Computer Networks 103 (2016) 165–180

Contents lists available at ScienceDirect

Computer Networks

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

Resource location based on precomputed partial random walks in dynamic networks

Víctor M. López Millán[a,*], Vicent Cholvi[b,1], Antonio Fernández Anta[d,2], Luis López[c,3]

[a] Universidad CEU San Pablo, Spain
[b] Universitat Jaume I, Spain
[c] Universidad Rey Juan Carlos, Spain
[d] Institute IMDEA Networks, Spain

ARTICLE INFO

Article history:
Received 23 November 2015
Revised 29 February 2016
Accepted 9 April 2016
Available online 13 April 2016

Keywords:
Resource location
Dynamic networks
Random walks
Complex networks

ABSTRACT

The problem of finding a resource residing in a network node (the *resource location problem*) is a challenge in complex networks due to aspects as network size, unknown network topology, and network dynamics. The problem is especially difficult if no requirements on the resource placement strategy or the network structure are to be imposed, assuming of course that keeping centralized resource information is not feasible or appropriate. Under these conditions, random algorithms are useful to search the network. A possible strategy for static networks, proposed in previous work, uses short random walks precomputed at each network node as partial walks to construct longer random walks with associated resource information. In this work, we adapt the previous mechanisms to dynamic networks, where resource instances may appear in, and disappear from, network nodes, and the nodes themselves may leave and join the network, resembling realistic scenarios. We analyze the resulting resource location mechanisms, providing expressions that accurately predict average search lengths, which are validated using simulation experiments. Reduction of average search lengths compared to simple random walk searches are found to be very large, even in the face of high network volatility. We also study the cost of the mechanisms, focusing on the overhead implied by the periodic recomputation of partial walks to refresh the information on resources, concluding that the proposed mechanisms behave efficiently and robustly in dynamic networks.

1. Introduction

Random walks are network routing mechanisms which have been extensively studied and used in a wide range of applications: physics, mathematics, population dynamics, bioinformatics, etc. [11,18,24]. Roughly speaking, they choose, at each point of the route, the next node uniformly at random among the neighbors of the current node.

Among the advantages of random walks when applied to communication networks is the fact they need only local information, avoiding the bandwidth overhead necessary in other routing mechanisms to communicate with other nodes. This is especially useful when there is no knowledge on the structure of the whole network, or when the network structure changes frequently. For these reasons, random walks have been proposed as a base mechanism for multiple network applications, including network sampling [9,16], network resource location [1,10,28,33], network construction [5,15,19–21], and network characterization [8,29,32].

The emergence of the peer-to-peer (P2P) architecture model has been proven useful in many applications in recent years. While structured P2P systems (e.g., Chord [31], CAN [27] –Content-Addressable Network–, Kademlia [23], etc.) provide efficient search mechanisms, they introduce a significant management overhead. In turn, unstructured systems have little management overhead and, consequently, have been considered in several scenarios (e.g., Gnutella [5,20], CAP [14] –Cluster-based Architecture for P2P–,

* Corresponding author. Tel.: +34 91 3726424 (ext. 4911); fax: +34 91 3724049.
E-mail address: vmlopez.eps@ceu.es (V.M.L. Millán).

1 Supported in part by the Spanish Ministerio de Economía y Competitividad under grant TIN2011-28347-C02-01.

2 Supported in part by Ministerio de Economia y Competitividad grant TEC2014-55713-R, Regional Government of Madrid (CM) grant Cloud4BigData (S2013/ICE-2894, co-funded by FSE & FEDER), NSF of China grant 61520106005, and European Commission H2020 grants ReCred and NOTRE.

3 Supported in part by the European Commission under projects NUBOMEDIA (FP7-ICT GA 610576), FI-WARE (FP7-ICT GA 285248) and FI-CORE (FP7-ICT GA 632893). This work is also supported by the Spanish Ministerio de Economía y Competitividad under grant REACTIVEMEDIA (TIN2013-41819-R) and by Comunidad de Madrid under grant CLOUD4BIGDATA (P2013/ICE-2894) cofounded by FSE & FEDER.

http://dx.doi.org/10.1016/j.comnet.2016.04.008
1389-1286/© 2016 Published by Elsevier B.V.

etc.). For such systems, searching techniques based on *flooding, supernodes* and *random walks* have been used. However, it is known that flooding mechanisms do not scale well [13], and supernode systems are vulnerable to supernodes failures (technical problems, attacks, censorship, etc.). Therefore, random walks have been used to search for resources held in the nodes of a network (e.g., [5,22]), a problem usually known as *resource location*. The problem consists of finding a node that holds a given resource, the *target node*, starting at some *source node*. The source node is checked for the resource: if it is not found locally, the search hops to a random neighbor, checking that node for the resource. The search proceeds through the network in this way, until the target node is reached.

Nevertheless, by using random walks, some nodes may be (unnecessarily) visited more than once, while other nodes may remain unvisited for a long time. Avoiding this problem is the main objective of our study.

1.1. The Dynamic resource location problem

In this work, we are concerned with the resource location problem in networks with dynamic behavior regarding both resources and nodes.

In particular, we consider scenarios in which resources are randomly placed in the nodes across the network. Then, on the one hand, we consider scenarios in which the instances of the resources may appear and disappear from a time instant to another, maybe at different nodes. On the other hand, we also consider scenarios in which the network nodes themselves may also leave and join the network.

In these scenarios, all the nodes of the network may launch independent searches for different resources (e.g., files) at any time, without the help of a centralized server, and we are interested in measuring the average performance of searches between *any* pair of nodes.

Our assumptions regarding dynamicity cover a wide range of scenarios. For instance, in P2P networks nodes represent users, which may leave and join the network quite often. Also, resources represent the shared files, which may appear and disappear from time to time.

1.2. Contributions

In this work, we use the technique of concatenating *partial random walks* (PWs) to generalize the resource location mechanisms introduced in [17] for static networks to the case of dynamic networks. In particular, this paper provides new analytical models that predict the behavior of the resource location mechanisms in scenarios with dynamic resources and in scenarios with dynamic nodes, along with new simulation experiments to validate the analytical results. In addition, a new analysis of the cost of the mechanisms in these scenarios is provided. We consider the two versions of the mechanisms proposed in [17] and adapt them to operate in the dynamic scenarios. In the first version, which we refer as *choose-first PW-RW*, the search mechanism first chooses one of the PWs at random and then checks its associated information for the desired resource. In the second version, which we refer as *check-first PW-RW*, the search mechanism first checks the associated resource information of all the PWs of the node, and then randomly chooses among the PWs with a positive result. It is clear that there are other choices regarding the search mechanisms that seem reasonable. However, the ones considered in our study follow very different approaches (one chooses first, and the other checks first). Therefore, that will allow us to check the strength of our approach in very different circumstances.

Then, we have studied their performance, considering the following aspects:

- *Dynamic resources:* We have developed an analytical mean-field model for both mechanisms when resources are dynamic. Expressions are given for the corresponding *expected search length* (i.e., the expected number of hops taken to find the resource, averaged over all source nodes, target nodes, and network topologies) of each mechanism. These expressions provide predictions as a function of several parameters of the model, such as the network structure (size and degree distribution), the resource dynamics, and those of the mechanisms operation.
  The predictions of the models are validated by simulation experiments in three types of randomly built networks: regular, Erdős-Rényi, and scale-free. These experiments are also used to compare the performance of both mechanisms, and to investigate the influence of the resource dynamics. We have compared the performance of the proposed search mechanisms with respect to random walk searches. For the *choose-first PW-RW* mechanism we have found a reduction in the average search length with respect to simple random walk ranging from around 57% to 88%. For the *check-first PW-RW* mechanism such a reduction is even bigger, achieving reductions above 90%.
- *Dynamic nodes:* We have also considered the case where network nodes may leave and join the network, and have provided both analytical and experimental results. We have found a reduction in the average search length with respect to simple random walks above 94% (using the *check-first PW-RW* mechanism).
- *Cost:* Finally, we have analyzed the cost of the PW-RW mechanisms, defined as the number of messages, taking into account the cost of searches themselves and the cost of precomputing the PWs in each recomputation interval. We have provided analytical expressions for the relation between the cost and the length of the recomputation interval, as well as for the interval length that minimizes this cost. We have found that the impact of the precomputation of PWs on the cost is not significant in a wide range of lengths of the precomputation interval depending on the dynamic behavior of the network and on the dynamic behavior of searches.

1.3. Related work

Da Fontoura Costa and Travieso [6] study the network coverage of three types of random walks: traditional, preferential to untracked edges, and preferential to unvisited nodes. Also, Yang [33] studies the search performance of five random walk variations: no-back (NB), no-triangle-loop (NTL), no-quadrangle-loop (NQL), self-avoiding (SA) and high-degree-preferential self-avoiding (PSA). Self-avoiding walks (SAW) are those that try not to visit nodes that have already been visited. Several variations of this idea have been studied, differing in the probability of revisiting a node. Some examples are: strict SAW, true or myopic SAW, and weakly SAW [4,30]. In [7], Das Sarma et al. propose a distributed algorithm to obtain a random walk of a specified length ℓ in a number of rounds[1] proportional to $\sqrt{\ell}$.

López Millán et al [17] propose a mechanism for resource location based on building random walks connecting together partial walks (PW) previously computed at each network node. However, the mechanisms in [17] are only valid when both nodes and resources have a static behavior, contrary to the approach we follow in this paper.

The remainder of this paper is arranged as follows. Section 2 and 3 respectively present the *choose-first PW-RW* and *check-first PW-RW* mechanisms in scenarios with dynamic

[1] A *round* is a unit of discrete time in which every node is allowed to send a message to one of its neighbors. According to this definition, a simple random walk of length ℓ would then take ℓ rounds to be computed.