Adaptive Probabilistic Search in Complex Dynamic Networks

Spiridoula V. Margariti

Department of Computer Engineering
T.E.I. of Epirus
Arta, Greece, GR-47100
Email: smargari@cse.uoi.gr

Vassilios V. Dimakopoulos

Department of Computer Science and Engineering
University of Ioannina
Ioannina, Greece, GR-45110
Email: dimako@cse.uoi.gr

***Abstract*—Probabilistic flooding search is a fundamental technique for a wide range of complex systems. However, search over these systems is quite challenging due to their dynamic and complex nature, which results from the interactions between participants. Here, we propose A2PF, an adaptive probabilistic search scheme, which is capable of adjusting its operation in dynamic network environments. It works in a distributed manner and each participating node exploits past query messages to reveal hidden attributes of the network topology (e.g., other nodes' degrees). Based on such estimates, and with a partial knowledge about the topology of its neighborhood, a node decides how to efficiently forward incoming query messages to other nodes. In order to quantitatively evaluate and confirm the performance of A2PF, we conduct detailed experiments in various network topologies.**

***Keywords*–Complex networks; dynamic networks; probabilistic flooding; search.**

I. INTRODUCTION

Search is a common and well known operation in numerous domains, including physical, biological, economic, communication social and peer-to-peer (p2p) systems [1]–[4]. The process of search unfolds when a node generates a question (or a query message); the message is transmitted, traversing the network until it gets answered or some termination condition is met. The query is about locating resources (e.g., data, services, information or other types of computational resources [5]) or target nodes.

The development of search strategies has a rich history [6]. It has become the subject of a substantial amount of research and as a result, quite a few schemes have been proposed. Flooding and its variants are traditionally used for search in real-world networks as information regarding target nodes or resource placement is inherently limited. Because flooding can indeed flood the network with messages, its extent is usually limited by a Time-To-Live (TTL) parameter. TTL is actually the maximum number of hops a query message is allowed to travel. One of the major drawbacks of flooding is the large number of produced messages which overload the network quickly [7].

Probabilistic search, which is the subject of this paper, is a class of dynamic search strategies which try to alleviate the deficiencies of flooding. In this type of search, each node propagates the query message with a given probability, termed *forwarding probability*. One of the major goals of probabilistic flooding strategies is to determine the forwarding probabilities so as to make the decision whether to further propagate a message or not, as efficient as possible. This decision may encode a property of the network or an estimation of some of the system characteristics, e.g., approximate knowledge of node distances or resource popularity.

Actual networks are usually formed by nodes and links between them that change over time [8]. The problem of performing search efficiently in such settings is vastly more difficult than in static networks. The search procedure now faces new restrictions and challenges as it has to deal with temporal links and *churn*. Churn [9] refers to the continuous process of node arrival and departure in/from the system.

In this paper, we propose a new search method which is applicable to dynamic networks. It is rooted in a previous proposal [10] but exploits local information about the generalized neighborhood of a node, which is collected from past queries that have been submitted by other nodes in the system. The main contributions of this work are the following:

- We propose Adaptive Advanced Probabilistic Search (*A2PF*), a novel adaptive search scheme which is capable of adjusting its operation parameters in dynamic networks.
- We implement a local mechanism that "learns" topological properties of the network from passing messages.
- We provide experimental evidence on dynamic networks, which show that A2PF is able of achieving superior performance in comparison to other probabilistic flooding protocols.

The rest of this paper is organized as follows. After discussing related works, Section II gives the system model and introduces our heuristic for unveiling characteristics of the network topology. The proposed A2PF strategy is then presented in Section III. Section IV presents experimental results and, finally, Section V concludes this work.

A. Related Work

Typically, researchers study the problem of search in networks which are modeled as static (non-evolving) graphs. Plain flooding propagates the query messages with forwarding probability equal to 1 and makes no use of knowledge about the structure of the network or any other of its properties. In modified BFS (mBFS) [11], a node propagates the query message only to a subset of its neighbors with a fixed forwarding probability. Adaptive Resource-based Probabilistic Search (ARPS) [12] varies the forwarding probabilities according to resource popularity and node degree distribution. Advanced Probabilistic Flooding (APF) [10] adjusts the forwarding probability according to the distance from the query initiator and the popularity of the requested resource.

However, the above approaches are not designed for many real-world cases, where the state of nodes and links change over time. Studies on structured and unstructured peer-to-peer (p2p) systems [9] [13] [14] try to understand the user behavior and examine the impact of churn on the network and its processes. Other works [15]–[17] study the limitations of distributed computations, such as counting, aggregation, summing or averaging in dynamic networks, using random walkers, gossip protocols or other probabilistic forwarding techniques.

Using simulation, Furness and Kolberg [18] investigate the impact of churn on the success rate of "blind" search techniques in structured p2p systems. Augustine et al [19], assuming limited churn and efficient storage, propose randomized distributed algorithms that guarantee the data retrieval with high probability even under high adversarial churn.

Finally, the works in [20]–[23] examine the completion time of the flooding mechanism on dynamic graphs, which are modeled as a fixed set of nodes with links whose birth and death is modeled after a Markovian process. These approaches do not capture other important network events, such as node arrivals and/or departures.

II. System Model and Methodology

Let $G_\tau = (V_\tau, E_\tau)$ be a graph that is defined by its node set $V_\tau = \{v_1, ..., v_N\}$ and edge set $E_\tau = \{(v_i, v_j)|v_i, v_j \in V_\tau\}$, representing our dynamic network at time τ. Here, we focus on the classical Erdős-Rényi (ER) network and on scale-free networks. The ER network is formed on N nodes and each pair of nodes is connected by an edge with some given probability p. In scale-free networks, the node degrees follow a power-law distribution.

Each node u in the network has a certain lifetime that is defined at node creation time. At a particular time τ, this node is *active* if it participates or *inactive* if it has left the system. Similarly, each neighbor v of an active node u can be either active or inactive. The state of each neighbor is modeled as an on/off process $Y_v(\tau)$ [24], where

$$Y_v(\tau) = \begin{cases} 1 & \text{neighbor } v \text{ is active at time } \tau, \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

At time τ, the degree of node u, can be calculated as

$$d(\tau) = \sum_{v \in \mathcal{N}(u)} Y_v(\tau)$$

where $\mathcal{N}(u)$ is the immediate neighborhood of u (i.e., the set of all nodes that have a connection with u).

Flooding search starts at a randomly picked node u. The node sends the query message to all its neighbors, and then the neighbors which do not know about the asked resource forward the message to their own neighbors, repeating the process until the whole network or a certain part of it has been reached. Intuitively, the flooding search process discovers new nodes in rounds. In step/round 1, it reaches the neighbors of u, in step 2 it reaches the neighbors of its neighbors, and so on. Each query message embeds a step (or hop) counter, which is incremented by 1 when the message gets forwarded to the next node, so that the process can be limited to some maximum allowed distance (TTL).

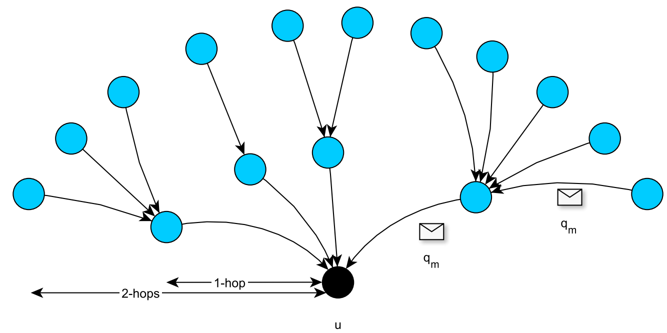

Figure 1. The flooding shape as it is viewed by node u. It is the result of information collected from incoming query messages q_m.

A. Average Degree at Distance t

As we discussed above, a node u with degree $d(\tau)$ knows that there exist $\sum_{v \in \mathcal{N}(u)} Y_v(\tau)$ nodes at distance 1. Suppose that the average number of nodes in distance t from a node u is n_t. During the flooding process, these nodes will be receiving the query message at exactly the t-th step. If we symbolize the set of these nodes by $\mathcal{N}_t(u)$, then the average degree of nodes at distance t from a node u is given by:

$$\overline{d}_t(\tau) = \sum_{v \in \mathcal{N}_t(u)} Y_v(\tau)$$

We are interested in determining n_t and $\overline{d}_t(\tau)$. However, in complex networks with arbitrary degree distributions, deriving analytically these quantities for distance $t > 1$ is not always precise or indeed possible [2] [25] [26]. In addition, such approaches are based on the assumption that the network is static. To overcome this problem, we propose a heuristic technique.

Consider an active node u which, during its lifetime, poses and receives query messages. Every node in the network is assumed to never re-forward any message that it forwarded in the past. In effect, each node implements a local log/monitor at each connection in order to limit retransmissions of the same message on that edge. Due to the nature of flooding-based schemes and the utilization of local monitor mechanisms, first-time message receptions occur through shortest paths. Thus, a node can learn its distance from the querying node simply by checking the hop counter in the message. Consider node u as illustrated in Figure 1. From incoming query messages, node u can learn the distances to other nodes in the network. This leads us to the following idea: it may be possible to discover additional network parameters through the information stored in query messages. For example, if querying nodes include their degree within the message, node u could also estimate the average degree of nodes at any distance t, simply by averaging up all the incoming degree information from messages with hop counter equal to t. This key insight allows node u to uncover structural information about the network topology.

Consequently, we enforce the following rules for all the nodes in the network:

- Every querying node u includes its degree into the query messages (q_m) it poses in network.
- Every node v which receives a query message q_m for the first time, at step t of the search process, stores

the embedded degree information in order to estimate the average degree of nodes at distance t.

B. Estimating the Average Degree at Distance t

Each node exploits observations from incoming query messages coming from a certain distance t, in order to estimate the average degree of nodes at this distance. This estimation can be simply the arithmetic mean of the k most recent observations (node degrees). Such an estimate assumes that all values are equally important. However, the degrees of nodes change over time due to the dynamic evolution of the network; in addition some of those k values may not be relevant any more because the corresponding nodes departed and are no longer active.

To obtain a better estimate, we resort to a *weighted* mean where we give more emphasis to recent values and progressively forget the past ones. For a participating node, let $\hat{d}_t$ represent the estimate of the average node degree at distance t. For each subsequent received value, the new estimation is calculated as a linear combination of the current one and the new incoming value, as follows:

$$\hat{d}_t(\text{new}) = (1 - w) \times \hat{d}_t(\text{current}) + w \times value$$

where $value$ is the newly received information about node degree at distance t, and w is the weight factor, $0 \leq w \leq 1$. Initially, when the first messages from distance t arrives, we set $\hat{d}_t = value$. The weight factor is a design parameter and determines how quickly the past values are forgotten. Its value should be tuned according to the characteristics of the network and the generated queries; for example, under a high churn rate and relatively infrequent queries, a value of $w > 0.5$ might be more appropriate so as to quickly "learn" the new node degrees.

III. Advanced Adaptive Probabilistic Search (A2PF)

Suppose that a node u poses a query message to the network using a flooding-based strategy. We can visualize the search space as extending in concentric circles, with node u at the center and the nodes on the circle of radius t being nodes lying t hops away from u. For example, the nodes in the first cycle ($t = 1$) are u's immediate neighbors; the second circle ($t = 2$) contains the immediate neighbors of u's neighbors, etc. When a node receives a query, it does not know if the query has been answered on another search path. Here, we exploit the main idea behind the original APF strategy [10], which chooses to propagate a query message only if the query has not been answered yet (with high probability). To do this, we need to know the number of nodes that have received the query up to that point, so as to estimate if the query is already answered or not.

Let n_t be the number of nodes that have received the query exactly at step t, and let N_t be the number of all visited nodes up to and including step t; clearly $N_t = \sum_{i=0}^{t} n_t$. According to APF, which assumes static networks, the number of nodes that have received the query at step $t+1$ can be calculated as follows:

$$n_{t+1} = (\overline{d}_t - 1) n_t \left(1 - \frac{N_t}{N}\right) \tag{2}$$

where $\overline{d}_t$ is the average node degree in distance t from u. A node that receives the query at step t, may then decide to forward it with probability $(1 - (N_t/N))^{qN}$, where q is a measure of the popularity of the required resource.

Notice, however, that it was implied that $\overline{d}_t$ is either known or can be (imprecisely) substituted by the average degree of the whole network. While the actual details are given in Section IV, Figure 2 can serve as an early illustration of how the average node degree varies according to the distance, as viewed by a node. In networks with power-law degree distributions, the average degree decreases abruptly after the first few hops. This means that the average node degree may be quite large during the first steps of the search process, converging to the network average degree in the following steps. Clearly, we need the knowledge of the average degree at distance t from a node so as to obtain an accurate estimation of the network coverage and guide the search process.

In dynamic networks, the number of nodes varies over time according to the churn characteristics while links between nodes may come and go. Let $N(\tau)$ be the number of nodes of network at time τ. Based on (2) we can express the number of nodes that have received the query at step $t+1$ in a dynamic network as follows:

$$n_{t+1} = (\overline{d}_t(\tau) - 1) n_t \left(1 - \frac{N_t}{N(\tau)}\right).$$

Similarly to the static case, the term $\overline{d}_t(\tau)$ is crucial for the successful calculation of n_{t+1}, especially for particular families of networks, such as scale-free ones. In the proposed A2PF strategy we use the estimator presented in Section II-B to approximate $\overline{d}_t(\tau)$:

$$n_{t+1} = (\hat{d}_t(\tau) - 1) n_t \left(1 - \frac{N_t}{N(\tau)}\right). \tag{3}$$

The forwarding probability is then given by

$$p_f(t) = \left(1 - \frac{N_t}{N(\tau)}\right)^{qN(\tau)}.$$

IV. Simulation

We use the *Armonia* simulation framework [27] to evaluate and compare our approach with other flooding-based strategies. Armonia offers a multitude of parametrized topologies, while also allowing the generation of additional ones. Armonia offers complete control over resource allocation and placement. Finally, it implements a large number of search protocols but also provides facilities for user-defined ones.

For our purposes here, we conduct our study on synthetic networks, focusing on classical Erdős-Rényi (ER) random graphs and Barabási (BA) power-law graphs. The networks can be static or dynamic. Initially, the network has N nodes; this number can vary in dynamic networks during simulation time. The nodes provide resources (with each resource having a number of replicas), which are uniformly distributed over the network. In dynamic networks, a node participates in the system for a certain time period, called the node *lifetime*, which is the elapsed time from its first appearance in the system until its departure. The lifetime of each node is defined at creation time by a Weibull distribution [13] with parameters a and b. New nodes join the network according to a Poisson distribution with arrival rate λ. A total of 1,000 queries are submitted to the system; we collect the measurements and average the results.

Before proceeding with the comparative study, we first present the average degree as a function of distance t in real-world datasets, obtained from the Stanford Network Analysis

 ISBN: 978-1-61208-680-4

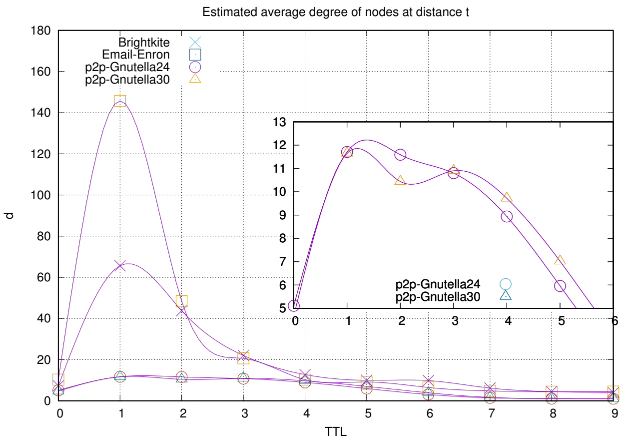

Figure 2. The overall average degree of nodes in distance t, measured in datasets from real networks.

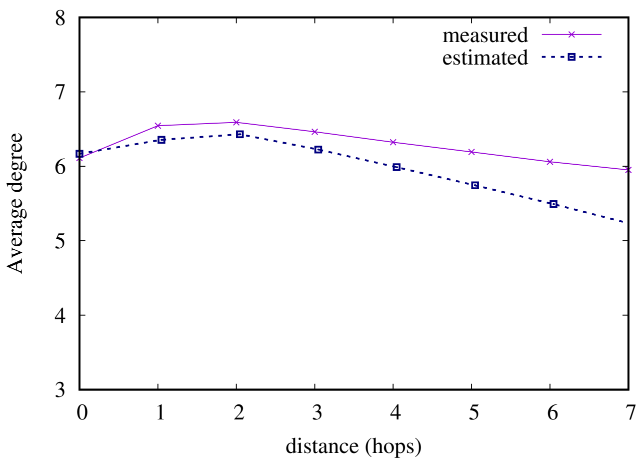

Figure 3. The overall average estimation of node degrees per hop distance vs the actually measured ones in a dynamic ER network of 1,000 nodes.

Platform [28]. Figure 2 shows the overall average degree per hop distance for the following traces: two gnutella network snapshots with 26,518 nodes and 36,682 nodes, captured at 2002, a trace of the email-Enron network with 36,518 nodes and a trace from the Brightkite social network with 58,228 nodes. We observe that the average degree 1 hop away is quite larger than the average network degree. This proves that knowledge of the average node degree in the network cannot serve as a good approximation of the node degree at arbitrary hop distances.

Next, we apply our heuristic estimator in a dynamic ER network with 1,000 nodes. The node lifetime is determined by a Weibull distribution with parameters $a = 7,200$ sec and $b = 4$ and the arrival rate is $\lambda = 2$. Nodes submit queries and collect statistics about other nodes' degrees. The estimator weight factor was empirically set to $w = 0.1$ to favor past estimations. Figure 3 shows the arithmetic mean of averages, over all nodes as estimated by our heuristic ("estimated" curve) while the other curve ("measured") shows the actual averages as a function of distance t for the same network. While the two curves do not coincide, the estimator proves to follow effectively the measured curve, even if the weight factor was not fine-tuned.

For our comparative study, we consider four flooding-based search policies, namely plain flooding (flood), mBFS [11], ARPS [12] and our proposed A2PF. All strategies are constrained by a TTL hop limit. Also, they are assumed to maintain a local monitor mechanism in order to avoid the re-forwarding of an already sent message. For the mBFS and ARPS strategies, the forwarding probabilities (p_f) follow the corresponding authors' guidelines. In particular, in mBFS the query is forwarded to 50% of a node's neighbors ($p_f = 0.5$), while in ARPS the value of p_f depends on resource popularity. For plain flooding $p_f = 1$. The forwarding probabilities of A2PF involve a different calculation in each node, using the estimations in Section III. We assume that the basic structural properties of the network do not change drastically over time, so that $N(\tau) \simeq N$, approximately balancing arrivals and departures.

The comparison is based on the performance of these search strategies on different networks. We assess the performance based on three metrics:

- The *probability of success*, which measures the probability that a query can locate the desired resource, given the TTL hop limit. A query is considered successful if it discovers at least one replica of the resource in question.
- The total number of *messages* that are transmitted, before the resource is located.
- The number of *duplicate messages*. A message is considered duplicate, or redundant, if it is received by the same node more than once (and thus does not contribute to the success of the search). It serves as a measure of the how efficiently a policy utilizes the network resources.

We present three different experiments. The corresponding simulation parameters are summarized in Table I.

The first ER model is generated with 30,000 nodes and high churn (45%). Resource popularity is the same for all resources and equal to $q = 0.0006$. Figure 4 shows the results (success rate on the left and number of duplicate messages on the right) obtained by simulating the four search strategies. While all strategies manage to locate at least one resource in the first few hops, only A2PF produces negligible duplicate messages. One of the key benefits of A2PF is its adaptive behavior on network changes.

Next, we use the model of Barabási-Albert [29] and generate a scale-free network with no churn. In scale-free networks, the node degrees can vary significantly with respect to the average network degree. In practice, this impacts heavily the search process. In this experiment we test the behavior of our approach on such networks. This particular network starts with three connected nodes. Each new incoming node is connected to two existing nodes ($m = 2$). New nodes prefer to link with nodes with more neighbors. The total number of nodes on the network is 30,000. Resource popularity is uniform and equal to $q = 0.0016$.

Figure 5 illustrates the simulation results. We observe that flooding is successful after 2 hops, ARPS and A2PF need 1 more hop, while mBFS needs a further 1 hop. At the same time, we observe an exponential growth of duplicate messages in the case of plain flooding and ARPS. In contrast, mBFS

TABLE I. SIMULATION NETWORK TYPES AND PARAMETERS

$\overline{d}$	*network*	*nodes*	*q*	*Flooding*	*mBFS*	*ARPS*
6	dynamic ER	$30,000$	0.0006	$p_f = 1$	$p_f = 0.5$	$p_f = 0.9$
4	BA	$30,000$	0.0016	$p_f = 1$	$p_f = 0.5$	$p_f = 0.8$
4	dynamic BA	$20,000$	0.005	$p_f = 1$	$p_f = 0.5$	$p_f = 0.7$

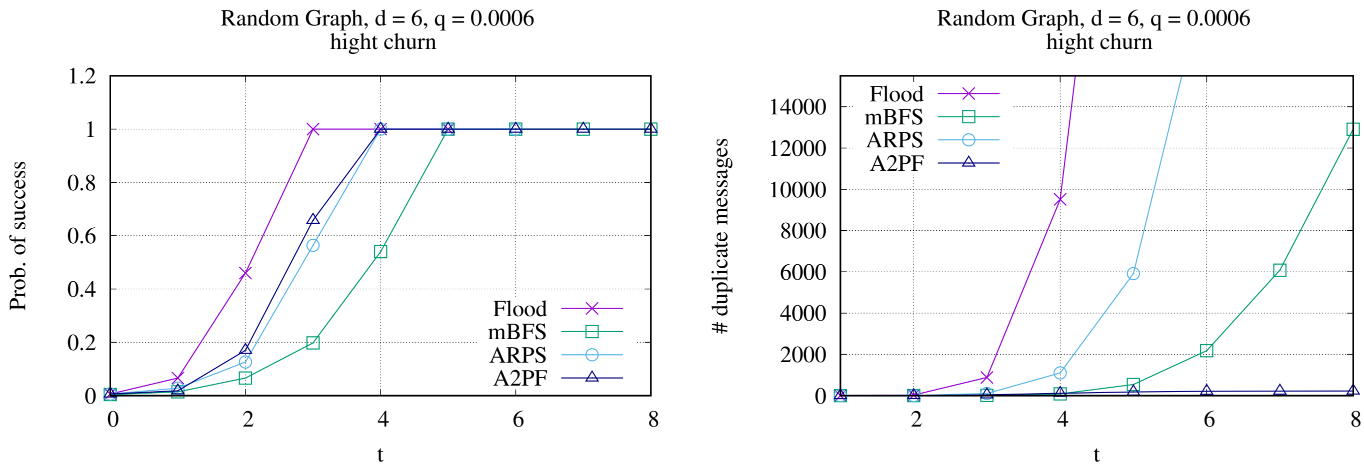

Figure 4. Success rate (left) and duplicate messages (right) in a dynamic ER network with 30,000 nodes, $\overline{d} = 6$ and $q = 0.0006$. The churn rate is churn 45%.

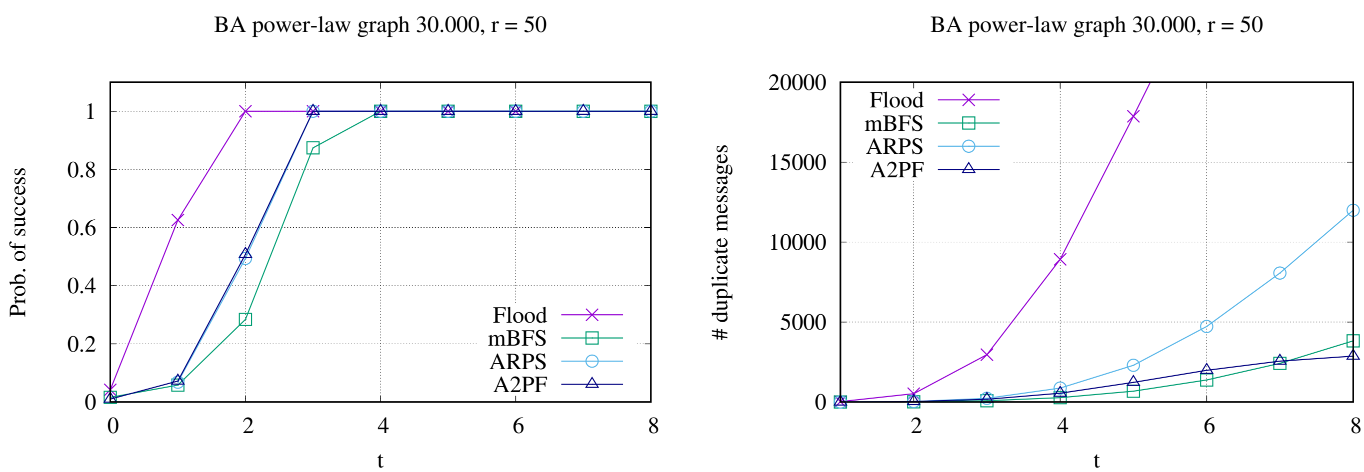

Figure 5. Success rate (left) and duplicate messages (right) in a BA power-law network with 30,000 nodes and $m = 2$, $q = 0.0016$.

and, especially, A2PF, manage to keep such messages in quite limited quantities.

Finally, we consider the search performance of the four strategies on a dynamic BA network with 20,000 nodes under high churn (45%). The experimental results are given in Figure 6. Flooding has the best success rate as a function of hop distance. At the same time, it pays the price of the largest number of redundant messages. ARPS and A2PF have similar success rates while A2PF produces the fewest number of messages among all strategies. mBFS does not seem to be very effective in this case.

All the above experiments indicate the consistent superiority of A2PF over the other three strategies (flooding, ARPS, mBFS) regarding the success rate and the number of redundant messages. The employment of the heuristic estimator is responsible for the success of A2PF, giving it the ability to moderate the redundant traffic.

V. CONCLUSION

In this work, we propose the A2PF, an adaptive probabilistic search strategy, which exploits knowledge collected from received queries at each node. A2PF is based on APF and embeds a new heuristic mechanism. The heuristic mechanism assists nodes in estimating the average degree of nodes $t \geq 2$ hops away from them. The estimations guide a node to make a knowledgeable decision whether to further propagate a query message or not. Our experimental results confirm that A2PF is fast and efficient.

 ISBN: 978-1-61208-680-4

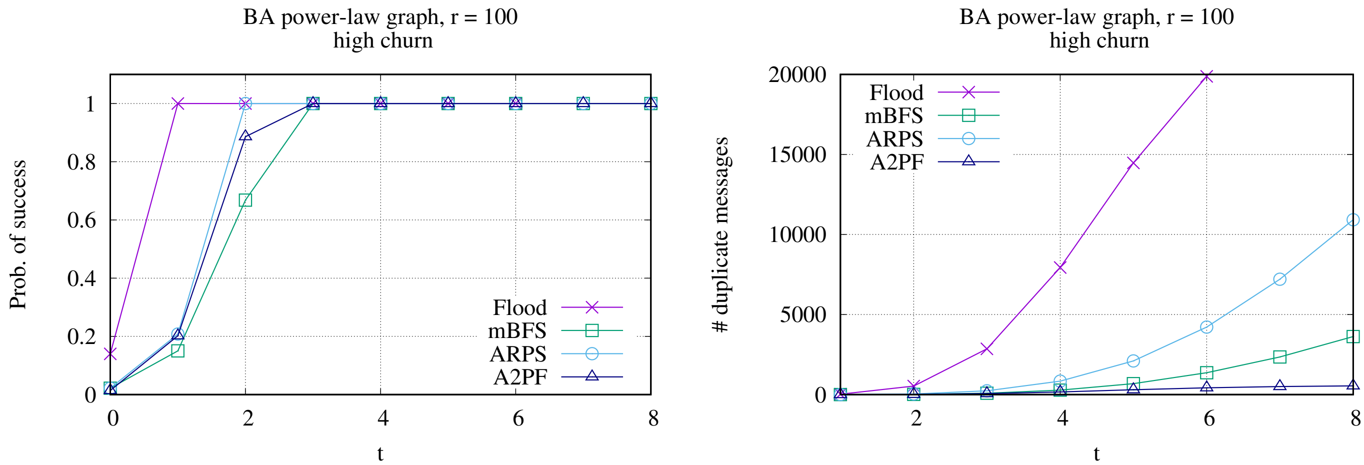

Figure 6. Success rate (left) and duplicate messages (right) in a BA power-law network with 20,000 nodes, $m = 2$ and $q = 0.005$. The churn rate is 45%.

REFERENCES

[1] A.-L. Barabási, Network science. Cambridge University Press, 2016.

[2] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics," Physics Reports, vol. 424, no. 4, 2006, pp. 175–308.

[3] M. O. Jackson, Social and economic networks. Princeton University Press, 2010.

[4] M. Newman, Networks: An Introduction. New York, NY, USA: Oxford University Press, Inc., 2010.

[5] W. Xu, Y. Guo, L. Shi, L. Liu, and B. Yuan, "A search strategy for social resource in decentralized social networks," in 2017 Fifth International Conference on Advanced Cloud and Big Data (CBD). IEEE, 2017, pp. 213–219.

[6] S. M. Thampi and C. S. K, "Survey of search and replication schemes in unstructured p2p networks," Network Protocols and Algorithms, vol. 2, no. 1, 2010, pp. 93–131.

[7] V. V. Dimakopoulos and E. Pitoura, "On the performance of flooding-based resource discovery," IEEE Transactions on Parallel and Distributed Systems, vol. 17, no. 11, 2006, pp. 1242–1252.

[8] P. Holme and J. Saramäki, "Temporal networks," Physics Reports, vol. 519, no. 3, 2012, pp. 97–125.

[9] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz, "Handling churn in a dht," in Proc. Annual Conference on USENIX Annual Technical Conference, ser. ATEC '04. Berkeley, CA, USA: USENIX Association, 2004, pp. 10–10.

[10] S. V. Margariti and V. V. Dimakopoulos, "On probabilistic flooding search over unstructured peer-to-peer networks," Peer-to-Peer Networking and Applications, vol. 8, no. 3, 2015, pp. 447–458.

[11] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti, "A local search mechanism for peer-to-peer networks," in Proceedings of the 11th International Conference on Information and Knowledge Management. ACM, 2002, pp. 300–307.

[12] H. Zhang, L. Zhang, X. Shan, and V. O. Li, "Probabilistic search in p2p networks with high node degree variation," in 2007 IEEE International Conference on Communications, 2007, pp. 1710–1715.

[13] D. Stutzbach and R. Rejaie, "Understanding churn in peer-to-peer networks," in Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement, 2006, pp. 189–202.

[14] Z. Yao, D. B. Cline, X. Wang, and D. Loguinov, "Unifying models of churn and resilience for unstructured p2p graphs," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 9, 2014, pp. 2475–2485.

[15] M. Jelasity, A. Montresor, and O. Babaoglu, "Gossip-based aggregation in large dynamic networks," ACM Transactions on Computer Systems (TOCS), vol. 23, no. 3, 2005, pp. 219–252.

[16] F. Kuhn, N. Lynch, and R. Oshman, "Distributed computation in dynamic networks," in Proceedings of the 42nd ACM Symposium on Theory of Computing. ACM, 2010, pp. 513–522.

[17] A. D. Sarma, A. R. Molla, and G. Pandurangan, "Distributed computation in dynamic networks via random walks," Theoretical Computer Science, vol. 581, 2015, pp. 45–66.

[18] J. Furness and M. Kolberg, "Considering complex search techniques in dhts under churn," in 2011 IEEE Consumer Communications and Networking Conference, 2011, pp. 559–564.

[19] J. Augustine, A. R. Molla, E. Morsy, G. Pandurangan, P. Robinson, and E. Upfal, "Storage and search in dynamic peer-to-peer networks," in Proceedings of the 25th Annual ACM Symposium on Parallelism in Algorithms and Architectures, 2013, pp. 53–62.

[20] A. E. Clementi, C. Macci, A. Monti, F. Pasquale, and R. Silvestri, "Flooding time in edge-markovian dynamic graphs," in Proceedings of the 27th ACM Symposium on Principles of Distributed Computing. ACM, 2008, pp. 213–222.

[21] H. Baumann, P. Crescenzi, and P. Fraigniaud, "Parsimonious flooding in dynamic graphs," Distributed Computing, vol. 24, no. 1, 2011, pp. 31–44.

[22] ——, "Flooding in dynamic graphs with arbitrary degree sequence," Journal of Parallel and Distributed Computing, vol. 74, no. 5, 2014, pp. 2433–2437.

[23] A. Clementi, R. Silvestri, and L. Trevisan, "Information spreading in dynamic graphs," Distributed Computing, vol. 28, no. 1, 2015, pp. 55–73.

[24] D. Leonard, Z. Yao, V. Rai, and D. Loguinov, "On lifetime-based node failure and stochastic resilience of decentralized peer-to-peer networks," IEEE/ACM Transactions on Networking, vol. 15, no. 3, 2007, pp. 644–656.

[25] M. Mayo, A. Abdelzaher, and P. Ghosh, "Long-range degree correlations in complex networks," Computational Social Networks, vol. 2, no. 1, 2015, pp. 1–13.

[26] L. Wang, S. Bayhan, J. Ott, J. Kangasharju, and J. Crowcroft, "Understanding scoped-flooding for content discovery and caching in content networks," IEEE Journal on Selected Areas in Communications, 2018.

[27] S. V. Margariti and V. V. Dimakopoulos, "Simulating search protocols in large-scale dynamic networks," in 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP). IEEE, 2016, pp. 176–183.

[28] J. Leskovec. Stanford network analysis project. [Online]. Available: http://snap.stanford.edu/index.html [retrieved: September, 2018]

[29] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," Science, vol. 286, no. 5439, 1999, pp. 509–512.