Searching method through biased random walks on complex networks

Sungmin Lee, Soon-Hyung Yook,[*] and Yup Kim[†]
Department of Physics and Research Institute for Basic Sciences, Kyung Hee University, Seoul 130-701, Korea

(Received 14 December 2008; revised manuscript received 10 March 2009; published 15 July 2009)

Information search is closely related to the first-passage property of diffusing particle. The physical properties of diffusing particle is affected by the topological structure of the underlying network. Thus, the interplay between dynamical process and network topology is important to study information search on complex networks. Designing an efficient method has been one of main interests in information search. Both reducing the network traffic and decreasing the searching time have been two essential factors for designing efficient method. Here we propose an efficient method based on biased random walks. Numerical simulations show that the average searching time of the suggested model is more efficient than other well-known models. For a practical interest, we demonstrate how the suggested model can be applied to the peer-to-peer system.

DOI: 10.1103/PhysRevE.80.017102 PACS number(s): 89.75.Hc, 05.40.Fb, 89.20.Hh

The interplay between network topology and dynamical process has been an important topic in complex network studies [1–7]. Examples include information search [1–4], diffusive particle systems [5], epidemic spreading [6], and coupled oscillators [7]. Among these studies, information search has attracted many researchers due to its possible applications in diverse fields ranging from communication networks to social networks. Especially, peer-to-peer (P2P) system is one of the popular examples of information search. In P2P systems, the wanted information is located at several nodes in a network, and nodes exchange the wanted information directly with each other. Many studies on the large-scale topology of P2P networks have uncovered that the distribution of a node with degree k follows the scale-free (SF) distribution $P(k) \sim k^{-\gamma}$, with $\gamma < 3$ [2,8–14], or highly skewed fat-tailed distributions [2,11–13,15]. In SF networks, several nodes have most of degrees or connections. These nodes are called hubs and many important properties of complex networks are dominated by them [9,14]. Since the performance of P2P protocol is crucially affected by the underlying topology [8], a good P2P protocol should take advantage of the underlying topology. However, in many popular P2P protocols, the underlying topology is not implemented. This poorly designed P2P protocols cause very significant problems by consuming the bandwidth of the Internet [16,17].

The two most popular algorithms used in many P2P applications [18–22] are (1) the flooding-based (FB) query-packet-forwarding algorithm and (2) the n-random walker (n-RW) model. The FB algorithm [8] spreads the query packets to all nodes within a preassigned diameter. Thus, this algorithm causes significant traffic congestion. In n-RW model [2,8], n query packets are generated and take random walks along the P2P connections. n-RW model can cause long waiting time because of the dynamical properties of RWs on complex networks [23]. Specifically, Adamic *et al.* suggested a diffusing particle model (1-RW) when the exact location of the wanted information is unknown in a network [2]. The searching efficiency of the model of Adamic *et al.* is closely related to the first-passage property of diffusing particle.

Recently, we have studied the survival probability $S(t)$ of prey particles in diffusive capture process on complex networks and successfully applied it to model the P2P system [4]. We suggested N lions-lamb (NLL) model inspired by recent discoveries in diffusive capture process [5,24]. In the NLL model, both the advantage of the underlying structure and the property of the diffusive capture process are used in order to improve the performance of information search. Not only the query packet but also the information packets are generated. All the generated packets take random walks. The NLL model has two benefits compared to the other models. First the amount of traffic is always constant and much less than the FB algorithm. Second it has much less searching time than that for n-RW model. However, the average searching time of the NLL model is still larger than that of the FB algorithm [4]. In this Brief Report, we introduce a biased NLL (BNLL) model based on biased random walks. Using the numerical simulations, we show that the average searching time of the BNLL model drastically decreases down to the level of the FB algorithm.

In the BNLL model, each node sends out an information packet whose main part consists of names of files stored in it [25], regardless of the existence of query events. Each of these packets takes biased random walks along the P2P connections. The probability that a walker at a node i moves to one of its nearest neighbors j is given by

$$P_{ij} = \frac{k_j^{\alpha}}{\sum_{l \in \Gamma_i} k_l^{\alpha}}. \tag{1}$$

Here, k_j is the degree of node j and Γ_i represents the set of i's nearest neighbors. The exponent α represents the degree of bias. For example, if $\alpha > 0$ then the walker prefers to move a node of large k_i. Independently, a randomly chosen node sends out one query packet to find a specific file. The query packet also takes the same biased random walks as the information packets. Thus, not only the querying packet moves but also the information of all files moves on the network. If

[*] syook@khu.ac.kr
[†] ykim@khu.ac.kr

the query packet meets an information packet that has the requested file name in its list, then the query packet is terminated but the information packet continues biased random walks for the next query.

A specific case, the one lamb and one lion problem has been studied [26]. In this model, two biased walkers are initially located at two randomly selected nodes. Then the probability to meet the walkers at node i of degree k_i is

$$P_i \sim k_i^{2(\alpha+1)}. \qquad (2)$$

Equations (1) and (2) imply that the bias to the larger degree of node becomes stronger when α increases. Thus, in general, there is a strong tendency that the walker (or packet) prefers to stay around hubs when $\alpha > 0$. As a result the hubs become strong attractors when $\alpha > 0$. Such effects enhance the searchability in P2P networks.

In the simulation, we assume n_f available files on the given network. Each node of the network is assumed to have one randomly chosen file among n_f files and sends out an information packet with the name of the file stored in it, its internet protocol (IP) address, etc. And a randomly chosen node sends out a query packet to find one randomly chosen item among n_f files. We consider two possible situations: (1) n_f is fixed for each given network and (2) $n_f = \rho N$. Here, N is the number of nodes in a given network.

In order to investigate the effect of underlying topology, we use two kinds of networks for P2P networks. One is the theoretical SF network with $\gamma = 2.4$ to generate the virtual P2P network. For this we use the method suggested by Goh *et al.* [27]. To compare the scalability, we use the networks of various sizes, $N = 10^3 - 10^6$. The other kind is the snapshot of a real Gnutella topology obtained from Refs. [10,11,15], which has 1 074 843 ($\approx 10^6$) nodes. The known characteristics of Gnutella network are as what follows: $P(k)$ distribution of Gnutella follows a power law or is fat tailed. This implies that there exist hubs. These hubs in the Guntella network play an important role. For example, it is closely related to the reachability of a query and also related to the mean distance to other pairs [15]. Therefore, the dynamical properties of the searching algorithm can be affected by the characteristics of these hubs. To extract Gnutella subnetworks having $N = 10^3 - 10^5$ nodes from the snapshot, we use random walk sampling method [28]. The subnetwork sampled by random walk method is known to inherit the topological properties from the original network. All quantities measured in the simulations are averaged over ten network realizations and 100 different histories for each network realization.

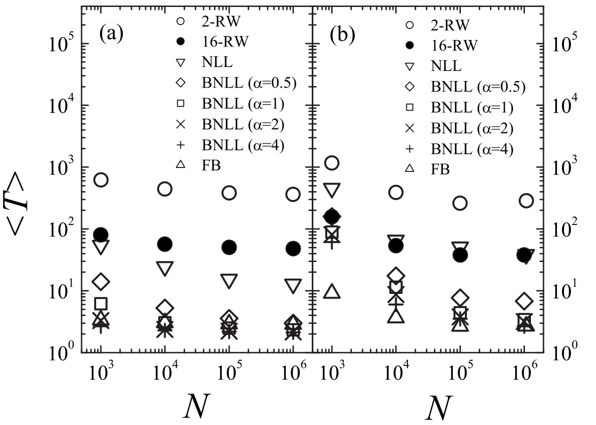

FIG. 1. Log-log plot of the average time $\langle T \rangle$ taken to find the requested information with a fixed value of n_f (=500) on SF networks with (a) $\gamma = 2.4$ and (b) real Gnutella networks.

In Figs. 1 and 2, we show the average searching time $\langle T \rangle$ for each model. The searching time T is defined by the time taken to find the requested information. All packets have preassigned time-to-live (TTL) counter t_{TTL}, which decreases by 1 when the packet is forwarded to other node. When $t_{TTL} = 0$, the packet is removed. We use infinite t_{TTL} in the measurement of $\langle T \rangle$. Since the searching time of the n-RW model depends on the value of n, we need a criterion for n. If there are q simultaneous query events, then the traffic of the n-RW model is qn and that of the BNLL model is $N + q$. Here, the network traffic $f(t)$ is defined as the total number of packets on the network at time t. Note that each query event in n-RW and BNLL is independent to other query events. Thus, $\langle T \rangle$ does not depend on q, but $f(t)$ generated by each model is affected by q. When $q > N$, the resulting $f(t)$ of n-RW exceeds that of BNLL if $n > 1$. On the other hand, if $q < N$ then $f(t)$ of n-RW drastically decreases compared to that of BNLL. Thus, one can increase n of the n-RW model to have the comparable $f(t)$ with the BNLL model when $q < N$. Since the known optimal value of n is 16 [8], we also compare $\langle T \rangle$ of the 16-RW model with that of the BNLL model in Figs. 1 and 2, and we find that $\langle T \rangle$ for $n = 16$ scales in the same way with that for $n = 2$. Thus, we choose the

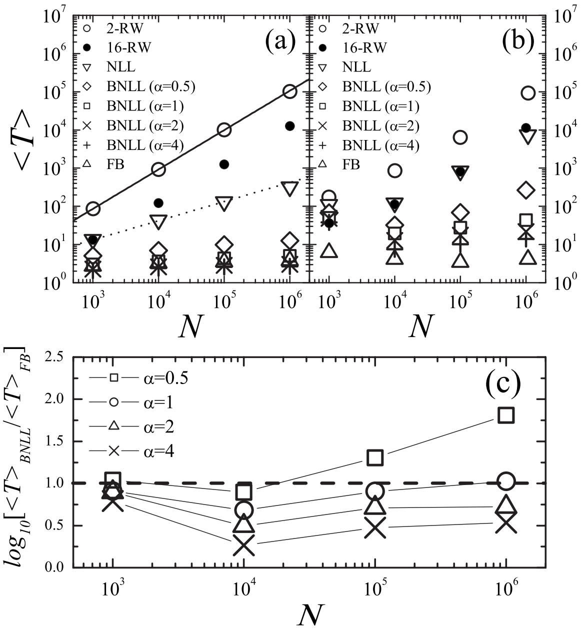

FIG. 2. (a) $\langle T \rangle$ for $n_f = \rho N$ ($\rho = 0.1$) on SF networks with $\gamma = 2.4$. Solid and dotted lines represent $\langle T \rangle \sim N^{1.0}$ and $\langle T \rangle \sim N^{0.5}$, respectively. (b) $\langle T \rangle$ for $n_f = \rho N$ ($\rho = 0.1$) on real Gnutella networks; $\langle T \rangle$ of 2-RW and NLLL models grows faster than the power law. (c) Plot of the $\log_{10}[\langle T \rangle_{\mathrm{BNLL}} / \langle T \rangle_{\mathrm{FB}}]$ against N for various α on real Gnutella networks. The dashed line represents $\log_{10}[\langle T \rangle_{\mathrm{BNLL}} / \langle T \rangle_{\mathrm{FB}}] = 1$.

condition $f_{n\text{-RW}}(t)=f_{\text{BNLL}}(t)$ when $q=N$ to compare $\langle T\rangle$ of both models under the condition with the same $f(t)$, so we fix $n=2$ in the following simulations. We use the parameters $\alpha=0.5$, 1, 2, and 4 to control the bias in the BNLL model.

We display $\langle T\rangle$'s for each model when $n_f=500$. As shown in Fig. 1(a), $\langle T\rangle$ decreases as N increases for the 2-RW, the NLL, and the BNLL ($\alpha=0.5,1$) models; while $\langle T\rangle$ for the FB algorithm and the BNLL ($\alpha=2,4$) model remains almost constant on SF networks. The $\langle T\rangle_{\text{BNLL}}$ model with $\alpha=0.5$ decreases much faster than those of the n-RW and the NLL models as N increases and approaches to $\langle T\rangle_{\text{FB}}$. When α increases further ($\alpha=1$), $\langle T\rangle_{\text{BNLL}}$ drastically decreases and becomes slightly less than $\langle T\rangle_{\text{FB}}$ for $N\geq 10^4$ on SF networks [Fig. 1(a)]. Moreover, when $\alpha\geq 2$, $\langle T\rangle_{\text{BNLL}}$ becomes almost the same as $\langle T\rangle_{\text{FB}}$ for any N. In Fig. 1(b), we display $\langle T\rangle$ for each model on the real Gnutella (sub)networks. As in the case of theoretical networks, the $\langle T\rangle_{\text{BNLL}}$ model with $\alpha=0.5$ and 1 on Gnutella network decreases faster than the n-RW and the NLL models. The BNLL model with $\alpha\geq 1$ shows almost the same efficiency as the FB algorithm in searching time for large N or $N\geq 10^6$. The average searching times on Gnutella networks also satisfy the inequality

$$\langle T\rangle_{\text{BNLL}}\approx\langle T\rangle_{\text{FB}}<\langle T\rangle_{\text{NLL}}<\langle T\rangle_{\text{2-RW}}, \tag{3}$$

when n_f is fixed. Therefore, as α increases, the BNLL model shows almost the same or better efficiency in searching time than the FB algorithm on both SF and Gnutella networks. This can be understood from the dynamical properties of RWs on complex networks. Since the probability that a RW visits a node with the degree k is proportional to k [23], the probability that a query packet finds a requested file at one node with degree k is proportional to k in the n-RW model. And in the NLL model, due to the random walking information packets, the probability is proportional to k^2 [5]. But in our BNLL model, the probability that two biased random walks meet at the same node with degree k is proportional to $k^{2(\alpha+1)}$ [Eq. (2)] [26]. Thus, the hubs in the BNLL model collect more packets and become stronger attractors than those in the NLL model as α increases. Therefore, $\langle T\rangle$'s of the 2-RW, the NLL, and the BNLL models decrease but $\langle T\rangle_{\text{2-RW}}-\langle T\rangle_{\text{BNLL}}$ and $\langle T\rangle_{\text{NLL}}-\langle T\rangle_{\text{BNLL}}$ increase as N increases.

We also measure $\langle T\rangle$'s for each model when $n_f=\rho N$ ($\rho=0.1$). The data in Fig. 2(a) show that $\langle T\rangle_{\text{2-RW}}$ increases almost linearly (or $\langle T\rangle\sim N^{\beta}$ with $\beta\sim 1.0$) and $\langle T\rangle_{\text{FB}}$ stays around 3 on the theoretic SF networks. And $\langle T\rangle_{\text{NLL}}$ grows as $\langle T\rangle_{\text{NLL}}\sim N^{\beta}$ with $\beta=0.5$ for small N, but β seems to be less than 0.5 or $\langle T\rangle$ becomes saturated to a fixed finite value for large N ($>10^5$) [24]. For the BNLL model, $\langle T\rangle_{\text{BNLL}}$ increases much slowly than those of the 2-RW and the NLL models. The increment of $\langle T\rangle_{\text{BNLL}}$ depends on α. As α increases, $\langle T\rangle_{\text{BNLL}}$ approaches to $\langle T\rangle_{\text{FB}}$. More specifically, $\langle T\rangle$ of each model satisfies the inequality $\langle T\rangle_{\text{FB}}<\langle T\rangle_{\text{BNLL}}<\langle T\rangle_{\text{NLL}}<\langle T\rangle_{\text{2-RW}}$ when $\alpha=0.5$. $\langle T\rangle_{\text{BNLL}}$ with $\alpha=1$, 2, and 4 drastically decreases and becomes almost the same as $\langle T\rangle_{\text{FB}}$ [see Fig. 2(a)].

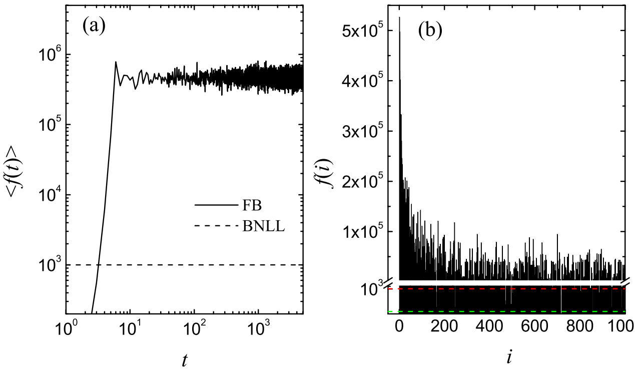

FIG. 3. (Color online) (a) Plot of the average traffic $\langle f(t)\rangle$ against t. (b) The traffic of FB algorithm measured at each node i, $f(i)$. The upper (red) and the lower (green) dashed lines represent the possible maximum traffic of BNLL and 2-RW models, respectively.

On the real Gnutella network, we are unable to find a power-law behavior of $\langle T\rangle$ for each model. However, $\langle T\rangle_{\text{2-RW}}-\langle T\rangle_{\text{BNLL}}$ and $\langle T\rangle_{\text{NLL}}-\langle T\rangle_{\text{BNLL}}$ also increase as N increases. Thus, the searching time of the BNLL model is more efficient than the 2-RW and the NLL models [see Fig. 2(b)]. As for $\langle T\rangle_{\text{BNLL}}$ on SF networks, the behavior of $\langle T\rangle_{\text{BNLL}}$ strongly depends on α. As α increases $\langle T\rangle_{\text{BNLL}}$ drastically decreases and becomes comparable to $\langle T\rangle_{\text{FB}}$. Figure 2(c) shows the plot of $\log_{10}[\langle T\rangle_{\text{BNLL}}/\langle T\rangle_{\text{FB}}]$ against N for various α on the real Gnutella networks. $\log_{10}[\langle T\rangle_{\text{BNLL}}/\langle T\rangle_{\text{FB}}]<1$ implies that $\langle T\rangle_{\text{BNLL}}$ has the same order of magnitude as $\langle T\rangle_{\text{FB}}$. As shown in Fig. 2(c), $\log_{10}[\langle T\rangle_{\text{BNLL}}/\langle T\rangle_{\text{FB}}]<1$ for $\alpha\geq 1$, and $\log_{10}[\langle T\rangle_{\text{BNLL}}/\langle T\rangle_{\text{FB}}]$ decreases as α increases. From Figs. 1 and 2, we show that the inequality $\langle T\rangle_{\text{BNLL}}<\langle T\rangle_{\text{NLL}}<\langle T\rangle_{\text{2-RW}}$ is always satisfied. Moreover, the BNLL model shows almost the same as the FB algorithm in searching time depending on α, when the underlying topology has the high degree of heterogeneity. We also find that there exists an optimal value of α, above which $\langle T\rangle$ does not drastically decrease as α increases for large N. This optimal value, α_{op}, depends on the underlying topology and n_f. For SF networks with $\gamma=2.4$, we find that $\alpha_{op}\simeq 1$ for both fixed n_f and $n_f=\rho N$ cases. On Gnutella networks, we find that $\alpha_{op}\simeq 1$ for fixed n_f and $\alpha_{op}\simeq 2$ when $n_f=\rho N$.

Let us discuss the traffic generated by each model. In Fig. 3(a), we compare $f(t)$ of FB algorithms to that of the BNLL model on SF networks with $N=10^3$ nodes and $n_f=5$. We assign $t_{TTL}=7$ for the FB algorithm. In order to prevent the overflow caused by a large number of packets in the FB algorithm, we assume that a new query event can occur when one of the query packets succeed to find the request file. In this case, the other query packets, which fail to find the request file, are forwarded until their t_{TTL}'s become 0. In the FB algorithm with finite t_{TTL}, the traffic generated during each query event is known to increase exponentially as $f(t=t_{TTL})\approx\langle k\rangle[\{\langle k^2\rangle-\langle k\rangle\}/\langle k\rangle]^{t_{TTL}-1}$, where $\langle k\rangle$ and $\langle k^2\rangle$ are the first and the second moments of the network degree distribution, respectively [4,9]. However, the traffic of the NLL and the BNLL models is always $N+1$ for successive query events. As shown in Fig. 3(a), the FB algorithm generates around 500 times more traffic than the BNLL model on the average [29]. If there are q simultaneous query events, then the average traffic for the FB algorithm increases by q times of the average traffic, but it becomes simply $q+N$ for the

NLL and the BNLL models. The traffic of the n-RW model is qn for q simultaneous query events. If $q=N$, then the traffic generated by the n-RW model can exceed the traffic of the BNLL model depending on the value of n. Thus depending on q the traffic generated by the n-RW model can exceed the traffic of the BNLL model.

Since the probability that a biased walker visits a node of degree k follows the Eq. (2), the hub can have a considerable amount of traffic. In order to compare the bottleneck traffic between the FB algorithm and the BNLL model, we measure the traffic generated by the FB algorithm at each node i, $f(i)$. By the definition of the static SF network (SFN) model [27], the smaller node index i has the larger k. As shown in Fig. 3(b), $f(i)$ of the largest hub $(i=1)$ exceeds 5×10^5, which is much larger than the possible maximum traffic of the BNLL model [$f_{max}=1001$ for $N=10^3$; see the (red) dashed line in Fig. 3(b)]. This huge amount of packets in the largest degree of node can cause severe traffic congestion in the Internet when the FB algorithm is applied. The mean-field argument for the traffic of each model is presented in Ref. [4].

This research was supported by the Kyung Hee University Research Fund in 2008 (Grant No. KHU-20080574).

[1] J. M. Kleinberg, Nature (London) **406**, 845 (2000).

[2] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman, Phys. Rev. E **64**, 046135 (2001).

[3] C. P. Herrero, Phys. Rev. E **71**, 016103 (2005); S.-J. Yang, *ibid.* **71**, 016107 (2005); H. P. Thadakamalla, R. Albert, and S. R. T. Kumara, *ibid.* **72**, 066128 (2005).

[4] S. Lee, S.-H. Yook, and Y. Kim, Physica A **385**, 743 (2007).

[5] S. Lee, S.-H. Yook, and Y. Kim, Phys. Rev. E **74**, 046118 (2006).

[6] R. Pastor-Satorras and A. Vespignani, Phys. Rev. Lett. **86**, 3200 (2001); Phys. Rev. E **63**, 066117 (2001).

[7] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, Phys. Rep. **424**, 175 (2006).

[8] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, *Search and Replication in Unstructured Peer-to-Peer Networks*, Proceedings of the 16th Annual ACM International Conference on Supercomputing, New York City, USA, 2002 (unpublished).

[9] R. Pastor-Satorras and A. Vespignani, *Evolution and Structure of the Internet: A Statistical Physics Approach* (Cambridge University Press, Cambridge, England, 2004).

[10] D. Stutzbach and R. Rejaie, *Capturing Accurate Snapshots of the Gnutella Networks*, Global Internet Symposium, 2005, Vol. 125.

[11] D. Stutzbach and R. Rejaie, *Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, (ACM, New York, 2005), p. 402, extended abstract.

[12] M. A. Jovanovic, F. S. Annexstein, and K. A. Berman, University of Cincinnati, Technical report, 2001 (unpublished).

[13] M. Ripeanu, I. Foster, and A. Iamnitchi, IEEE Internet Comput. **6**, 50 (2002).

[14] R. Albert and A.-L. Barabábasi, Rev. Mod. Phys. **74**, 47 (2002).

[15] D. Stutzbach and R. Rejaie, IEEE/ACM Trans. Netw. **16**, 267 (2008).

[16] http://www.theregister.co.uk/2003/10/14/edonkey_rides_like_the_wind/

[17] D. Plonka, Uw-Madison Napster Traffic Measurement, http://net.doit.wisc.edu/data/Napster, 2000.

[18] J. Buford and K. Ross, *P2P Overlay Design Overview*, IETF P2P-SIP Ad Hoc Meeting, 2005 (unpublished).

[19] BitTorrent Protocol Specification, http://www.bittorrent.org/protocol.html

[20] W. Nejdl *et al.*, *EDUTELLA: A P2P Networking Infrastructure Based on RDF* (ACM Press, New York, 2005).

[21] Gnutella Protocol Specification v0.4 and v0.6.

[22] http://www.kazaa.com

[23] J. D. Noh and H. Rieger, Phys. Rev. Lett. **92**, 118701 (2004).

[24] S. Kwon, S. Lee, and Y. Kim, Phys. Rev. E **73**, 056102 (2006).

[25] In practice, the names of the files in the information packet can be restricted to those which are intended to be shared by the owner for the security.

[26] S. Kwon, S. Yoon, and Y. Kim, Phys. Rev. E **77**, 066105 (2008).

[27] K.-I. Goh, B. Kahng, and D. Kim, Phys. Rev. Lett. **87**, 278701 (2001).

[28] S. Yoon, S. Lee, S.-H. Yook, and Y. Kim, Phys. Rev. E **75**, 046114 (2007).

[29] Since $\langle f(t)\rangle$ of FB algorithm is an exponential function of t_{TTL}, $\langle f(t)\rangle$ of FB can be smaller than that of BNLL when t_{TTL} is very small. However, when t_{TTL} is less than or equal to the obtained $\langle T\rangle$'s in Figs. 1 and 2, many packets fail to find the requested information and $\langle T\rangle$ diverges. Thus, t_{TTL} should be much larger than the obtained $\langle T\rangle$ ($\approx$4–5). As a result, $\langle f(t)\rangle$ of FB always exceeds that of BNLL when $\langle T\rangle$ is finite.