ESTIMATING THE SIZE OF PEER-TO-PEER NETWORKS USING LAMBERT'S W FUNCTION.

Javier Bustos-Jimenez, Nicolas Bersano
Escuela de Ingenieria Informatica. Universidad Diego Portales
Av. Ejercito 441, Santiago, Chile.
javier.bustos@inf.udp.cl,nbersano@al.udp.cl

Elisa Schaeffer
Universidad Autonoma de Nuevo Leon. FIME - Posgrado en Ingenieria de Sistemas.
AP 126-F. Ciudad Universitaria, San Nicolas de los Garza, NL 66450. Mexico.
elisa@yalma.fime.uanl.mx

Jose Miguel Piquer
Departamento de Ciencias de la Computacion (DCC). Universidad de Chile.
Blanco Encalada 2120, Santiago, Chile.
jpiquer@dcc.uchile.cl

Alexandru Iosup
Parallel and Distributed Systems Group,
Faculty of Engineering, Mathematics and Computer Science, Delft University of Technology,
Mekelweg 4, 2628 CD, Delft, The Netherlands
A.Iosup@tudelft.nl

Augusto Ciuffoletti
Department of Computer Science. University of Pisa
Corso Italia 40, 56125 Pisa - Italy.
augusto@di.unipi.it

Abstract In this work, we address the problem of locally estimating the size of a Peer-to-Peer (P2P) network using local information. We present a novel approach for estimating the size of a peer-to-peer (P2P) network, fitting the sum of new neighbors discovered at each iteration of a breadth-first search (BFS) with a logarithmic function, and then using Lambert's W function to solve a root of $a\ln(n)+b-n=0$, where n is the network size. With rather little computation, we reach an estimation error of at most 10 percent, only allowing the BFS to iterate to the third level.

Keywords: Peer-to-Peer, Network size, Estimation

1. Introduction

In this work, we address the problem of locally estimating the size of a Peer-to-Peer (P2P) network using local information. Our approach is motivated by a *breadth-first search* (BFS) at a node to calculate the total number of "unknown" nodes in the joint neighborhoods of the already discovered nodes. The rest of this article is organized as follows: in Section 2 we present the algorithm for P2P network-size estimation. In Section 3 we evaluate of our algorithm in terms of the estimation error (in percentages of the real network size). Finally, conclusions and future work are presented in Section 4.

1.1 Motivation

The potential applications of accurate estimates of network size are various. One broad class of applications are the numerous algorithms that either require knowledge of the network size or greatly benefit from such information, which is the case in some routing tasks. Essentially, in many distributed protocols the knowledge of the network size is implicitly assumed.

For example, when estimating the information spread or the gossiping coverage [11], knowing the network size helps. These algorithms have direct applications in the field of P2P recommendation schemes. Knowing the network size is also useful in estimating the latency of gossip based broadcast, especially in setting up a time-to-live (TTL) mechanism for gossips.

Secondly, many parameters that control the provisioning of resources in commercial P2P applications (such as data or video on demand) should be based on the network size [7]. To give a concrete example, the performance of BitTorrent can be improved by selecting the choking/unchoking count and rate dynamically, which eliminates a a potential source of poor performance [8].

Horowitz and Malkhi [9]propose a scheme for dynamically estimating network size at each node of the network as the network evolves (that is, nodes join and leave the network), by maintaining an estimate of the logarithm of network size. We, however, aim to estimate directly and more accurately the actual number of nodes in the network on demand, whenever a node happens to need that information, without maintaining activity at all nodes as all times.

1.2 Data Validity

For modelling Peer-to-Peer networks, we used the Delft BitTorrent DataSet 2[1], which is the outcome of a large-scale measurement of the BitTorrent network during one week in May 2005 [6]. The data set tracks over 450,000

[1] The Delft BitTorrent DataSet 2 is available online at `http://multiprobe.ewi.tudelft.nl/`

BitTorrent users coming from PiratesBay, the largest BitTorrent community at the time of the measurements.

The measurement tracked peers participating in any swarm of size 40 or above. By tracking peers interested in a broad range of file types and sizes, the measurement captured the characteristics of a world-wide community of users.

There are over 35 million BitTorrent peer events recorded in the data set, making it the largest publicly available P2P dataset. The measurement also included the Internet routes used by peers to exchange information. This information was gathered through multi-sourced traceroutes and spanned about 20 million IP addresses.

2. Peer-to-Peer network size

The work of Faloutsos et al. [12]studied the structure of the Internet in terms of different distributions, including the "hop-count" distribution of how many nodes are at a given distance.

It was one of three seminal papers — along with the small-world networks of Watts and Strogatz [13]and the scale-free networks of Barabási and Albert [14]that gave rise to extensive research of model and properties of *nonuniform networks* [15]. Nonuniform networks are graph models of real-world systems where the edges are *not* placed uniformly at random among the vertices. In such systems, the structure of the network is typically nontrivial and the network behavior is complex. Among such networks are practically all P2P systems — usually not every possible pair of peers in the network is equally likely to be neighbors.

Returning to the problem of network-size estimation, upon plotting the the number of new nodes discovered upon visiting each vertex during a BFS, we discovered a fuzzy, hard-to-model function, but when we instead summed the contributions of the neighbors and plotted the values

$$S_i(v) = \sum_{j=1}^{i} |\Lambda_i(v)| \tag{1}$$

with $\Lambda_i(v)$ of Equation 5 for a start vertex, we noted that the resulting plot was soft, approachable function. In the rest of this paper we present the results obtained by studying the function $\Lambda_i(v)$ varying the starting node v. $S_i(v)$ eventually levels off as $\Lambda_i(v)$ falls to zero, meaning that the entire network has been traversed and new nodes can no longer be found.

Let us model the network as a graph $G = (V, E)$, with $|V| = n$ being the total number of nodes. We denote the set of neighbors of a node $v \in V$ by

$$\Gamma(v) = \Gamma_1(v) = \{w \mid (u, w) \in E\}, \tag{2}$$

we define the second-neighborhood of v as

$$\Gamma_2(v) = \bigcup_{w \in \Gamma(v)} \Gamma(w) \tag{3}$$

and recursively from thereon the k-distance neighborhood of v as

$$\Gamma_k(v) = \bigcup w \in \Gamma_{k-1}(v)\Gamma(w). \tag{4}$$

One can compute iteratively for a starting node $v \in V$ the following value for $i = 1, 2, 3, \ldots$ until the value drops to zero:

$$\Lambda_i(v) = \Gamma_k(v) \setminus \bigcup_{j=1}^{k-1} \Gamma_j(v), \tag{5}$$

that is, the number of new nodes reached at step i of the BFS.

In BFS, nodes "pass on messages" to their neighbors, who then pass messages on to their neighbors – the messages being in a sense recursive procedure calls. This process resembles that of how a *rumor* spreads in a population, although in BFS the nodes pass the information or the request onto *all* of their neighbors, whereas rumors tend to spread less efficiently. Supposing that each node was to pass the message to just one random neighbor, the process of spreading would effectively be a random walk. Pittel [1]studies the coverage that a rumor achieves given the number of "rounds" of gossip-passing. Another good analogy comes from *epidemic spreading*, where it is of interest to estimate the number of infected individuals at a given time [2]. Analogies to epidemic spreading have been applied in data-base system design [3].

We use the *Lambert's W function* [4], which is the inverse function of $f(w) = we^w$ and cannot be directly expressed in terms of elementary functions, to *estimate* the network size based on the form of plots of accumulated sums of $|\Lambda_i(v)|$ for different nodes v.

The details on how this is done are presented in Section 2, where we justify the application of the W function in estimating the network size based on the shape of the plot of the accumulated sums for fixed v, which has the shape of $f(i) = a\log(i) + b$. Our choice of the W function bases on the work of Corless et al. [5], where a recurrence relation is used to calculate W, providing fast convergence to the desired result in less than 100 iterations.

In a P2P network, technically every node could explicitly calculate the network size by running a BFS (see pseudo-code in Figure 1).

In large networks, this is a long and communication-intensive procedure, as each edge of the network must be traversed. The size of `list` at each iteration is the number of new nodes discovered, $|\Lambda_i(v)|$. We found that the shape of the curve of $S_i(v)$ (Equation 1) can be approximated by

$$S_i(v) \approx y = a\ln(i) + b \tag{6}$$

```
procedure BFS(node s, graph G)

queue q;

// put the starting node s into the queue
put(q, s);

N = 1; // initial known network size

// while the queue is not empty
while (!empty(q)) {

  // retrieve the first element of the queue
  v = get(q);

  // mark the node v to have been visited
  mark_visited(v);

  // retrieve the list of neighboring nodes to v
  list = neighbours(v);

  // remove from the list all previously visited nodes
  list = remove_visited(list);

  // add to the known network size the newly encountered nodes
  n = n + size(list);

  // add the elements of list to the end of the queue
  put(q, list);
}
```

Figure 1. The BFS algorithm that would need to be executed to determine the size of the network n given the network as a graph G and a starting node s.

as is shown in Figure 1. We found an even better fit with a function with an additional term with $\ln(\ln(i))$, but we could only work with such a function using numerical methods. Nevertheless, the simpler form of Equation 6 serves our purposes and gives a reasonable fit, we saw no need to complicate the fitted function further.

The value of $S_i(v)$ as it levels off is exactly n, as the maximum value of $S_i(v)$ is the size of the whole network. Hence, the points that interest us are those with $S_i(v) = n$, which is necessarily the case for $S_n(v) = n$, although the leveling off can occur much earlier, depending on the network structure[2].

The point $S_n(v) = n$ corresponds in the fitted function (Equation 6) to the solution of this equation:

$$n = a\ln(n) + b. \tag{7}$$

[2] When i reaches diam(G), the diameter of G, which is the maximum distance in terms of number of edges on the shortest paths between any two nodes, necessarily the BFS has reached the entire network regardless of the starting vertex v. Typically diam$G \ll n$, although in the worst case diam$G = n - 1$.

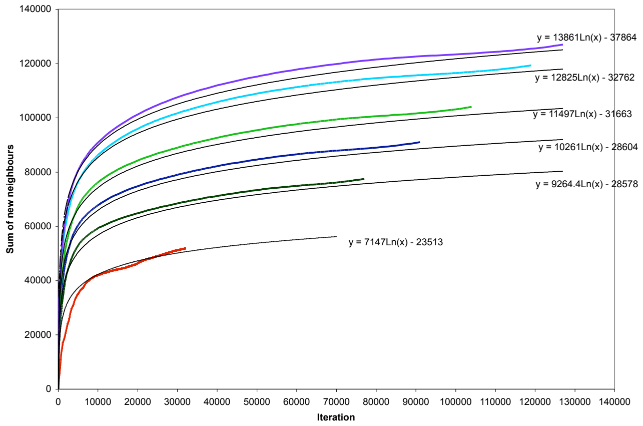

Figure 2. Accumulated sum of new neighbors (in color) along with their representative function (drawn black)

Equation (8) in itself is solved by

$$n = -a \cdot W(-\frac{e^{-b/a}}{a}), \tag{8}$$

where $W(z)$ is the Lambert's function [4](also known as the *Omega function*). Lambert's W function can be calculated by the following recurrence relation [5],

$$w_{j+1} = w_j - \frac{w_j e^{w_j} - z}{e^{w_j}(w_j + 1) - \frac{(w_j+2)(w_j e^{w_j} - z)}{2w_j+2}}, \tag{9}$$

included in mathematical software packages such as Matlab and Maple.

Our equation $a\ln(n) + b - n = 0$ (Eq. 7) has two roots (see Figure 3), and the network size is given by the *second root*. To obtain this second root using Lambert's W function, is its non-principal value which has to be calculated. In our case, as $z \in [-1/e, -0.1]$, an initial value of $w_0 = -2$ should be used in the recurrence (Eq. 9).

Seeking to simplify the situation, we studied the correlations of the coefficients a and b of Equation 6 in the fits to the real-world data obtained by computing the values of $S_i(v)$ from different nodes of the data set up to $i = n$. As a positive surprise, we found a strong, stable correlation:

$$b = c_1 a + c_2, \tag{10}$$

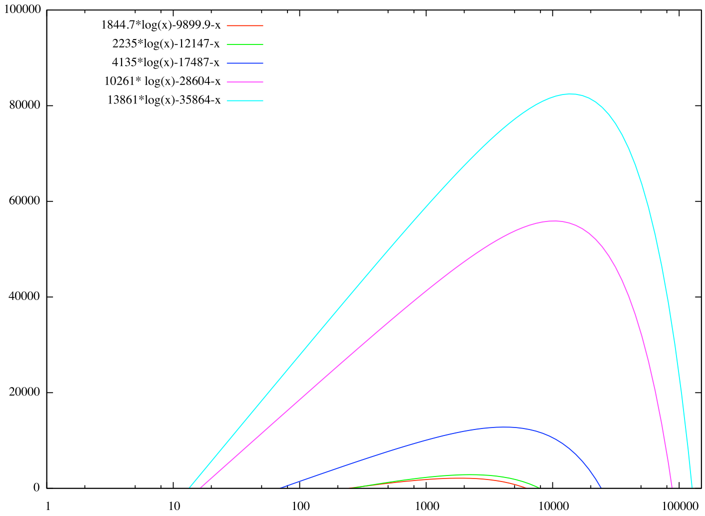

Figure 3. The two roots of $a\ln(x) + b - x = 0$

where c_1 and c_2 are constants that do not appear to depend strongly on the starting node v and could be obtained numerically from a small data set of the P2P network. From our data set, we estimated the values of $c_1 = -2.0552$ and $c_2 = -7840$ that turned out to be well-behaving. We can therefore replace the constant a by the following substitution:

$$a = \frac{y - c_2}{\ln(x) + c_1}. \tag{11}$$

The precision of our estimate of n will depend on the correctness of the estimation of a. Once the estimation algorithm reaches a stable value for a, the estimate of n can be computed. The estimation of a is done by starting a BFS at a node v, setting a_0 at an initial guess (we used $5,000$), computing at each iteration i the value

$$a' = \frac{S_i(v) - c_2}{\ln(i) + c_1}, \tag{12}$$

based on which a_i is computed as

$$a_i = \frac{1}{i}((i-1)a_{i-1} + a'). \tag{13}$$

The BFS is cut off when $a_i \approx a_{i-1}$, that is, when the estimate of a no longer significantly changes. In the next section, we study at which number of iterations i the estimate of a, and hence the estimate of n, typically stabilizes.

Should this iteration count be significantly smaller than the diameter of the P2P network (around 14 in our data set), the estimation method would provide an estimate on the network size with less communication overhead than the straightforward method. Another option would be to simply give a TTL for the BFS to explicitly cut it off after a fixed number of iterations.

3. Evaluation of the estimation method

We evaluate our algorithm by computing the estimation error, defined as

$$\epsilon = \frac{100|\text{real size} - \text{estimated size}|}{\text{real size}}\% \tag{14}$$

by each new neighbor visited and by the depth-level i of the BFS. Figure 4 shows that, for small networks (with less than $80,000$ nodes), running $5,000$ iterations (10% of whole network) is sufficient to reach an estimation error of 10 percent, and around $26,000$ iterations suffice for larger networks (with over $100,000$ nodes) to achieve same estimation error.

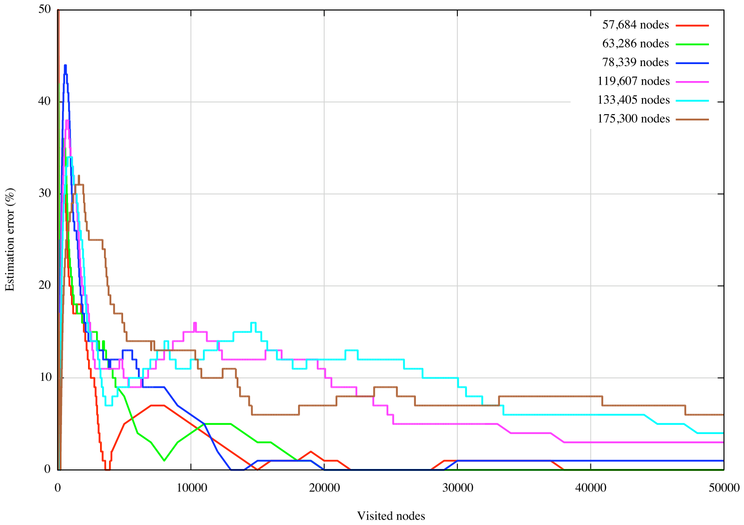

Figure 4. Estimation error (% of the network real size, Eq. 14) by iteration.

Using a fixed TTL for the BFS, our experiments show that for all networks, finishing the third depth-level of BFS, the estimation error is at most 10 percent (see Figure 5). This is a promising result for quickly estimating the size of a P2P network.

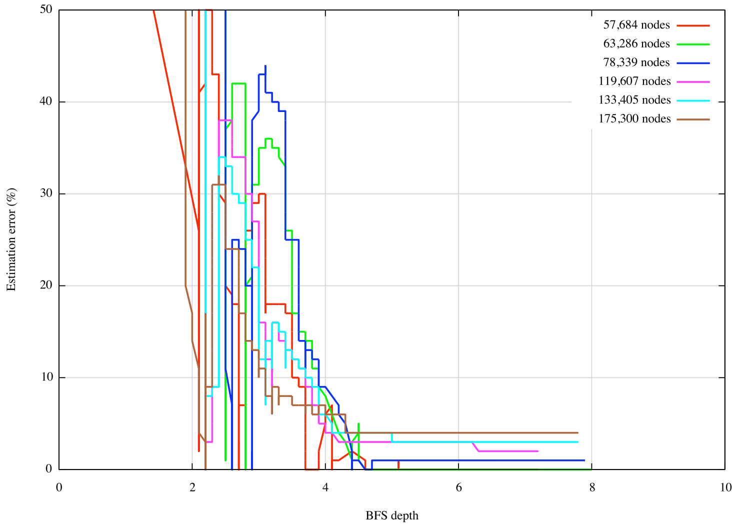

Figure 5. Estimation error (% of network real size, Eq. 14) using a fixed TTL for the BFS.

4. Conclusions and Future Work

We presented a novel approach for estimating the size of a peer-to-peer (P2P) network, fitting the sum of new neighbors discovered at each iteration of a breadth-first search (BFS) with a logarithmic function, and then using Lambert's W function to solve a root of $a\ln(n)+b-n=0$, where n is the network size. With rather little computation, we reached an estimation error of at most 10 percent, only allowing the BFS to iterate to the third level.

As future work, we plan to study the effect of fitting instead a function $d\ln(\ln(x))$ that have a better fit to the real data. This function is more difficult to manage analytically, but could enable us to fine-tune of the parameters of the proposed method. Additionally, possible effects of network structure, such as the presence of clustering, on the values of c_1 and c_2 (of Equation 10) are of interest. We also plan to implement our estimation algorithm in real systems, where it can be of direct practical use.

Acknowledgments

This work was partially funded by CoreGrid NoE and NIC Labs. The work of Dr. Schaeffer is partially funded by PROMEP.

References

[1] Boris Pittel. “On Spreading a Rumor.” *S*IAM Journal on Applied Mathematics, Vol. 47 (1): 213–223, 1987.

[2] Romualdo Pastor-Satorras and Alessandro Vespignani. “Epidemic spreading in scale-free networks.” *Physical Review Letters, Vol. 86 (14): 3200-3203, 2001.*

[3] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. “Epidemic algorithms for replicated database maintenance.” *In Proceedings of the Symposium on Principles of Distributed Computing, pp. 1–12, August 1987.*

[4] Leonhard Euler. “De serie Lambertina Plurimisque eius insignibus proprietatibus.” *Acta Acad. Scient. Petropol. 2, 29–51, 1783.* Reprinted in Euler, L. Opera Omnia, Series Prima, Vol. 6: Commentationes Algebraicae. Leipzig, Germany: Teubner, pp. 350–369, 1921.

[5] Rob Corless et al. “On the Lambert W function” *In Advanced Computational Mathematics. Volume 5, 1996.*

[6] Alexandru Iosup, Pawel Garbacki, Johan Pouwelse, and Dick Epema. “Correlating Topology and Path Characteristics of Overlay Networks and the Internet". *In Proceedings of Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2006), 16–19 May 2006, Singapore.*

[7] Alexandru Iosup, Pawel Garbacki, and Dick Epema. “Provisioning and Scheduling Resources for World-Wide Data-Sharing Services”, *In Proceedings of IEEE International Conference on e-Science and Grid Computing (e-Science), December 2006.*

[8] Arnaud Legout, Nikitas Liogkas, Eddie Kohler, and Lixia Zhang. “Clustering and sharing incentives in BitTorrent systems”, *In Proceedings of the 2007 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS 2007, San Diego, California, USA, June 12–16, 2007.*

[9] Keren Horowitz and Dahlia Malkhi. “Estimating network size from local information” *Information Processing Letters, Vol. 88 (5): 237–243, 2003.*

[10] Dorian Arnold, Gary Pack, and Barton Miller. “Tree-based overlay networks for scalable applications”. *In Proceedings of 20th International Parallel and Distributed Processing Symposium (IPDPS 2006), 25–29 April 2006, Rhodes Island, Greece.*

[11] Márk Jelasity, Spyros Voulgaris, Rachid Guerraoui, Anne-Marie Kermarrec, and Maarten van Steen. “Gossip-based peer sampling”. *In ACM Transactions on Computer Systems. Vol. 25 (3), 2007.*

[12] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. “On Power-law Relationships of the Internet Topology”, *In Proceedings of the ACM SIGCOMM’99 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, Cambridge, MA, USA,* pp. 251–262. ACM Press, 1999.

[13] Duncan J. Watts and Steven H. Strogatz. “Collective Dynamics of ’Small World’ Networks”, *In Nature. Vol. 393 (6684): 440–442, 1998.*

[14] Albert-László Barabási and Réka Albert. “Emergence of scaling in random networks”, *In Science. Vol. 268: 509–512, 1999.*

[15] Satu Elisa Schaeffer. “Algorithms for nonuniform networks”. *Doctoral thesis, Helsinki University of Technology, Department of Computer Science and Engineering, Espoo, Finland, April 2006.*