IEICE TRANS. INF. & SYST., VOL.E94–D, NO.9 SEPTEMBER 2011

LETTER

Assessing the Impact of Node Churn to Random Walk-Based Overlay Construction

Kyungbaek KIM[†a)], ***Member***

SUMMARY Distributed systems desire to construct a random overlay graph for robustness, efficient information dissemination and load balancing. A random walk-based overlay construction is a promising alternative to generate an ideal random scale free overlay in distributed systems. However, a simple random walk-based overlay construction can be affected by node churn. Especially, the number of edges increases and the degree distribution is skewed. This inappropriate distortion can be exploited by malicious nodes. In this paper, we propose a modified random walk-based overlay construction supported by a logistic/trial based decision function to compensate the impact of node churn. Through event-driven simulations, we show that the decision function helps an overlay maintain the proper degree distribution, low diameter and low clustering coefficient with shorter random walks.

key words: *random walk, random scale-free overlay, node churn, decision function*

1. Introduction

In these days, many distributed systems exploit overlay networks where each node is connected to each other with overlay links. The topology of overlay networks is very diverse based on the purpose of the system (e.g. a tree, a mesh, a random scale-free, and etc. [1]–[4]). In this paper, we focus on a random scale-free overlay which has a small diameter and a small clustering-coefficient. With a random scale-free overlay, distributed file systems/content distribution systems can search/disseminate data efficiently and effectively [2]–[4]. Also, it can improve the fault resilience against random overlay failure [1].

To obtain an ideal random scale-free overlay, a node maintains the given number of neighbors which spread uniformly on the participated nodes in a distributed system. One way getting a random neighbor uniformly is using a centralized sampling service which has a global view of the system [11], [12]. However, in distributed systems where a node only knows a small fraction of nodes, such as its overlay neighbors, the centralized sampling service is very unlikely.

A possible alternative is using a random walk based sampling [5]. To get a random neighbor, a node initiates a random walker and forwards the walker to one of its current overlay neighbors. This random walk continues until the walker visits sufficient number of nodes. The distribution of the last node of a sufficiently long random walk ($O(\log n)$) over a fast-mixing graph exhibits a stationary distribution, especially a uniformly random distribution [6]. According to this, a random walk based overlay construction can build a random scale free overlay in distributed systems.

However, nodes in distributed systems may be unreliable and frequently join/leave. Because of the dynamic behavior of participated nodes, a random scale-free overlay constructed by random walk based neighbor selection can be biased from the *initial overlay* which is ideally constructed by all nodes of the system without node churn. If a node keeps the number of neighbors above D to build an overlay, it tries to get a new random neighbor whenever it detects that its number of neighbors drops below D caused by node leave. In this case, the new neighbors are selected from the nodes which are currently online. Let us assume that there is a random-scale overlay graph $G(V, E)$, where $|V| = N$ and $|E| = M$. If some nodes leave, the overlay can be modified to $G'(V', E)$, where $|V'| = N' < N$ and $|E| = M$. Note that the modified overlay G' has fewer nodes but it still has same number of edges. After a while, if the nodes rejoin, each of them tries to pick its neighbors without any consideration of the modified overlay. Eventually, the overlay can be changed to $G''(V, E')$, where $|V| = N$ and $|E'| = M' > M$. That is, the total number of edges of the overlay increases because of node churn. Moreover, the additional edges crowd on long lived nodes and they can get high maintenance overhead. If malicious attackers live long intentionally, they gather many neighbors and exploit them to subvert the system.

In this paper, we propose a modified random walk-based overlay construction supported by a decision function which helps an overlay keep the properties of the initial overlay. To pick a random neighbor, a node initiates a random walker and forwards it to the next node which is picked among its current neighbors by a random walk-based walking function. When the random walker visits a node, it refers a decision function to pick this node as a new neighbor of its initiator. The decision function makes a decision based on the current number of neighbors of a node and the length of a random walk. As considering the number of neighbors, the constructed overlay can keep the properties of the initial overlay under node churn. The length of a random walk is related to fast construction of an overlay. The random walker keeps walking until any one node is selected as a new neighbor.

Manuscript received March 10, 2011.
[†]The author is with the Department of Computer Science, University of California, Irvine, USA
a) E-mail: kyungbak@uci.edu
DOI: 10.1587/transinf.E94.D.1830

2. Modified Random Walk-Based Neighbor Selection with a Decision Function

2.1 Walking Function

Before describing the detail of a decision function, we explore random walk-based walking functions. The purpose of a walking function is choosing the next node from a set of neighbors to which a random walker is forwarded, and eventually supports that the random walker can visit nodes in an overlay uniformly at random. We considered two walking functions: random walk-based function (RW) and re-weighted random walk-based function (RWRW).

In RW, the next node is selected uniformly at random from the neighbors of a node. The walking probability that a node i selects a node j which is one of its neighbor (wp_{ij}) is only dependent on i's degree like Eq. (1). This walking probability seems to be uniform from the local view of a single node, but it can be biased to the high degree nodes from the global view. Since high degree nodes has more chance to be selected than low degree nodes, a random walker may visit the high degree nodes more often than others.

$$wp_{ij} = \frac{1}{d_i}, j \in N_i, (N_i : \text{a set of neighbors of } i) \tag{1}$$

To impede the biased walking, RWRW re-weight the walking probability which is inversely proportional to the degree of node j like Eq. (2). For re-weighting the probability, RWRW requires degree information of neighbors. As more degree a node has, its walking probability becomes smaller. Despite the nodes with high degree can get more chance to be a next node than other nodes, their walking probability is smaller than others. According to this, RWRW can visit nodes more uniformly at random than RW.

$$wp_{ij} = \frac{1/d_j}{\sum_{k \in N_i} 1/d_k}, j \in N_i \tag{2}$$

2.2 Decision Function

The main purpose of a decision function is controlling the number of degree of a node to obtain the desired properties of an overlay. A decision function also affects the length of a random walk required to get a new neighbor. A decision function generates the decision probability (dp_i) that a node i can be a new neighbor in (0,1) based on the current degree of a node (d_i) as well as the length of the current random walk (t). To make a decision whether a random walker picks the currently visiting node i as a new neighbor, the random walker takes a random value in (0,1). If the random value is smaller than dp_i, the node i is selected as a new random neighbor.

We consider a decision function called a logistic/trial-based decision function (LT) like the Eq. (3). LT decision function exploits a logistic function which considers the current degree of a node (d_i) as well as the given degree (D). Two leftmost terms in Eq. (3) shows the probability value calculated by the logistic function. If d_i is smaller than D, dp_i becomes big. Otherwise, if d_i approaches to or exceeds D, dp_i decreases enormously.

If a decision function only leans to the logistic function, it takes very long random walk to obtain a new random neighbor. To reduce the required length of a random walk, LT decision function exploits the length of a random walk (t) for calculating dp_i like two rightmost terms in Eq. (3). Whenever a random walker is forwarded to a next node, t is increased by 1. As t increases, dp_i should increase to expedite picking up a new random neighbor. In LT decision function, dp_i increases logarithmically by t, and the ratio of increment is adjust by γ ($\gamma > 0$).

Since LT decision function is composed of two parts, there is a trade-off between effectiveness and fastness. If γ is big, the probability value related to t dominates dp_i, and the required length of a random walk decreases. But a random walker loses the chance to find a better neighbor which can meet the desired properties of an overlay. On the other hand, if γ is small, the probability value related to d_i dominates dp_i until t becomes very big. In this case, a random walker can have more chance to select a better neighbor, but it takes a long random walk. More detail comparison is described in Sect. 3.

$$dp_i = 1 - \frac{1}{1 + e^{D-d_i}} + 1 - \frac{1}{e^{\gamma \cdot t}}, \text{ where } d_i \geq 1, t \geq 0 \tag{3}$$

3. Evaluation

We evaluate the effectiveness of the proposed decision function to random walk-based neighbor selection under node churn through event-based simulations. We implemented a simulator emulating dynamic node behaviors by assigning on-time/off-time of a node at the time of its join/leave event. The distribution of on-time/off-time is the exponential distribution with given average values. With our given setting, around 2% of total number of nodes join or leave the system within a unit time period. Whenever a new node joins, it tries to get D neighbors. For a node to leave, its neighbor nodes, whose number of neighbors becomes below D, add more neighbors up to D. We set D as 4 and use 10000 nodes.

In results, "Initial" means the initial degree distribution of the overlay generated by using ideal random neighbor selection. "RW" and "RWRW" means the degree distribution for random walk based neighbor selection and re-weighted random walk based neighbor selection after 1000 unit time, respectively. For the simple random walk-based neighbor selection, the length of a random walk set to 14. "RW-LT" and "RWRW-LT" represents the modified random walk based neighbor selections with the proposed decision function, LT.

We consider three properties of a random overlay graph: Degree Distribution, Diameter and Clustering coefficient. The degree of a node is the number of neighbors, in a undirected graph. For effective and efficient information dissemination, the degree distribution should be balanced and

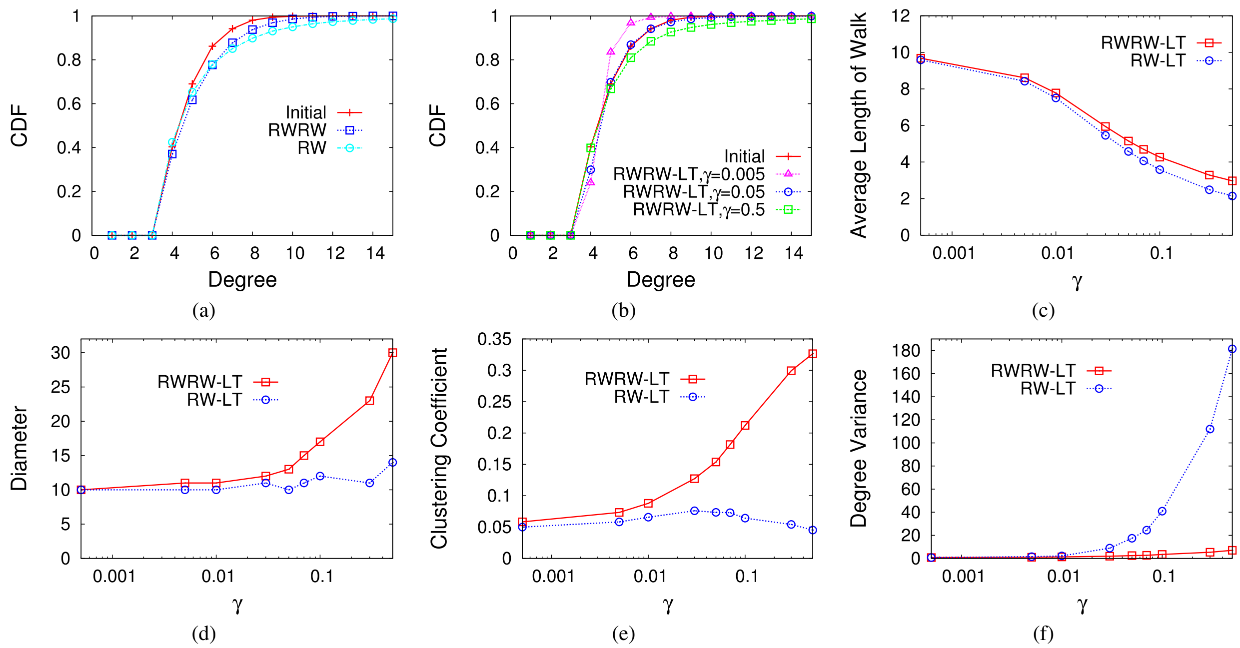

Fig. 1 Results of event-driven simulations with 10000 nodes. Except Initial case, parameters are measured after 1000 unit time. On every one unit time, around 2% of membership of an overlay changes.

uniformly distributed. That is, it is desirable to have a degree distribution with low variance. Diameter is the maximum path length between any two nodes in an overlay graph. As diameter is smaller, nodes in the graph communicate with each other faster. Clustering coefficient represents the likelihood that two neighbor nodes, A and B, share another node C as their neighbor. It is desired to have small diameter as well as small clustering coefficient for a good expander network.

3.1 Node Churn and LT Decision Function

The Fig. 1 (a) represents the degree distribution of the overlay constructed by random walk based neighbor selection. We observed that RWRW and RW are skewed from the initial distribution. This skewed distribution means that the number of internal edges increases. Specifically, there are 25654 edges at the initial time, but after 1000 unit time it increases to 26879 and 28554 for RWRW and RW, respectively. That is, around 5–11% of additional edges are required to maintain an overlay during churn. The main reason is that a random walk tends to focus on few long lived nodes having high degree. Despite RWRW considers the heterogeneity of node degree during random walking, it can not resolve the skewed distribution.

The Fig. 1 (b) shows that how LT decision function helps an overlay keep the degree distribution under node churn. When γ is 0.05, RWRW-LT almost fits its degree distribution to the initial state, and the number of edges becomes 25818 which is almost same to the initial state. Since of the logistic function, LT decision function suppresses that high degree nodes obtain more neighbors. Also, we observed that the overlay constructed by using LT decision function achieves low diameter and low clustering-coefficient like Fig. 1 (d) and 1 (e). For RW-LT, we got very similar results and we omitted their results.

3.2 Effect of γ

The Fig. 1 (c) represents the average length of a random walk for a node to get a random neighbor as a function of γ. The length of a random walk can be thought as the number of messages, and it should reduce to minimize maintenance overhead. As we discussed in Sect. 2, the length of a random walk can be controlled by the LT decision function, especially, by γ. As γ increases, a random walker picks a random neighbor earlier. On the other hand, if γ increases, a random walker wanders over an overlay until it meets a node having fewer neighbors than D.

In the aspect of reducing overhead as well as walking time, big γ can be preferred. But it can incur side effects such as skewed degree distribution, high diameter, high clustering coefficient or high degree variance. That is, a too short random walker can not visit enough number of nodes to find a better random neighbor. The side effect of big γ depends on the walking function. It affects diameter and clustering coefficient for RWRW, and affects degree variance for RW. For RW, the probability that a random walker visits higher degree nodes is more than the other nodes. With a short random walk, RW picks higher degree nodes most likely and their degree keeps increasing by time. But these higher degree nodes can help an overlay keep low diameter and low clustering coefficient like Fig. 1 (d) and 1 (e). On the other hand, RWRW walking function forwards a random walker more uniformly than RW, and the node variance is not affected much by the length of a random walk like

Fig. 1 (f). But, with a short random walk, it is hard to reach high degree nodes and the diameter (also, the clustering coefficient) of a constructed overlay becomes high.

Too small γ is not a good choice either. As γ decreases less than 0.03, the overhead keeps increasing, but there is no benefit to other parameters like Fig. 1 (d), 1 (e), and 1 (f). A sufficiently long random walk helps a random walker obtain a random neighbor which locates far away over an overlay, and keeps low diameter as well as low clustering coefficient. Also, it gives moare chance to get a new neighbor having fewer number of neighbors rather than high degree nodes, and reduces node variance. However, once the lengh of a random walk becomes more than the sufficient length, the effectiveness of LT decision function becomes saturated.

According to this, we set γ as the value which makes the probability that a sufficiently long random walk ($O(\log n)$) is accepted around 0.5. In our simulation using 10000 nodes, we can set γ as 0.05 to obtain desired properties overlay with low overhead.

4. Related Works

There are some previous efforts related to the overlay construction in distributed systems. Some efforts exploited distributed structures, especially, Hamilton cycles ([7], [8]) or Interleaved Spanning Trees ([9]), to obtain a new random neighbors. In the case of high node churn, these approaches require very high overhead. [10] proposed a membership exchange-based overlay management. But it incurs long convergence time which becomes severe under high node churn. [5] presented link weighting methods based on node heterogenity to enhnace the performance of random walk-based overlay construction. In this paper, we mostly focus on how to minimize the impact of node churn on the random walk-based overlay construction, even with a link weighting method.

5. Conclusion

In distributed systems, node churn is inevitable. The distributed overlay construction is viable through a random walk based neighbor selection, but high node churn distorts the degree distribution of the constructed overlay and increases the required number of edges up to 11%. We propose a logistic/trial based (LT) decision function to support a random walk-based overlay construction to maintain the overlay having desired properties. LT decision function can control the length of the random walk by adjusting γ and it affects the properties of the constructed overlay.

References

[1] K. Kim and N. Venkatasubramanian, "Assessing the impact of geographically correlated failures on overlay-based data dissemination," Proc. IEEE Globecom 2010, Dec. 2010.
[2] M. Deshpande, B. Xing, I. Lazardis, B. Hore, N. Venkatasubramanian, and S. Mehrotra, "CREW: A gossip-based flash-dissemination system," Proc. ICDCS 2006, 2006.
[3] R. Morselli, B. Bhattacharjee, A. Srinivasan, and M. Marsh, "Efficient lookup on unstructured topologies," Proc. ACM PODC 2005, 2005.
[4] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Gossip algorithms: Design, analysis and applications," Proc. IEEE INFOCOM, 2005.
[5] V. Vishnumurthy and P. Francis, "On heterogeneous overlay construction and random node selection in unstructured P2P networks," Proc. IEEE INFOCOM, 2006.
[6] M. Mitzenmacher and E. Upfal, Probability and Computing, Cambridge University Press, 2005
[7] C. Law and K. Siu, "Distributed construction of random expander networks," Proc. INFOCOM 2003, 2003.
[8] C. Gkantsidis, M. Mihail, and A. Saberi, "Random walks in peer-to-peer networks: Algorithms and evaluation," Performance Evaluation, vol.63, no.3, pp.241–263, March 2006.
[9] A. Young, J. Chen, Z. Ma, A. Krishnamurthy, L. Peterson, and R.Y. Wang, "Overlay mesh construction using interleaved spanning trees," Proc. INFOCOM 2004, 2004.
[10] S. Voulgaris, D. Gavidia, and M. Steen, "CYCLON: Inexpensive membership management for unstructured P2P overlays," J. Network and systems Management, vol.13, no.2, pp.197–217, June 2005.
[11] A.S. Rodionov and H. Choo, "On generating random network structures: Connected graphs," Proc. ICOIN 2004, 2004.
[12] R. Milo, N. Kashtan, S. Itzkovitz, M.E.J. Newman, and U. Alon, "On the uniform generation of random graphs with prescribed degree sequences," arXiv, arXiv:cond-mat/0312028v, May 2004.