A Near-Optimal Algorithm Attacking the Topology Mismatch Problem in Unstructured Peer-to-Peer Networks

Hung-Chang Hsiao, *Member*, *IEEE Computer Society*, Hao Liao, and Po-Shen Yeh

Abstract—In an unstructured peer-to-peer (P2P) network (e.g., Gnutella), participating peers choose their neighbors randomly such that the resultant P2P network mismatches its underlying physical network, resulting in the lengthy communication between the peers and redundant network traffics generated in the underlying network. Previous solutions to the topology mismatch problem in the literature either have no performance guarantees or are far from the optimum. In this paper, we propose a novel topology matching algorithm based on the Metropolis-Hastings method. Our proposal is guided by our insight analytical model and is close to the optimal design. Specifically, we show that our proposal constructs an unstructured P2P network in which a broadcast message, originated by any node v, reaches any other node u by taking approximately the only physical end-to-end delay between v and u. In addition, our design guarantees the exponential broadcast scope. We verify our solution through extensive simulations and show that our proposal considerably outperforms state-of-the-art solutions.

Index Terms—Unstructured peer-to-peer systems, topology mismatch, location awareness, broadcast.

1 Introduction

Peer-to-peer (P2P) networking is an emerging technique for next-generation network applications. P2P networks (or *overlays*) are application-level networks built on top of end systems, which provide message routing and delivery. Although P2P networks (e.g., CAN [1], Chord [2], Pastry [3], and Tapestry [4]) can be structured in a well-organized fashion to guarantee the quality of services (e.g., the number of overlay hop count of routing a message), *unstructured* P2P networks are widely deployed in the mass market [5], [6], [7].

A critical function offered by an unstructured P2P network is the broadcasting of a message. Since the peers participating in an unstructured overlay interconnect with one another in a random fashion, one often resorts to the *time-to-live* (TTL) mechanism for broadcasting. Consider a peer v that demands the dissemination of a message to peers elsewhere in a system and first floods the message to its *neighbors*. By neighbors of v, we mean those peers that have direct end-to-end connections to v in the overlay network. Upon receiving a flooded message, the peer (say, u) decreases the associated TTL value by 1, and then, replicates the message to its neighbors if the reduced TTL value remains positive, and if u has not ever forwarded the same message [8].

Pioneering studies (e.g., Liu et al. [9]) presented that more than 70 percent of communication paths in an unstructured overlay do not exploit their underlying physical network topology (i.e., the Internet), leading to lengthy communication, and thus, *the topology mismatch problem*. In particular, the work in [9] showed that a naive mass market overlay network (i.e., Gnutella [8]) without matching the physical underlay can introduce network traffic that is approximately four times that of a perfectly matching overlay. As P2P network traffic dominates 70 percent of the total traffic in the Internet [7], resolving the topology mismatch problem can remarkably relieve the burden of the Internet.

Consider an unstructured overlay network represented as an undirected graph $G = (V, E)$, where the set of nodes (i.e., participating peers) and edges (i.e., overlay links) between nodes are denoted by V and E, respectively. Any node v in V may flood messages to the nodes elsewhere in the system. Given $G = (V, E)$, we aim to improve G such that the message flooding in G becomes efficient. By efficiency, we mean to reduce the *broadcast delay* from any node v to any node $u \in V - \{v\}$. Particularly, we intend to minimize the following:

$$\min_{p \in \mathcal{P}_{v \sim u}} \ell_p, \tag{1}$$

where $\mathcal{P}_{v \sim u} \subseteq G$ represents the set of all paths induced by the flooded messages due to v toward u, and ℓ_p denotes the total delay required for traversing the path $p \in \mathcal{P}_{v \sim u}$.

Let p be any path in $\mathcal{P}_{v \sim u}$, given any v and $u \in V - \{u\}$. Denote the number of edges (or the *hopcount*) on p by $|p|$. Our second objective in this study is to minimize $|p|$ as much as possible. Precisely, we are to minimize the following:

$$\left| \arg \min_{p \in \mathcal{P}_{v \sim u}} \ell_p \right|. \tag{2}$$

Minimizing $|p|$, in turn, maximizes the *broadcast scope* of disseminating a message, where the broadcast scope is

• *The authors are with the Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan 701, Taiwan. E-mail: hchsiao@csie.ncku.edu.tw.*

Manuscript received 16 Apr. 2009; revised 27 Aug. 2009; accepted 4 Sept. 2009; published online 30 Oct. 2009.
Recommended for acceptance by A. Boukerche.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2009-04-0167. Digital Object Identifier no. 10.1109/TPDS.2009.160.

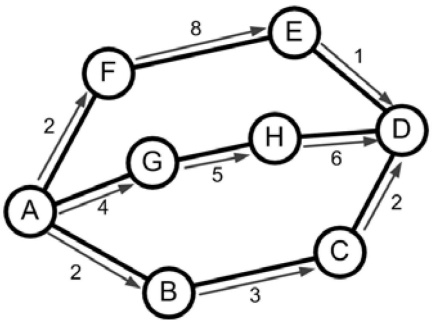


Fig. 1. An example of an unstructured P2P network $G = (V, E)$, where $V = \{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}, \mathtt{G}, \mathtt{H}\}$, $E = \{\mathtt{AB}, \mathtt{AF}, \mathtt{AG}, \mathtt{BC}, \mathtt{EF}, \mathtt{GH}, \mathtt{CD}, \mathtt{DE}, \mathtt{DH}\}$, and the number aside from any $vu \in E$ denotes the delay of forwarding a message from v to u.

defined as the number of distinct peers receiving the message with a specified TTL value. Notably, enlarging the broadcast scope is particularly important for P2P search applications since this increases the probability of discovering a requested object [9], [10], [11].

In the following discussion, the message broadcasting protocol as mentioned above is called the *scoped broadcast* (or broadcast for short) in this paper. Fig. 1 depicts an example of an unstructured P2P network that implements the scoped broadcast. Consider node A in the network. If $TTL = 3$, then the messages originated by A will follow three communication paths, namely, $\mathcal{P}_{\mathtt{A}\sim\mathtt{D}} = \{\mathtt{ABCD}, \mathtt{AGHD}, \mathtt{AFED}\}$, toward D. Among ABCD, AGHD, and AFED, the path ABCD has the minimal communication latency. That is, D accepts the message from the shortest path ABCD and discards those from paths AGHD and AFED. Here, $|\arg\min_{p\in\mathcal{P}_{\mathtt{A}\sim\mathtt{D}}}\ell_p|$ is 3, and the scope of the broadcast message due to A is 7.

1.1 Our Contributions

In this paper, we present an analytical model to show that there exists unstructured P2P networks minimizing (1) for any pair of participating peers v and u. In particular, a broadcast message originating from v with $TTL = \Theta(\ln^c \ell_{vu})$ takes the "nearly optimal" delay of $\Theta(\ell_{vu} + \ln^c \ell_{vu})$ to reach u, where ℓ_{vu} represents the end-to-end delay[1] from v to u, and $c\ (1 < c < 2)$ is a small constant. In addition, our analytical result concludes that such networks guarantee the exponential broadcast scope.

Second, motivated by our analytical results, we propose a fully decentralized algorithm based on the *Metropolis-Hastings* method [12], [13] to construct the intended overlay networks. In our proposal, any peer in the system connects to as many as possible of its geographically closest peers, subject to its maximum number of connections. In addition, compared with distant peers, the peers in the proximity of the physical network connect to one another in higher probability. We show that our algorithm constructs the intended overlay networks in a lightweight fashion, and each participating peer implementing our algorithm requires only local knowledge.

Third, we perform extensive simulations to investigate the performance of our proposed algorithm against the state-of-the-art solutions of Liu et al. [9], [10], [11] and Hsiao et al. [14] designed for attacking the topology mismatch problem. Our simulation results reveal that our design is effective and validates the analytical results. Additionally, while our proposal maintains a broadcast scope comparable to those in [9], [10], [11], [14], our design outperforms prior solutions remarkably in terms of broadcast delay. Moreover, our proposal strives to minimize broadcast delay and maximize broadcast scope in a dynamic environment in which peers may come and go freely. Together with the performance investigation for Gnutella search application, we further conclude that our topology matching algorithm clearly outperforms the existing solutions in [9], [14] in terms of the number of messages introduced to the physical network and the query response time.

1. The end-to-end delay from node v to node u is the latency of routing a message from v to u in the underlying network (e.g., the Internet).

1.2 Road Map

The remainder of this paper is organized as follows: Section 2 discusses related work. We provide our design rationale in Section 3, and then, present our proposal in Section 4. Our proposal is evaluated through extensive simulations, with the simulation results given in Section 5. Section 6 concludes our study with possible future research directions.

2 Related Work

Liu et al. [9], [10], [11] present deterministic algorithms for the topology mismatch problem between unstructured P2P networks and underlying networks. To optimize the P2P network topology (denoted by G), the studies [9], [10], [11] suggest adding a new overlay link to G and removing an existing one from G iteratively such that the net operations can reduce the total delay cost of overlay links. While Liu et al.'s solutions are elegant and require only local knowledge for each peer, their proposals provide no performance guarantee. In contrast, we present a novel proposal that is driven by rigorous performance analysis. Unlike Liu et al.'s algorithms, the latency of routing a scoped broadcast message from any node v to another other node u in our presented overlay network approximates the minimum (i.e., the end-to-end delay from v to u), which is independent of the number of nodes in the system. We compare our proposal to Liu et al.'s solution in extensive simulations, and the simulation results validate the effectiveness of our proposal, showing that our design significantly outperforms Liu et al.'s.

Hsiao et al. [14] present a randomized algorithm to identify a family of unstructured P2P networks that match physical networks reasonably well. Consider a P2P network $G = (V, E)$ constructed by the algorithm in [14]. Let the end-to-end delay between any two nodes in the physical network be no more than $\mathcal{L}$. Hsiao et al. [14] rigorously show that any two nodes in G have the routing delay of $\leq\Theta(\mathcal{L})$ in expectation. In contrast, the communication delay between any two nodes v and u in our proposed network is no more than $\Theta(\ell_{vu} + \ln^c \ell_{vu})$ in the probability of $\geq\Theta(1 - \frac{\ln \ell_{vu}}{\ell_{vu}^{\mathcal{K}}})$, where $\mathcal{K}$ depends on the number of overlay links maintained by each peer. Here, ℓ_{vu} is the end-to-end delay from node v to node u in the physical network and c is a small constant within 1 and 2. To our best knowledge, the

unstructured P2P networks we present in this paper approximate the optimal design in terms of broadcast delay to tackle the topology mismatch problem.

Small-world networks exhibit low diameter [15], [16], [17]. By diameter, we mean the maximum hopcount of routing a message on the shortest path[2] between any two nodes in a given graph network $G = (V, E)$. Although a small-world network G has a low diameter, the "delay" of routing a message between any two nodes in G may not be necessarily small, considering that G is layered on top of a physical network (e.g., the Internet). Without relying on rigorous performance analysis, Merugu et al. in their seminal study [18] conclude that there exist some instances of small-world networks that can match their physical network topologies. Merugu et al. [18] do not detail how to create such a small-world P2P network, however. In contrast, in our study, we discuss how an unstructured P2P network that well matches the physical network topology can be constructed. Our proposed algorithm is motivated by a rigorously analytical model.

Kleinberg et al. [19] present a unicast routing algorithm for a small-world network based on the d-dimensional lattice. Consider a two-dimensional lattice (i.e., $d = 2$) in which any node i in the lattice has a coordinate of the form $(i.x, i.y)$. A node v in the two-dimensional lattice-based small-world network connects to another node w as its neighbor if either $|v.x - w.x| = 1$ or $|v.y - w.y| = 1$. v additionally connects to a *long-distance neighbor* u in the network with the probability of

$$\frac{\ell_{vu}^{-2}}{\sum_{\forall u} \ell_{vu}^{-2}}$$

(where $\ell_{vu} = |v.x - u.x| + |v.y - u.y|$), assuming that v has global knowledge regarding the network. In [19], upon receiving a unicast message m, x simply forwards m to a node y picked among x's neighbors such that y is closest to m's destination in terms of euclidean distance in the lattice. In contrast, our work differs in Kleinberg et al.'s study in that we assume no substrate (e.g., the two-dimensional lattice) available to our proposal. While Kleinberg et al.'s work aims at unicast routing in a static environment where the participating nodes are labeled with network coordinates and have global network knowledge, in our study, the nodes only have local knowledge and they cooperatively construct a network in a distributed manner for message broadcasting.

Landmark ordering is a scheme that estimates the network coordinate for Internet hosts [20], [21], [22]. The scheme in [22], for example, first deploys k landmark nodes, and each landmark has a unique index number. To compute the coordinate of an Internet host j, j measures the delay to each of the k landmark nodes. The measured delays are then sorted in increasing (or decreasing) order, resulting in a landmark vector $[l_1, l_2, \ldots, l_k]$, where l_is $(1 \leq i \leq k)$ represent the indices of landmarks. If j is relatively closer to the landmark l_{i_1} than the landmark l_{i_2}, then $i_1 < i_2$ in the vector. Clearly, such landmark vectors form a k-dimensional coordinate space. If two nodes have "similar" landmark vectors, then the two nodes are physically close. Our proposal presented in this paper is orthogonal to the landmark vector scheme. On the one hand, our design does not assume the availability of network coordinate service; on the other, if the network coordinate service is available to our proposal, the participating nodes in our network may depend on their landmark vectors to estimate their message delays and determine whether they shall connect to one another or not.

2. Here, the path length is measured in terms of hopcount.

TABLE 1
The Notations Frequently Used in This Paper

Notation	Description
$G = (V, E)$	the input graph to our algorithms as discussed in Section 4.1 and 4.2
ℓ_{vu}	the end-to-end delay from v to u
$\mathcal{B}_v(k)$	the set of nodes having the end-to-end delay of $\leq k$ from v
$\mathcal{B}_v(1)$	the geographically closest nodes to v
$\mathcal{L}$	the maximum end-to-end delay between any two nodes in V
α, β	the given constants for the end-to-end power-law delay model
$\mathcal{W}_v$	the $\frac{\mathcal{K} \ln \mathcal{L}}{\beta}$ nodes connected by v in addition to $\mathcal{B}_v(1)$
$\mathcal{K}$	the system parameter dependent of $\lvert\mathcal{W}_v\rvert$ for all $v \in V$
$\mathcal{G}_v$	$\{u \lvert vu \in E\}$ in the input graph G
$\mathcal{G}_v^2$	$\{w \lvert u \in \mathcal{G}_v, uw \in E\} - \mathcal{G}_v - \{v\}$ in the input graph G
deg_v	the number of neighbors currently maintained by v in the input graph G
$\mathcal{M}$	the maximum number of neighbors connected by any node v in the input graph G, that is, $deg_v \leq \mathcal{M}$ for all $v \in V$
$\mathcal{T}$	the system parameter used by the algorithm discussed in Section 4.2

Given the set $\mathcal{V}$ of physical network locations, an end-to-end delay function $\mathcal{I} : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$, and an overlay graph $G = (V, E)$, Qiu et al. [23] intend to find a one-to-one function $\mathcal{F} : V \rightarrow \mathcal{V}$ that assigns participating peers in V to the locations in $\mathcal{V}$ of the physical network such that the delay of routing a message between any two nodes in G is minimized. The objective of our study is to construct an overlay topology $G = (V, E)$ and is thus orthogonal to [23]. That is, our algorithm generates G, and such G may be further improved using the solution suggested in [23].

3 Preliminaries

This section provides the design rationale that motivates our proposal presented later in Section 4. We first give a model for analyzing the performance of scoped broadcast in a practical end-to-end network delay model (Section 3.1), and then, discuss our analytical results (Section 3.2).

Table 1 summarizes the notations frequently used in this paper for easy reference.

3.1 Broadcast in the Power Law Delay Model

Prior studies in [24], [25], [26], [27] presented that the network latency distribution between Internet end hosts are likely to follow the *power law expansion*, and early seminal works (e.g., [19], [28], [29]) have relied on the power law latency expansion characteristic of the underlying network to analyze the performance of their algorithms. In our

study, we thus assume the network latency between nodes participating in a P2P network as follows:

Definition 1. *Given positive constants α and β, a set of nodes V follows the α-power-law latency expansion if for each node $v \in V$, the number of nodes (denoted by $\chi_v(z)$) that have the latency no more than z to v is*

$$\chi_v(z) = \beta z^{\alpha}. \tag{3}$$

The set of these βz^{α} nodes is denoted by $\mathcal{B}_v(z)$.

Without loss of generality, we let the minimum and maximum delays between any two nodes be 1 and $\mathcal{L}$, respectively. To estimate the values of $\mathcal{L}$, α and β could rely on the existing schemes (e.g., that in [24]), and we assume that such values are known to each participating peer in our study. Throughout the paper, we denote the network latency from u to v as ℓ_{uv}. For tractable analysis, we assume that the triangle inequality (i.e., $\ell_{uv} + \ell_{vw} > \ell_{uw}$ for any distinct nodes u, v, and w) holds in our analytical model. However, we will relax such assumption in our simulations, as will be discussed later in Section 5. Notice that our analytical results presented in this section can be generalized to the case that $\beta_1 z^{\alpha} \le \chi_v(z) \le \beta_2 z^{\alpha}$, where $\beta_1 < \beta_2$.

Lemma 1. *Let each node $v \in V$ connect to all nodes in $\mathcal{B}_v(1)$. If a node $s \in V$ initiates the message flooding with $TTL = k$, then the number of nodes receiving the flooded messages from s is βk^{α}, and these nodes are all the nodes in the set $\mathcal{B}_v(k)$.*

Lemma 1 is trivial, and we omit its detailed proof.

Theorem 1. *Let each node $v \in V$ connect to all nodes in $\mathcal{B}_v(1)$. In addition, v connects to $\frac{\mathcal{K}\ln\mathcal{L}}{\beta}$ nodes, each (denoted by w) picked from $V - \{v\}$ with the probability of*

$$\frac{1}{\alpha\beta\ell_{vw}^{\alpha}\ln\mathcal{L}}. \tag{4}$$

Consider that a node $s \in V$ broadcasts a message with TTL of

$$4(\alpha\ln\mathcal{L})^{\frac{1}{\alpha}}, \tag{5}$$

and that any node $d \in V - \{s\}$. If $(\frac{\ell_{sd}}{2^i}) \ge TTL$ (where i is an integer $\le \log_2 \ell_{sd}$), there exists at least one connection between $\mathcal{B}_s(TTL)$ and $\mathcal{B}_d(\frac{\ell_{sd}}{2^i})$ with a constant probability no less than

$$1 - \frac{1}{\mathcal{L}^{\mathcal{K}}}. \tag{6}$$

Proof. We first show that the probability that v connects to w is well defined. Since the maximum latency between any two nodes is $\mathcal{L}$:

$$\int_{x=1}^{\mathcal{L}} \alpha\beta x^{\alpha-1}\frac{1}{\alpha\beta x^{\alpha}\ln\mathcal{L}}dx = \frac{1}{\ln\mathcal{L}}\int_{x=1}^{\mathcal{L}}\frac{1}{x}dx = 1.$$

Since any two nodes in $\mathcal{B}_s(TTL)$ and $\mathcal{B}_d(\frac{\ell_{sd}}{2^i})$, respectively, have their end-to-end communication latency no more than $2(TTL + \frac{\ell_{sd}}{2^i})$, there exists a connection between the two nodes with a probability no less than

$$\frac{1}{\alpha\beta\big(2\big(TTL + \frac{\ell_{sd}}{2^i}\big)\big)^{\alpha}\ln\mathcal{L}}.$$

If $TTL \le (\frac{\ell_{sd}}{2^i})$, then a node in $\mathcal{B}_s(TTL)$ connects to any node in $\mathcal{B}_d(\frac{\ell_{sd}}{2})$ with the probability of

$$\begin{aligned}
&\ge \beta\left(\frac{\ell_{sd}}{2^i}\right)^{\alpha} \times \frac{1}{\alpha\beta\big(2\big(TTL + \frac{\ell_{sd}}{2^i}\big)\big)^{\alpha}\ln\mathcal{L}} \\
&> \beta\left(\frac{\ell_{sd}}{2^i}\right)^{\alpha} \times \frac{1}{\alpha\beta\big(\frac{\ell_{sd}}{2^{i-2}}\big)^{\alpha}\ln\mathcal{L}} \\
&= \frac{1}{\alpha 4^{\alpha}\ln\mathcal{L}}.
\end{aligned}$$

By Lemma 1, any node in $\mathcal{B}_s(TTL)$ receives a message flooded by s. Therefore, the probability that no connection between $\mathcal{B}_s(TTL)$ and $\mathcal{B}_d(\frac{\ell_{sd}}{2^i})$ exists is

$$\begin{aligned}
&< \left(1 - \frac{1}{\alpha 4^{\alpha}\ln\mathcal{L}}\right)^{\beta(TTL)^{\alpha}\frac{\mathcal{K}\ln\mathcal{L}}{\beta}} \\
&\le \left(e^{-\frac{1}{\alpha 4^{\alpha}\ln\mathcal{L}}}\right)^{\beta(TTL)^{\alpha}\frac{\mathcal{K}\ln\mathcal{L}}{\beta}} \\
&= \frac{1}{\mathcal{L}^{\mathcal{K}}}.
\end{aligned}$$

□

Corollary 1. *Let s originate a broadcast message m with a TTL of*

$$t(\mathcal{C}+1) - 1 = \Theta\big((\ln\ell_{sd} - \ln\ln\mathcal{L})(\ln\mathcal{L})^{\frac{1}{\alpha}}\big). \tag{7}$$

Then, node d receives the message m due to s with a probability no less than

$$1 - \frac{t-1}{\mathcal{L}^{\mathcal{K}}} = 1 - \Theta\left(\frac{\ln\ell_{sd} - \ln\ln\mathcal{L}}{\mathcal{L}^{\mathcal{K}}}\right), \tag{8}$$

where $t = \log_2\ell_{sd} - (1 + \frac{\log_2\alpha}{\alpha} + \frac{\log_2\ln\mathcal{L}}{\alpha})$ and $\mathcal{C} = 4(\alpha\ln\mathcal{L})^{\frac{1}{\alpha}}$. Such a flooded message m takes a total latency no more than

$$(4t-2)\mathcal{C} + 2\ell_{sd} = \Theta\Big((\ln\ell_{sd} - \ln\ln\mathcal{L})(\ln\mathcal{L})^{\frac{1}{\alpha}} + \ell_{sd}\Big). \tag{9}$$

Proof. Let $f_1 = s \rightsquigarrow f_2 \rightsquigarrow \cdots \rightsquigarrow f_t = d$ be the path concatenating the subpaths, $f_1 \rightsquigarrow f_2, f_2 \rightsquigarrow f_3, \ldots, f_{t-1} \rightsquigarrow f_t$, that the flooded message m traverses. By Theorem 1, there exists a connection between a node in $\mathcal{B}_{f_1=s}(\mathcal{C})$ and $f_2 \in \mathcal{B}_d(\frac{\ell_{sd}}{2^1})$ with a probability no less than $1 - \frac{1}{\mathcal{L}^{\mathcal{K}}}$ (see Fig. 2). Upon receiving m from a node in $\mathcal{B}_{f_1}(\mathcal{C})$, f_2 performs similarly, and there is a connection between a node in $\mathcal{B}_{f_2}(\mathcal{C})$ and $f_3 \in \mathcal{B}_d(\frac{\ell_{sd}}{2^2})$ with the probability of $\ge 1 - \frac{1}{\mathcal{L}^{\mathcal{K}}}$. Iteratively, the probability that a connection between a node in $\mathcal{B}_{f_{t-1}}(\mathcal{C})$ and $f_t = d \in \mathcal{B}_d(\mathcal{C})$ exists is $\ge 1 - \frac{1}{\mathcal{L}^{\mathcal{K}}}$. More specifically, due to $\mathcal{C} = 4(\alpha\ln\mathcal{L})^{\frac{1}{\alpha}} = \frac{\ell_{sd}}{2^{t-1}}$ and $t = \log_2\ell_{sd} - (1 + \frac{\log_2\alpha}{\alpha} + \frac{\log_2\ln\mathcal{L}}{\alpha})$, the flooded message m initiated by s traverses nodes in $\mathcal{B}_{f_i}(\mathcal{C})$, where $i = 1, 2, 3, \ldots, t$. Consequently, m visits at most the $t\mathcal{C} + (t-1)$ nodes on a path from s to d with a probability no less than $1 - \frac{t-1}{\mathcal{L}^{\mathcal{K}}}$.

On the other hand, since m traverses the nodes in $\mathcal{B}_{f_i}(\mathcal{C})$ by taking the delay of $\le 2\mathcal{C}$ for $i = 1, 2, 3, \ldots, t$, and

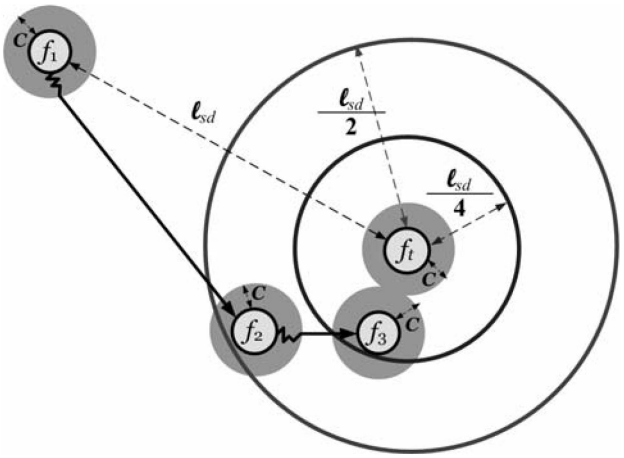


Fig. 2. The idea of the proof.

the delay of a link between f_j and f_{j+1} (where $j = 1, 2, 3, \ldots, t-1$) is $\leq 2 \times (\mathcal{C} + \frac{\ell_{sd}}{2^j})$, the total delay of sending the flooding message m is thus $\leq 2t\mathcal{C} + \sum_{j=1}^{t-1} 2 \times (\mathcal{C} + \frac{\ell_{sd}}{2^j}) < (4t-2)\mathcal{C} + 2\ell_{sd}$, yielding the proof. □

Remark 1. By letting TTL be equal to

$$4(\alpha \ln \ell_{sd})^{\frac{1}{\alpha}}, \tag{10}$$

the probability that there exists at least one connection between $\mathcal{B}_s(TTL)$ and $\mathcal{B}_d(\frac{\ell_{sd}}{2^t})$ is at least

$$1 - \frac{1}{{\ell_{sd}}^{\mathcal{K}}}. \tag{11}$$

The proof is similar to that in Theorem 1, and we omit the details of the proof. By Remark 1, we further have the result as follows:

Corollary 2. *Let s originate a scoped broadcast with a TTL of*

$$\Theta\big((\ln \ell_{sd} - \ln\ln \ell_{sd})(\ln \ell_{sd})^{\frac{1}{\alpha}}\big) = \Theta\big((\ln \ell_{sd})^{1+\frac{1}{\alpha}}\big). \tag{12}$$

Then, the probability that node d receives a flooded message from s is at least

$$1 - \Theta\left(\frac{\ln \ell_{sd} - \ln\ln \ell_{sd}}{{\ell_{sd}}^{\mathcal{K}}}\right), \tag{13}$$

and the message takes a total delay of

$$\Theta\big((\ln \ell_{sd})^{1+\frac{1}{\alpha}} + \ell_{sd}\big). \tag{14}$$

3.2 Discussions

We summarize and discuss our analytical results as follows, and these results motivate our algorithm presented in the following section.

First, Corollary 1 concludes that there exists a network topology $G = (V, E)$ such that a scoped broadcast message takes $\Theta((\ln \ell_{sd} - \ln\ln \mathcal{L})(\ln \mathcal{L})^{\frac{1}{\alpha}})$ hops to reach a receiving node d from a source node s in a probability no less than $1 - \Theta(\frac{\ln \ell_{sd} - \ln\ln \mathcal{L}}{\mathcal{L}^{\mathcal{K}}})$ if

1. each node $v \in V$ connects to all nodes in $\mathcal{B}_v(1)$,
2. v additionally connects to $\frac{\mathcal{K}\ln \mathcal{L}}{\beta}$ nodes, and each of these nodes, w, connected by v is selected from V with the probability of $\frac{1}{\alpha\beta\ell_{vw}^{\alpha}\ln \mathcal{L}}$.

Second, we may have a smaller hopcount of sending a scoped broadcast message. However, the event of sending such a scoped broadcast message is at a lower probability. Precisely, a scoped broadcast message takes $\Theta(\ln \ell_{sd})^{1+\frac{1}{\alpha}}$ hops to reach a receiving node d from a source node s in the probability of $\geq 1 - \Theta(\frac{\ln \ell_{sd} - \ln\ln \ell_{sd}}{{\ell_{sd}}^{\mathcal{K}}})$ (see Corollary 2).

Finally, we note that a source node s may inject a message with $TTL = \Theta(\ell_{sd})$ to reach a node d, and such message takes the only total delay of $\Theta(\ell_{sd})$ (i.e., the nodes in $\mathcal{B}_v(1)$ for all vs on the path from s to d help relay the message). However, Corollary 1 (or Corollary 2) presents that s can flood a message with only $TTL = \Theta((\ln \ell_{sd} - \ln\ln \mathcal{L})(\ln \mathcal{L})^{\frac{1}{\alpha}})$ (or $\Theta((\ln \ell_{sd})^{1+\frac{1}{\alpha}})$), compared with $TTL = \Theta(\ell_{sd})$, by slightly increasing the total latency up to $\Theta((\ln \ell_{sd} - \ln\ln \mathcal{L})(\ln \mathcal{L})^{\frac{1}{\alpha}})$ (or $\Theta((\ln \ell_{sd})^{1+\frac{1}{\alpha}})$).

Remark 2. Let s originate a scoped broadcast with $TTL = \Theta((\ln \mathcal{L} - \ln\ln \mathcal{L})(\ln \mathcal{L})^{\frac{1}{\alpha}}) = \Theta(\ln \mathcal{L})^{1+\frac{1}{\alpha}}$. Let d be the most distant node in the system having the delay of $\mathcal{L}$ to s. Then, d receives a flooded message from s at the latency of up to $\Theta((\ln \mathcal{L} - \ln\ln \mathcal{L})(\ln \mathcal{L})^{\frac{1}{\alpha}} + \mathcal{L}) = O((\ln \mathcal{L})^{1+\frac{1}{\alpha}} + \mathcal{L})$ in the probability of $\geq 1 - O(\frac{\ln \mathcal{L}}{\mathcal{L}^{\mathcal{K}}})$.

Due to Remark 2, we further conclude the following:

Remark 3. If the total number of nodes in the system is $\mathcal{N}$ and $\mathcal{K} \geq \alpha$, then s can broadcast a message to d with $TTL = O((\ln \mathcal{N})^{1+\frac{1}{\alpha}})$ in the probability of $\geq 1 - O(\frac{\ln \mathcal{N}}{\mathcal{N}})$. This represents that the broadcast scope increases exponentially.

4 Our Proposal and Implementation

As we mentioned in Section 1, given $G = (V, E)$, our goal in this study is to reshape G such that for any $v, u \in V$, $\min_{p \in P_{v \sim u}} \ell_p$ and $|\arg\min_{p \in P_{v \sim u}} \ell_p|$ are minimized as much as possible. In Section 3, we conclude that: 1) if each node $v \in V$ links to $\frac{\mathcal{K}\ln \mathcal{L}}{\beta}$ nodes (denoted by the set $\mathcal{W}_v$), each $w \in \mathcal{W}_v$ is picked from V with the probability of $(\alpha\beta\ell_{vw}^{\alpha}\ln \mathcal{L})^{-1}$ and 2) if each node $v \in V$ additionally connects to all nodes in its $\mathcal{B}_v(1)$, then the resultant overlay network guarantees $\min_{p \in P_{v \sim u}} \ell_p = \Theta(\ell_{vu} + \ln^c \ell_{vu})$ and $|\arg\min_{p \in P_{v \sim u}} \ell_p| = \Theta(\ln^c \ell_{vu})$, where $1 < c < 2$.

In the following discussion, we consider any node $v \in V$ that implements our proposal. We will describe in this section a distributed algorithm to construct $\mathcal{W}_v$ and $\mathcal{B}_v(1)$.

4.1 Constructing $\mathcal{W}_v$

Consider any $G = (V, E)$. To construct $\mathcal{W}_v$ for any $v \in V$, our proposal is based on the *Metropolis-Hastings* method [12], [13] in order to sample nodes from V. In particular, v issues a biased random walker to the network G. Let x be a peer in G that receives the walker. Then, x dispatches the walker to one (say y) of its overlay neighbors, denoted by $\mathcal{G}_x$ ($\mathcal{G}_x = \{y \mid xy \in E\}$), according to the following probability:

$$\Pr_{x,y} = \begin{cases} \phi\, T_{x,y} \min\left(1, \frac{\pi_v(y) T_{y,x}}{\pi_v(x) T_{x,y}}\right), & \text{if } y \in \mathcal{G}_x, \\ 0, & \text{if } y \notin \mathcal{G}_x, \\ 1 - \sum_{w \neq x} \Pr_{x,w}, & \text{if } x = y, \end{cases} \quad (15)$$

and

$$T_{x,y} = \frac{1}{deg_x}, \quad (16)$$

where ϕ is an arbitrarily positive value, and $0 < \phi < 1$ such that $1 - \sum_{w \neq x} \Pr_{x,w} > 0$. Here, $\pi_v(w) = \frac{1}{\alpha \beta \ell_{vw}^{\alpha} \ln \mathcal{L}}$ is the intended probability that v selects the node w from V.

Notably, (15) allows a unique stationary probability distribution such that v picks w from V with the probability of $\pi_v(w)$. This is because: 1) the Markov chain defined by the probability transition matrix $[\Pr_{x,y}]$ is aperiodic (each node has a positive probability to pick itself); 2) irreducible (there is a positive probability to pick any node from V); and 3) time reversible ($\pi_v(x)\Pr_{x,y} = \pi_v(y)\Pr_{y,x}$ for all $x \neq y \in V$) [30].

As the Markov chain defined by (15) allows sampling any node w with the intended probability of $\pi_v(w) = \frac{1}{\alpha \beta \ell_{vw}^{\alpha} \ln \mathcal{L}}$, v takes time to approximate the probability distribution $[\pi_v(w)]$ for all $w \in V$. In particular, in our proposal, a node v issues a random walker, say $\jmath$, to sample nodes. $\jmath$ visits the nodes in the system and updates $\mathcal{W}_v$ maintained by v. If $|\mathcal{W}_v| = \frac{\mathcal{K} \ln \mathcal{L}}{\beta}$, v replaces an element in $\mathcal{W}_v$ in a FIFO fashion. Once $[\pi_v]$ is approximated, the nodes sampled by $\jmath$ with the probability close to the intended. In the following, we thus analyze the time steps required by $\jmath$ to approach the distribution $[\pi_v]$.

We first define the following notations:

Definition 2. *The variation distance between two probability distributions D_1 and D_2 on a countable state space S is given by*

$$\|D_1 - D_2\| = \frac{\sum_{x \in S} |D_1(x) - D_2(x)|}{2}, \quad (17)$$

where the factor $\frac{1}{2}$ guarantees that the variation distance is between 0 and 1.

Definition 3. *Let D be the stationary probability distribution of a Markov chain with state space S. Let D_x^t be the probability distribution of the state of the chain starting at x after t time steps. Given any small constant ϵ, we define the mixing time of the Markov chain as*

$$\tau(\epsilon) = \max_{x \in S} \tau_x(\epsilon), \quad (18)$$

where

$$\tau_x(\epsilon) = \min\{t : \|D_x^t - D\| \leq \epsilon\}. \quad (19)$$

By Definition 3, $\tau_x(\epsilon)$ is the first time step t at which the variation distance between D_x^t and D is no more than ϵ. Among $\tau_x(\epsilon)$s, $\tau(\epsilon)$ is the maximum over all xs in S. If $\tau(\epsilon)$ is polynomial in the problem size (i.e., $\beta \mathcal{L}^{\alpha}$) and $\ln(\epsilon^{-1})$, then the Markov chain is *rapidly mixing* [31].

Theorem 2. *Given $G = (V, E)$, in our proposal, the random walker $\jmath$ initiated by any node $v \in V$ takes*

$$\tau(\epsilon) = deg_{\max} \cdot \mathcal{D} \cdot O\left(\mathcal{L}^{\alpha} \ln^2 \mathcal{L} - \ln \epsilon\right), \quad (20)$$

to approximate the probability distribution $[\pi_v]$ with the variation distance ϵ, where $\mathcal{D}$ represents the diameter of G and $deg_{\max}$ is the maximum degree of a node in V.

Proof. Let $\mathcal{MC}$ be any Markov chain with the state transition matrix $[\Pr_{x,y}]$ and the stationary probability distribution $[\pi_v(x)]$. Denote $\pi_{\min}$ as the smallest stationary probability of visiting a state in $\mathcal{MC}$. Denote $Q(e) = Q(s_1 s_2) = \pi_v(s_1)\Pr_{s_1,s_2}$ as the conditional probability for the state transition from s_1 to s_2 in $\mathcal{MC}$. Let Γ be a collection of simple paths γ_{xy} between all distinct states x and y in $\mathcal{MC}$, that is, $\Gamma = \{\gamma_{xy} | \forall x \neq y\}$. For a given Γ, define $\rho(\Gamma) = \max_e \frac{1}{Q(e)} \sum_{\gamma_{xy} \ni e} (\pi_v(x)\pi_v(y))$, and $\mathcal{D}(\Gamma)$ is the maximal number of edges connecting any two states in Γ. Then, due to [32], [33], $\tau(\epsilon) \leq \rho(\Gamma)\mathcal{D}(\Gamma)\ln(\frac{1}{\pi_{\min}\epsilon})$ for any Γ.

Consequently, in our proposal, since $Q(e) \geq \frac{\phi \pi_{\min}}{deg_{\max}}$ (where $\pi_{\min} = \frac{1}{\alpha \beta \mathcal{L}^{\alpha} \ln \mathcal{L}}$), $\mathcal{D}(\Gamma) \leq \mathcal{D}$, and

$$\begin{aligned} \rho(\Gamma) &= \max_e \frac{1}{Q(e)} \sum_{\gamma_{xy} \ni e} (\pi_v(x)\pi_v(y)) \\ &\leq \frac{deg_{\max}}{\phi \pi_{\min}} \cdot \left(\sum_{x \in V} \pi_v(x)\right)^2 \\ &= \frac{deg_{\max}}{\phi \pi_{\min}}, \end{aligned}$$

the proof follows. □

Consider an unstructured P2P network G to be optimized by our proposal. Theorem 2 presents that v discovers $\mathcal{W}_v$ in G in a rapid fashion. This is because a typical G is likely to be a random graph [9], [10] such that its diameter $\mathcal{D}$ is small. In particular, a random graph with $\mathcal{N}$ nodes has a diameter of $O(\ln \mathcal{N})$ with high probability [34] (where $\mathcal{N} = \beta \mathcal{L}^{\alpha}$), and a random graph is connected if the maximum degree of a node is $O(\ln \mathcal{N})$ (i.e., $deg_{\max} = O(\ln \mathcal{N})$). Consequently, a random walker $\jmath$ initiated from any node $v \in V$ can collect $\mathcal{W}_v$ in a fast fashion because of the rapid mixing time of $O(\mathcal{L}^{\alpha} \ln^c \mathcal{L} - \ln \epsilon)$, where c is a constant.

We remark that our proposal requires the measurement of ℓ_{vu} between any two nodes v and u. To minimize the overhead of measuring ℓ_{vu}, we may rely on the network coordinate service (e.g., [20], [21]) to estimate the communication delay between nodes. Second, it is possible that a random walker $\jmath$ is lost owing to peer failure or departure. To address this, $\jmath$ shall proactively inform its originating peer v its mostly recent k locations (say peers $l_1, l_2, \ldots, l_k$). If v did not receive any responses from $\jmath$, v recovers $\jmath$ by installing a new walker at l_1. If l_1 fails and cannot accept $\jmath$, then v requests l_2 to receive the walker. v performs this iteratively until it can find a peer among the k ones to accept the new walker. Otherwise, v initiates a new walker at its location. In either case, the new walker would take time no larger than that given in (20) to approximate the intended probability distribution for sampling peers. Our proposal opts to install a new walker at the recent locations visited by the failure walker $\jmath$. This is because $\jmath$ may have taken time to visit nodes and have reached those nodes that are likely to be included in $\mathcal{W}_v$.

4.2 Constructing $\mathcal{B}_v(1)$

In addition to constructing $\mathcal{W}_v$, v shall link to the nodes in $\mathcal{B}_v(1)$. Clearly, given any $G=(V,E)$, $\mathcal{G}_v=\{u|vu\in E\}$ is *not* likely to be identical to $\mathcal{B}_v(1)$, resulting in $\sum_{u\in\mathcal{G}_v}\ell_{vu} > \sum_{u\in\mathcal{B}_v(1)}\ell_{vu}$. Thus, our goal is to reformat G and generate $G'=(V,E')$ such that $\sum_{vu\in E'}\ell_{vu}$ in G' is minimized.

On the other hand, while minimizing $\sum_{vu\in E'}\ell_{vu}$, we shall avoid introducing extra resources (i.e., overlay connections) to reformat G, and meanwhile, to prevent G from being disconnected due to the removal of overlay connections.

Therefore, given $G=(V,E)$, we reshape G and output $G'=(V,E')$ by solving the problem as follows:

$$\min \quad \sum_{v\in V}\sum_{u\in V-\{v\}} x_{vu}\ell_{vu}, \tag{21}$$

$$\text{s.t.} \quad \sum_{v\in V}\sum_{u\in V-\{v\}} x_{vu} = |E|, \tag{22}$$

$$deg_v \le \mathcal{M}, \tag{23}$$

$$\sum_{vu\in\mathcal{E}(\mathcal{U})} x_{vu} \ge 1 \quad \forall \mathcal{U}\neq\emptyset\subset V, \tag{24}$$

$$x_{vu}\in\{0,1\} \quad \forall v\neq u\in V, \tag{25}$$

where x_{vu} is a binary variable indicating whether the edge vu shall appear in G' or not, deg_v is the number of connections currently maintained by v, and $\mathcal{E}(\mathcal{U})=\{vu|v\in\mathcal{U},u\notin\mathcal{U}\}$ is the set of overlay links incident to the nodes in the subset $\mathcal{U}\subset V$. Equation (24) indicates that for any subset $\mathcal{U}\subset V$, there is at least one overlay link connecting the two components $\mathcal{U}$ and $V-\mathcal{U}$, guaranteeing the connectivity of G'. Overall, our objective is to compute x_{vu}s such that (21) is minimized subject to the constraints in (22), (23), (24), and (25). We assume that the initial G is scalable in which the number of connections each node maintains is bounded from above by $\mathcal{M}$.

To solve the above problem, we again rely on the Metropolis-Hastings method. Let

$$\Delta T = T_{G'} - T_G, \tag{26}$$

where

$$T_G = \sum_{vu\in E}\ell_{vu}, \tag{27}$$

and

$$T_{G'} = \sum_{vu\in E'}\ell_{vu}. \tag{28}$$

Our proposal works as follows:

- If $\Delta T<0$, $G'=(V,E')$ is accepted.
- Otherwise, accept $G'=(V,E')$ with the following probability:

$$e^{-\frac{\Delta T}{\mathcal{T}}}, \tag{29}$$

where $\mathcal{T}$ is a system parameter.

To construct $G'=(V,E')$, in our proposal, each node $v\in V$ first picks any node $u\in\mathcal{G}_v$ and $w\in\mathcal{G}_v^2=\{w|u\in\mathcal{G}_v, uw\in E\}-\mathcal{G}_v-\{v\}$, and it then performs the following:

- If $\ell_{vw}-\ell_{vu}<0$, then v disconnects u and connects to w.
- The probability that v connects to w and disconnects u is $e^{-\frac{\ell_{vw}-\ell_{vu}}{\mathcal{T}}}$, otherwise.

Obviously, $\mathcal{G}_v$ and $\mathcal{G}_v^2$ represent the "one-hop neighbors" and "two-hop neighbors" of v. It suffices for v to collect and maintain its local knowledge (i.e., $\mathcal{G}_v^2$ in the given G) to perform the algorithm.

Notice that our proposal will not disconnect the nodes in V. This is because a node v removes a link to its neighbor in $\mathcal{G}_v$; meanwhile, it connects to another in $\mathcal{G}_v^2$.

Second, each node v in our proposal accepts an extra connection subject to $deg_v<\mathcal{M}$. This is not explicitly specified in the above algorithm for ease of discussion.

Third, from the perspective of any single node v, v restructures G, and then, generates $G'=(V,E')$. If $\ell_{vw}<\ell_{vu}$, and thus, $\Delta T=T_{G'}-T_G=\ell_{vw}-\ell_{vu}<0$, then G' contains $E'=E\cup\{vw\}-\{vu\}$. Otherwise, $E'=E\cup\{vw\}-\{vu\}$ is accepted with the probability of $e^{-\frac{\Delta T}{\mathcal{T}}}=e^{-\frac{\ell_{vw}-\ell_{vu}}{\mathcal{T}}}$.

Fourth, the nodes participating in the system perform our algorithm in parallel. This is because any node $v\in V$ helps reshape G by simply tailoring the subgraph $\mathcal{H}=(V_\mathcal{H},E_\mathcal{H})\subset G$, where $V_\mathcal{H}=\{v\}\cup\mathcal{G}_v\cup\mathcal{G}_v^2$ and $E_\mathcal{H}=E\cap\{vu\mid v\in V_\mathcal{H},u\in V_\mathcal{H}\}$.

Fifth, in our proposal, each node v performs the above algorithm iteratively. Given G, v improves G, resulting in G'. Then, let $G=G'$, and v repeats the algorithm for such G. According to Metropolis et al. [12] and Hastings [13], if each node performs our algorithm for a sufficient time, then G' is constructed with the approximate probability of

$$\frac{e^{-\frac{T_{G'}}{\mathcal{T}}}}{\mathcal{A}}, \tag{30}$$

where $\mathcal{A}=\sum_{f\in\mathcal{F}}e^{-\frac{T_f}{\mathcal{T}}}$, and $\mathcal{F}$ includes all feasible G's satisfying the constraints in (22), (23), (24), and (25). Because of (30), our proposal constructs G' to have a small $T_{G'}$ in a higher probability compared with those of G's with a relatively larger $T_{G'}$. On the other hand, $\mathcal{T}$, a system parameter in our proposal, determines the probability of finding G' with a designated value $T_{G'}$. The smaller the $\mathcal{T}$ is, the higher the probability of G' to have a smaller $T_{G'}$. However, this runs at the risk of being trapped in a local optimum[3] [35]. In Section 5, we will investigate the time steps required to determine our intended topology in simulations, and we experimentally study the effect of varying $\mathcal{T}$.

4.3 Discussions

Given the input graph $G=(V,E)$ to our algorithms presented in Sections 4.1 and 4.2, each node v in our proposal constructs and discovers $\mathcal{W}_v$ and $\mathcal{B}_v(1)$, respectively, leading to our intended network $G^\star=(V,E^\star)$.

- As mentioned in Section 4.1, G is likely to be a random graph, and it allows rapid mixing time to

3. By a "local" optimum, we mean that there exists some $\mathcal{B}_v^\star(1)\neq\mathcal{B}_v(1)$ such that $\sum_{u\in\mathcal{B}_v^\star(1)}\ell_{vu}<\sum_{u\in\mathcal{B}_v(1)}\ell_{vu}$.

sample the nodes from G for $\mathcal{W}_v$ for any $v \in V$. Hence, $\mathcal{W}_v$ shall be constructed as early as possible. In particular, our proposal creates $\mathcal{W}_v$ and $\mathcal{B}_v(1)$ in parallel without synchronization in a potentially dynamic, large-scale P2P environment.

- Once $\mathcal{W}_v$ is discovered, v links to the nodes in $\mathcal{W}_v$. The nodes in $\mathcal{W}_v$ that accept the connections requested by v treat these connections as extra overlay links, resulting in $G^{\mathcal{W}} = (V, E \cup E^{\mathcal{W}})$, where $E^{\mathcal{W}} = \{vu | v \in V, u \in \mathcal{W}_v\}$. Notably, each node additionally connects to only $\frac{\mathcal{K}\ln\mathcal{L}}{\beta}$ nodes.
- v discovers $\mathcal{B}_v(1)$ and links to the nodes in $\mathcal{B}_v(1)$ using the algorithm we mentioned in Section 4.2, provided *only* $G = (V, E)$. This would reshape G to $G' = (V, E')$. We finally have $G^\star = (V, E^\star)$ in which $E^\star = E' \cup E^{\mathcal{W}}$.
- Once we have $G^\star$, the participating nodes in $G^\star$ iterate the algorithms in Sections 4.1 and 4.2 due to peers' arrival and departure. On the one hand, the random walker issued by any node v samples the nodes with the intended probability (i.e., $[\pi_v]$ in (15)) in a rapid fashion since the diameter of $G^\star$ is $O(\ln^c \mathcal{N})$, where $1 < c < 2$ (see Remark 3). On the other hand, the input graph to our algorithm presented in Section 4.2 becomes $G'' = (V'', E'')$ in which V'' is the set of alive nodes in $G^\star$ and $E'' = \{vu | v \in V'', u \in V'', vu \in E'\}$ excludes the existing links in $\{vu | v \in V'', u \in V'', vu \in E^{\mathcal{W}}\}$.

5 Simulations

We have developed an event-driven simulator to study the performance of our proposed network. In the following, Section 5.1 details our simulation setting. In Section 5.2, we compare our proposal to prior overlay topology matching algorithms that are briefly reviewed in Section 5.1.1. In addition to the overheads introduced by the investigated overlay topology matching algorithms, Section 5.3 discusses the performance of our proposal operating in a dynamic environment, where participating peers may come and go freely. Finally, we discuss in Section 5.4 the impact of $\mathcal{W}_v$ and $\mathcal{B}_v(1)$ on our proposal (for all $v \in V$). We also investigate the effect of varying the system parameter $\mathcal{T}$ for our proposed network.

5.1 Simulation Setting

In our simulations, the number of nodes participating in the system is up to $\mathcal{N} = 100{,}000$, and the default is 30,000. We simulate a static environment in which the nodes participating in the network do not join and leave. We also assess our proposal in a dynamic environment, and the peers in such an environment have a lifetime with the exponential distribution of the mean equal to t minutes (e.g., $t = 20$ and 200) [36], [37].

Since previous studies (e.g., [36], [38], [39]) concluded that the peers in an unstructured peer-to-peer network, represented by $G = (V, E)$, are randomly interconnected, we generate in the simulations Gs as random graphs [34] using the algorithm suggested in [39]. In G, each node v connects to at most $\mathcal{M} = 6$ neighbors (i.e., the maximum number of connections v can maintain). For a fair comparison between prior topology matching algorithms (discussed later in Section 5.1.1) and our proposal, each simulated peer in our design can also link to six peers at most. In particular, in our proposed overlay, each peer v can only connect up to three peers for its $\mathcal{B}_v(1)$, while the size of vs $\mathcal{W}_v$ is no more than 3. However, unlike $\mathcal{W}_v$, v exploits its $\mathcal{B}_v(1)$ by rewiring the peers in an overlay network provided to our proposal. In the simulations, such a provided overlay network is a random graph in which each node only connects to three neighbors at most.

The end-to-end delays between the nodes in G in our simulations are generated by BRITE (the Barabasi model) [40] and PlanetLab [41]. As mentioned in Section 3, the delays between end hosts follow the power law latency distribution with parameters α and β [24], [25], [26], [27]. We approximate in this study the end-to-end delays due to BRITE with $\alpha = 2$ and $\beta = 0.001$. As for the end-to-end delays in PlanetLab, α and β are 0.8 and 500, respectively. We note that the publically available trace for the end-to-end host delays in PlanetLab, containing about 490 nodes, includes only about 490×490 delay metrics. We thus assign in our simulations each simulated peer to one, picked uniformly at random, among the 490 Planet nodes. Finally, in addition to the end-to-end delays due to BRITE and PlanetLab, we further benchmark our proposal using our synthesized end-to-end delays with $\alpha = 2$ and $\beta = 4$. Notably, except the end-to-end delays generated by BRITE, the triangle inequality for end-to-end delays due to PlanetLab and the one we synthesize is *not* hold.

5.1.1 Topology Matching Algorithms

The topology matching algorithms we investigate in this study are as follows:

- `Random+THANCS`: Let u, v, and w be any three nodes in a given overlay network $G = (V, E)$, and v connects to both u and w in G. THANCS [10], a localized, deterministic topology matching algorithm, improves G by performing the following. In the case that there is no overlay link that exists between u and w, if $\ell_{uv} + \ell_{vw} > \ell_{uw}$, and w can accommodate one extra connection, then u introduces a new link to w and removes its existing connection with v. If the connection between u and w appears in G, and if $\ell_{uv} + \ell_{vw} < \ell_{uw}$, then such connection will be eliminated. In this paper, we denote the random graph enhanced by THANCS as `Random+THANCS`.
- `Hsiao` and `Hsiao+THANCS`: Hsiao et al. present in [14] a randomized topology matching algorithm in which each participating peer samples nodes independently and uniformly at random in the system, and connects to those geographically close such that the resultant overlay (denoted by `Hsiao` in this paper) can match the physical network topology. Hsiao et al. in [14] show that their constructed P2P network guarantees the diameter of $\Theta(\ln\mathcal{N})$ hop-count. While further considering the end-to-end delays with the power law distribution, if each peer performs polylogarithmic samples and chooses those geographically close as its neighbors, then the expected delay of sending a message between

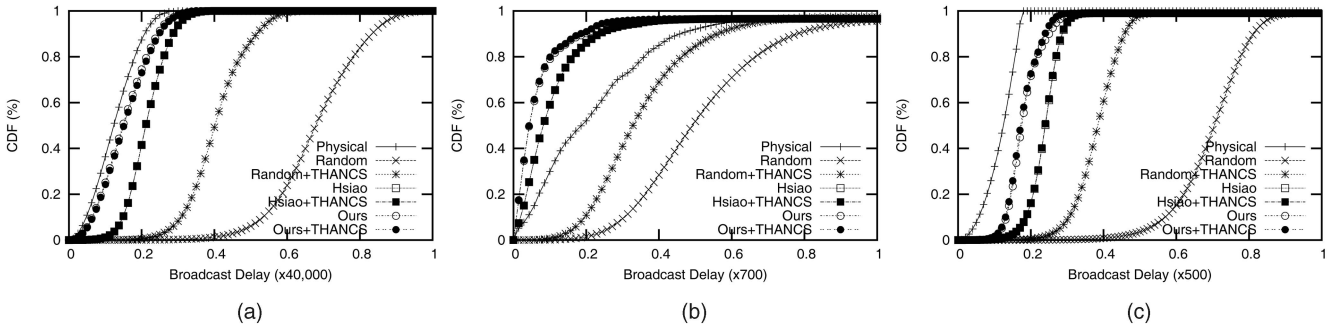


Fig. 3. The broadcast delay (each peer can connect up to six neighbors). (a) BRITE. (b) PlanetLab. (c) Synthesis.

any two nodes in the resultant overlay network is a constant no more than $\mathcal{L}$ [14]. Here, $\mathcal{L}$ is the maximum end-to-end delay between peers in the physical network. In addition to `Hsiao`, our simulation study investigates `Hsiao+THANCS` that optimizes `Hsiao` by having each participating peer further performs THANCS.

- `Ours` and `Ours+THANCS`: `Ours` represents the P2P network built by our proposal presented in this paper, and `Ours+THANCS` optimizes `Ours` by THANCS.

Note that in our study, we have calibrated and optimized the parameters relevant to the topology matching algorithms investigated. Given an overlay network $G = (V, E)$, each peer in `Random+THANCS` performs the THANCS algorithm with 3,000 rounds. This is because in our simulations, THANCS cannot further improve a given overlay topology with more than 3,000 rounds. In contrast, each participating peer in `Hsiao` issues 10 random walkers, each taking 100 walk steps, to uniformly sample the peers in G. `Hsiao+THANCS` intends to improve `Hsiao` by then having each peer perform THANCS for an additional 3,000 rounds. Regarding `Ours`, in default, each peer v operates our algorithm presented in Section 4.2 for 100 rounds to exploit its $\mathcal{B}_v(1)$, while it takes 100 random walk steps to build its $\mathcal{W}_v$. As for `Ours+THANCS`, each peer v first takes 100 rounds and 100 walk steps to construct its $\mathcal{B}_v(1)$ and $\mathcal{W}_v$, respectively. Then, v additionally takes 3,000 rounds to perform THANCS.

We finally notice that the default value of $\mathcal{T}$ (see Section 4.2) is four in our simulations.

5.2 Comparative Study

5.2.1 Broadcasting Delay and Scope

Fig. 3 depicts the broadcast delay, that is, $\min_{p \in \mathcal{P}_{v \sim u}} \ell_p$, for any $v \in V$ and $u \in V - \{v\}$ (see (1)), in the optimized overlay topologies, namely, `Random+THANCS`, `Hsiao`, `Hsiao+THANCS`, `Ours`, and `Ours+THANCS`. The coordinate (x, y) in Fig. 3 denotes the percentage of nodes y receiving a scoped broadcast message with the delay no more than x. We also include `Random` into Fig. 3 to represent the random graph for optimization. Moreover, `Physical` as shown in the plot denotes the end-to-end host delay in the physical network. We notice that in the experiment, the number of nodes $\mathcal{N}$ in the system is 30,000, and the simulation results discussed in this section investigate the static network environment wherein peers do not come and go freely. We will study the scenario that peers dynamically join and leave in Section 5.3.

Fig. 3 reveals that our proposal (i.e., `Ours`) considerably outperforms `Random+THANCS`, `Hsiao`, and `Hsiao+THANCS`. Particularly, in Fig. 3a, given the end-to-end delay between peers satisfying the triangle inequality (i.e., the BRITE topology), our proposal approximates the minimum (i.e., `physical`[4]); this not only validates our theoretical results (i.e., Corollary 2 in Section 3), but also shows the effectiveness of our design motivated by our analytical model. Interestingly, Fig. 3b shows that for the PlanetLab topology not satisfying the triangle inequality, both Hsiao et al.'s and our proposal perform better than `physical`, allowing the latency of routing a broadcast message to be smaller than that in the physical network. Notably, as shown in Fig. 3, the performance gain due to THANCS (i.e., `Random+THANCS`, `Hsiao+THANCS`, and `Ours+THANCS`) heavily depends on the target overlay (i.e., `Random`, `Hsiao`, and `Ours`) to be optimized. While THANCS can greatly improve `Random`, it only slightly improves `Hsiao` and `Ours`. This is mainly because THANCS does not provide the performance guarantee for the improvement of a given overlay topology.

Fig. 4 presents the "averaged" broadcast scope for the studied networks. In this experiment, each node broadcasts a message with $TTL = \infty$, and we then compute the averaged broadcast scope as

$$\frac{\sum_{v \in V} |\mathcal{S}_v(k)|}{\mathcal{N}},$$

where $\mathcal{S}_v(k)$ denotes the set of distinct nodes receiving the broadcast messages; they originate from a randomly selected node v at the hopcount of k ($k = 1, 2, 3, \ldots$). Specifically, in Fig. 4, the coordinate (x, y) denotes the percentage of nodes y receiving a broadcast message at the hopcount of $\leq x$. As can be seen in Fig. 4, `Random`, `Random+THANCS`, `Hsiao`, `Hsiao+THANCS`, `Ours`, and `Ours+THANCS` are comparable in that they all exhibit the exponential broadcast scope. While `Random` performs well due to the random graph, `Ours` as presented in this paper provides the rigorous performance guarantee that routing a broadcast message between any two nodes takes a polylogarithmic hopcount, thus having the exponential broadcast scope (see Remark 3).

5.2.2 Effects of Varying the Number of Neighbors

We investigate the effects of varying the number of neighbors of a peer. In the experiment discussed in this

4. `Physical` in Fig. 3a has the lower bound of broadcast delay.

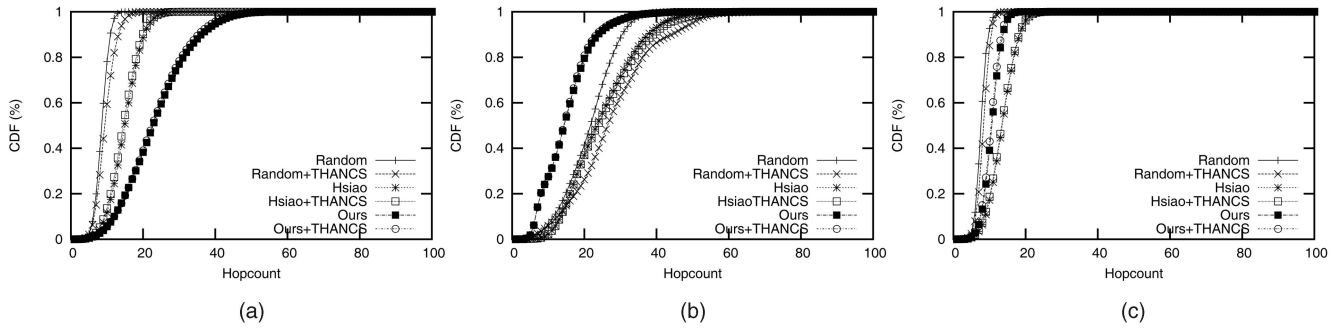


Fig. 4. The broadcasting scope (each peer can connect up to six neighbors). (a) BRITE. (b) PlanetLab. (c) Synthesis.

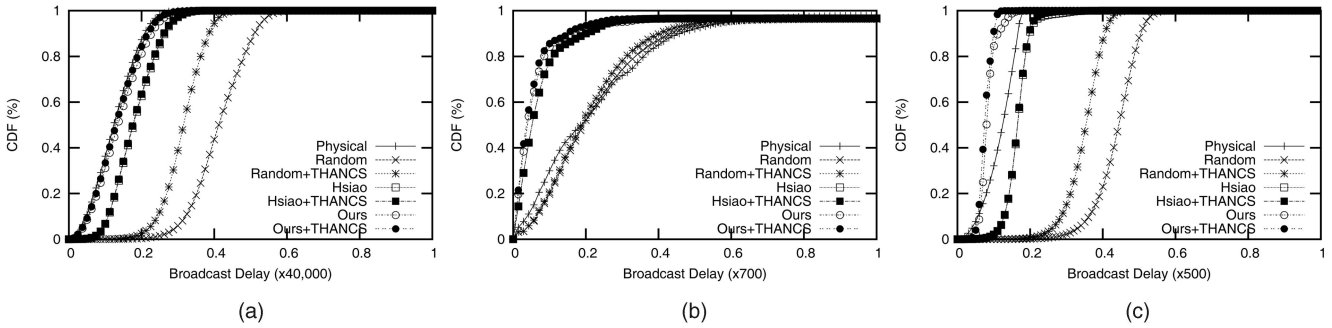


Fig. 5. The broadcast delay (each peer can connect up to 12 neighbors). (a) BRITE. (b) PlanetLab. (c) Synthesis.

section, each peer links to 12 neighbors at most. Fig. 5 depicts the simulation result. As compared to Fig. 3, Fig. 5 reveals that the broadcast delay in `Random` and `Random+THANCS` decreases due to the increase in the number of paths between any two nodes in the overlay. Second, similar to that presented in Fig. 3, our proposal ($|\mathcal{W}_v| = 9$ and $|\mathcal{B}_v(1)| = 3$ in the experiment for each $v \in V$) performs very well and consistently outperforms `Random`, `Random+THANCS`, `Hsiao`, and `Hsiao+THANCS`, provided that there are a relatively larger number of neighbors for each peer. Third, as Corollary 2 concluded in Section 3, for the end-to-end delay between nodes to satisfy the triangle inequality, the latency of routing a broadcast message approximates the minimum in higher probability if each peer v can connect to more neighbors in $\mathcal{W}_v$. Fig. 5a validates this, where `Physical` and `Ours` are quite comparable.

We do not include in this paper the simulation result for the broadcast scope due to the investigated overlays (where each peer can link to up to 12 neighbors). This is because the simulation result is similar to that in Fig. 4.

5.2.3 Effects of Varying the Number of Participating Peers

Corollary 2 states that our proposal minimizes the delay of sending a broadcast message between any two nodes v and u, and approximates the minimum (i.e., the lower bound of sending a message from v to u, ℓ_{vu}) independent of the number of nodes participating in the system. We are thus interested in the effects of varying the number of participating peers in terms of broadcast delay in this section.

Fig. 6 presents the simulation result for `Random`, `Random+THANCS`, `Hsiao`, and `Ours`, given the end-to-end delays between peers due to BRITE, by varying the number of nodes from 10,000 to 100,000. In Fig. 6, we measure

$$\sum_{v \neq u \in V} \left(\frac{\frac{\min_{p \in \mathcal{P}_v \rightsquigarrow_u} \ell_p}{\ell_{vu}}}{\binom{|V|}{2}} \right),$$

which is the averaged ratio of the delay of routing a broadcast message from any node v to any other node u on the shortest path in the overlay to that of sending the message from v to u in the physical network.

As Fig. 6 depicts, our proposal performs very well and approximates the minimum delay of sending a message between any two nodes. This validates the effectiveness of our design as motivated by our analytical result in Section 3 and confirms Corollary 2. Second, while in [14], Hsiao et al. propose an overlay network with rigorous performance guarantee in which the latency of routing a broadcast message between any two nodes v and u is no more than $\Theta(\mathcal{L})$ in expectation, what we present in this paper outperforms `Hsiao`, and the delay of sending a message between v and u in our design is close to the minimum ℓ_{vu}.

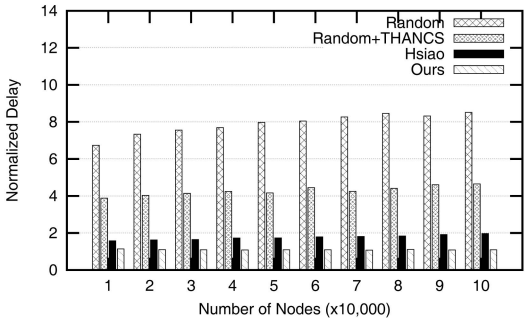


Fig. 6. The effect of varying the number of peers.

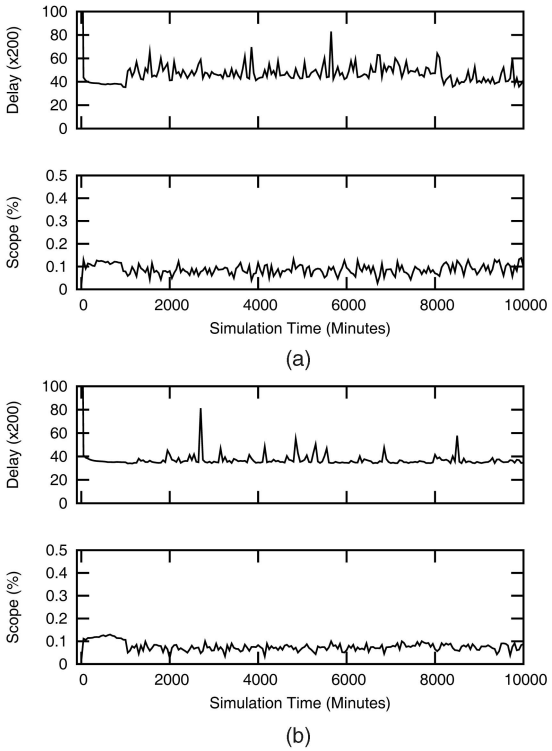


Fig. 7. The broadcasting delay and scope (with $TTL = 7$) by varying the mean lifetime of a peer for our proposal. (a) $t = 20$. (b) $t = 200$.

5.3 System Dynamics

5.3.1 Broadcast Delay and Scope

Fig. 7 depicts the simulation results for our proposed network operating in a dynamic network environment, where the lifetime of a peer in the system follows the exponential distribution with a mean of $t = 20$ and $t = 200$ minutes. In the experiment, each participating peer v performs our algorithm as discussed in Section 4.2 every 1 minute to discover $\mathcal{B}_v(1)$. Meanwhile, the random walker issued by v takes one walk step per minute to construct its $\mathcal{W}_v$. Note that in this experiment, we first generate a random graph with $\mathcal{N} = 30{,}000$ nodes, and these peers do not leave in the beginning of 1,000 minutes of the simulation. During the period of 1,000 minutes, each peer v takes a total of 1,000 rounds in performing the algorithm in Section 4.2 to exploit its $\mathcal{B}_v(1)$, and a total of 1,000 walk steps to discover its $\mathcal{W}_v$. Peers may start to leave after 1,000 minutes, depending on their lifetime. If a peer v newly joins the system, v connects up to six neighbors picked uniformly at random from the system. In the experiment, we measure and average $\min_{p \in \mathcal{P}_{v \sim u}} \ell_p$ for any two nodes v and u in the network. Additionally, we compute the scope of broadcasting a message, originated by a randomly selected node, with $TTL = 7$. The end-to-end delay between peers in this experiment is due to BRITE.

The simulation result shown in Fig. 7 illustrates that our design strives to stably maintain the performance metrics—broadcast delay and scope—in a dynamic environment. Specifically, even if peers have a smaller mean lifetime (e.g., $t = 20$), our proposal tends to maintain the broadcast delay and scope comparable to those in a static environment (i.e., that of the first 1,000 simulation minutes), concluding that the peers implementing our proposal are able to repair the network in a rapid fashion.

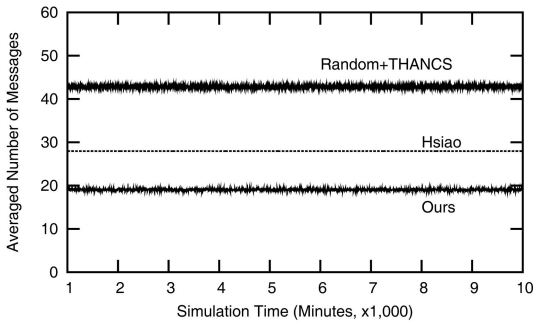


Fig. 8. The message overhead for the system where the mean lifetime of a peer is $t = 20$ minutes.

5.3.2 Message Overhead

Fig. 8 compares the averaged number of messages generated by a participating peer in our proposed network to those in `Random+THANCS` and `Hsiao`. In this experiment, the system operates for 10,000 minutes, and the lifetime of a participating peers follows the exponential distribution with a mean of $t = 20$ minutes. The experimental result for a larger mean lifetime (e.g., $t = 200$) is similar to what Fig. 8 depicts, and is thus omitted.

Similar to the experiment setting discussed previously in Section 5.3.1, in this experiment, a peer in `Random+THANCS` performs the THANCS algorithm every one simulated minute of the system. In contrast, in `Hsiao`, a joining peer v issues 10 random walkers once it participates in the system. These 10 walkers sample nodes uniformly at random from the network for v, and each of the walkers takes one walk step per simulated minute. v may update its neighbors every one simulated minute, depending on whether the sampled peers are "valid" to be connected or not.[5] As for our proposal, any participating peer v originates a single biased random walker to sample the peers for its $\mathcal{W}_v$ (where $|\mathcal{W}_v| = 3$). Meanwhile, v performs the algorithm in Section 4.2 for discovering $\mathcal{B}_v(1)$ per simulated minute.

Fig. 8 shows that the averaged number of messages generated by each peer v in our proposal is clearly less than that of `Random+THANCS`. This is because in `Random+THANCS`, v needs to collect its one and two-hop neighbors such that the THANCS algorithm can be performed. In contrast, in our proposal, in addition to the only one message introduced per simulated minute due to the random walker of v, v collects one and two-hop neighbors for computing its $\mathcal{B}_v(1)$. However, compared with up to six neighbors maintained by v in `Random+THANCS`, v in our proposal links to no more than three neighbors in its $\mathcal{B}_v(1)$, and thus, introduces less traffic.

Our proposal also outperforms `Hsiao` in terms of the message overhead. This is mainly because each peer v in `Hsiao` relies on a number of random walkers to sample a

5. Basically, if a peer maintains up to γ neighbors, then with a high probability, the peer needs to sample $O(\ln^{\gamma+\Theta(1)} \mathcal{N})$ nodes, and among these samples, v then picks γ as its neighbors. Interested readers may refer to [14] for the details.

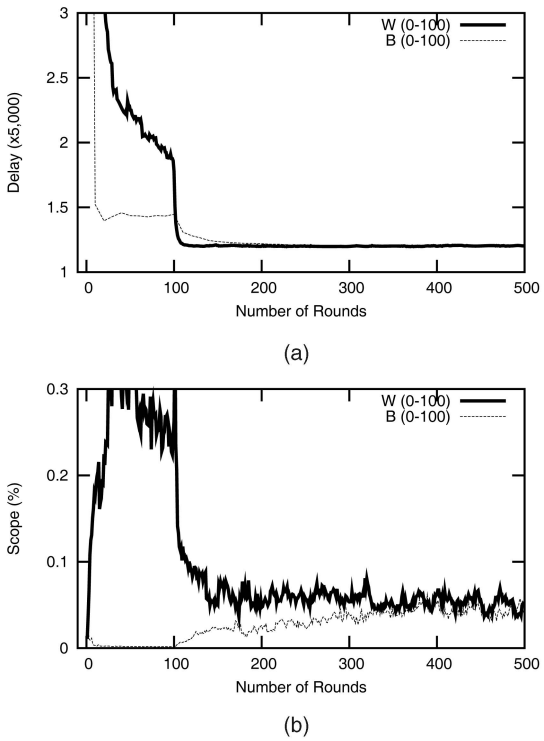


Fig. 9. The effects of $\mathcal{W}_v$ and $\mathcal{B}_v(1)$. (a) Broadcast delay. (b) Broadcast scope ($TTL = 7$).

sufficient number of valid peers as its neighbors, taking more messages than those generated by our proposal.

5.4 Impact of $\mathcal{W}_v$ and $\mathcal{B}_v(1)$ on Our Proposal

5.4.1 Building $\mathcal{W}_v$ and $\mathcal{B}_v(1)$ in Parallel

As mentioned in Section 4, each peer v issues a biased random walker for constructing its $\mathcal{W}_v$, and the random walker approximates the intended probability distribution (i.e., $[\pi_v]$ in (15)) in a rapid fashion if it traverses in a random graph (see Theorem 2). Hence, we suggest to discover $\mathcal{W}_v$ and $\mathcal{B}_v(1)$ in parallel.

Fig. 9 illustrates the simulation result in case $\mathcal{W}_v$ and $\mathcal{B}_v(1)$ are *not* exploited simultaneously. In this experiment, each participating peer v only issues its random walker to create its $\mathcal{W}_v$ for the first 100 simulation rounds, and the walker takes one walk step per round. v then enables the algorithm presented in Section 4.2 to discover its $\mathcal{B}_v(1)$. More specifically, after the first 100 rounds of the simulation, v invokes the algorithm in Section 4.2 per simulation round in addition to one walk step by v's walker. This is denoted by $\mathtt{W}(0-100)$ in Fig. 9. In contrast to $\mathtt{W}(0-100)$, $\mathtt{B}(0-100)$ represents that v solely invokes the algorithm in Section 4.2 to exploit its $\mathcal{B}_v(1)$ in the beginning of the 100 simulation rounds, and it then discovers both its $\mathcal{W}_v$ and $\mathcal{B}_v(1)$ in parallel.

In this experiment, in addition to the averaged broadcast delay, i.e.,

$$\frac{\sum_{v \neq u \in V} \min_{p \in \mathcal{P}_v \sim u} \ell_p}{\binom{|V|}{2}},$$

the broadcast scope of disseminating a message, originated by a randomly picked node, with $TTL = 7$ is measured. The experimental result presented in Fig. 9 is for the end-to-end host delay due to BRITE.

The simulation result reveals that if each node v collects its $\mathcal{W}_v$ first, then the broadcast scope increases considerably (see $\mathtt{W}(0-100)$ in Fig. 9b). This is because v's random walker converges the intended probability distribution for discovering its $\mathcal{W}_v$ in a rapid fashion by taking about 50 simulation rounds. The broadcast scope is reduced when v starts to collect its $\mathcal{B}_v(1)$. As discussed in Section 5.2, the broadcast scope in our proposal remains to be exponential. As compared to $\mathtt{W}(0-100)$, the broadcast scope in $\mathtt{B}(0-100)$ gradually increases and becomes stable by taking $\sim$300 simulation rounds (from the 100th round to the 400th one). This is mainly because v's random walker has difficulty in approximating the intended probability distribution for collecting the neighbors in its $\mathcal{W}_v$. This simulation result confirms our result in Theorem 2 and the effectiveness of our design in which $\mathcal{W}_v$ and $\mathcal{B}_v(1)$ are built in parallel.

Regarding averaged broadcast delay, we observe a similar scenario. $\mathtt{W}(0-100)$ takes a few rounds to minimize the broadcast delay (see a few rounds after the 100th one in Fig. 9a). In contrast, $\mathtt{B}(0-100)$ requires extra $\sim$100 rounds (from the 100th one to the 200th one) to further reduce the broadcast delay once the discovery of $\mathcal{W}_v$ is enabled.

5.4.2 Effects of Varying $\mathcal{T}$

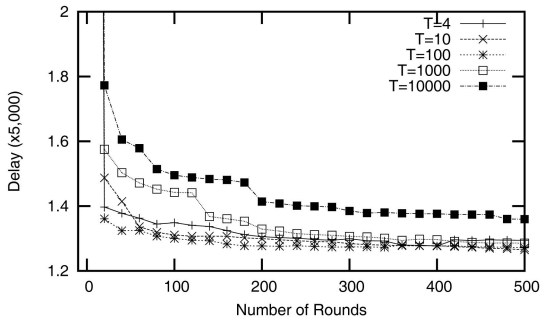
Section 4.2 concludes that the quality of $\mathcal{B}_v(1)$ for each $v \in V$ strongly depends on the system parameter, $\mathcal{T}$. If $\mathcal{T}$ is small, then v may not discover the optimal neighbor set for $\mathcal{B}_v(1)$. v may then have difficulty escaping such a local optimum and exploiting other $\mathcal{B}_v(1)$s with better qualities. In contrast, if $\mathcal{T}$ is large, $\mathcal{B}_v(1)$ is likely to include nodes randomly picked from the network regardless of the locations of the nodes. Consequently, the overlay formed by $\mathcal{B}_v(1)$s ($\forall v \in V$) may include a subnetwork resembling a random graph independent of the physical network topology. Hence, we investigate the effects of different $\mathcal{T}$ values in this section.

Fig. 10 illustrates our simulation result for $\mathcal{T} = 4$, 10, 100, 1,000, and 10,000. For each $v \in V$, $\mathcal{W}_v$ and $\mathcal{B}_v(1)$ are constructed in parallel. As Fig. 10a presents, when $\mathcal{T}$ is large (e.g., $\mathcal{T} = 10{,}000$), the averaged broadcast delay apparently increases. This indicates that our proposal operating in a large $\mathcal{T}$ tends to create a random network graph. This is confirmed by Fig. 10b since the broadcast scope also increases, given a large $\mathcal{T}$.

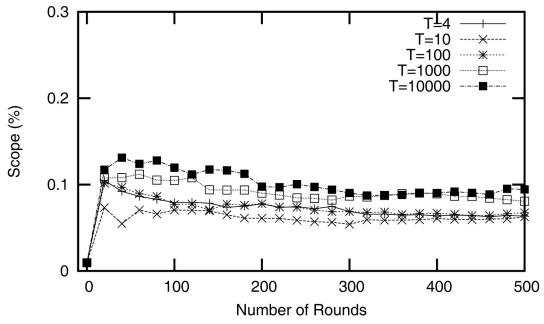
Second, when $\mathcal{T}$ decreases (e.g., $\mathcal{T} = 4, 10$, and 100), the broadcast delay is reduced accordingly, and it becomes stable in a rapid fashion after the 100th simulation round. This represents that each peer v is likely to be trapped in a local optimum in order to discover its $\mathcal{B}_v(1)$. Although the discovered $\mathcal{B}_v(1)$ may not be optimal, and exploiting $\mathcal{B}_v(1)$, in turn, decreases the broadcast scope (see Fig. 10b), the resultant overlay performs well in terms of broadcast delay.

5.5 Application Scenario: Gnutella

In this section, we investigate the efficiency of our proposal for the Gnutella search application [8]. We assume that there are m data objects, denoted by the set $\mathcal{S}$, in the system. We denote the relative popularity of an object $i \in \mathcal{S}$ by p_i. That is, $\sum_{i=1}^{m} p_i = 1$. As suggested in [42], the popularity of objects in $\mathcal{S}$ follows the Zipf-like distribution. Specifically, p_i

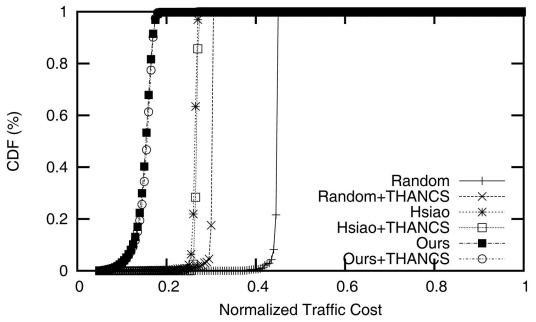


(a)

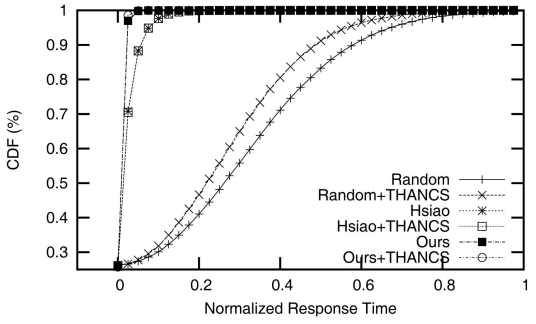


(b)

Fig. 10. The effect of varying $\mathcal{T}$. (a) Broadcast delay. (b) Broadcast scope ($TTL = 7$).



(a)



(b)

Fig. 11. The Gnutella application scenario. (a) Traffic cost. (b) Response time.

is proportional to $\frac{1}{i^{\gamma}}$, where $0.63 < \gamma < 1.24$ [43]. In addition, each object $i \in \mathcal{S}$ is replicated on r_i nodes in the system and the total number of objects in the system is $\sum_{i=1}^{m} r_i = \Gamma$. Here, r_i is proportional to $\sqrt{p_i}$ due to [44].

In the experiments, the performance metrics we measure are *traffic cost of a query* and *response time of a query*. Consider a query q. While the traffic cost of a query q represents the sum of the numbers of messages traversing the links connecting the network routers in the physical network due to q, the response time for q denotes the time elapsed of receiving a first reply since q is issued [9], [10], [11]. Note that

1. the simulation results discussed as follows are based on the BRITE topology,
2. the Gnutella network we simulate comprises 30,000 peers,
3. the total number of queries issued to the simulated Gnutella network is 300,000 (each participating peer generates 10 queries on average),
4. $m = 1{,}000$, $\gamma = 1$, and $\Gamma = 30{,}000$ in the experiments.

Figs. 11a and 11b depict the simulation results for the traffic cost and the query response time, respectively. For the Gnutella search application, the simulation results show that our proposal clearly outperforms `Random+THANCS` and `Hsiao` in terms of the traffic cost (normalized to the maximum one among the costs of all queries) and the query response time (normalized to the maximum one among the response delays of all queries). This demonstrates that an overlay well matching the physical underlay can greatly minimize the Gnutella traffic generated to the physical network and can considerably shorten the query response time.

6 Summary

We have presented in this paper a novel, decentralized topology matching algorithm for unstructured P2P networks. Our topology matching algorithm is driven by a rigorously analytical model. The unique feature of our algorithm is that it allows the communication delay of routing a scoped broadcast message from node v to node u to approximate the end-to-end delay from v to u in the physical network. To the best of our knowledge, our proposal is the first design driven by the rigorous analytical model, which approximates the optimum. In addition, our design guarantees the exponential broadcast scope.

We have assessed our proposal in extensive simulations. The simulation results not only reveal that our proposal is effective in reformatting an unstructured P2P network such that the resultant overlay matches well the physical network, but they also validate our theoretical results. In addition, we have compared our proposal to prior solutions, including those of Liu et al. [9] and Hsiao et al. [14]. While Liu et al.'s, Hsiao et al.'s, and our proposal have comparable broadcast scope, our design outperforms these prior solutions considerably in terms of broadcast delay. Furthermore, we have quantitatively investigated the performance of our proposal for the Gnutella search application, and the simulation results conclude that our proposal clearly outperforms the existing solutions in [9] and [14] in terms of the traffic cost and the query response time.

Acknowledgments

The authors are grateful to the anonymous reviewers who have provided them with valuable comments to improve

their study. This work was partially supported by the National Science Council, Taiwan, under Grant 97-2221-E-006-132 and 98-2221-E-006-096.

REFERENCES

[1] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network," *Proc. ACM SIGCOMM '01,* pp. 161-172, Aug. 2001.

[2] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," *Proc. ACM SIGCOMM '01,* pp. 149-160, Aug. 2001.

[3] A. Rowstron and P. Druschel, "Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems," *Lecture Notes in Computer Science,* pp. 161-172, Springer, Nov. 2001.

[4] B.Y. Zhao, L. Huang, J. Stribling, S.C. Rhea, A.D. Joseph, and J.D. Kubiatowicz, "Tapestry: A Resilient Global-Scale Overlay for Service Deployment," *IEEE J. Selected Areas in Comm.,* vol. 22, no. 1, pp. 41-53, Jan. 2004.

[5] X. Liao, H. Jin, Y. Liu, L.M. Ni, and D. Deng, "Scalable Live Streaming Service Based on Interoverlay Optimization," *IEEE Trans. Parallel and Distributed Systems,* vol. 18, no. 12, pp. 1663-1674, Dec. 2007.

[6] M. Ripeanu, I. Foster, and A. Iamnitchi, "Mapping the Gnutella Network," *IEEE Internet Computing,* vol. 6, no. 1, pp. 50-57, Jan./Feb. 2002.

[7] S. Sen and J. Wang, "Analyzing Peer-to-Peer Traffic Across Large Networks," *IEEE/ACM Trans. Networking,* vol. 12, no. 2, pp. 219-232, Apr. 2004.

[8] Gnutella, http://rfc-gnutella.sourceforge.net/, 2010.

[9] Y. Liu, L. Xiao, X. Liu, L.M. Ni, and X. Zhang, "Location Awareness in Unstructured Peer-to-Peer Systems," *IEEE Trans. Parallel and Distributed Systems,* vol. 12, no. 2, pp. 163-174, Feb. 2005.

[10] Y. Liu, "A Two-Hop Solution to Solving Topology Mismatch," *IEEE Trans. Parallel and Distributed Systems,* vol. 19, no. 11, pp. 1591-1600, Nov. 2008.

[11] Y. Liu, L. Xiao, and L.M. Ni, "Building a Scalable Bipartite P2P Overlay Network," *IEEE Trans. Parallel and Distributed Systems,* vol. 18, no. 9, pp. 1296-1306, Sept. 2007.

[12] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, "Equations of State Calculations by Fast Computing Machines," *J. Chemical Physics,* vol. 21, no. 6, pp. 1087-1092, June 1953.

[13] W.K. Hastings, "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrika,* vol. 57, no. 1, pp. 97-109, Apr. 1970.

[14] H.-C. Hsiao, H. Liao, and C.-C. Huang, "Resolving the Topology Mismatch Problem in Unstructured Peer-to-Peer Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 20, no. 11, pp. 1668-1681, http://doi.ieeecomputersociety.org/10.1109/TPDS.2009.24, Nov. 2009.

[15] J.M. Kleinberg, "Navigation in a Small World," *Nature,* vol. 406, p. 845, Aug. 2000.

[16] S. Milgram, "The Small-World Problem," *Psychology Today,* vol. 2, pp. 60-67, 1967.

[17] D.J. Watts and S.H. Strogatz, "Collective Dynamics of Small-World Networks," *Nature,* vol. 393, pp. 440-442, June 1998.

[18] S. Merugu, S. Srinivasan, and E. Zegura, "Adding Structure to Unstructured Peer-to-Peer Networks: The Use of Small-World Graphs," *J. Parallel and Distributed Computing,* vol. 65, no. 2, pp. 142-153, Feb. 2005.

[19] J.M. Kleinberg, "The Small-World Phenomenon: An Algorithm Perspective," *Proc. 32nd ACM Ann. Symp. Theory Computing (STOC '00),* pp. 163-170, May 2000.

[20] V.N. Padmanabhan and L. Subramanian, "An Investigation of Geographic Mapping Techniques for Internet Hosts," *Proc. ACM SIGCOMM '01,* pp. 173-185, Aug. 2001.

[21] T.S.E. Ng and H. Zhang, "Predicting Internet Network Distance with Coordinates-Based Approaches," *Proc. IEEE INFOCOM '02,* pp. 170-179, June 2002.

[22] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologically-Aware Overlay Construction and Server Selection," *Proc. IEEE INFOCOM '02,* pp. 1190-1199, June 2002.

[23] T. Qiu, G. Chen, M. Ye, E. Chan, and B.Y. Zhao, "Towards Location-Aware Topology in Both Unstructured and Structured P2P Systems," *Proc. 36th Int'l Conf. Parallel Processing (ICPP '07),* Sept. 2007.

[24] H. Zhang, A. Goel, and R. Govindan, "An Empirical Evaluation of Internet Latency Expansion," *ACM SIGCOMM Computer Comm. Rev.,* vol. 35, no. 1, pp. 93-97, Jan. 2004.

[25] H. Zhang, A. Goel, and R. Govindan, "Improving Lookup Latency in Distributed Hash Table Systems Using Random Sampling," *IEEE/ACM Trans. Networking,* vol. 13, no. 5, pp. 1121-1134, Oct. 2005.

[26] G. Phillips, S. Shenker, and H. Tangmunarunkit, "Scaling of Multicast Trees: Comments on the Chuang-Sirbu Scaling Law," *ACM SIGCOMM Computer Comm. Rev.,* vol. 29, no. 4, pp. 41-51, Oct. 1999.

[27] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, "Network Topology Generators: Degree-Based vs. Structural," *Proc. ACM SIGCOMM '02,* pp. 147-159, Aug. 2002.

[28] C.G. Plaxton, R. Rajaraman, and A.W. Richa, "Accessing Nearby Copies of Replicated Objects in a Distributed Environment," *Proc. Ninth ACM Symp. Parallel Algorithms and Architectures (SPAA '97),* pp. 311-320, June 1997.

[29] D.R. Karger and M. Ruhl, "Finding Nearest Neighbors in Growth-Restricted Metrics," *Proc. 34th ACM Ann. Symp. Theory Computing (STOC '02),* pp. 741-750, May 2002.

[30] S.M. Ross, "Markov Chains," *Introduction to Probability Models,* ninth ed., pp. 185-280, Academic Press, 2007.

[31] M. Mitzenmacher and E. Upfal, "Coupling of Markov Chains," *Probability and Computing: Randomized Algorithms and Probabilistic Analysis,* pp. 271-294, Cambridge Univ. Press, 2005.

[32] P. Diaconis and D. Stroock, "Geometric Bounds for Eigenvalues of Markov Chains," *Annals of Applied Probability,* vol. 1, no. 1, pp. 36-61, Feb. 1991.

[33] A. Sinclair, "Improved Bounds for Mixing Rates of Markov Chains and Multicommodity Flow," *Combinatorics, Probability and Computing,* vol. 1, no. 4, pp. 351-370, Dec. 1992.

[34] B. Bollobás, *Random Graphs,* second ed. Cambridge Univ. Press, 2001.

[35] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, "Optimization by Simulated Annealing," *Science,* vol. 220, no. 4598, pp. 671-680, May 1983.

[36] S. Saroiu, P.K. Gummadi, and S.D. Gribble, "Measurement Study of Peer-to-Peer File Sharing Systems," *Proc. Ninth SPIE/ACM Multimedia Computing Networking (MMCN '02),* Jan. 2002.

[37] K.P. Gummadi, R.J. Dunn, S. Saroiu, S.D. Gribble, H.M. Levy, and J. Zahorjan, "Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload," *Proc. 19th ACM Symp. Operating Systems Principles (SOSP '03),* pp. 314-329, Oct. 2003.

[38] G. Pandurangan, P. Raghavan, and E. Upfal, "Building Low-Diameter Peer-to-Peer Networks," *IEEE J. Selected Areas in Comm.,* vol. 21, no. 6, pp. 995-1002, Aug. 2003.

[39] S. Jin and H. Jiang, "Novel Approaches to Efficient Flooding Search in Peer-to-Peer Networks," *Computer Networks,* vol. 51, no. 10, pp. 2818-2832, July 2007.

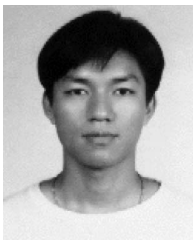
[40] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: An Approach to Universal Topology Generation," *Proc. Ninth Int'l Workshop Modeling, Analysis, and Simulation of Computer and Telecomm. Systems (MASCOTS '01),* pp. 346-353, Aug. 2001.

[41] PlanetLab, http://www.planet-lab.org/, 2010.

[42] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and Replication in Unstructured Peer-to-Peer Networks," *Proc. ACM Int'l Conf. Supercomputing (ICS '02),* pp. 84-95, June 2002.

[43] K. Sripanidkulchai, "The Popularity of Gnutella Queries and Its Implications on Scalability," O'REILLY P2P openp2p.com, http://www.oreillynet.com/topics/p2p/gnutella, 2010.

[44] E. Cohen and S. Shenker, "Replication Strategies in Unstructured Peer-to-Peer Networks," *Proc. ACM SIGCOMM '02,* pp. 177-190, Aug. 2002.



Hung-Chang Hsiao received the PhD degree in computer science from the National Tsing-Hua University, Taiwan, in 2000. He has been an assistant professor in computer science and information engineering at the National Cheng-Kung University, Taiwan, since August 2005. He was also a postdoc researcher in computer science at the National Tsing-Hua University, from October 2000 to July 2005. His research interests include peer-to-peer computing, overlay networking, and grid computing. He is a member of the IEEE Computer Society.

Hao Liao received the BS degree in electrical engineering from the National Cheng-Kung University, Taiwan, in 2004. He is currently working toward the PhD degree in computer science and information engineering at the National Cheng-Kung University. His research interests include peer-to-peer computing, grid computing, algorithm design, and analysis.

Po-Shen Yeh received the BS degree in computer science and information engineering at the National Chung-Hsing University, Taiwan, in 2007, and the MS degree in computer science and information engineering from the National Cheng-Kung University, Taiwan, in 2009. His research interests include algorithm design and analysis for peer-to-peer networks.