, which eventually increases the search performance. The only advantage of degree-based topology is that the number of maximum degree nodes is slightly higher than our popularity-based topology. Even though this feature helps degree-based topology for improved search performance, it is not enough to perform better than our popularity-based topology

1.1. Contributions

Popularity of the nodes has significant impact in the self-organization of the overlay topology. Power-law graphs have super-hubs which can be considered as the “center” of the topology. The most profound property of these super-hubs is their large degree. Yet the research shows mixed results [7] on power the existence of power-law behavior in P2P networks. It was reported [8,9] that power-law behavior emerges in unstructured P2P networks if nodes that are close to the edge of the topology are excluded. Several studies [10–12] showed small-world topology for P2P networks. Small world topologies do not have super-hubs. But, the “center” is the node(s) where betweenness centrality is the maximum. Even for small-world networks, betweenness centrality is higher at nodes equally farther away from the leaf nodes, which usually causes those nodes to be at the cross-section of many end-to-end paths and hence have a higher degree [13].

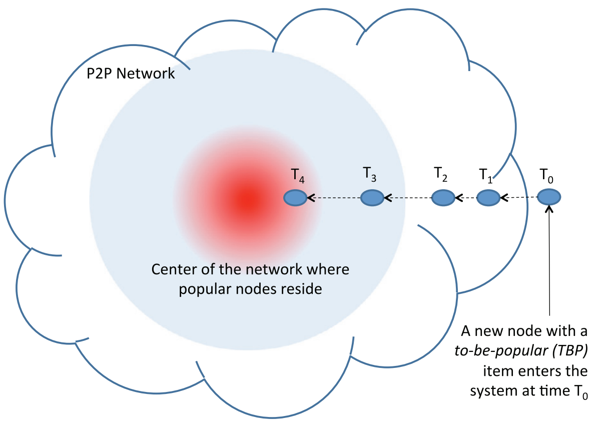
P2P Network
T_4
T_3
T_2
T_1
T_0
Center of the network where popular nodes reside
A new node with a *to-be-popular (TBP)* item enters the system at time T_0

Fig. 2. Illustration of "topological gravitation" on a joining node with a to-be-popular (TBP) item: A larger time difference $T_4 - T_0$ can significantly degrade the search efficiency on the new TBP item.

So, in general, a node residing in a central location of the topology will have a large degree. More detailed models for quantifying a nodes centrality could involve measures like betweenness centrality, closeness centrality, and degree centrality [14]. In our study, we use a simple model and consider the node degree as the main indicator of how central the node is within the network topology.

A newly joining node will have few connections at first and the number of its connections will increase in time depending on its popularity. Since higher degree nodes are more central to the overall topology, this change in a new node's degree can also be considered as a movement towards the "center" of the P2P overlay topology over time. That is, the new node will gravitate into the center of the overlay topology as illustrated in Fig. 2. One of the purposes of this work is to find how the popularity of a node affects this "topological gravitation" of that node into the center of the network in time. The key difference of our work is that it sheds light into the transient behavior of search efficiency as the new nodes with different popularities join the overlay. Prior work on the search efficiency only explored spatial efficiency of the search assuming that the overlay topology stands still over time. But, as the size of the overlay is getting bigger, consideration of both temporal and spatial dynamics is needed to understand the effectiveness and behavior of unstructured P2P networks at larger scales. Other contributions include:

- Guidelines for generating scale-free topologies based on popularity: We introduce a model that embeds item popularity into network topology and this information is used when a node joins or leaves.
- Search efficiency on the generated scale-free topologies: Through extensive simulations, we studied efficiency of Normalized Flooding (NF) on the topologies generated by our model.
- Transient properties of popularity-based topology growth: We investigated how search performance is affected when a new item (e.g., Twitter hashtag or meme) is trending to become popular.

2. P2P topology and search

Performing efficient decentralized search is a fundamental problem in Peer-to-Peer (P2P) systems. There has been significant amount of research in developing robust self-organizing P2P topologies that support efficient search. The rate at which the search is successful is directly proportional to the efficiency of the search. Search success rate depend upon the capability of preserving nature during churn in the networks, which is difficult to obtain in peer-to-peer networks. Thus most of the practical solutions involve unstructured approaches while attempting to maintain the structure at various levels of protocol stack as unstructured P2P networks do not impose any structure on the overlay networks. Peers in these networks connect in an ad-hoc fashion. Such networks can be easily constructed as a new peer that wants to join the network can copy existing links of another node and then form its own links over time.

Unstructured P2P overlays are inherently flexible in their neighbor selection and routing mechanisms. They can actually make use of the neighbor information to localize the communication pattern of the system. These networks can create topologies that are able to withstand to random failures as well as targeted malicious attacks. In addition to these robustness measures efficiency of search in P2P networks is also heavily relied upon the topological characteristics. The best search efficiency in realistic networks can be achieved when the topology is scale-free (power-law) in terms of some of the key factors such as the node degree, i.e., the number of connections a peer has with its neighbors in the overlay topology. Establishing such scale-free topological patterns has been done with the phenomena of "rich-get-richer". In essence, the key to achieving a scale-free topology structure is to quantify and use how popular a node/peer is in the overall network. With a high-level description, being popular corresponds to being rich and means that new nodes joining the network more likely to select a popular existing node as a neighbor. The node degree has traditionally been used to make this quantification of how popular a node is in the network.

The distinctive dimension we explore in this work is to seek an answer to the following question: *"What are other potential ways of quantifying a node's popularity with more accuracy?"* We consider the popularity of items each node has and aim to arrive at a finer granularity quantification of the node popularity. The intuition is that if popularity quantification is done better, it becomes possible to construct topologies so that the efficiency of search in unstructured P2P networks increases. Based upon the popularity of items on a node the popularity of the node can be determined as a sum of the individual popularities of the items. The popularity of an item can be based on (i) the number of times it was queried/downloaded or (ii) an explicit measure provided by an external entity that is introducing the item to the network. The latter approach is particularly useful when a completely new item is posted on a node since the former approach will have to spend some sizable amount of time to establish an accurate popularity value for that new item. As an example, if a video file for a famous and recent event is being posted for the first time, the owner of the file should better associate a higher popularity value to the item, which can be done in latter approach. A hybrid of the two approaches can be done as well by combining the number of queries and the explicit popularity value entered by an external entity. In this study, we assume the popularity value of an item is readily available.

In an unstructured P2P network, if a peer wants to find a desired piece of data/item in the network, the query has to be flooded though the network to find as many peers as possible that share that data. This can be a disadvantage as the query may not always be resolved. So a single query can generate a number of query messages floating in the network, thereby increasing the consumption of network resources to an extent that can limit the scalability and efficiency of the network. It becomes very important to choose the right search method for the system. With this in mind, selecting a search technique can be a compulsion instead of a choice. Simple blind flooding mechanisms depend on the network structure. An irregular graph causes a significant deterioration in the search performance by increasing the number of messages floating in the system. There can be a sudden increase in the number of flooded messages, if flooding process reaches a node with high degree. Hence, a new technique has been introduced to subsidize this problem, known as Normalized Flooding (NF) [15]. It is similar to flooding but the nodes send the messages to at most m (minimum number of links in the network) neighbors. We used NF to search items in our simulation experiments.

3. Popularity based topology growth

In terms of graph theory or networks, node degree is defined as the number of connections or links the node has to the other nodes. Degree distribution is one of the most important factors to be considered for the study of networks, either theoretical or real. Degree distribution, denoted by $P(k)$, of a network is defined as the probability distribution of the degrees of nodes, k, in the entire network. $P(k)$ can be represented as the fraction of nodes with degree k. For a network with a total of n nodes, if n_k of the nodes have degree k, then $P(k) = n_k/n$ is a way of estimating the degree distribution, $P(k)$.

3.1. Motivation: join and leave under network dynamics

Before getting into more specifics of our model, it is important to discuss our design motivations. Networks should show robustness against a dynamic behavior, e.g., it should survive various kinds of churn. The source of churn for unstructured P2P networks is that nodes can join or leave the network at any point of time. The reasons why a node left the network range from a simple closing of the browser to some technical reasons. It is not only hard to predict when a node is about to leave and join but it is also difficult to have control over the leave and join. There have been various researches performed in order to estimate the average time a node stays in the P2P network. This dynamic behavior gives rise to various practical concerns, such as:

- When a new peer joins, how should it construct its list of neighbors?
- When a new peer leaves, how should its neighbors rewire themselves to the network?

It is very important to address these concerns in order to ensure the robustness and search efficiency of the P2P network. There are various protocols and algorithms suggested for diluting the effect of churn created by joining and leaving nodes, but most of them use global information about the network in order to construct and rewire the network. This is how our model stands out as we construct and rewire the network based on local parameters discussed more in the next sections. In our approach, we do not rewire the topology unless there is a node join or leave. If any neighbor of a node leaves, our leave algorithm tries to regain that lost connection by applying a popularity-based preferential attachment. We did not consider directly adjusting a node's degree based on the popularity of the items on it. Our approach indirectly increases or decreases the node's degree as the sum of popularity of the items on it.

While keeping the join/leave actions local is more practical and scalable in terms of protocol overhead, maintaining global features of the network is challenging. Our earlier work showed such an example challenge in maintaining a scale-free topology while trying to adhere to hard cutoffs for node degrees [16,17]. We were able to show that one can effectively grow a scale-free topology and attain good search efficiency (i.e., decreased amount of time needed to find an item) for while keeping actions within a few hops of nodes joining or leaving the P2P network. Since the resulting topology was a mostly power-law one, the search in those scale-free networks were yielding better performance for "popular" items, which is the hallmark of unstructured P2P systems. However, the recently emerged social networking applications, such as Twitter and Facebook, brought new phenomena of trends and memes [3,4]. Our main focus in this paper is *to re-design our scale-free topology growth techniques so that popular as well as to-be-popular (TBP) items can be found quickly.*

3.2. Preferential attachment with popularity and hard cutoffs

A widely known technique for growing scale-free topologies is preferential attachment (PA). The PA model was introduced by Barabasi and Albert [2] to explain how the power-law degree distribution in complex real-world networks emerges. It is used to generate random scale-free networks. In this algorithm, the network begins with an initial network of m_0 nodes where $m_0 > 1$. The degree of each node in the initial network should be at least 1; otherwise, it may remain disconnected from the rest of the network. New nodes are added to the network one at a time. Each new node is connected to m existing nodes with a probability that is proportional to the number of links that the existing nodes already have. Formally, the probability p_i that the new node is connected to node is [18] $p_i = k_i/\sum_j k_j$ where k_i is the degree of node i. heavily linked nodes tend to quickly accumulate even more links, while nodes with only a few links are unlikely to be chosen as the destination for a new link. The new nodes have a "preference" to attach themselves to the already heavily linked nodes.

We revised the PA model by adding the concept of popularity to the network topology. The popularity of each node in the network can be determined based upon various factors. In our proposed algorithm, each node will have items (i.e., downloadable items) that can be queried by other nodes. Each item has got its own popularity, which, in practice, might depend upon factors like the number of times the item was queried, the number of times the item was downloaded or the number of nodes that have the same item (also called replication of objects in a network). In our design, the popularity of a node is the sum of the popularities of each item that a node owns. After calculating the popularities of items across different nodes, a node's popularity is calculated by summing the popularities of items residing on that node. The overhead of this calculation is negligible since it only involves summation of the item popularities on a node. Furthermore, this calculation is done reactively when a neighbor leaves or a new node tries to connect. In terms of the memory, there will be a popularity value stored for each item, which will occupy a few bytes of space per item. Since the item sizes will be significantly larger, the memory overhead will be negligible as well.

When a node wants to join a network, it chooses its neighbors based on the "hybrid preferential attachment". The term "hybrid" is used in our concept because we use both degree and popularity factors at the same time. The traditional preferential attachment uses node degrees to determine how the nodes connect with each other. This approach has assumed that a node's popularity is mainly determined by its degree, due to the conventional wisdom that a node should be more connected if it is more popular.

Our work focuses much upon examining the performance of the degree based and popularity based algorithms used for peer selection in unstructured P2P systems. In our model we parameterize the followings:

(i) the ad-hoc behavior of the nodes by using probability that a node leaves, μ, which is also called "churn" here;
(ii) the amount of local information to be used at the time of join within the radius of proximity from the point the node joins, τ_j (i.e., the node knows about the local topology covering τ_j hops away from the point node joins the network);
(iii) the amount of local information to be used at the time of leave within the radius of proximity from the location of the neighbor of the leaving node, τ_l (i.e., each neighbor of the leaving node knows about the local topology covering τ_l hops away from itself);
(iv) the maximum number of downloadable items a node can have in the network, N_i (i.e., each node will have items that will be requested by other nodes in the network);
(v) the maximum number of links to be stored by peers as hard cut off k_c, for the degree of a peer in the network as compared to natural cutoff emerging from finite-size effects; and
(vi) the weightage, w, to specify the importance of nodes' popularity in the preferential attachment process.

As explained above, the regular PA process uses nodes' degrees to identify the existing node, to which a newcomer will establish a connection with. We use the weight parameter, w, to steer the PA process so that it uses nodes' popularities along with their degrees when making this critical decision for newcomers. In particular, to explore a balance between node degrees and node popularities, we use the weightage, w, across degree and popularity when determining the total preference probability of a node in the preferential attachment process. If the total weightage is given to the popularity (i.e., $w = 100$) then the node with highest popularity is preferred. It becomes possible to give different weightage to the popularity or degree by changing w. This approach also means that a newly joining node will need to have information about its potential neighbor's popularity and degree in some cases. The weightage parameter, w, plays a critical role in our algorithm because it may result in significantly different set of neighbors for a newly joining node – particularly when there is disparity between nodes' degrees and popularities, which may happen when new trends or memes are settling into the P2P network. A mathematical and detailed explanation of our algorithm follows next.

3.3. The algorithm

Before looking further into the algorithm, we start with a more detailed and technical understanding of its parameters:

- G: graph of the existing network of M links and N nodes
- M: the number of links in the existing network
- N: the number of nodes in the existing network
- N_{target}: the number of nodes in the network at the end of the growth
- *Number of items*, N_i: the maximum number of items that may exist in node i
- *Churn rate*, μ: the probability of a node leaving/deleted from the network
- *Hard cutoff*, k_c: the maximum limit on the degree (or the number of links) a node can have
- *TTL at join*, τ_j: the horizon of the nodes in the network, that a new node can connect to
- *TTL at leave*, τ_l: the horizon of the nodes in the network, that the neighbors of the deleted/leaving node can connect to
- *Number of stubs*, m: the minimum degree (or the number of links) a node must have in the network
- *Popularity weightage*, w: the percentage of nodes' popularity in determining the probability of attachment in the preferential attachment process
- $PA(G1, G2, w)$: a function that performs preferential attachment to $G1$ using the nodes in $G2$ with a popularity weightage of w, and returns the number of successful new links

- *Random*(a, b): a function that returns a uniformly distributed random number between a and b

3.3.1. PA with weightage and hard cutoff

During the PA process, an incoming node connects to an existing node according to its degree with respect to the entire set of existing nodes – as was explained in Section 3.2. We revise the PA process so that it uses the popularity of the existing nodes when calculating the probability of attachment, p_i, to an existing node i. We use the weightage, w, to design a hybrid (degree- and popularity-based) PA.

We use the weightage w to merge popularity and degree of the existing nodes into one normalized value of probability of attachment, H_i. If $w = 100$, then the node with the highest popularity is preferred. Likewise, $w = 0$, then the node with the highest degree is preferred. More specifically, we first calculate the the probability of attachment based on node degrees, which is

$$p_i = k_i \Big/ \sum_j k_j \tag{1}$$

where k_i is the degree of node i. Next, we calculate the probability of attachment based on node popularities as follows

$$q_i = r_i \Big/ \sum_j r_j \tag{2}$$

where r_i is the popularity of the node calculated based upon each item's popularity generated by a Zipf distribution. Then, we normalize these probabilities into one probability of attachment using the weightage factor, w:

$$H_i = \frac{w \times p_i + (100 - w) \times q_i}{100} \tag{3}$$

Once the normalized probability of attachment, H_i, is calculated for all the nodes in a neighborhood, we finally apply the regular PA process using these normalized probabilities.

3.3.2. Weighted join

A newly added node, first, selects a random node from the existing network and connects to it. Then, with the provided value of τ_j (may vary from 1 to 3), it constructs a set of nodes that are τ_j or fewer hops away from the node it has recently connected to. The set does not contain the nodes that already have degree equal to the hard cutoff, k_c. Next, it randomly selects a node from this set and connects to it, with a probability proportional to its degree and popularity. The probability is normalized by the total degree and total popularity and the percentage of preference, w, to the popularity of the nodes in the set. Subsequently, until its degree reaches to the m (minimum degree), it selects other nodes from the set and tries to connect to it with the probabilities calculated in the first step. If there is no left over nodes in that particular set and the degree of the newly added node is still smaller than m, then it will select a random node again from the entire network and continue the same process until its degree is greater than or equal to m but smaller than k_c. A pseudo-code of this "weighted join" process is given in Algorithm 1.

Algorithm 1 Join Algorithm.

```
Void WeightedJoin($N_i$,$\tau_j$, $w$){
    $N \leftarrow N + 1$
    $numberOfLinks \Leftarrow 0$
    while $numberOfLinks < m$ do
        $N_{rand} \Leftarrow Randomize(1, N)$ // Pick a random node from the existing network
        $myG \Leftarrow getSubGraph(N_{rand}, \tau_j)$ // Get the subgraph including neighbor nodes of $N_{rand}$ up to $\tau_j$ hops away
        $numberOfLinks \Leftarrow numberOfLinks + PA(myG, N_i, w)$
    end while
}
```

Algorithm 2 Leave Algorithm.

```
Void WeightedLeave($N_{del}$, $\tau_l$, $w$){
    $myG \Leftarrow getSubGraph(N_{del}, \tau_l)$ // Get the subgraph including neighbor nodes of $N_{del}$ up to $\tau_l$ hops away
    $immediateNeighbors \Leftarrow getSubGraph(N_{del}, 1)$ // Get the subgraph including immediate neighbor nodes of $N_{del}$ up to 1 hop away
    $Remove(N_{del})$ //Delete $N_{del}$ from the existing network
    $N \leftarrow N - 1$
    for Each node $N_{im}$ in $immediateNeighbors$ do
        $PA(myG, N_{im}, w)$
    end for
}
```

3.3.3. Weighted leave

Our model also considers the churn created by the leaving nodes and performs a rewiring process in order to maintain the reachability, robustness of the network and to minimize vulnerability to link/node losses. We randomly select a node with a probability of μ and delete it from the network. During this delete/leave operation, we perform a "weighted leave" within the local neighborhood of the leaving node. In particular, the immediate neighbors of the deleted node select a node randomly from set of existing nodes reachable at τ_l hops or fewer from the deleted node and connect to it by applying the rules of our modified PA as described earlier. Similar to the join process, the leave process also does not include the nodes with the degree equal to the hard cutoff, k_c, in the set of nodes which are the neighbors of the deleted node.

A pseudo-code of this "weighted leave" process is given in Algorithm 2. We designed the leave algorithm to address vulnerability to link losses. When a node leaves the topology, its neighbors loose a link. Whenever a node looses a link due to a leaving neighbor, the node recovers that lost link using by rewiring it according a localized PA. This way, node failures/leaves are handled without causing a major harm to the overall performance. We evaluated our approach under high churn rates, where 10% and 30% of the nodes leave the network at a time. The results in Section 4 show that the performance does not suffer significantly even under such high churn rates.

Algorithm 3 Growth Algorithm.

```
Void Grow($N_{target}$, $\mu$, $\tau_j$, $\tau_l$, $w$){
    $i \leftarrow m+1$
    **while** $i < N_{target}$ **do**
      $WeightedJoin(N_i, \tau_j, w)$
      **if** $i == N_{target}$ **then**
          $break$
      **end if**
      // Decide if a node should be removed
      **if** $Random(0,1) < \mu$ **then**
          $N_{del} \Leftarrow Random(1,N)$ // Randomly choose a node to remove to inject churn into the network
          $WeightedLeave(N_{del}, \tau_l, w)$
      **end if**
      $i \leftarrow i+1$
    **end while**
}
```

3.3.4. Growth with churn

To simulate how the P2P topology grows, we devise an algorithm as detailed in Algorithm 3. The growth process continues until a target network size, N_{target}, is reached. We start with a substrate graph of a few nodes, then go into a loop which adds one node and removes a node with a probability of μ at each iteration. To add or remove a node, we call the weighted join and weighted leave procedures. Since $1-\mu$ nodes are added at every iteration, the number of times the loop repeats is

$$H_i = \frac{N_{target} - s}{1-\mu} \tag{4}$$

where the number of nodes in the substrate graph was s. Obviously, the growth process takes longer time if the churn, μ, is larger.

3.4. A sample scenario

The following example gives an idea of how join and leave operations are done in our approach. When a new node enters the system, it makes a modified preferential attachment to a set of nodes which are the neighbors of the randomly selected node, n_{rand}, in the system. In this example, we consider that the join and leave operations are done based on only popularities, i.e., $w = 100$. Let's assume that n_{rand} has a popularity of 40, which is the sum of all the items' popularities residing in n_{rand}. Next, we gather the topology information about the n_{rand}'s neighbors, that are τ_j hops away from n_{rand}. Let's assume τ_j is 1, then, we need to find the neighbors which are one hop away. The number of the neighbors depends on the degree of n_{rand}, since the τ_j value is 1. Let's assume there are 6 neighbors, $n_{1..6}$, with the popularity values $r_{1..6}$ = [20, 80, 30, 60, 110, 50]. The weightage factor of all the nodes are equal to their popularity values since we are assuming $w = 100$ in this example. Otherwise, we would need to apply (3) to calculate the normalized probabilities of attachment to each node in the neighborhood. So, we calculate the probability of attachment based on popularity using (2). For example, the probability of attachment n_5 is 110/(20+80+30+60+110+50+40) = 28%. The lowest attachment probability will belong to n_1, which is 20/(20+80+30+60+110+50+40) = 5%. Using these probabilities, our new node makes connections to these nodes. The number of connections should be higher than minimum degree, m, and lower than hard cut-off, k_c. If somehow, a new node cannot make enough connections to satisfy the minimum degree requirement, then we select another node randomly and repeats the procedures explained above until at least m connections are made.

When a node wants to leave, we first find the neighbors based on τ_l value. Lets use the example above with $\tau_j = 1$ to remove n_{rand}. The neighbors are going to be the same as above which have the popularities $r_{1..6}$. When we remove n_{rand}, some neighbors (the ones that have a connection to n_{rand}) lose one of their connections. In this particular example, all node $n_{1..6}$ will loose a connection. In order to restore these connections, we rewire the lost connection via preferential attachment. For each neighbor $n_{1..6}$ that lost a connection, we try to make a connection to the rest of the neighbors. That is, for n_i, we apply a modified preferential attachment to the rest of the nodes in $n_{1..6} \backslash n_i$. If no connections could be made, then a join operation is done for that node as explained in the previous paragraph.

4. Performance evaluation

We have conducted several tests to see how our popularity-based topology growth performs. It is necessary to know the terms explained in Section 3.3 to understand the results.

4.1. Experiment setup

As it was explained in Section 3.3, we have several parameters that we use to test our topology growth mechanisms. Table 1 shows these parameters and the value ranges we used in our simulations. The growth process starts from a manually configured substrate network. We used a well connected substrate network consisting of 10 nodes with a minimum degree of 3 and a diameter of 3. Further, we varied the churn rate, μ, during the topology growth. The churn rate shows the probability of a node leaving the system. That means, if the churn rate is $\mu = 0.1$, there is 10% probability that one node will leave the topology whenever a new node joins the topology.

In our simulations we took various values for the number of items a node has, N_i, 10, 25, 50 and 100. Ear-

Table 1
Parameters and their value range.

Parameter	Value range
N_{target}	10,000
Churn rate, μ	0.1, 0.3
Minimum degree, m	1, 3
Number of items, N_i	$0 < N_i < 100$
Popularity weightage, w	0, 25, 50, 75, 100
TTL at join, τ_j	1, 2
TTL at leave, τ_l	1, 2
Hard cutoff, k_c	20, 50, 100

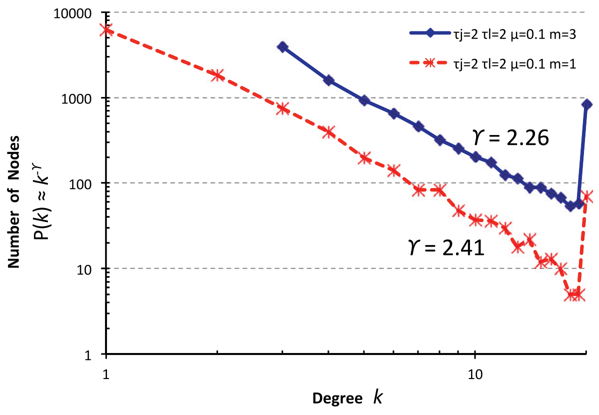

Fig. 3. Effect of minimum degree on degree distribution, where $w = 100$ and $k_c = 20$.

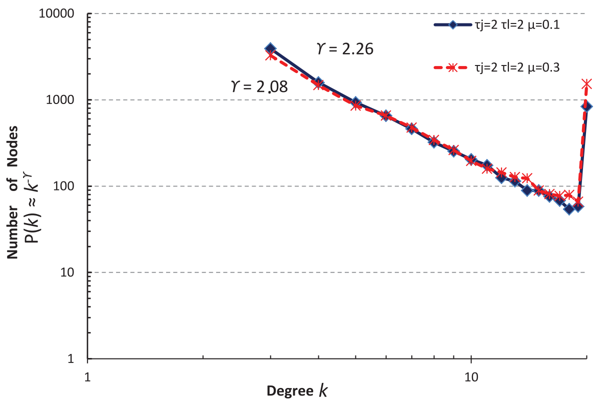

Fig. 4. Effect of churn on degree distribution in P2P topology, where $m = 3$, $w = 100$, and $k_c = 20$.

lier studies [19] showed that popularity can be modeled with the Zipf distribution. Popularities of the items are generated by using the Zipf distribution ranging between 1 and 100. Our simulations had two phases: (i) topology generation and (ii) item search. For topology generation, we repeated the simulations three times, resulting in three different topologies for the same for each combination of parameters in Table 1. Then, we performed searches on the grown topologies using normalized flooding [15].

4.2. Topological behavior: degree distributions

We first observe the degree distributions of topologies grown with our popularity-based method. Fig. 3 shows the effect of minimum degree, m, on degree distributions. The distributions follow a power law trend with a jump on the hard cutoff, $k_c = 20$. The nice feature of these distributions is that the power law trend is essentially maintained even though the nodes are strictly forbidden to have a degree greater than 20. Higher m results in a better constructed P2P topology where the number of low degree nodes is low and the number of high degree nodes is high. This is also observed with the power exponent, γ, for these degree distributions, where lower m yields a smaller γ, i.e., 2.26 versus 2.41. A smaller γ, in this case, means a less steep trend of the power law distribution and thus more nodes having higher degree. Existence of higher degree nodes make the topology more suitable for broadcast-based search, as will observe later too.

We, then, look at the effect of churn rate on the topologies. Fig. 4 shows the churn effect on the degree distribution of P2P topology. Higher churn rates usually causes topology to be unstable due to broken links. In our framework, we handled the churns in a way that broken links are replaced by the new ones by reattaching the immediate neighbors of the leaving node with the topology. As a result, we see better constructed topology under high churn rates in our framework. More churn causes our approach causes to rewire the topology and refine it towards a more power law trend, as observed with a smaller γ for the higher churn rate $\mu = 0.3$.

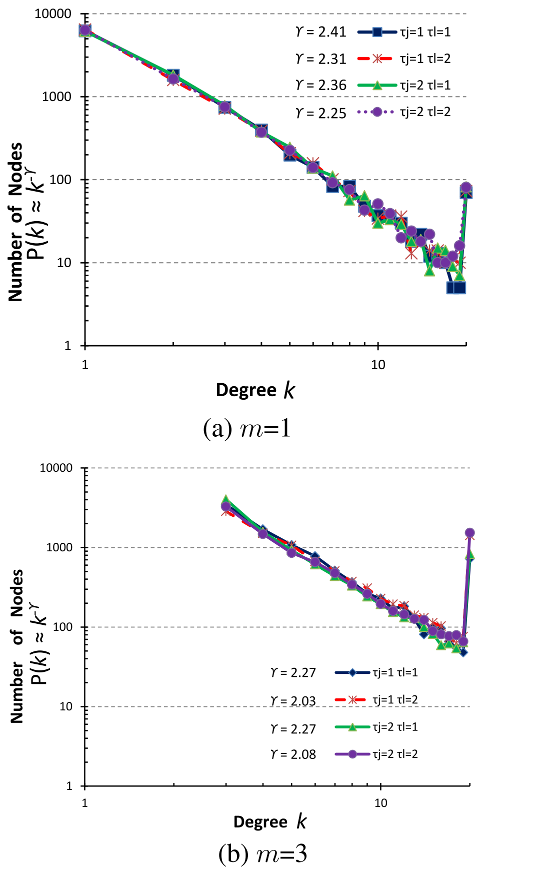

Fig. 5. Degree distribution in P2P topology, where $w = 100$ and $k_c = 20$.

We ran tests to observe the effects of different values of join and leave parameters, i.e., τ_j and τ_l. Fig. 5(a) shows the degree distribution in P2P topology where minimum degree is 1. In this case, we do not see a considerable difference on the degree distribution for different τ_j and τ_l. The γ value stays roughly the same even though τ_j and τ_l increases. The main reason for this is the strict minimum degree constraint which prevents well-constructed topology. However, Fig. 5(b) shows the same tests for the

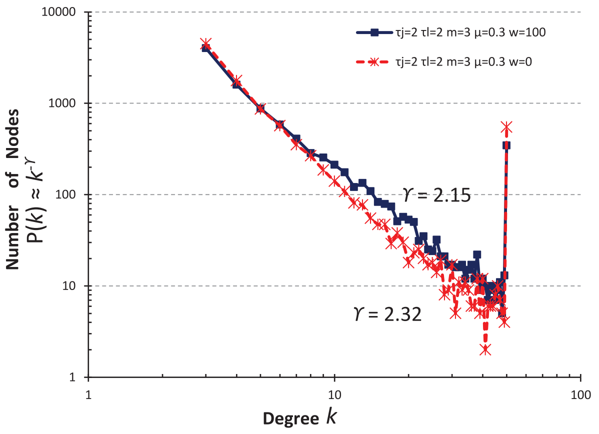

Fig. 6. Comparison of degree- and popularity-based topologies in terms of degree distributions in P2P topology where $k_c = 50$.

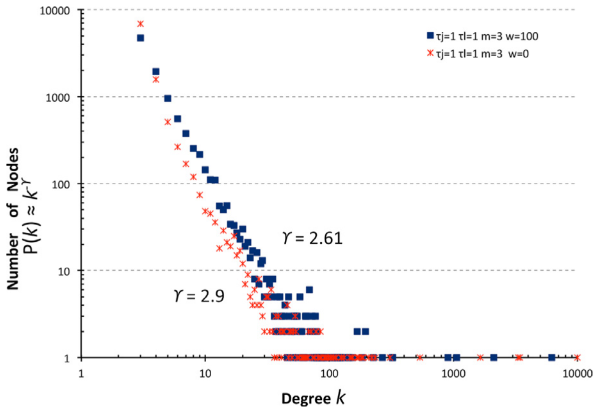

Fig. 8. Comparison of degree- and popularity-based topology construction in terms of degree distribution in P2P topology where there is no cutoff.

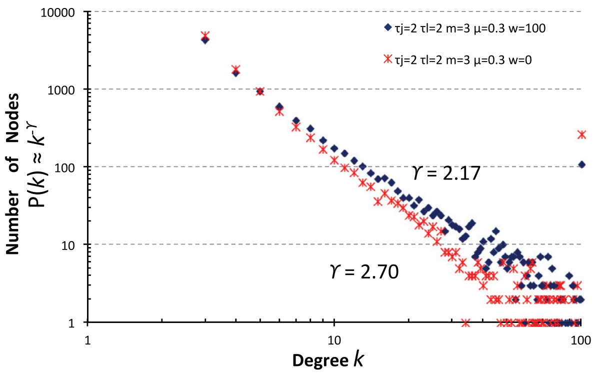

Fig. 7. Comparison of degree- and popularity-based topologies in terms of degree distributions in P2P topology where $k_c = 100$.

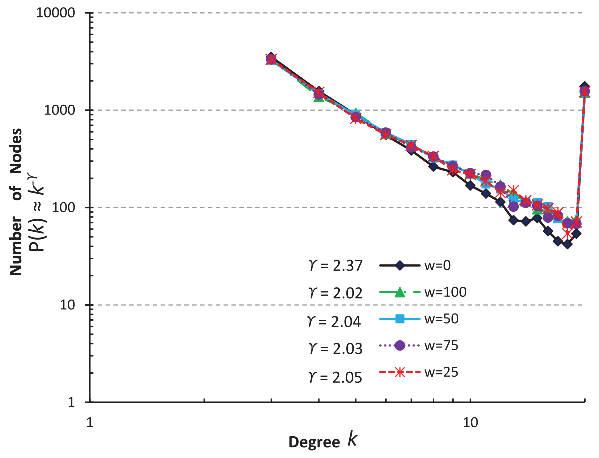

Fig. 9. Comparison of degree distributions based on percentage of popularity and degree, where $\tau_j = 2$, $\tau_l = 2$, $m = 3$, $k_c = 20$, and $\mu = 0.3$.

minimum degree of 3. In this one, we see the effect of join and leave parameters more clearly. We see that the higher the leave parameter, τ_l, the better the topology with a smaller γ. We also see that leave parameter is more effective than the join parameter in terms of topology construction.

In the following tests we have increased the cutoff to see how our framework responds. Fig. 6 shows the degree distributions for the cutoff value $k_c = 50$. We compared degree- and popularity-based constructed topologies in this figure. We see that degrees are better distributed in the popularity-based topology with a smaller γ and more nodes with larger degrees. Fig. 7 compares the same properties for a higher cutoff value $k_c = 100$. As expected, power law trends and a more structured degree hierarchy among the nodes are more observable when the cutoff is larger. As a result of this, the γ values are larger in Fig. 7 than the ones in Fig. 6. Interestingly, in both cases, popularity-based topology growth (i.e., $w = 100$) is clearly better in attaining a power law trend, indicating a topology for better search performance.

Fig. 8 again compares degree- and popularity-based constructed topologies where there is no hard cutoff. Since cutoff is removed and smaller join and leave parameters (τ_j and τ_l) are used, the degree distributions are steeper. However, we still see the similar pattern with the previous tests, meaning that popularity-based topology is better constructed since the degree distribution is better with a smaller γ, more suited for broadcast-based search.

We conducted tests to see how a hybrid approach would perform in constructing a topology. When $w = 0$, our technique results in a topology that is purely based on node degree. On the other extreme, when $w = 100$, the technique uses only the node popularities to grow the topology. We varied w and found that the popularity-only ($w = 100$) case gives the best results. Fig. 9 shows these results where the weightage w is varied. We see that if popularity is given more weight (i.e., larger w) in the topology construction, the resulting topology is better than the one which uses only degrees for topology construction. This is due to smallest the power exponent of the degree distribution for $w = 100$. The smaller power exponent in degree distribution results in a more flat topology as the nodes are more well-connected. According to the literature [17,20], search in lower power exponent topologies are expected to be better. Fig. 9 shows that the number of minimum or close-to-minimum degree nodes is high-

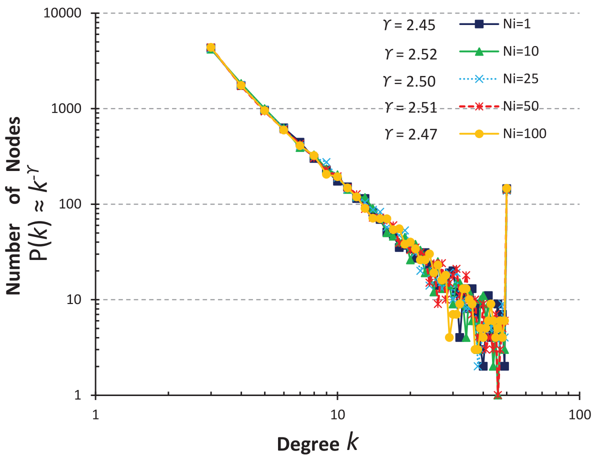

Fig. 10. Effect of maximum item number of each node to the degree distribution where $m = 3$, $k_c = 50$, $\mu = 0.3$, $\tau_l = 2$, and $\tau_j = 2$.

est in the entirely degree-based topology ($w = 0$) since the power exponent is the largest for that case. This is not desirable since the nodes with only few neighbors (i.e., smaller degree) will negatively affect the search performance of the topology. As the node degree increases in Fig. 9, the topologies with higher popularity weight w have more nodes with higher degree, which is indicated by the decreasing power exponent γ. Having high degree nodes results in more connected topology, which eventually increases the search performance. For that reason, we will later experiment with (in Section 4.3) popularity-only ($w = 100$) topology and compared it to the degree-only case ($w = 0$) to observe the improvement in the search performance.

We wanted to see if the number of items on nodes has an effect on topology construction. For that reason, we conducted tests for different maximum number of items, N_i, on nodes. Fig. 10 shows the results of this experiment. We see that maximum number of items on each node does not really affect the topology. So, our popularity-based approach is applicable to nodes with many items.

4.3. Search performance

One of the important factors that indicates the quality of the unstructured P2P topology is the amount of time it takes to find items on that P2P topology. Earlier studies indicate very close relevance between the search performance and the characteristics of the underlying topology [1,2,17,19]. In order to compare our popularity-based P2P topology with the degree-based P2P topology, we conducted searches using normalized flooding [15] and observed the performance in terms of search coverage, query response time and messaging overhead. We further looked at the performance of our approach in finding a popular item.

In order to make a more granular observation of the the effect of the item popularity on the search performance, we distributed the item popularity according to Zipf ranging between 1 and 500. We repeated the search simulations 10 times, resulting in 10 different topologies for each case. We performed the topology growth with $m = 3$, $\mu = 0.3$, $\tau_l = 2$, and $\tau_j = 2$. We grew the topologies with two cutoff values, $k_c = 100$ and $k_c = 100$, and no cutoff. Then, we repeated the search using normalized flooding (NF) 50 times with two different normalization parameters: NF2 and NF3. In NF2, a node forwards the search query to 2 of its neighbors while in NF3 to 3 neighbors. Since the minimum degree of the topologies we grew were $m = 3$, we did not simulate NF with normalization parameter greater than 3. We stopped the NF process at 150 hops, i.e., the TTL for the search queries was set to 150.

4.3.1. Coverage

A crucial indicator of the search performance in unstructured P2P systems is the number of distinct nodes a search query can reach to after a fixed number of hops. Therefore, we counted the number of distinct nodes visited for different TTL values on both topologies. The results, as in Fig. 11, clearly show that the number of distinct nodes visited by our popularity-based topologies is sizably better. The popularity-based approach outperforms more in NF2 than NF3, which shows that the topology shapes into a more diverse form where fewer number of search queries can pay back more. Further, the popularity-based approach outperforms more as the cutoff increases. When there is no cutoff, the difference is very large, e.g., the number of distinct nodes in the popularity-based approach is about 38% more (7071 vs. 9729) 126% more (3522 vs. 7967) in NF3 and NF2, respectively, when TTL is 20. The fact that the benefit of the popularity-based approach reduces as cutoff gets smaller shows that the cutoff on the

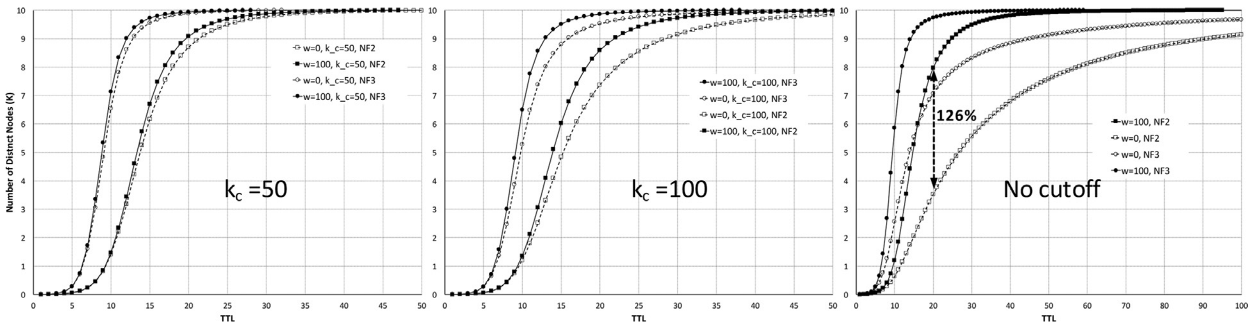

Fig. 11. Number of distinct nodes reached in Normalized Flooding for popularity-based ($w = 100$) and degree-based ($w = 0$) topologies. ($m = 3$, $\mu = 0.3$, $\tau_l = 2$, and $\tau_j = 2$).

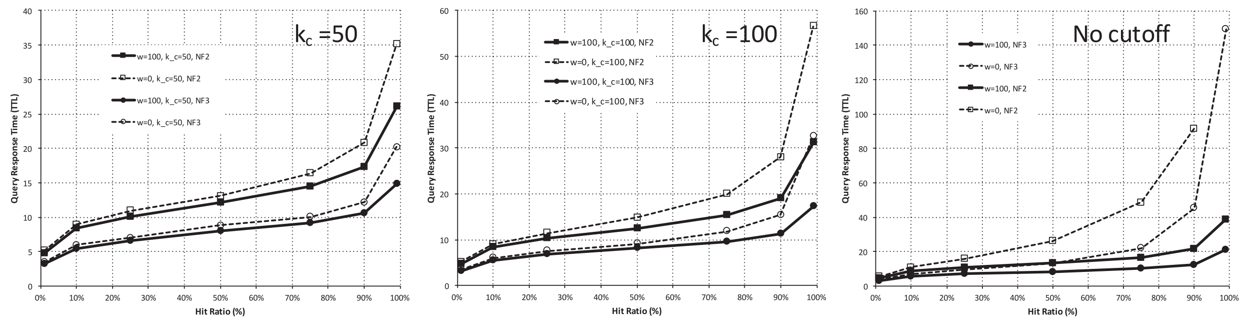

Fig. 12. Query response time (QRT) in Normalized Flooding for popularity-based ($w = 100$) and degree-based ($w = 0$) topologies. ($m = 3$, $\mu = 0.3$, $\tau_l = 2$, and $\tau_j = 2$).

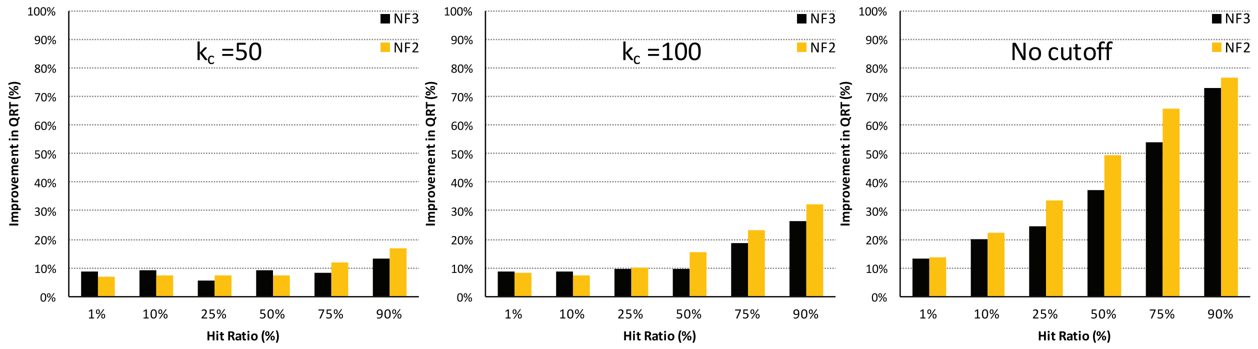

Fig. 13. Improvement in QRT due to the popularity-based approach. ($m = 3$, $\mu = 0.3$, $\tau_l = 2$, and $\tau_j = 2$).

node degree dominates and forces the topology to be more flat. This means the degree-based and popularity-based approaches end up producing similar topologies. Given that the cutoff (i.e., the maximum degree of a node) in P2P systems and social networks can easily grow to 500 or more, the benefit of the popularity-based approach will be more profound in practice.

4.3.2. *Query Response Time (QRT)*

Although the coverage of distinct visited nodes in a search indicates the overall performance of a search, a more detailed look at the query response time (QRT) (i.e., the amount of time the first response to a search query) is needed. So, we explored the benefit of the popularity-based approach in terms of QRT. We measured QRT in terms of the number of hops (i.e., TTL) it takes to reach an item. Since NF is a flooding-based search technique, finding an item in the topology requires the search query (with the address of the query's originator included) to reach a node with the item. Once the node with the item receives the query, it can contact the query originator directly or send the item via the path the search query came through. Either way, the QRT will be driven by the number of hops it takes to reach an item. Thus, we evaluate the QRT performance based on TTL.

We used the Zipf distribution of the items' popularity for determining the frequency of queries on an item. Fig. 12 shows average TTL it takes to reach 1%, 10%, 25%, 50%, 75%, 90%, and 99% of the items for three different cutoff settings. The popularity-based approach consistently improves QRT, but with a more pronounced improvement when cutoff is larger. This is in line with the performance in terms of the number of distinct nodes observed in Fig. 11. Fig. 13 shows this more clearly in terms of the percentage improvement the popularity-based approach attains in QRT. An interesting result emerging from Fig. 13 is that the popularity-based approach helps much more as the expected probability of finding an item increases. Since the query distribution over the items is skewed (with Zipf), more of the queries go to popular more items. Thus, the popularity-based approach's help becomes more obvious as more hit probability is expected from the queries.

Fig. 14 shows the hit ratio versus the popularity of items under three cutoff settings. The results are shown at three TTL values 8, 12, and 16. While the popularity-based topologies ($w = 100$) result in higher hit probabilities for more popular items, the hit ratios in the degree-based topologies ($w = 0$) are unaffected by the items' popularity. This result is intuitive and clearly shows that the popularity-based topology generation works as expected. Further, the difference in the hit ratios become more obvious as TTL gets larger in NF2. Yet, beyond a particular TTL, the benefit of the popularity-based topology generation diminishes. Since NF3 explores more of the topology within a small TTL, the benefit of the popularity-based approach gets smaller quickly as TTL increases in NF3. It is also observable that the popularity-based approach results in smaller hit probabilities for unpopular items, but it improves the QRT (Fig. 13) by pulling the nodes with more popular items towards the center of the topology. This is good tradeoff in improving the search performance in an

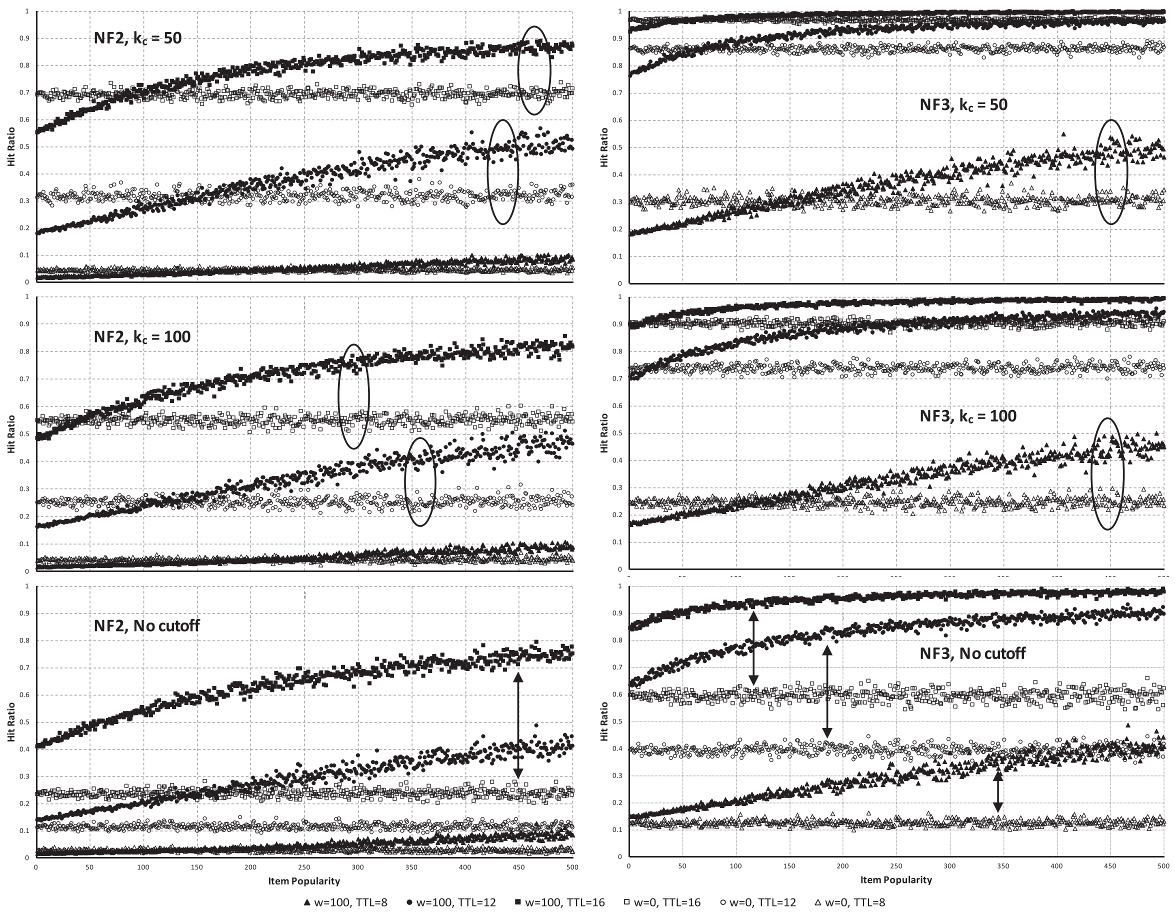

Fig. 14. Hit ratio against the popularity of items at TTL values 8, 12, and 16. ($m = 3$, $\mu = 0.3$, $\tau_l = 2$, and $\tau_j = 2$).

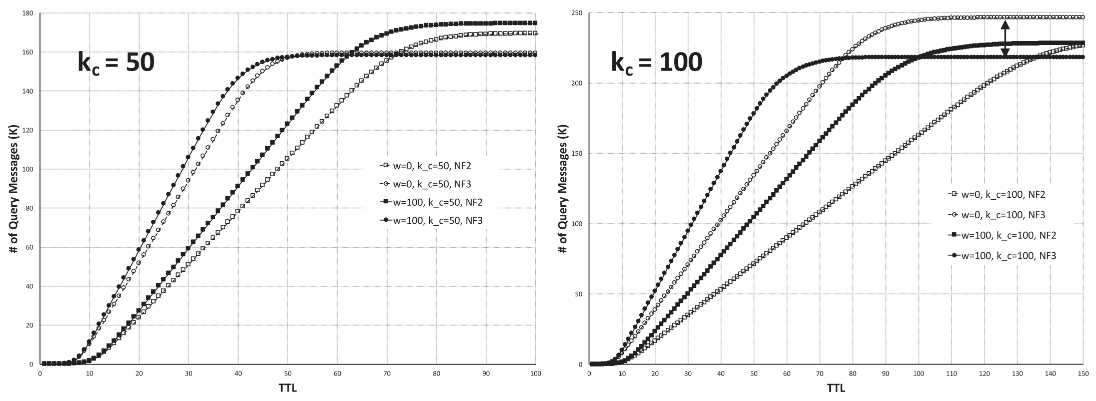

Fig. 15. Number of query messages for cutoffs $k_c = 50$ and $k_c = 100$. ($m = 3$, $\mu = 0.3$, $\tau_l = 2$, and $\tau_j = 2$).

unstructured P2P system where average performance matters more than finding a rare item.

4.3.3. Messaging overhead

Since it is not possible to perform to-the-point searches in unstructured P2P systems, the traditional approaches are flooding-based search techniques such as NF. If the number of query messages being flooded around is not kept under control, these flooding-based searches can cause a significant amount of messaging overhead. During our simulations, we counted the number of query messages that have been sent since the beginning of a search. Fig. 15 shows the number of messages sent in NF2 and NF3 in the popularity-based and the degree-based topologies

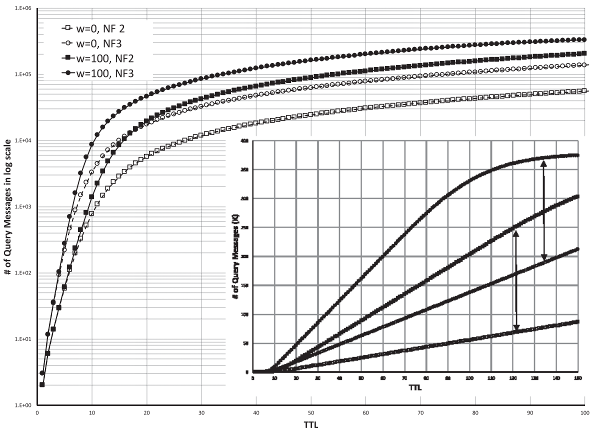

Fig. 16. Number of query messages when there is no cutoff. ($m = 3$, $\mu = 0.3$, $\tau_l = 2$, and $\tau_j = 2$).

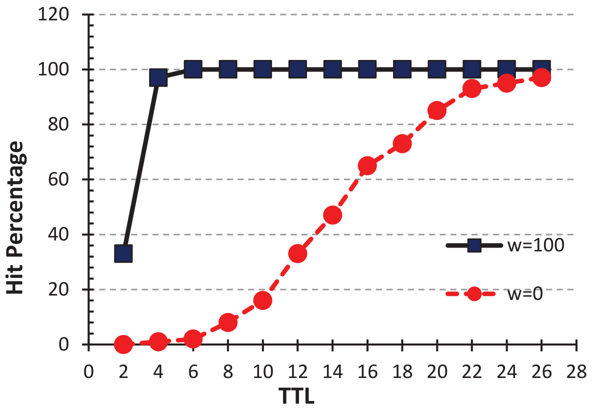

Fig. 17. Hit percentage of a highly popular item. ($m = 3$, $\mu = 0.3$, $\tau_l = 2$, and $\tau_j = 2$).

when the cutoff is $k_c = 50$ and $k_c = 100$. The popularity-based approach causes more messages to be sent around earlier in the search but later converges to a similar count towards the end of the search as TTL grows larger. The same result appears in Fig. 16, when there is no cutoff on the node degree. The log scale plot shows that the popularity-based and the degree-based approaches eventually converge to a similar messaging overhead, but, for small TTLs (which are more realistic), the difference can be as large as 3–4 times against the popularity-based approach. This is due to the fact that the popularity-based approach explores the topology faster in terms of the number of distinct nodes (Fig. 11). In return of this "early" overhead the QRT is significantly smaller (by 50% or more as shown in Fig. 13) than the degree-based approach. This is a desired tradeoff for unstructured P2P systems where response time is more important than the number of query messages flooding in the network at an instant. As long as the eventual number of query messages is under control, which seems to be roughly similar to the degree-based case, early flooding is not going to cause noticeably harm to the P2P system's performance.

4.3.4. Finding a popular item

Another search performance criteria that needs to be taken into account is whether a search is successful for a specific item or not. This criterion is also related with the previous one since the number of visited nodes is directly related with the success of the search. We ran the tests in degree-based and popularity-based topologies where both topologies have no hard cutoff. We searched for a specific item which has the highest popularity in the topology. Here, we wanted to show how our topology favors an extremely popular item after entering the topology. In order to show our topology's capabilities, we removed the cutoff constraint in this test. The node that has this special item becomes very popular and gravitates towards the center of the topology in a very short time. The node's degree grows very quickly; and therefore, it will be very easy to find this item. Having a high degree might overload the node in practice. For the purposes of observing the growth in the node's degree, we assumed that the nodes can handle large degrees and have not considered a particular overload/degree threshold as it depends on many factors such as the bandwidth, CPU and memory.

As it can be seen in Fig. 17, proposed topology is much better than the degree-based topology in locating popular items. The success rate of the popularity-based topology is almost 100% for TTL value of 4 and above. This result shows that our topology favors highly popular items so that they can be found easily in the system. The key takeaway is that popularity-based topology growth, if done carefully, can enable significant improvements in the search of popular items (Fig. 17) while providing sizable improvements in the search of any item (Fig. 11).

4.4. Catching trends and TBP items

We conducted tests to see how our popularity-based topology handles the nodes that contain to-be-popular (TBP) or high popularity items. It is expected that the P2P topology adapts itself so that the nodes with popular items move to towards the "center" of the topology, as was sketched in Fig. 1. So, when everything is settled the P2P topology should have popular items at its center since the items around the center will be easier to find. To the best of our knowledge, existing systems do not use an unstructured P2P topology growth/adaptation approach to handle such TBP items. Most of the existing solutions use cloud-based centralized management of items' popularity and their corresponding search performance. In our approach, we are exploring what is possible by a pure P2P system employing topology adaptation to handle trends or memes.

4.4.1. Highly popular items

We experimented with various scenarios involving highly popular items and observed the hit ratio and degree of the node holding these highly popular items. To observe the increase/decrease trends in the node degrees, we did not enforce a hard cutoff on the node degrees in these experiments.

Entering for the first time: We wanted to see how long it takes for our topology growth process to recognize a

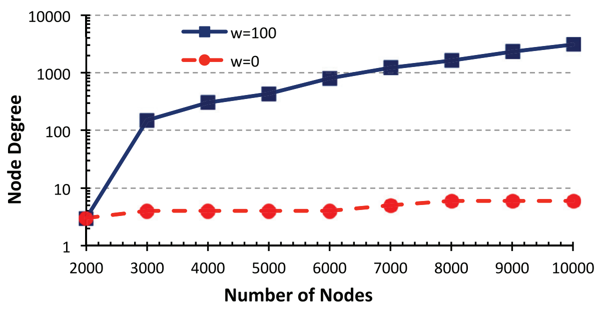

Fig. 18. Temporal changes in the degree of a node holding a highly popular item entering for the first time. ($m = 3$, $\mu = 0.3$, $\tau_l = 2$, and $\tau_j = 2$).

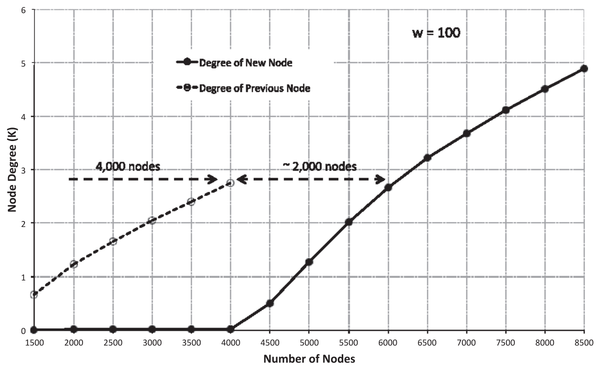

Fig. 19. Degree of the previous and the new nodes of a moving highly popular item. ($m = 3$, $\mu = 0.3$, $\tau_l = 2$, and $\tau_j = 2$).

highly popular item that just entered the system and gravitate it towards the center of the topology. This resembles to the earlier phases of a new trend or meme in social networks. In order to show that, we inserted an item, that has the highest popularity in the topology, to a leaf node when there are 2000 nodes in the system. We then checked the degree of that node as topology grows. Fig. 18 shows changes in the degree of the node, containing the item that has the highest popularity in the topology, as topology grows. As it can be seen in the figure, degree of that node sharply increases as the topology grows and reaches to slightly above 3000. On the other hand, if we construct the topology using degrees, we see that the degree of that node almost does not change at all, meaning that it will not favor the popular items.

Moving to a neighbor node: Robustness to dynamic changes is crucial in P2P system performance as the peer nodes can leave at will. We looked at the scenario where a node holding a highly popular item leaves. We assumed that one of the neighbors of the leaving node, which we call the "previous node", takes over the highly popular item as the "new node". We observed the degree of the previous and the new nodes for the highly popular item. As shown in Fig. 19, the node holding the highly popular item leaves the system when the number of nodes is 4000. Until then, the new node's degree stays mostly constant. But, when the move happens, the new node's degree quickly increases. It takes about an increase of 2000 nodes (i.e., a relative measure of time) during the

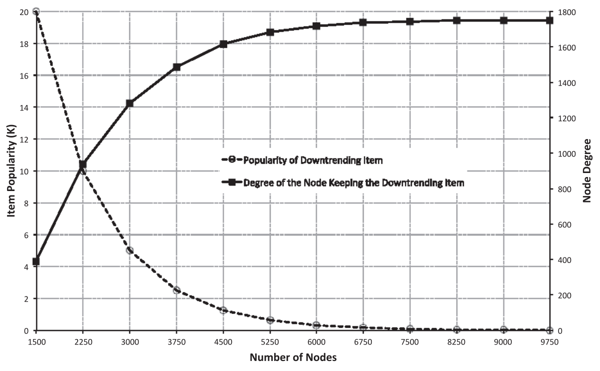

Fig. 20. Degree of the node holding a highly popular downtrending item. ($m = 3$, $\mu = 0.3$, $\tau_l = 2$, and $\tau_j = 2$).

growth process to recover the degree in the previous node, which is about 2700. This is relatively a good performance given that the previous node took about 4000 nodes of time in the growth process. Note that the degree-based approach (i.e., $w = 0$) does not adapt the node degrees based on popularity and this moving scenario would cause the highly popular item to stay in a very low degree node similar to the result in Fig. 18.

Downtrending: Another interesting scenario is a "flash crowd" style increase followed by a fast decrease in the popularity of an item. To observe the performance of our popularity-based approach, we inserted a highly popular item (with popularity value 20K) to a leaf node and then halved its popularity at every 750 nodes increase in the topology. Fig. 20 shows increase in the degree of the node holding the downtrending highly popular item. The node degree saturates around 1750 when the popularity of the downtrending item goes below 1K. This is an expected result and verifies the intuition that the popularity-based approach respects the popularity of the items; and hence, stops increasing the degree of the node as the popularity of the items on it diminishes. An important property of our approach here is that it does not reduce the node's degree even after the popularity of the item on it becomes miniscule. This is because our join and leave algorithms gravitate the nodes towards the center based on their popularity, but do not push them away. So, in that sense, our node gravitation is a one directional "force". Introducing the other repulsive direction is possible as a future work, but requires careful balancing of these two forces. We kept our design simple and assumed that nodes will only move away from the center under two circumstances: (i) a higher popularity node arrives and gets placed at a more central position and this indirectly pushes the previously central nodes away, and (ii) the central nodes simply leave the system.

4.4.2. TBP item

We experimented with a TBP item which has gradually increasing popularity. Since the experiments above already investigated abrupt increases or decreases in item popularity, we focused on gradual increase of popularity in this experiment. As the topology grows to 10K nodes, we linearly increased a specific TBP item's popu-

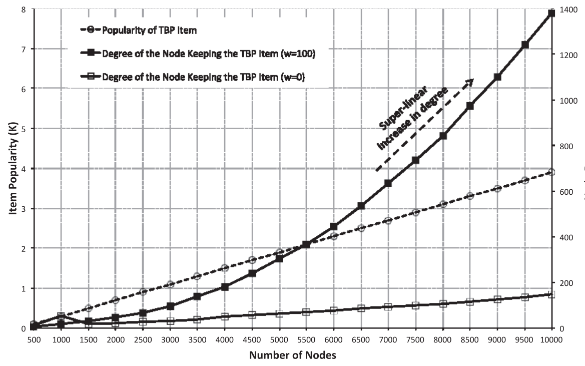

Fig. 21. Degree of the node holding a TBP item. ($m = 3$, $\mu = 0.3$, $\tau_l = 2$, and $\tau_j = 2$).

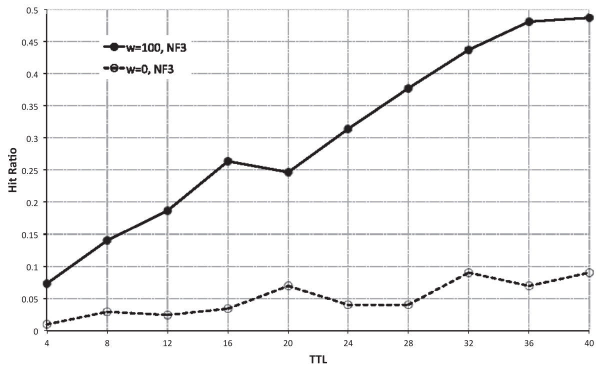

Fig. 22. Hit ratio for a TBP item. ($m = 3$, $\mu = 0.3$, $\tau_l = 2$, and $\tau_j = 2$).

larity with a rate of 200 units at every 200 nodes increase in the topology size. We, then, observed how the degree of the node holding this TBP item grows. As Fig. 21 shows, the popularity-based approach ($w = 100$) increases the node's degree super-linearly while the degree-based approach ($w = 0$) makes a small linear increase. The outcome that the popularity-based approach reacts with a super-linear increase in the node's degree to a linear increase in the node's popularity is very promising. In particular, for TBP items that may start with a sluggish increase in their popularity, the search performance will be very good. Fig. 22 verifies this intuition. It shows the hit ratio for searching the TBP item using NF3 after the topology is fully grown to 10K nodes. Hit ratio for the TBP item in the popularity-based topology is 5 times more than the degree-based topology.

5. Related work

P2P networking has generated tremendous interest across worldwide among both the Internet users and networking professionals. It can be explained as the "type of network in which each peer has equivalent/similar capabilities and responsibilities". It totally differs from the client/server architectures, in which some of the peers are dedicated for serving others [21].

P2P networks inherit three important properties: self-organization, symmetric communication and distributed control. A self-organizing P2P network automatically adapts and configures to the join, leave and failure of peers. It has the capability to survive the churn in the network. Symmetric communication relates to the fact that each peer in the network acts as both client and server. P2P networks have no centralized point of control as the devised protocols are completely distributed. The body of P2P research can be divided into several categories including topology generation, storage, search, security and applications. In terms of organization of nodes, the P2P systems can be-classified into three different categories [19]: centralized, distributed but structured and decentralized and unstructured. In the centralized P2P systems there is a central directory server which users can submit queries. This central server is used for indexing functions and to bootstrap the entire system-, while interactions among peers are still strictly peer-to-peer. Search in such centralized P2P systems is typically very fast as the server(s) undertake the job of maintaining the list of available keys across the peers participating in the network. Although this has similarities with a structured architecture, the connections are not determined by any algorithm and are left to the underlying network i.e. the Internet. These types of P2P networks provide most efficient and comprehensive search possible for the requested files but have single point of failure and have scalability problems.[22] The very first prominent and popular peer-to-peer file sharing system, Napster, was an example of the centralized model.

Other P2P systems are decentralized and have no central server. The hosts form an ad hoc network among them and send their queries to their peers. Of these decentralized designs, some are structured in that they have close coupling between the P2P network topology and the location of data for a sampling of these designs. Distributed Hash Tables (DHTs) [23], Chord [23] and CAN [24] are examples of such distributed and structured P2P systems.

Decentralized and unstructured P2P networks have no centralized directory or any control over the topology of the network. Furthermore, file/item placement is random in these types and no strict or loose enforcement of these placements to the topological structure exists. Unstructured P2P networks are heavily used in today's Internet, mostly for file and content sharing. Freenet [25] and early implementations of the Gnutella protocol are examples of the decentralized and unstructured model. In large scale and highly dynamic unstructured P2P systems, resource location is inefficient due to lack of knowledge about the destination [26]. Search method used in these kinds of systems is blind flooding or random walk [27], which does not provide any guarantee in finding an item existing on the system. On the plus side, such networks can be easily constructed as a new peer that wants to join the network can copy existing links of another node and then form its own links over time.

Designing P2P systems with an awareness of content popularity has been considered in various contexts. In [28,29], authors constructed a hybrid system which combines structured and unstructured P2P systems. In order to increase search success rate, they insert to insert items into the system based on their popularity. By placing popular items at more connected and reachable nodes, they aim to attain higher search performance. Their work optimizes the search on an already constructed topology; whereas, we attempt to make the item popularity a first-class citizen of the P2P system and construct and update the topology by explicitly considering the popularity.

Another design approach that got attention by researchers is smart indexing of the content according to its popularity so that the search on P2P systems is improved. Hybrid Adaptive Search [30]using a gossip exchange method was proposed for unstructured P2P overlays. It constructs a hybrid overlay that efficiently combines topology awareness and interest-based links instead of random or DHT invoked connections. Each peer maintains an index table of files with the fields of content index and age, in which popularity of an item defined by its age. Following on a similar vein, a DHT-based partial indexing scheme, which improves the chance of finding unpopular items was proposed in [31]. In this scheme, the overlay network is constructed based on peers' interests. Another work [32] presents a strategy to prune routing indices based on popularity of resources on an already constructed P2P topology. In another work [33], content replication method that prioritizes popularity, is proposed in order to improve network traffic and storage consumption. The main difference of our work from these prior proposals is to adapt the P2P topology according to content popularity.

Understanding popularity of items and content has attracted quite a lot of recent interest from the research community. In particular, there is an increased amount of research on online social networks in terms of trends and memes. Cha et al. [34] tried to measure user influence on Twitter. Asur et al. [3] studied trending topics in Twitter in terms of persistence and decay. They found that the resonance of the content with the users of the social network plays a major role in causing trends. Yang and Leskovec [35] studied temporal patterns associated with online content and how the content's popularity grows and fades over time in Twitter. Characterization of emerging trends in Twitter is studied by Naaman et al. [36]. They did their study on a large dataset of Twitter messages belonging to a specific geographic area and identified important dimensions according to which trends can be categorized. Detection of trending topics on Twitter can be done by TwitterMonitor [37] using Twitter stream.

6. Conclusions and future work

In this paper, we introduced a popularity-based unstructured P2P topology design that use local information unlike traditional scale-free topology generation mechanisms using global topology information. We conducted tests to see how it compares with degree-based P2P topology growth. We observed that degree distribution in popularity-based P2P topology is better than the degree-based P2P topology in terms of power law and scale-free behavior. Attesting this observation, the test results show that the search performance is better in our design. We also incorporated item popularity into our overlay topology design without compromising scalability in order to treat popular items or to-be-popular (TBP) items better. We showed that TBP items are treated better in our popularity-based overlay topology so that they can be found easily in the topology.

Several future research directions are possible. We are currently using static popularities for the items in the system, given by the user. In order to prevent malicious intent, a reputation system like EigenTrust [38] can be used. Another future work that could be done is to incorporate dynamic popularity for the items into our design. The item popularity can be increased or decreased based on several factors including search potential, time, and replication. We believe that replication of the resource items that a new node have will slow down its movement towards the center of the network since the nodes that copied its resources will also move towards the center.

We modeled item popularity using Zipf distribution based on prior studies [19]. However, a more custom and dynamic model should be designed based on real applications as well as topological dynamics. For instance, a tweet's popularity and a Facebook post's popularity can be modeled differently. While the former is mainly driven by the views, the latter may better be modeled by the popularity of the person who posted it. Further, topological properties may have significant effect on the popularity modeling. In addition to the number of queries for an item, factors like the location on the topology and the degree of the node hosting the item should be considered while developing a more fine-grained model for item popularity. Our technique will be more successful if such application-specific item popularity models are developed.

What we have done is an initial step in laying down what is possible with a pure P2P design without using any centralized management. It will be interesting to see how close this type of popularity-based P2P topology adaptation and growth can be to the cloud-based centralized management of trends and memes in social networks. For instance, a node in our system could be made to periodically rewire itself and apply the modified preferential attachment at every rewire. As it is, we are making the nodes apply preferential attachment only when they join or leave. So, if the frequency of a node rewiring itself is tuned, the time it takes for the P2P topology to adapt itself to cater the popular or TBP items better may be reduced. An interesting future work would be to explore tradeoff between the gains possible with such a design and the messaging overhead due to increased rewiring.

Finally, our approach adjusts nodes' degrees using item popularities as a parameter for preferential attachment. This is an indirect approach and does not directly update the nodes' connections (i.e., the degree) to their neighbors based purely on the item popularities. A more direct update of the nodes' degrees with respect to item popularity may prove to be better. However, such a direct update of the node degrees could make the topology too heavily dependent on the item popularity model. If the popularity of an item is calculated with a significant error, it may cause disruptive changes in the topology too quickly. The design of such a scheme requires careful handling of this particular issue.

References

[1] E.K. Lua, J. Crowcroft, M. Pias, R. Sharma, S. Lim, et al., A survey and comparison of peer-to-peer overlay network schemes., IEEE Commun. Surv. Tutor. 7 (1-4) (2005) 72–93.
[2] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (5439) (1999) 509–512.
[3] S. Asur, B.A. Huberman, G. Szabo, C. Wang, Trends in social media: persistence and decay., in: ICWSM, 2011.
[4] M. Mathioudakis, N. Koudas, Twittermonitor: trend detection over the twitter stream, in: Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, ACM, 2010, pp. 1155–1158.
[5] J. Leskovec, L. Backstrom, J. Kleinberg, Meme-tracking and the dynamics of the news cycle, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD '09, ACM, New York, NY, USA, 2009, pp. 497–506, doi:10.1145/1557019.1557077.
[6] H. Guclu, M. Yuksel, Scale-free overlay topologies with hard cutoffs for unstructured peer-to-peer networks, in: Proceedings of the 27th International Conference on Distributed Computing Systems, 2007. ICDCS'07., IEEE, 2007.32–32
[7] M. Ripeanu, Peer-to-peer architecture case study: Gnutella network, in: Proceedings of International Conference on Peer-to-Peer Computing, 2001, pp. 99–100.Linkoping, Sweden
[8] C. Xie, G. Chen, A. Vandenberg, Y. Pan, Analysis of hybrid P2P overlay network topology, Comput. Commun. 31 (2) (2008) 190–200.
[9] C. Xie, Y. Pan, Analysis of large-scale hybrid peer-to-peer network topology, in: Proceedings of IEEE GLOBECOM, 2006, pp. 1–5.San Francisco, CA
[10] A. Malatras, State-of-the-art survey on P2P overlay networks in pervasive computing environments, J. Netw. Comput. Appl. 55 (2015) 1–23.
[11] D. Stutzbach, R. Rejaie, S. Sen, Characterizing unstructured overlay topologies in modern P2P file-sharing systems, IEEE/ACM Trans. Netw. 16 (2) (2008) 190–200.
[12] G.H.L. Fletcher, H.A. Sheth, K. Börner, Agents and Peer-to-Peer Computing, in: Lecture Notes in Computer Science, vol. 3601, Springer Berlin Heidelberg, 2005, pp. 14–27.
[13] L. Li, D. Alderson, W. Willinger, J. Doyle, A first principles approach to understanding the internet's router-level topology, in: Proceedings of ACM SIGCOMM, 2004.
[14] Centrality, (Wikipedia).
[15] C. Gkantsidis, M. Mihail, A. Saberi, Hybrid search schemes for unstructured peer-to-peer networks, in: Proceedings IEEE INFOCOM, vol. 3, 2005, pp. 1526–1537, doi:10.1109/INFCOM.2005.1498436.vol. 3
[16] D. Kumari, H. Guclu, M. Yuksel, Ad-hoc limited scale-free models for unstructured peer-to-peer networks, Peer-to-Peer Netw. Appl. 4 (2) (2011) 92–105.
[17] H. Guclu, M. Yuksel, Limited scale-free overlay topologies for unstructured peer-to-peer networks, IEEE Trans. Parallel Distrib. Syst. 20 (5) (2009) 667–679.
[18] R. Albert, A.-L. Barabási, Statistical mechanics of complex networks, Rev. Modern Phys. 74 (1) (2002) 47.
[19] Q. Lv, P. Cao, E. Cohen, K. Li, S. Shenker, Search and replication in unstructured peer-to-peer networks, in: Proceedings of the 16th international conference on Supercomputing, ACM, 2002, pp. 84–95.
[20] R. Cohen, S. Havlin, Scale-free networks are ultra-small, Phys. Rev. Lett. 90 (5) (2003) 058701–058705.
[21] R. Rodrigues, P. Druschel, Peer-to-peer systems, Commun. ACM 53 (10) (2010) 72–82, doi:10.1145/1831407.1831427.
[22] Gnutella Inc., http://www.gnutella.com/, 2004,
[23] I. Stoica, R. Morris, D. Karger, F. Kaashoek, H. Balakrishnan, Chord: a scalable Peer-To-Peer lookup service for internet applications, in: Proceedings of ACM SIGCOMM, 2001, pp. 149–160.
[24] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker, A scalable content-addressable network, in: Proceedings of ACM SIGCOMM, 2001.
[25] Freenet Project, http://freenetproject.org.
[26] H. Mashayekhi, J. Habibi, Combining search and trust models in unstructured peer-to-peer networks, J. Supercomput. 53 (1) (2010) 66–85, doi:10.1007/s11227-009-0341-2.
[27] Y. Liu, L. Xiao, X. Liu, L. Ni, X. Zhang, Location awareness in unstructured peer-to-peer systems, IEEE Trans. Parallel Distrib. Syst. 16 (2) (2005) 163–174, doi:10.1109/TPDS.2005.21.
[28] M.M. Kamran, S. Khorsandi, A popularity-based globally structured hybrid peer-to-peer network, in: Proceedings of the International Conference on Information Networking and Automation (ICINA), 2010, vol. 1, IEEE, 2010, pp. V1–411.
[29] M.M. Kamran, S. Khorsandi, Popularity estimation in a popularity-based hybrid peer-to-peer network, in: Proceedings of the 13th International Conference onAdvanced Communication Technology (ICACT), 2011, IEEE, 2011, pp. 399–404.
[30] L. Sharifi, S. Khorsandi, A popularity-based query scheme in p2p networks using adaptive gossip sampling, Peer-to-Peer Netw. Appl. 6 (1) (2013) 75–85.
[31] R. Zhang, Y.C. Hu, Assisted peer-to-peer search with partial indexing, IEEE Trans. Parallel Distrib. Syst. 18 (8) (2007) 1146–1158.
[32] S. Bressan, A.N. Hidayanto, Z.A. Hasibuan, Exploiting local popularity to prune routing indices in peer-to-peer systems, in: Proceedings of the Sixteenth International Workshop on Database and Expert Systems Applications, 2005, IEEE, 2005, pp. 790–795.
[33] Y. Kawasaki, N. Matsumoto, N. Yoshida, Popularity-based content replication in peer-to-peer networks, in: Computational Science–ICCS 2006, Springer, 2006, pp. 436–443.
[34] M. Cha, H. Haddadi, F. Benevenuto, P.K. Gummadi, Measuring user influence in twitter: the million follower fallacy., ICWSM 10 (2010) 10–17.
[35] J. Yang, J. Leskovec, Patterns of temporal variation in online media, in: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, in: WSDM '11, ACM, New York, NY, USA, 2011, pp. 177–186, doi:10.1145/1935826.1935863.
[36] M. Naaman, H. Becker, L. Gravano, Hip and trendy: characterizing emerging trends on twitter, J. Am. Soc. Inf. Sci. Technol. 62 (5) (2011) 902–918.
[37] M. Mathioudakis, N. Koudas, Twittermonitor: trend detection over the Twitter stream, in: Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, ACM, 2010, pp. 1155–1158.
[38] S.D. Kamvar, M.T. Schlosser, H. Garcia-Molina, The eigentrust algorithm for reputation management in p2p networks, in: Proceedings of the 12th international conference on World Wide Web, ACM, 2003, pp. 640–651.

Gurhan Gunduz received the BS degree in computer engineering from Ege University, Izmir, Turkey, and the MS and PhD degrees in Computer Science from Syracuse University, Syracuse, NY. He is an assistant professor at Pamukkale University, Denizli, Turkey. His research is on computer networks, distributed systems and their applications.

Murat Yuksel is an Associate Professor at the CSE Department of The University of Nevada - Reno (UNR), Reno, NV. Prior to joining UNR, He was with the ECSE Department of Rensselaer Polytechnic Institute (RPI), Troy, NY as a Postdoctoral Research Associate and a member of Adjunct Faculty until 2006. He received a B.S. degree from Computer Engineering Department of Ege University, Izmir, Turkey in 1996. He received M.S. and Ph.D. degrees from Computer Science Department of RPI in 1999 and 2002 respectively. His research interests are in the area of networked, wireless, and computer systems with a focus on big-data networking, optical wireless, network management and optimization, cloud-assisted routing, network architectures and economics, and peer-to-peer. He has been on the editorial board of Computer Networks, Elsevier. He published more than 100 papers at peer-reviewed journals and conferences and is a co-recipient of the IEEE LANMAN 2008 Best Paper Award. He is a senior member of IEEE, life member of ACM, and was a member of Sigma Xi and ASEE.