Resolving the Topology Mismatch Problem in Unstructured Peer-to-Peer Networks

Hung-Chang Hsiao, *Member*, *IEEE Computer Society*, Hao Liao, and Cheng-Chyun Huang

Abstract—Prior studies show that more than 70 percent of communication paths in a popular unstructured peer-to-peer (P2P) system (i.e., Gnutella) do not exploit the physical network topology, leading to the topology mismatch problem, and thus, lengthen communication between participating peers. While previous efforts in solving overlay topology matching problems do not guarantee the bounds of performance metrics (e.g., the communication delay between any two overlay peers and the broadcasting scope of any participating peer), in this paper, we present a novel topology matching algorithm that has provable performance qualities. In our proposal, each participating node creates and manages a constant number of overlay connections to other peers in a distributed manner. In rigorous performance analysis, we show that 1) the expected overlay communication delay between any two nodes in our P2P network is a constant; 2) in addition, any joining node has the exponential broadcasting scope in expectation; 3) furthermore, a participating node takes a polylogarithmic overhead to exploit the physical network locality and maintain its flooding scope. Together with extensive simulations, we present our proposal that significantly outperforms two recent solutions, i.e., THANCS and mOverlay, in terms of overlay communication latency and/or broadcasting scope.

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

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. Popular applications based on P2P networks include file sharing, distributed computing, media streaming applications, etc., which rely on their P2P network infrastructures for message routing, information search, and/or content delivery.

P2P network infrastructures are key building blocks for designing and implementing successful P2P applications. Potential P2P substrates are based on *distributed hash tables*, or DHTs. Examples of DHTs are CAN [1], Chord [2], Pastry [3], and Tapestry [4]. P2P networks based on DHTs, namely *structured overlays*, often rely on deterministically logical topologies (e.g., the d-dimensional lattice in CAN and the ring in Chord) to organize participating peers such that performance quality (e.g., the average hopcount of routing a message) can be guaranteed.

Unlike structured overlays, *unstructured P2P networks* (e.g., Gnutella [5]) do not rely on any deterministically logical topologies to organize participating peers. Peers in an unstructured overlay can interconnect to one another freely. Due to the absence of implementing deterministically logical structures, resource discovery in an unstructured network depends on message flooding. Typically, a message originator floods query messages with a positive time-to-live (or the TTL for short) value. Upon receiving a query message, the receiver decreases the associated TTL value by one and forwards the message to its neighbors except the one sending this message. A peer stops forwarding a query message if the associated TTL value is zero.

While previous studies (e.g., [6], [7]) have shown that unstructured P2P systems are popular and generate a large portion of traffic in the Internet, Liu [8] concluded that more than 70 percent of communication paths in an unstructured overlay do not exploit the physical network topology (i.e., the Internet topology), resulting in a topology mismatch between the overlay and the underlying network. This was mainly attributed to peers in an unstructured network that randomly choose their neighbors [8], [9], [10].

In this study, we are interested in improving the performance of unstructured P2P systems such that clients in an unstructured P2P system not only perceive low service time but also minimize traffic generated in the Internet. Specifically, consider an unstructured overlay network represented as an undirected graph $G = (V, E)$, where the sets of nodes (i.e., participating peers) and edges (i.e., overly links) between nodes are denoted by V and E, respectively. Any node v in V can perform searching by flooding its query messages to nodes within its *k-hop scope* (denoted by $S_v(k)$), where $S_v(k)$ includes those nodes receiving the query messages originated by v using $TTL = k$ (k is a predefined system parameter). In this study, we aim to build an unstructured P2P network "topology" such that 1) the communication latency of any two peers in the network is minimized as much as possible and 2) the number of nodes (i.e., $|S_v|$) in a k-hop scope of any node v is enlarged as much as possible. Precisely, we define the *averaged communication latency* for any two nodes in an overlay $G = (V, E)$ as

- *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 8 Aug. 2008; revised 25 Jan. 2009; accepted 4 Feb. 2009; published online 5 Feb. 2009.
Recommended for acceptance by K. Hwang.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2008-08-0302.
Digital Object Identifier no. 10.1109/TPDS.2009.24.*

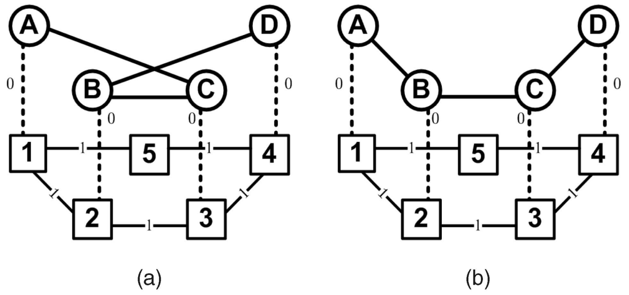


Fig. 1. An example illustrates two overlays (each consists of four peers, A, B, C, and D) over the same underlying network comprising five routers and implementing the shortest path routing algorithm (we assume that the communication delays between routers are identical and that each peer in the overlay takes zero latency to send a message to its connected router).

$$\frac{\sum_{u\in V, v\in V, u\neq v} \sum_{(i,j)\in P_{u\sim v}} d_{i,j}}{n \times (n-1)}, \tag{1}$$

where $P_{u\sim v} \subseteq G$ is the shortest path in G starting at any node $u \in V$ and ending at any node $v \in V - \{u\}$, $d_{i,j}$ represents the latency of an overlay link connecting two adjacent nodes in $P_{u\sim v}$, and $n = |V|$. We also define the *averaged broadcasting scope* as

$$\frac{\sum_{u\in V} |S_u(k)|}{n}. \tag{2}$$

We intend to construct an unstructured network G that not only minimizes the averaged communication latency, but also maximizes the averaged broadcasting scope. We believe that such a network topology G reduces the response time for a client to receive search results, and in the meantime, minimizes the traffic introduced to the Internet due to the reduction of communication delay between overlay peers. In addition, the quality of query results will be improved accordingly since the number of nodes receiving query messages is maximized and the probability of successfully resolving queries is thus enlarged [8], [9], [10].

Fig. 1 depicts an example illustrating two unstructured networks over the same underlying network. Assume that $TTL = 1$. In Fig. 1a, the averaged communication latency and the averaged broadcasting scope are 1.67 and 1.5, respectively, while both metrics are, respectively, 1 and 1.5 in Fig. 1b. Clearly, the overlay in Fig. 1b outperforms that in Fig. 1a.

Liu [8] showed that given Δ, δ, and ϵ (where ϵ is a small positive constant), the construction of an unstructured overlay G is that 1) any participating node in G has the degree within $[\Delta - \epsilon, \Delta + \epsilon]$ and 2) the averaged communication latency in G, which is no more than δ, is $\mathcal{NP}$-*complete*. Notably, a random regular graph in which nodes have degrees within $[\Delta - \epsilon, \Delta + \epsilon]$ is a good expander [11], [12], [13], thus guaranteeing a large broadcasting scope for any node in the graph. However, to the best of our knowledge, building unstructured overlay networks based on random graph techniques (e.g., the designs in [11], [12], [13]) does not take physical communication delays into consideration.

1.1 Our Contributions

The overlay mismatch problem is a fundamental issue. In this paper, we tackle the fundamental problem and offer a novel network construction algorithm for unstructured P2P networks. Our network construction algorithm operates in a distributed manner and builds the network, which minimizes the averaged communication latency between nodes in the network and maximizes the averaged broadcasting scope.

First, we summarize our contributions as follows: while any node in our network maintains a constant number of neighbors, our constructed network has provable performance metrics (i.e., the expected communication latency and the expected broadcasting scope). Through rigorous performance analysis, our network guarantees that the expected communication latency is $\Theta(1)$ regardless of the network size (i.e., the number n of nodes joining the network) and the expected broadcasting scope is $\Omega(2^{TTL})$.

Second, we evaluate our proposal in extensive simulations. We compare our proposal with recent works presented by Liu [8] and Zhang et al. [14]. Our simulation results reveal that our design significantly outperforms the solutions in [8], [14] in terms of the expected communication latency and/or broadcasting scope. While the efforts of Liu [8] and Zhang et al. [14] provided elegant network construction algorithms for building locality-aware unstructured P2P systems, our study, to the best of our knowledge, presents a first topology matching algorithm having provable performance metrics for unstructured networks.

Third, our network construction algorithm operates in a lightweight fashion that any node in our network takes the polylogarithmic number of messages to exploit its physical network locality and maintain its exponential broadcasting scope.

1.2 Roadmap

The remainder of the paper is organized as follows: Section 2 discusses related works. We present our topology matching algorithm in Section 3. Section 4 provides the theoretical performance analysis for our proposal. We also evaluate our proposal in simulations and the simulation results are given in Section 5. We summarize our study in Section 6 with possible future research directions.

2 Related Work

Liu et al. in [9] presented a premier study for the topology mismatch problem between unstructured P2P networks and their underlying networks. For optimizing the P2P network topology G such that neighboring peers in the resultant overlay G' perceive low communication latencies without shrinking their search scopes, Liu [8], Liu et al. [9], [10] proposed that any three nodes in G form a three-edge ring subnetwork $\mathcal{G}$, and among the three edges in $\mathcal{G}$, the one having the longest delay is removed without partitioning the three nodes. To minimize the overhead of discovering a three-edge ring network and to maximize the broadcasting scope of any node, Liu et al. suggested that the three nodes forming the ring shall be those nodes having their overlay hopcount distance no more than two in G. While Liu et al.'s solution is simple and elegant, their solution does not provide guarantees for the performance metrics like the averaged communication delay as defined in (1) and the averaged broadcasting scope in (2). In contrast, we present an unstructured network that guarantees the expected communication latency and the expected broadcasting scope. Our proposal is compared against the state-of-the-art solution given by Liu [8]. The simulation results reveal that our proposal significantly outperforms Liu's proposal in terms of expected communication latency. Regarding the broadcasting scope, our proposal is comparable to Liu's if his solution is based on random regular graphs as discussed in the following.

Random d-regular graphs [15] are good candidates for unstructured P2P network topologies. A random d-regular graph is a graph picked uniformly at random from all d-regular graphs (i.e., each node in the graph has the degree of d). Bollobás and Vega presented a centralized construction of random regular graphs based on the configuration model in [16] and showed that the probability of every random regular graph having the logarithmic diameter (in terms of hopcount) approaches one if the number of nodes n in the graph goes to infinity. However, it may not be pragmatic in constructing random regular graphs in a P2P environment with dynamic entities. For example, a recent study by Law and Siu [11] proposed a distributed construction for random $2d$-regular graphs based on d Hamiltonian cycles. Each node needs to join each of the d Hamiltonian cycles. However, the proposal in [11] requires obtaining global locks on Hamiltonian cycles when nodes join, which is clearly impractical in a dynamic P2P environment. In contrast, other studies, e.g., [12], [13], have presented less sophisticated algorithms for building random (d, ϵ)-regular graphs, where each node in the graph has the degree within $[d - \epsilon, d]$. Although random d-regular or (d, ϵ)-regular graphs are attractive to unstructured P2P systems due to low diameter and large flooding scope, nodes connect one another with the same probability, and therefore, cannot minimize their communication latencies. In Section 5, we will show that the averaged broadcasting scope provided by our proposal is comparable to that of random regular graphs. However, our constructed network considerably outperforms random regular graphs in terms of the averaged communication latency.

A recent study [14] closely relevant to ours presented *mOverlay*—a locality-aware unstructured P2P network. In mOverlay, peers in the proximity form a group and mOverlay maintains the invariant that nodes allocated to the same group $\mathcal{G}$ have the identical network latency to each of the d neighboring groups of $\mathcal{G}$. Although mOverlay presents an interesting construction for locality-aware P2P network, it provides no rigorous performance guarantee for the communication latency for any route in the network.

Small-world networks exhibit low diameter and high cluster coefficient[1] [17], [18], [19]. Previous works (e.g., [20], [21], [22]) presented distributed constructions for building small-world-based P2P networks. While [21], [22] depended on the high cluster coefficient exhibited by small-world networks to handle flash crowd and to enhance system connectivity, Merugu et al. [20] suggested to cluster nearby nodes and to maintain a few distant nodes for each peer. The simulation results illustrated in Merugu et al.'s study [20] concluded that the instance of small-world networks that can exploit physical network locality exists. However, they did not detail on how to design and optimize such a small-world-based P2P network. Unlike them, our proposal in the paper presents a locality-aware construction for unstructured networks. We detail how to construct a flooding-based P2P network that minimizes the expected communication latency as defined in (1) and maximizes the expected broadcasting scope as defined in (2). In addition to simulations, we provide a rigorous, insight performance analysis for our design.

In unstructured P2P networks, nodes may form cycles. If nodes on the cycles are within a broadcasting scope, then redundant query messages will be introduced, and thus, will generate extra overhead to the underlying network. Regarding this, ACE [23] and Lightflood [24] proposed to eliminate cycles from unstructured P2P systems. Notably, given an optimized overlay topology G, we may additionally include the cycle removing algorithms as presented in [23], [24] to further enhance G.

Our work is also orthogonal to [25]. Conceptually, given the set $\mathcal{V}$ of physical network locations, a distance function $\mathcal{D} : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ and an overlay graph $G = (V, E)$, a one-to-one function $\mathcal{F} : V \rightarrow \mathcal{V}$ assigns participating peers in V to the locations in $\mathcal{V}$ of the physical network, which determines the overall performance of G in accordance with the metric defined in (1). Qiu et al. [25] intended to find $\mathcal{F}$ for minimizing the metric in (1). Our study seeks to construct $G = (V, E)$. Thus, once our overlay G is constructed, we may further optimize G using the algorithm presented in [25].

3 Topology Matching Algorithm

We present our proposal in this section. The notations frequently used in the paper are defined in Section 3.1. Section 3.2 details our topology matching algorithm. We then discuss the implementation of our algorithm in Section 3.3. A working example is given in Section 3.4 to illustrate our idea. The performance analysis of our proposal is discussed later in Sections 4 and 5.

3.1 Preliminaries

Fig. 2 shows example of our constructed P2P network $G = (V, E)$ consisting of eight peers (i.e., $|V| = 8$) and 10 connections (i.e., $|E| = 10$) between these peers. The variable G is an undirected graph.[2] In G, each peer $u \in V$ has at most γ distinct IDs, denoted by $u^{(1)}.id, u^{(2)}.id, \ldots, u^{(\gamma)}.id$, where $u^{(i)}.id$ $(1 \leq i \leq \gamma)$ is the ith ID of u. The variable u in G maintains Δ links at most to other peers ($\Delta = 5$ in Fig. 2) and Δ is a system parameter. The degree of a peer $u \in V$ is denoted by $\deg_u$ and $\deg_u \leq \Delta$. For example, in Fig. 2, peer C with IDs $C^{(1)}.id = 20$ and $C^{(2)}.id = 24$ has $\deg_c = 4$, assuming that $\gamma = 2$.

[1] The *cluster coefficient* for a node v in a graph $G = (V, E)$ is defined as the ratio of the number of existing connections between vs neighboring nodes to $m \times (m - 1)$, where m is the number of neighboring nodes of v. Small-world networks exhibiting high cluster coefficient represent that neighboring nodes of any node v likely connect one another.

[2] The Gnutella network can be represented as an undirected graph since neighboring nodes in the network may ping and pong messages to one another [5].

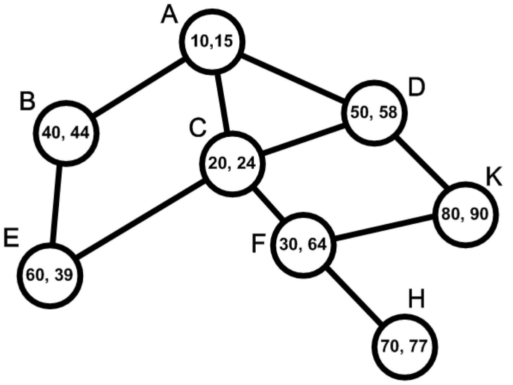


Fig. 2. An example of our unstructured P2P network, where the number in the left (right) of a node represents the first (second) ID of that node.

Given a value r (where $1 \le r \le \gamma$), $u^{(r)} \overset{-}{\rightsquigarrow} v^{(r)}$ denotes a path that is a subgraph of G in which the assembling nodes on the path have their rth IDs in decreasing order. For instance, in Fig. 2, the path of $H^{(1)} \overset{-}{\rightsquigarrow} A^{(1)}$ is $70 \rightarrow 30 \rightarrow 20 \rightarrow 10$. A node w assembling the path $u \overset{-}{\rightsquigarrow} v$ is denoted by $w \in u^{(r)} \overset{-}{\rightsquigarrow} v^{(r)}$. Clearly, $C \in H^{(1)} \overset{-}{\rightsquigarrow} A^{(1)}$, for example. Similarly, a routing path comprising nodes with IDs in increasing order is denoted by $u^{(r)} \overset{+}{\rightsquigarrow} v^{(r)}$. For example, $C^{(2)} \overset{+}{\rightsquigarrow} K^{(2)} = 24 \rightarrow 64 \rightarrow 90$ in Fig. 2.

Note that we denote $\mathcal{H}_u$ as the path $u^{(1)} \overset{-}{\rightsquigarrow} v^{(1)}$, where v represents the unique node having the smallest ID in the system. Here, the unique node with the smallest ID is the bootstrap node,[3] which helps nodes join the system. For example, in Fig. 2, $\mathcal{H}_H = H^{(1)} \overset{-}{\rightsquigarrow} A^{(1)} = 70 \rightarrow 30 \rightarrow 20 \rightarrow 10$.

Finally, when we say that among the nodes in the set $\mathcal{S} \subseteq V$, a node $u \in \mathcal{S}$ is the "physically closest" to a given node $v \in V - \mathcal{S}$ if in the underlying network there does not exist any node $w \in \mathcal{S} - \{u\}$ such that the end-to-end communication delay between w and v is smaller than that between u and v.

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

TABLE 1
The Notations Frequently Used in This Paper

Notation	Description
$u^{(i)}.id$	the i-th ID of node u
Δ	the maximal number of connections available to a node
$\deg_u$	the number of connections node u currently maintains
$u^{(r)} \overset{-}{\rightsquigarrow} v^{(r)}$	the routing path from u towards v assembled by nodes with their r-th IDs in decreasing order
$u^{(r)} \overset{+}{\rightsquigarrow} v^{(r)}$	the routing path from u towards v assembled by nodes with their r-th IDs in increasing order
$\mathcal{H}_u$	$\mathcal{H}_u = u^{(1)} \overset{-}{\rightsquigarrow} z^{(1)}$, where z is the node (i.e., the bootstrap) having the smallest ID in the system
$B_u^{(red)}$	the red neighboring set of node u
$B_u^{(blue)}$	the blue neighboring set of node u
B_u	$B_u = B_u^{(red)} \cup B_u^{(blue)}$
γ	the maximal number of nodes in $B_u^{(red)}$
β	the maximal number of nodes in $B_u^{(blue)}$
$G^{(red)}$	$G^{(red)} = \left(V, E^{(red)}\right) \subseteq G$, where $E^{(red)} = \left\{(i,j) \mid i \in V, j \in B_i^{(red)}\right\}$
$\max_u^{(red)}$	the maximal number of connections node u can maintain in $G^{(red)}$
$\deg_u^{(red)}$	the number of connections node u currently maintains in $G^{(red)}$

3.2 Algorithm

3.2.1 Overview

Consider an unstructured P2P network $G = (V, E)$ that our algorithm, detailed in the following sections, constructs. $G = (V, E)$ comprises two subnetworks $G^{(red)} = (V^{(red)}, E^{(red)})$ and $G^{(blue)} = (V^{(blue)}, E^{(blue)})$. Specifically, $G^{(red)}$ contains all the nodes in G (i.e., $V^{(red)} = V$) and ensures that there exists at least one path between any two nodes in G such that G is connected. In contrast, $G^{(blue)}$ in our design may not include all the nodes in G, i.e., $V^{(blue)} \subseteq V$. The nodes in $G^{(blue)}$ are those nodes in $(V, E - E^{(red)})$ that have free connections for linking to other nodes. The nodes in $G^{(blue)}$ fully utilize their remaining connections. Consequently, the created overlay links in both $G^{(red)}$ and $G^{(blue)}$ together form E of G (i.e., $E = E^{(red)} \cup E^{(blue)}$).

In our design, $G^{(red)}$ may generate more protocol overhead than $G^{(blue)}$. Thus, our proposal suggests to create a less number of overlay links in $G^{(red)}$. We guarantee that the overlay routing delay between any two nodes in $G^{(red)}$ is constant in expectation. If there exists an overlay path between two nodes in $G^{(blue)}$, the communication delay of such a path is also proved to be constant in expectation.

3.2.2 Node Joining

Algorithm 1 details how a node u joins our network $G = (V, E)$. Basically, a node u in our network partitions its neighbors, denoted by B_u, into two subsets, namely the *red set* (denoted by $B_u^{(red)}$) and the *blue set* ($B_u^{(blue)}$). When u joins the network, it immediately creates its $B_u^{(red)}$ and joins G by connecting to nodes in $B_u^{(red)}$. The maximal number of nodes in the red neighbor set is γ. In parallel, u discovers its blue neighbor set $B_u^{(blue)}$ such that $B_u^{(blue)}$ has β members at most. We will detail later in Section 4 that γ is picked as small as possible to minimize the system operational overhead and β can thus be set to utilize the remaining connections available to a peer.

Clearly, the overlay links between nodes in red neighboring sets form a subgraph $G^{(red)} = (V, E^{(red)}) \subseteq G = (V, E)$, where $E^{(red)} = \{(i,j) | i \in V, j \in B_i^{(red)}\}$. We denote the degree of a node u in $G^{(red)}$ by $\deg_u^{(red)}$ and the maximal degree that u can contribute to $G^{(red)}$ is denoted by $\max_u^{(red)}$. Hence, u can connect to $\beta = \Delta - \max_u^{(red)}$ blue neighbors at most. Note that the notation $B_u^{(red)}$ shall not be confused, which only includes those nodes that u "proactively" requests to connect. Thus, $\deg_u^{(red)}$ may be not less than γ since in addition to $B_u^{(red)}$, u maintains the overlay links due to $u \in B_v^{(red)}$ for some vs. Similarly, $B_u^{(blue)}$ denotes the nodes that are proactively linked by u.

When constructing $B_u^{(red)}$, u first samples s_γ nodes (denoted by the set $\mathcal{R}$) uniformly at random from V. Given a random value r ($1 \le r \le \gamma$), each sampled node $v \in \mathcal{R}$ then discovers a routing path starting from itself toward a node z having the smallest (or the largest) ID in the system. Precisely,

[3] Most peer-to-peer systems such as Gnutella [5] rely on a bootstrap (or a number of bootstraps) to help nodes participate in the system.

v finds a path $v^{(r)} \overset{-}{\rightsquigarrow} z^{(r)}$ (or $v^{(r)} \overset{+}{\rightsquigarrow} z^{(r)}$). Thereafter, v includes a node $w \in v^{(r)} \overset{-}{\rightsquigarrow} z^{(r)}$ (or a node $w \in v^{(r)} \overset{+}{\rightsquigarrow} z^{(r)}$) 1) having the ID smaller than $u^{(r)}.id$ (i.e., $w^{(r)}.id < u^{(r)}.id$) and 2) not exceeding its degree bound in $G^{(red)}$ (i.e., $\deg_w^{(red)} < \max_w^{(red)}$) into the candidate set $\mathcal{S}^{(red)}$. Among the nodes in $\mathcal{S}^{(red)}$, u only adds the physically closest to its $B_u^{(red)}$.

```
input  : $G = (V, E)$ and $u$
output : $G' = (V', E')$, where $V' = V \cup \{u\}$,
         $E' = E \cup_{v \in B_u} (u, v)$, and $B_u \subseteq V$ is $u$'s neighbors,
         where $B_u = B_u^{(red)} \cup B_u^{(blue)}$
while $|B_u^{(red)}| < \gamma$ do
    $\mathcal{S}^{(red)} \leftarrow \emptyset$;
    $\mathcal{R} \leftarrow s_\gamma$ nodes sampled uniformly at random from $V$;
    for each $v \in \mathcal{R}$ do
        $u.id \leftarrow \textsc{Random}(1, \mathrm{MaxID})$;
        $r \leftarrow \textsc{Random}(1, \gamma)$;
        $\mathcal{S}^{(red)} \leftarrow \mathcal{S}^{(red)} \cup \{w\}$, where
        $w \in v^{(r)} \overset{-}{\rightsquigarrow} z^{(r)} \vee w \in v^{(r)} \overset{+}{\rightsquigarrow} z^{(r)}$,
        $w^{(r)}.id < u^{(r)}.id$, and $\deg_w^{(red)} < \max_w^{(red)}$;
    Let $v \in \mathcal{S}^{(red)}$ be the node physically closest to $u$;
    $B_u^{(red)} \leftarrow B_u^{(red)} \cup \{v\}$;
$\beta \leftarrow \Delta - \max_u^{(red)}$;
for $i = 1$ to $\beta$ do
    $\mathcal{S}^{(blue)} \leftarrow \emptyset$;
    $\mathcal{S}^{(blue)} \leftarrow s_\beta$ nodes each $w$ is sampled with the probability
    of $\Pr(\mathcal{H}_w, n)$ from $V$, where $n = |V|$;
    Let $v \in \mathcal{S}^{(blue)}$ be the node physically closest to $u$;
    $B_u^{(blue)} \leftarrow B_u^{(blue)} \cup \{v : \deg_v - \max_v^{(red)} > 0\}$;
```

Algorithm 1. JOIN—A peer u picks nodes in $G = (V, E)$ to connect.

First, we note that finding a node w on $v^{(r)} \overset{-}{\rightsquigarrow} z^{(r)}$ or on $v^{(r)} \overset{+}{\rightsquigarrow} z^{(r)}$ is based on a newly generated ID of u (in Algorithm 1, RANDOM$(1, R)$ picks an ID from 1 to R independently and uniformly at random). u maintains at most γ distinct IDs, denoted by $u^{(1)}.id, u^{(2)}.id, \ldots, u^{(\gamma)}.id$. Second, either $w \in v^{(r)} \overset{-}{\rightsquigarrow} z^{(r)}$ or $w \in v^{(r)} \overset{+}{\rightsquigarrow} z^{(r)}$ can occur since $u^{(r)}.id$ is either smaller or larger than $v^{(r)}.id$.

For constructing $B_u^{(blue)}$, u first samples s_β nodes (represented by $\mathcal{S}^{(blue)}$) from V and each $w \in \mathcal{S}^{(blue)}$ is then picked with the probability of $\Pr(\mathcal{H}_w, n)$. Notably, $\Pr(\mathcal{H}_w, n)$ is independent of $\Pr(\mathcal{H}_{\tilde{w}}, n)$ for any $\tilde{w} \in V - \{w\}$. Among the nodes in $\mathcal{S}^{(blue)}$, u selects the physically closest node v and includes v into $B_u^{(blue)}$. As we have mentioned, the blue neighboring set of a node u is used to utilize the available connections provided by u.

3.2.3 System Evolving

Since participating nodes may depart or fail, an active peer u in our system collects its $B_u^{(red)}$ periodically. We will discuss later in this paper the period Δt that u collects $B_u^{(red)}$. On the other hand, if u detects the departure/failure of any neighboring node w in its $B_u^{(red)}$, u simply picks a newly discovered node to replace w. However, this is subject to the degree constraint and ID ordering as we discussed for Algorithm 1. We will later in Section 4.3.3 show that our protocol guarantees the connectivity of the system since there exists at least one path, which consists of the node with the smallest ID between any two nodes in the network.

Similarly, an active peer u in our system collects its $B_u^{(blue)}$ periodically. u replaces failure nodes with those newly discovered in $B_u^{(blue)}$.

3.3 Implementation

Algorithm 1 has presented the details of constructing neighbors for a node. We discuss how to implement the algorithm in a distributed manner in this section.

For constructing and maintaining $B_u^{(red)}$ of any $u \in V$, u needs to sample a number of nodes (i.e., $\mathcal{R}$) uniformly at random from the set of nodes participating in the system. For sampling nodes uniformly at random, we implement the Markov Chain Monte Carlo method [26], [27]. Precisely, in our implementation, each node x in the network maintains the following transition probabilities:

$$\Pr_{x,y} = \begin{cases} \frac{1}{M}, & \text{if } y \in B_x, \\ 0, & \text{if } y \notin B_x, \\ 1 - \frac{|B_x|}{M}, & \text{if } x = y, \end{cases} \tag{3}$$

where M is any number not less than $n = |V|$. We assume that the number of nodes n in the system is known, which may be approximated using the proposals such as [28]. Thus, u can issue a random walker to sample nodes based on the transition probabilities in (3) via any existing node as an entry point.[4] If the system operates for a sufficiently long time,[5] the nodes in V are sampled with the same probability (i.e., $\frac{1}{n}$). This is because, if for any $x, y \in V$ (where $x \neq y$) their stationary probabilities π_x and π_y are equal, then the Markov Chain associated with the transition probabilities defined in (3) is time reversible due to $\pi_x \Pr_{x,y} = \pi_y \Pr_{y,x}$, resulting in a unique, uniform stationary distribution. The random walker associated with u periodically visits nodes in the system when u remains in the system. u samples nodes through its walker. Since u and its random walker ping and pong messages to one another, us random walker leaves the system due to the departure/failure of u.

On the other hand, when constructing $B_u^{(red)}$, each sampled node $v \in \mathcal{R}$ is requested to issue a route toward the node z having the smallest (or the largest) ID, i.e., v discovers the route path $v^{(r)} \overset{-}{\rightsquigarrow} z^{(r)}$ or the path $v^{(r)} \overset{+}{\rightsquigarrow} z^{(r)}$. For discovering the path $v^{(r)} \overset{-}{\rightsquigarrow} z^{(r)}$, each node $p \in V$ implements the message forwarding algorithm as shown in Algorithm 2. In Algorithm 2, upon receiving a route discovery message issued by $v \in \mathcal{R}$, node p forwards the message to its neighbor having the rth ID smaller than $p^{(r)}.id$. The route discovery message is forwarded until it reaches a node z whose neighboring nodes in B_z cannot relay the message further, thus resulting in the path $v^{(r)} \overset{-}{\rightsquigarrow} z^{(r)}$.

For constructing the route path $v^{(r)} \overset{+}{\rightsquigarrow} z^{(r)}$, we perform similar operations as shown in Algorithm 2 by finding a node q having $q^{(r)}.id > p^{(r)}.id$. That is, p forwards the message to its neighbor with the rth ID larger than $p^{(r)}.id$. This message is forwarded toward the peer having the largest ID.

[4] Similar to most unstructured P2P systems (e.g., Gnutella), our design depends on a bootstrap node (or a number of bootstraps) that provides an entry point helping a joining/rejoining node to participate in the system.

[5] Random walks have rapid *mixing time* (i.e., the time converged to the targeted stationary probability distribution) if the mixing time is a polynomial in the variation distance and the problem size (here, it is the size of the network) [27]. Zhong et al. [29] show that random walks have rapid mixing time to converge to uniform distribution in most P2P networks.

```
input : $p$
Let $q$ be a node $\in B_p^{(red)}$ and $q^{(r)}.id < p^{(r)}.id$;
if $\{q\} \neq \emptyset$ then
    $p$ forwards the message to $q$;
else
    $p$ stops forwarding the message;
```

Algorithm 2. FORWARD—A peer p helps discover the route path $v^{(r)} \overset{-}{\rightsquigarrow} z^{(r)}$.

Notably, in Algorithm 1, given r and $v \in \mathcal{R}$, u creates and maintains $\mathcal{S}^{(red)}$, the candidate set from which u finds the physically closest node to connect, by including a node w having 1) the rth ID smaller than $u^{(r)}.id$ and 2) the degree less than $\max_w^{(red)}$ on the path $v^{(r)} \overset{-}{\rightsquigarrow} z^{(r)}$ (or the path $v^{(r)} \overset{+}{\rightsquigarrow} z^{(r)}$). In this paper, to simplify our implementation for sampling $\mathcal{S}^{(red)}$, an issued route for discovering $v^{(r)} \overset{-}{\rightsquigarrow} z^{(r)}$ (or $v^{(r)} \overset{+}{\rightsquigarrow} z^{(r)}$) terminates once we find a node meeting the constraints of ID ordering and node degree bound.

Similar to $\mathcal{S}^{(red)}$ discovered by a node u, u creates and maintains an s_β-node set $\mathcal{S}^{(blue)}$ and then includes the physically closest node in $\mathcal{S}^{(blue)}$ into $B_u^{(blue)}$. To create $\mathcal{S}^{(blue)}$, each $w \in \mathcal{S}^{(blue)}$ is picked from V based on the random walk as mentioned above. Then, u selects w into its $\mathcal{S}^{(blue)}$ according to the probability distribution $\Pr(\mathcal{H}_w, n)$. Note that this probability distribution depends on $\mathcal{H}_w$ and $n = |V|$. For resolving $\mathcal{H}_w$ of any node $w \in V$, in our implementation, $w \in V$ needs to periodically perform the route $w^{(1)} \overset{-}{\rightsquigarrow} z^{(1)}$. This can be accomplished by implementing Algorithm 2. Based on $\mathcal{H}_w$ and n, w can then determine its probability (i.e., $\Pr(\mathcal{H}_w, n)$) for being selected into $\mathcal{S}^{(blue)}$ by u.

We will show later in Section 4 that there exists a nonuniform probability distribution $\Pr(\mathcal{H}_w, n)$ such that the expected communication latency between any two nodes in the network is a constant.

3.4 An Example

Fig. 3 depicts an example to show how our algorithm works, where the initial network $G = (V, E)$ is as shown in Fig. 2. We assume in this example that $\gamma = 2$, $\beta = 1$, $s_\gamma = 3$, $s_\beta = 3$, $\Delta = 5$, and $\max_u^{(red)} = 4$ for all $u \in V$.

In Fig. 3a, node Q with IDs $Q^{(1)}.id = 45$ and $Q^{(2)}.id = 73$ joins the network G, which samples nodes $\mathcal{R} = \{C, E, H\}$ uniformly at random from G due to $s_\gamma = 3$. Nodes in $\mathcal{R}$ then help discover routes toward the node having the smallest ID (i.e., $A^{(1)}.id = 10$) or the largest ID (i.e., $K^{(2)}.id = 90$). In this example, given $r = 1, r = 1$, and $r = 2$, the corresponding nodes C, E, and H discover the paths $20 \rightarrow 30 \rightarrow 80$ ($C^{(1)} \overset{+}{\rightsquigarrow} K^{(1)}$), $60 \rightarrow 40 \rightarrow 10$ ($E^{(1)} \overset{-}{\rightsquigarrow} A^{(1)}$), and $77 \rightarrow 64 \rightarrow 24 \rightarrow 15$ ($H^{(2)} \overset{-}{\rightsquigarrow} A^{(2)}$), respectively. The first nodes meeting the constraints of ID ordering and node degree bound on the paths toward node K, A, and A are F, B, and F, respectively, and $\mathcal{S}^{(red)}$ is thus $\{B, F\}$. Since $\gamma = 2$, node Q then connects to the two closest nodes in $\mathcal{S}^{(red)}$, i.e., $B_Q^{(red)} = \{B, F\}$ (see Fig. 3b).

For discovering the blue neighboring set (i.e., $B_Q^{(blue)}$) of node Q, in accordance with a given probability distribution $\Pr(\mathcal{H}_w, n)$, the random walker issued by node Q visits nodes in G, resulting in the set $\mathcal{S}^{(blue)} = \{D, H, K\}$ (Fig. 3c). We recall that $\mathcal{H}_w$ is the route $w^{(1)} \overset{-}{\rightsquigarrow} z^{(1)}$. For example, in Fig. 3c, node K has $\mathcal{H}_K = K^{(1)} \rightsquigarrow A^{(1)} = 80 \rightarrow 30 \rightarrow 20 \rightarrow 10$. Subsequently, node Q links to the node in $B_Q^{(blue)} = \{H\}$, which is assumed to be the physically closest to Q, among $\mathcal{S}^{(blue)} = \{D, H, K\}$.

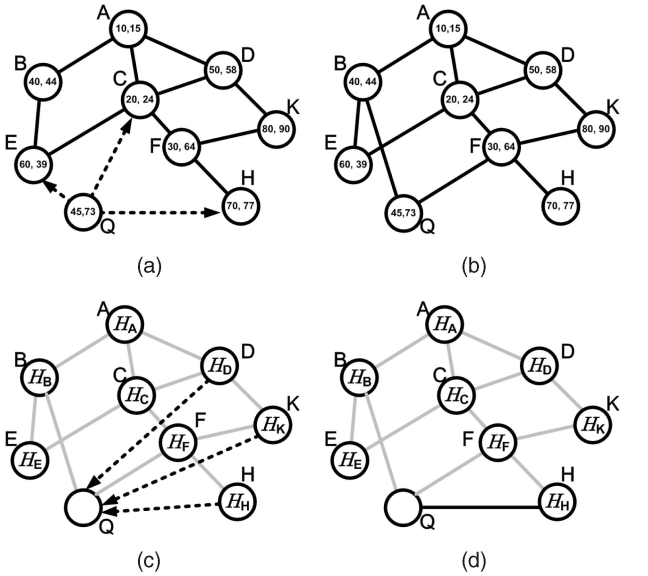


Fig. 3. An example to illustrate Algorithm 1, where (a) node Q requests nodes in $\mathcal{R} = \{C, E, H\}$ to help build $\mathcal{S}^{(red)} = \{B, F\}$, (b) node Q then connects to the nodes in $B_Q^{(red)} = \{B, F\}$, (c) node Q has $\mathcal{S}^{(blue)} = \{D, H, K\}$, and (d) node Q connects to the physically closest node in $\mathcal{S}^{(blue)}$ (i.e., $B_Q^{(blue)} = \{H\}$).

4 Theoretical Performance Analysis

We provide an insight performance analysis of our proposal in this section. Particularly, our design shown in Algorithm 1 strongly depends on the parameters, γ, β, s_γ, and s_β. We will determine the values of these system parameters in this section. Our design is compared to earlier proposals in simulations and the simulation results are discussed in Section 5.

We first provide the network latency model considered in our study (see Section 4.1). Section 4.2 then models the lifetime of a peer. The theoretical analysis results are then presented in Section 4.3.

4.1 Network Latency Model

Prior studies in [30], [31], [32] presented that the latency distribution between Internet end-hosts are likely to follow the *power-law expansion*. In this study, we are thus concerned with the power-law latency expansion graphs.

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

Let the maximum delay between any two nodes in the graph be $\mathcal{L}$. Then, the probability distribution of $\chi_v(z)$ is

$$\Pr(\mathcal{Z} \le z) = \left(\frac{z}{\mathcal{L}}\right)^\alpha, \tag{4}$$

where $\mathcal{Z}$ is the random variable that represents the latency from v to any node u in the graph. Without loss of generality, we let $\mathcal{L} = 1$ in this study. Note that α is typically a small value. For example, $\alpha = 0.4$ in our simulations (see Section 5).

The result stated in the following helps estimate the expected communication latency between any two nodes in our unstructured P2P network. The details for analyzing the expected communication latency, as defined in (1), are given in Section 4.3.

Lemma 1. *Let $Z_1, Z_2, \dots, Z_k$ be independent, identical random variables over $[0,1]$ and follow probability distribution with the α-power-law latency expansion, $\Pr(\mathcal{Z} \leq z) = z^\alpha$. Let $Z_{min} = \min\{Z_1, Z_2, \dots, Z_k\}$. Then, the expected value Z_{min} is $\mathbf{E}[Z_{min}] \leq k^{\frac{-1}{\alpha}}$.*

Proof. Since $Z_1, Z_2, \dots, Z_k$ are independent and $Z_{min} = \min\{Z_1, Z_2, \dots, Z_k\}$, we have

$$\begin{aligned}\Pr\big(Z_{min} > z\big) &= \Pr\big(\min\{Z_1, Z_2, \dots, Z_k\} > z\big)\\ &= \Pr\left(\bigcap_{i=1}^{k}\big(Z_i > z\big)\right)\\ &= \prod_{i=1}^{k}\Pr\big(Z_i > z\big)\\ &= (1 - z^\alpha)^k.\end{aligned}$$

Since $1 - a \approx e^{-a}$ (when $0 < a < 1$ and a is sufficiently small), it follows that

$$\begin{aligned}\mathbf{E}[Z_{min}] &= \int_0^1 \Pr\big(Z_{min} > z\big)dz\\ &= \int_0^1 \left(e^{-z^\alpha}\right)^k dz\\ &= \int_0^1 e^{-\left(k^{\frac{1}{\alpha}}z\right)^\alpha} dz\\ &= \frac{1}{\omega}\int_0^s e^{-z^\alpha} dz,\end{aligned}$$

where $\omega = k^{\frac{1}{\alpha}}$. Since $\int_0^\omega e^{-z^\alpha} dz \leq 1, \mathbf{E}[Z_{min}] \leq \frac{1}{\omega}$, and the proof follows. □

4.2 Peer Lifetime Model

Recent measurement studies [33], [34] of real P2P systems (i.e., Gnutella [5] and Napster [35]) provide evidence that peers have lifetimes approximating the exponential distribution reasonably well [12]. In the following analysis, we assume that the system follows the $M/M/\infty$ queuing model in which the arrival rate of peers is according to a Poisson distribution with the parameter λ. The lifetimes for peers are independent and exponentially distributed with the parameter μ. The number of peers in the system at time t is denoted by $M(t)$.

Lemmas 2 and 3 presented as follows conclude the number of peers, denoted by n, in the system that operates for a sufficiently long time, given the $M/M/\infty$ model with the peer arrival rate λ and the mean lifetime $\frac{1}{\mu}$ of a peer.

Lemma 2. *The number of peers in the system at time t is $\frac{M(t)}{2} \leq M(t) \leq \frac{3M(t)}{2}$ with the probability $\geq 1 - \mathcal{O}(n^{-\Omega(1)})$.*

Proof. Since

$$\Pr(M(t) = j) = e^{-\lambda tp}\frac{(\lambda tp)^j}{j!},$$

where $p = \int_0^t \frac{e^{-u(t-x)}}{t}dx$ due to the uniformity of the arrival time in $[0,t], \Pr\big(M(t) = j\big)$ is thus a Poisson distribution with the parameter λtp (see [27]). That is, $\mathbf{E}[M(t)] = \lambda tp$. We let $\mathbf{E}[M(t)] = n$. By Chernoff bound [27], we have

$$\begin{aligned}\Pr\left(M(t) \geq \frac{3}{2}n\right) &\leq \frac{e^{-n}(en)^{\frac{3}{2}n}}{(\frac{3}{2}n)^{\frac{3}{2}n}}\\ &= \left(\sqrt{\frac{8e}{27}}\right)^n,\end{aligned}$$

and

$$\begin{aligned}\Pr\left(M(t) \leq \frac{1}{2}n\right) &\leq \frac{e^{-n}(en)^{\frac{1}{2}n}}{(\frac{1}{2}n)^{\frac{1}{2}n}}\\ &= \left(\sqrt{\frac{e}{2}}\right)^{-n}.\end{aligned}$$

Therefore, when $n > 3, \Pr\big(|M(t) - n| \geq \frac{1}{2}n\big) < \left(\frac{1}{2}\right)^n$. The proof thus follows. □

Lemma 3. *Let $\frac{\lambda}{\mu} = n$. If $t \geq \frac{n}{\mu}$, then $1 - e^{-n} \leq \frac{\mathbf{E}[M(t)]}{n} < 1$.*

Proof. Since

$$\begin{aligned}\mathbf{E}[M(t)] &= \lambda tp\\ &= \lambda t\int_0^t \frac{e^{-\mu(t-x)}}{t}dx\\ &= \lambda t\frac{1}{\mu t}(1 - e^{-\mu t}),\end{aligned}$$

if $t \geq \frac{n}{\mu}$, then $\frac{\lambda}{\mu}(1 - e^{-n}) \leq \mathbf{E}[M(t)] < \frac{\lambda}{\mu}$. The proof follows. □

In the following discussion, we assume that the system has been operated for a sufficiently long time (that is, the system has been operated at the least time of dn, where d is a constant) and the expected number of peers in the system is denoted by n.

4.3 Analytical Results

In this section, we provide the analytical results for our proposal. Our major performance results present that if we select the values of γ, β, s_γ, and s_β carefully, then 1) our unstructured network has the constant communication latency between any two nodes in expectation independent of the system size (see Theorems 1 and 2) and 2) the expected broadcasting scope of any node is $\Omega(2^{TTL})$ (Corollary 4). 3) We will also show that a participating node in our design takes polylogarithmic overhead to maintain the network (Corollary 5).

As mentioned in Section 4.1, we assume $\mathcal{L} = 1$ to simplify the discussion. We first assess the expected communication latency between any two nodes in the network. The expected communication latency between any two nodes using only overlay links of nodes in red neighboring sets is discussed in the next section. We then estimate the expected communication latency using only links of nodes in blue neighboring sets (Section 4.3.2). Both analytical results lead to conclude that our network maintains the constant communication latency between any two nodes in expectation.

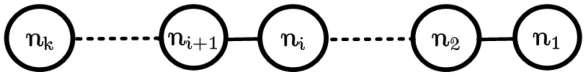


Fig. 4. A possible route path from node n_1 toward n_k having the smallest ID.

In the following discussion, let $G = (V, E)$ be the network that our algorithm constructs.

4.3.1 Network with Only Red Neighboring Sets

Lemma 4. *The expected value of $|u^{(r)} \rightsquigarrow z^{(r)}|$ is $\mathbf{E}[|u^{(r)} \rightsquigarrow z^{(r)}|] = \ln n + \Theta(1)$ for all $u \in V$, where $n = |V|$ and $|u^{(r)} \rightsquigarrow z^{(r)}|$ is the path length of $u^{(r)} \rightsquigarrow z^{(r)}$.*

Proof. Given an r value ($1 \leq r \leq \gamma$), Fig. 4 shows a possible route path P from any node $u = n_1$ (with $n_1^{(r)}.id$) toward node n_k (with $n_k^{(r)}.id$) having the smallest ID. Since nodes pick their IDs uniformly at random, the probability of $n_2^{(r)}.id < n_1^{(r)}.id$ is thus $\frac{1}{2}$. Similarly, the probability of $n_3^{(r)}.id < n_2^{(r)}.id$ is $\frac{1}{3}$. Consequently, the probability of $n_{i+1}^{(r)}.id < n_i^{(r)}.id$ is $\frac{1}{i+1}$.

Define the random variable X_i (where $2 \leq i \leq n-1$) as follows:

$$X_i = \begin{cases} 1, & \text{if } n_{i+1}^{(r)}.id < n_i^{(r)}.id, \\ 0, & \text{otherwise.} \end{cases}$$

Clearly, $|u^{(r)} \rightsquigarrow z^{(r)}| = \sum_{i=2}^{n-1} X_i$. Then, we have

$$\begin{aligned} \mathbf{E}[|u^{(r)} \rightsquigarrow z^{(r)}|] &= \sum_{i=2}^{n-1} X_i \\ &= \sum_{i=2}^{n-1} \frac{1}{i} \\ &= \ln n + \Theta(1) \end{aligned}$$

and the proof thus follows. □

Corollary 1. *$|u^{(r)} \rightsquigarrow z^{(r)}| = \mathcal{O}(\ln n)$ for all $u \in V$ w.h.p.*[6]

Proof. By Chernoff bound [27], we have

$$\begin{aligned} \Pr(|u^{(r)} \rightsquigarrow z^{(r)}| \geq 6\mathbf{E}[|u^{(r)} \rightsquigarrow z^{(r)}|]) &\leq 2^{-(\ln n + \Theta(1))} \\ &= \frac{1}{n} + \Theta(1). \end{aligned}$$

□

Lemma 5. *1) Discovering a node for $\mathcal{S}^{(red)}$ needs to sample no more than $6^\gamma \ln^\gamma n$ nodes in expectation and 2) if $s_\gamma \geq 6^\gamma \ln^{\gamma+1} n$, then $|\mathcal{S}^{(red)}| \geq 1$ w.h.p.*

Proof. Consider any node $u \in V$. u picks an s_γ-node set $\mathcal{R}$. Each node $v \in \mathcal{R}$ discovers a routing path toward the node having the smallest (or the largest) ID among IDs of nodes in V. Given an r value, let P be the set of the paths originating at any $v \in \mathcal{R}$. For each $p \in P$, let node $w \in \mathcal{S}^{(red)}$ of u be on the path p that first satisfies $w^{(r)}.id < u^{(r)}.id$ and $\deg_w^{(red)} < \max_w^{(red)}$.

By Corollary 1, we have the path length of any $p \in P$ equal to $\mathcal{O}(\ln n)$ w.h.p. The probability that there is at least one node $w \in p$ with $w^{(r)}.id < u^{(r)}.id$ and $\deg_w^{(red)} < \max_w^{(red)}$ is no less than $\mathcal{O}(\frac{1}{\ln^\gamma n})$. This is because the probability that from the node (e.g., n_k in Fig. 4) with the smallest ID on p, the most distant node w (e.g., n_1 in Fig. 4) on p having both $w^{(r)}.id < u^{(r)}.id$ and $|B_w^{(red)}| = \gamma < \max_w^{(red)}$ is $\geq \mathcal{O}(\frac{1}{\ln^\gamma n})$. The probability that such a node w can be selected into $\mathcal{R}$ is thus at least $\mathcal{O}(\frac{1}{\ln^\gamma n})$.

Let X_i be a random variable and is defined as follows:

$$X_i = \begin{cases} 1, & \text{if } \mathcal{S}^{(red)} = \mathcal{S}^{(red)} \cup \{w\}, \\ 0, & \text{otherwise.} \end{cases}$$

Clearly, $\Pr(X_i = 1) \geq \mathcal{O}(\frac{1}{\ln^\gamma n}) = \frac{1}{c^\gamma \ln^\gamma n}$, where $c = 6$ (see the proof in Corollary 1). Let $X = \sum_{i=1}^{s_\gamma} X_i$ be the number of paths such that in each path, there exists a node w to be included into $\mathcal{R}$. Since each joining node picks an ID uniformly at random for discovering $w \in v^{(r)} \rightsquigarrow z^{(r)}$ or $w \in v^{(r)} \overset{+}{\rightsquigarrow} z^{(r)}$ ($\forall v \in \mathcal{R}$), X is thus a geometric random variable with parameter $\geq \frac{1}{6^\gamma \ln^\gamma n}$ and, thus, we have $\mathbf{E}[X] \leq 6^\gamma \ln^\gamma n$. Moreover, since $1 + a \leq e^a$, we have

$$\begin{aligned} \Pr(X \geq 1) &= 1 - \Pr(X = 0) \\ &= 1 - \Pr\left(\sum_{i=1}^{s_\gamma} X_i = 0\right) \\ &\geq 1 - \prod_{i=1}^{s_\gamma}\left(1 - \frac{1}{6^\gamma \ln^\gamma n}\right) \\ &\geq 1 - \left(e^{\frac{-1}{6^\gamma \ln^\gamma n}}\right)^{s_\gamma}. \end{aligned}$$

Therefore, if $s_\gamma \geq 6^\gamma \ln^{\gamma+1} n$, then $\Pr(X \geq 1) \geq 1 - \frac{1}{n}$, yielding the proof. □

We thus have the following result.

Corollary 2. *If $\gamma \geq 1$ and $s_\gamma \geq \gamma 6^\gamma \ln^{\gamma+1} n$, then $|\mathcal{S}^{(red)}| \geq \gamma$ w.h.p.*

Remark 1. Both Corollaries 1 and 2 state that if $\mathcal{S}^{(red)}$ of u contains at least γ ($\gamma \geq 1$) physically closest peers, u needs to sample at least $\mathcal{O}(\ln^{\gamma+1} n)$ nodes from the network w.h.p. and each of the samples individually takes the overhead of $\mathcal{O}(\ln n)$ hops w.h.p. to discover a node satisfying the constraints of ID ordering and node degree bound.

We are now ready to prove the expected communication latency between any two nodes in the network. In Theorem 1, the message communication between any two nodes only depends on the overlay links of nodes in red neighboring sets of participating nodes.

Theorem 1. *Let $|B_u^{(red)}| = \gamma$ for all $u \in V$. If $s_\gamma \geq \gamma 2^{\alpha+\gamma} 3^\gamma \ln^{\alpha+\gamma+1} n$, then the expected communication latency ℓ between any two nodes through only overlay links of nodes in $B_u^{(red)}$ ($\forall u \in V$) is $\Theta(1)$.*

Proof. By Corollary 1, $\mathbf{E}(|u^{(r)} \rightsquigarrow z^{(r)}|) = \mathcal{O}(\ln n)$ w.h.p., given any r value. Since any route $u^{(r)} \rightsquigarrow z^{(r)}$ is toward the node with the smallest ID (that is the ID of the bootstrap), a route path thus has the length of $2 \times \mathcal{O}(\ln n)$ w.h.p. By Lemma 1, if each node in $\mathcal{S}^{(red)}$ of any node u is picked independently

[6] The abbreviated term "w.h.p." in this paper stands for "with high probability," which is the probability no less than $1 - \mathcal{O}(n^{-\Omega(1)})$.

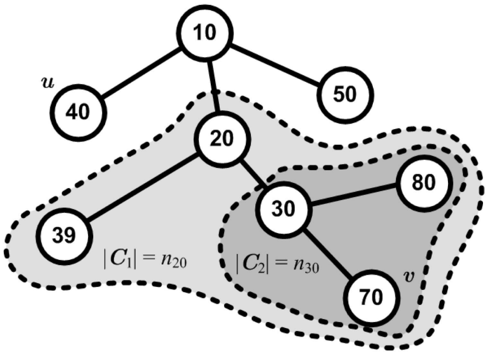


Fig. 5. A network consists of the only overlay links that connect nodes' first red neighbors, where $n_{20} = 5$ and $n_{30} = 3$.

and uniformly at random from $k \geq (2 \ln n)^{\alpha}$ samples, then the expected communication latency ℓ of sending a message between any two nodes through only overlay links of nodes in $B_u^{(red)}$ ($\forall u \in V$) is $\ell = 2 \ln n \times k^{\frac{-1}{\alpha}} \leq 1$. Consequently, by Corollary 2, for $|B_u^{(red)}| = \gamma$ and $\ell \leq 1$, we have to sample $k\gamma 6^{\gamma} \ln^{\gamma+1} n$ nodes, concluding the proof.□

4.3.2 Network with Only Blue Neighboring Sets

For building blue neighboring sets, any node $w \in V$ needs to estimate n_w so that a remote node u can be based on $\Pr(\mathcal{H}_w, n)$ to determine whether its $B_u^{(blue)}$ shall include w. Notably, n_w is the number of nodes having $\mathcal{H} \supseteq \mathcal{H}_w$. Fig. 5 depicts the notion of n_w. Notably, as mentioned in Section 3.1, $\mathcal{H}_w = w^{(1)} \rightsquigarrow z^{(1)}$, where z is the bootstrap and has the smallest ID.

In this section, we discuss 1) how to estimate n_w accurately and 2) how to determine $\Pr(\mathcal{H}_w, n)$ such that the communication delay between any two nodes through overlay links of nodes in blue neighboring sets is a constant in expectation.

We estimate n_w based on the following result.

Lemma 6. *Let $X_1, X_2, \ldots, X_m$ be independent, identical Poisson random variables. $\mathbf{E}[X_i] = \mathcal{E}$ for all $i = 1, 2, \ldots, m$. For any δ $(0 < \delta < 1)$, if $m \geq \frac{3 \ln(\frac{2}{\delta})}{\epsilon^2 \mathcal{E}}$, then $\Pr\left(|\frac{1}{m}\sum_{i=1}^{m} X_i - \mathcal{E}| \geq \epsilon\mathcal{E}\right) \leq \delta$.*

Proof. Let $X_1, X_2, \ldots, X_m$ be independent, identical Poisson trails such that $\Pr(X_i) = \mathcal{E}$ (where $i = 1, 2, \ldots, m$). Let $X = \sum_{i=1}^{m} X_i$. We then have the Chernoff bound [27] as follows:

$$\begin{cases} \Pr(X \geq (1+\epsilon)\mathbf{E}[X]) \leq e^{\frac{-\mathbf{E}[X]\epsilon^2}{3}}, & \text{for } 0 < \epsilon \leq 1, \\ \Pr(X \leq (1-\epsilon)\mathbf{E}[X]) \leq e^{\frac{-\mathbf{E}[X]\epsilon^2}{2}}, & \text{for } 0 < \epsilon < 1. \end{cases}$$

We thus have

$$\begin{cases} \Pr\left(\frac{X}{m} \geq \frac{(1+\epsilon)\mathbf{E}[X]}{m}\right) \leq e^{\frac{-\mathbf{E}[X]\epsilon^2}{3}}, & \text{for } 0 < \epsilon < 1, \\ \Pr\left(\frac{X}{m} \leq \frac{(1-\epsilon)\mathbf{E}[X]}{m}\right) \leq e^{\frac{-\mathbf{E}[X]\epsilon^2}{3}}, & \text{for } 0 < \epsilon < 1. \end{cases}$$

Since $\mathbf{E}[X_i] = \mathcal{E}$ and $\mathbf{E}[\sum_{i=1}^{m} X_i] = m\mathcal{E}$ by the linearity of expectations,

$$\Pr\left(\left|\frac{X}{m} - \mathcal{E}\right| \geq \epsilon\mathcal{E}\right) \leq 2e^{\frac{-m\mathcal{E}\epsilon^2}{3}}.$$

Therefore, if $m \geq \frac{3 \ln(\frac{2}{\delta})}{\epsilon^2 \mathcal{E}}$, the proof follows. □

Remark 2. For estimating n_w, Lemma 6 suggests that we sample m nodes independently and uniformly at random. If

$$m \geq \frac{3 \ln(\frac{2}{\delta})}{\epsilon^2 \left(\frac{n_w}{n}\right)}, \tag{5}$$

then the estimation of n_w is bounded within

$$[(1-\epsilon)n_w, (1+\epsilon)n_w]$$

with the probability $\geq 1 - \delta$, where $0 < \delta < 1$.

Lemma 7. *Consider the subnetwork $G' = (V, E') \subseteq G = (V, E)$, where $E' = \bigcup_{u \in V} B_u^{(blue)}$. If $\Pr(\mathcal{H}_w, n) \geq \frac{n_w^{-1}}{\ln n}$ and $|B_u^{(blue)}| \geq 1$, $\forall u \in V$, then a message takes $\ln^2 n$ hopcount in expectation to reach any node in V.*

Proof. We will show that a message takes $\ln n$ "steps" in expectation to reach its destination, and that in each step, the expected number of nodes visited is $\ln n$.

Assume that u is a node originating a message to route toward a destination v. We define *upstream nodes* of v as any node w on the path from v toward the node with the smallest ID. Precisely, the set of upstream nodes of v is denoted by $\{w_1, w_2, \ldots, w_k\}$, where $w_1^{(1)}.id < w_2^{(1)}.id < \cdots < w_k^{(1)}.id$, w_1 is the node directly connected to the bootstrap and w_k is the immediate upstream node of v. Given an upstream node $w_i (1 \leq i \leq k)$ of v, let $\mathcal{C}_i$ denote the set of nodes on the paths toward the bootstrap through w_i. Clearly, $v \in \mathcal{C}_i$ for all $i = 1, 2, \ldots, k$. For example, in Fig. 5, v (i.e., node 70) has upstream nodes $\{w_1 = 20, w_2 = 30\}$, $\mathcal{C}_1 = \{20, 30, 39, 70, 80\}$, $\mathcal{C}_2 = \{30, 70, 80\}$, $n_{20} = 5$, and $n_{30} = 3$.

In our proposal, if u includes a node w into its $B_u^{(blue)}$ with the probability of $\Pr(\mathcal{H}_w, n) \geq \frac{n_w^{-1}}{\ln n}$, then the probability that u picks any node $x_1 \in \mathcal{C}_1$ into its $B_u^{(blue)}$ with the probability of no less than $|\mathcal{C}_1| \times \frac{n_{w_1}^{-1}}{\ln n} = \frac{1}{\ln n}$, where $|\mathcal{C}_1| = n_{w_1}$. u forwards the message to x_1 and x_1 then forwards the message to its neighbor $x_2 \in B_{x_1}^{(blue)}$. Similarly, the probability that x_1 has x_2 in $\mathcal{C}_1$ is $\geq \frac{1}{\ln n}$. Consequently, the expected number of nodes required to successfully forward the message to a node in $\mathcal{C}_1$ is no more than $\ln n$.

We then perform this iteratively to forward the message to nodes in $\mathcal{C}_2, \mathcal{C}_3, \ldots, \mathcal{C}_k$. Since k is of $\ln n$ in expectation (Lemma 4), the expected hopcount for routing a message from u to v is thus $\ln^2 n$. □

Both Remark 2 and Lemma 7 conclude the following.

Corollary 3. *Let $m \geq \frac{3 \ln(\frac{2}{\delta})}{\epsilon^2 \left(\frac{n_w}{n}\right)}$ and the estimation of n_w be $\tilde{n}_w$. If*

$$\Pr(\mathcal{H}_w, n) \geq \frac{1 - \epsilon}{\tilde{n}_w \ln n} \tag{6}$$

and $\beta = 1$, then a message takes $\ln^2 n$ hopcount in expectation to reach its destination.

Notably, 1) in our implementation, any node $w \in V$ resolves its $\mathcal{H}_w = w^{(1)} \rightsquigarrow z^{(1)}$. Given any $\mathcal{C}_i$ (see the

TABLE 2
The Statistics for $\frac{n_w}{n} = \frac{3}{\ln n}$ and $m = \ln^3 n$

n	$\frac{m}{n}$	$\frac{n_w}{n}$	ϵ	$1-\delta$
$10,000$	0.0781316	0.3257208	0.3295051	0.9998
$100,000$	0.0152600	0.2605766	0.2947183	0.99998
$1,000,000$	0.0026369	0.2171472	0.2690397	0.999998
$10,000,000$	0.0004187	0.1861262	0.2490824	0.9999998
$100,000,000$	0.0000625	0.1628604	0.2329953	0.99999998

definition in the proof of Lemma 7) with respect to any node v, we can thus determine whether w is in $\mathcal{C}_i$ by checking $w^{(1)} \rightsquigarrow z^{(1)}$. 2) Let the number of the samples that are in $\mathcal{C}_i$ be $|\tilde{\mathcal{C}_i}|$. We then compute the ratio of $\frac{|\tilde{\mathcal{C}_i}|}{m}$. Since we depend on previously proposed schemes (e.g., the scheme in [28]) to estimate n, n_w can thus be approximated by $\frac{|\tilde{\mathcal{C}_i}| \times n}{m}$. 3) For minimizing the overhead required of performing sampling, we let $\frac{n_w}{n} \geq \frac{3}{\ln n}$ and $m \geq \ln^3 n$ such that $\epsilon \leq (\ln n)^{\frac{-1}{2}}$ and $\delta = \frac{2}{n}$. This allows us to estimate $n_w \geq \frac{3n}{\ln n}$ precisely, though it may not be accurate for approximating $n_w < \frac{3n}{\ln n}$. Table 2 illustrates the statistical results for the metrics $\frac{m}{n}, \frac{n_w}{n}$, ϵ, and $1 - \frac{2}{n}$ based on $\frac{n_w}{n} = \frac{3}{\ln n}$ and $m = \ln^3 n$, given various ns.

Theorem 2. *Let $|B_u^{(blue)}| \geq \beta$ for all $u \in V$. If $s_\beta \geq \beta \ln^{2\alpha} n$, then the expected communication latency ℓ between any two nodes through only overlay links of nodes in $B_u^{(blue)}$ $(\forall u \in V)$ is $\Theta(1)$.*

Proof. By Corollary 3, the expected communication latency between any two nodes using only links of nodes in $B_v^{(blue)} (\forall v \in V)$ is $\ln^2 n$. By Lemma 1, if elements in $\mathcal{S}^{(blue)}$ maintained by any node u are picked independently and uniformly at random from $k \geq \ln^{2\alpha} n$ samples, then the expected communication latency ℓ of sending a message between any two nodes through only overlay links of nodes in $B_v^{(blue)}$ $(\forall v \in V)$ is $\ell = \ln^2 n \times k^{\frac{-1}{\alpha}} \leq 1$. Consequently, for $|B_u^{(blue)}| \geq \beta$ and $\ell \leq 1$, we have to sample $s_\beta \geq \beta \ln^{2\alpha} n$ nodes, concluding the proof. □

4.3.3 Connectivity

A random graph is connected if each node in the graph contributes $\mathcal{O}(\ln n)$ connections [15]. While our proposal generates randomized networks, we show that our network is connected w.h.p. even if each node in the network links to a constant number of nodes. More specifically, in our proposal, there exists at least one path connecting any two nodes in $G^{(red)} \subset G$ through the node with the smallest ID.

Theorem 3. *Let t be the current system time. Our overlay network is connected w.h.p. within the system time interval $[t, t + \frac{\ln \gamma - \ln(\ln 6 + \ln n + \ln \ln n)}{\mu}] \approx [t, t + \frac{\ln \gamma}{\mu}]$.*

Proof. Since nodes in red neighbor sets join on paths toward the node (i.e., the bootstrap) with the smallest ID, it suffices to show the connectivity of any path toward the bootstrap.

In our proposal, each node maintains γ nodes on paths toward the bootstrap. Consider the system time $t + \Delta t$. Since any of such paths has the path length of $6 \ln n$ w.h.p. and the expected lifetime of any participating peer is $\frac{1}{\mu}$ (see Section 4.2), the probability that there exists at least a path toward the bootstrap is

$$f = \left(1 - \left(1 - e^{-\mu \Delta t}\right)^\gamma\right)^{6 \ln n}$$
$$\approx \left(e^{-(1 - e^{-\mu \Delta t})^\gamma}\right)^{6 \ln n}$$
$$\approx \left(\left(e^{-Q}\right)^\gamma\right)^{6 \ln n},$$

where $Q = e^{-e^{-\mu \Delta t}}$.

It is easy to show that if $\Delta t \leq \frac{\ln \gamma - \ln(\ln 6 + \ln n + \ln \ln n)}{\mu}$, then

$$Q^\gamma \leq \frac{1}{6n \ln n}.$$

That is,

$$f = e^{-Q^\gamma 6 \ln n}$$
$$\geq e^{\frac{-1}{n}}$$
$$\approx 1 - \frac{1}{n},$$

yielding the proof. □

Remark 3. Since any node may leave or depart the system at any time, Theorem 3 indicates that a participating node u needs to periodically collect its $B_u^{(red)}$ every time interval of $\frac{\ln \gamma}{\mu}$ such that u can maintain up to γ alive red neighbors and this results in the system that is connected w.h.p.

4.3.4 Broadcasting Scope

Corollary 4. *The expected broadcasting scope of any node is $\Omega(2^{TTL})$.*

Proof. By Lemma 4, since the expected hopcount of routing a message (with only overlay links of nodes in red neighboring sets) between any two nodes is $\Theta(\ln n)$, then the expected broadcasting scope of any node is thus $\geq 2^{TTL}$. □

4.3.5 Overhead

The overheads of building our network include the "implicit" overhead due to estimating n_u for any $u \in V$ and the "explicit" overhead due to constructing $B_u^{(red)}$ and $B_u^{(blue)}$. As we discussed in Section 4.3.2, approximating n_u takes $\ln^3 n$ messages in our implementation. By Theorem 1, creating $|B_u^{(red)}| = \gamma$ requires $\gamma 2^{\alpha+\gamma} 3^\gamma \ln^{\alpha+\gamma+1} n$ samples, while $\beta \ln^{2\alpha} n$ samples are required for $|B_u^{(blue)}| = \beta$ due to Theorem 2. Assuming that sampling a node for $B_u^{(red)}$ or $B_u^{(blue)}$ takes one message, we conclude the following result.

Corollary 5. *If $|B_u^{(red)}| = \gamma$ and $|B_u^{(blue)}| = \beta$, then the averaged overhead of joining a node takes $\max\{\mathcal{O}(\ln^{\alpha+\gamma+2} n), \mathcal{O}(\ln^{2\alpha} n)\}$ without considering the implicit overhead due to estimating n_w for all $w \in V$.*

Proof. It suffices to show the explicit overhead due to building $B_u^{(red)}$ for joining a node u. Since u needs to sample $|\mathcal{R}| = s_\gamma = \gamma 2^{\alpha+\gamma} 3^\gamma \ln^{\alpha+\gamma+1} n$ nodes such that the expected communication latency between any two nodes through only overlay links of nodes in $B_w^{(red)} (\forall w \in V)$ is a constant (see Theorem 1). For each $v \in \mathcal{R}$, either $v^{(r)} \overset{-}{\rightsquigarrow} z^{(r)}$ or $v^{(r)} \overset{+}{\rightsquigarrow} z^{(r)}$ is performed, taking $\leq 6 \ln n$ messages w.h.p. Consequently, creating $B_u^{(red)}$ takes $\leq \gamma 2^{\alpha+\gamma+1} 3^{\gamma+1}$ $\ln^{\alpha+\gamma+2} n$ messages, concluding the proof. □

We finally conclude the parameters γ and β as follows for our design.

Remark 4. Corollary 5 indicates that our design shall minimize γ (and thus $\max_u^{(red)}$ for any $u \in V$) and utilize the remaining connections available to any node by maximizing β.

5 Simulations

We have developed an event-driven simulator to study the performance of our constructed network. The performance metrics we are interested in include the averaged communication latency between any two nodes, as defined in (1), and the averaged broadcasting scope of participating peers (see (2)), given the number of nodes participating in the system, the mean lifetime of the joining peers, and the maximal degree Δ of a node.

In our simulations, the number of nodes participating in the system is up to $n = 100{,}000$. Each participating peer has a lifetime with a mean of 150 minutes [34], [36]. The lifetime follows exponential distribution. We have also studied the scenario, where participating peers have smaller lifetime values (e.g., 30 minutes [36]). We observe similar simulation results and only report the results for the mean lifetime of 150 minutes in this paper.

We simulate the end-to-end delay between any two nodes using the performance data collected from PlanetLab [32], a world-wide scale experimental testbed. Since our simulator simulates up to 100,000 peers and the publicly available PlanetLab topology[7] only includes 500 nodes, we thus assign each simulated peer to one, picked uniformly at random, among the 500 locations.

As discussed in Section 4.1, the PlanetLab topology follows the power-law latency expansion distribution [30]. In our performance study, we approximate the probability distribution for communication latency, denoted by the random variable $\mathcal{Z}$, between any two PlanetLab nodes as $\Pr(\mathcal{Z} \leq z) = \left(\frac{z}{\mathcal{L}}\right)^{0.4}$, where $\mathcal{L}$ is the maximum delay between any two nodes in the PlanetLab topology.

Given a system size n, we perform extensive simulations with 100 simulation runs. Each run takes 150 minutes[8] and generates a snapshot of the simulated network topology. We measure the performance metrics (i.e., the averaged communication latency and the averaged broadcasting scope) for each snapshot topology. The performance results discussed in this section are the averages of the measured metrics.

Notably, in the simulations, our proposal is compared to two representative solutions, namely *THANCS* [8] and *mOverlay* [14]. We briefly sketch our implementations for THANCS and mOverlay as follows.

5.1 Local Search Method: THANCS

Consider a network $G = (V, E)$ created by the THANCS algorithm [8]. Let N_v be the set of neighbors of any node v in the overlay (i.e., $N_v = \{u|(u, v) \in E, \forall u \in V\}$) and N_v^2 be the node having the hopcount distance of two from v (i.e., $N_v^2 = \{w|(w, u) \in E, \forall w \in V, \forall u \in N_v\} - N_v - \{v\}$). Denote the end-to-end network latency between v and u by $d_{v,u}$. Then, any node v in THANCS performs the following:

- v removes a link $(v, w) \in E$ for all $w \in N_v^2$ if there exists $u \in N_v$ such that $(u, w) \in E$, $d_{v,w} > d_{v,u}$, and $d_{v,w} > d_{u,w}$.
- v removes a link $(v, u) \in E$ for all $u \in N_v$ if there exists $w \in N_v^2$ such that $(u, w) \in E$, $d_{v,u} > d_{v,w}$, and $d_{v,u} > d_{u,w}$.
- v adds a link (v, w) to E (i.e., $E = E \cup \{(v, w)\}$) if there exist $u \in N_v$ and $w \in N_v^2$ such that $(v, w) \notin E$, $(u, w) \in E$, and $d_{v,u} > d_{v,w}$ (or $d_{u,w} > d_{v,w}$).

Clearly, THANCS is a *local search method* [37], which intends to find a local optimum solution. Any node v in THANCS explicitly creates links with low delays and removes those with high latencies by exploiting its local knowledge (that is, nodes within vs 2-hop scope).

Notably, 1) in our implementation for THANCS, an extra link can be included into G that is subject to the degree constraints of both end nodes of the connection. 2) THANCS is designed to improve a given overlay $G = (V, E)$. Since prior studies (e.g., [38]) show that Gnutella networks can be approximated by random graphs, we generate Gs, which are random graphs, by implementing the centralized algorithm presented in [39] subject to the degree bounds of nodes in V. 3) To the best our knowledge, the state-of-the-art solution, i.e., THANCS [8], presented by Liu considerably outperforms those in [9], [10].

5.2 Clustering Approach: mOverlay

Let $G = (V, E)$ be the overlay network the mOverlay algorithm [14] constructs. Basically, G is formed by clusters. Each cluster in G has a leader node, and a node in a cluster connects to a randomly picked node in the same cluster. Let C_i (where $i = 1, 2, \cdots, k$) represent the set of k clusters in the system and L_i denote the leader node of cluster C_i. The network delay between two clusters C_i and C_j is denoted by $d_{i,j}$, which measures the distance between the representative leaders L_i and L_j. We define N_{C_i} as the neighboring clusters of cluster C_i. That is, $N_{C_i} = \{C_j|(L_j, L_i) \in E, \forall L_j \in V\}$.

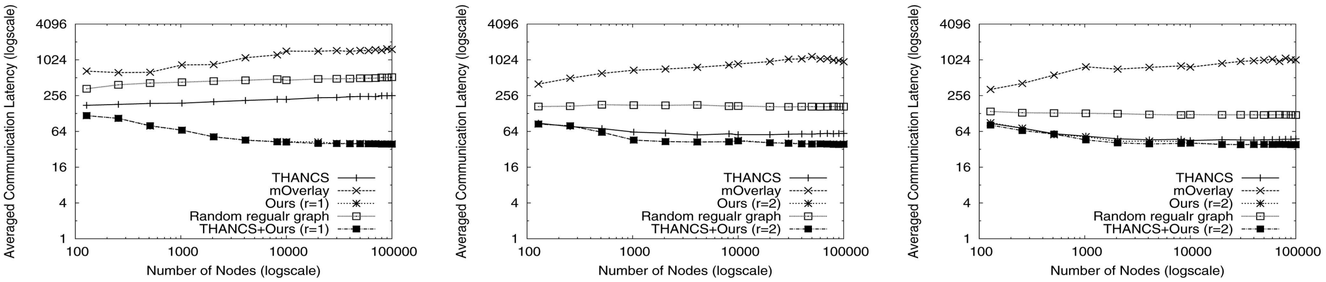
For participating in G, a newly joining node v performs as follows:

1. v first finds an entry point, say node w, to help its joining. Typically, w is a leader node of cluster C_w.
2. v becomes a member of C_w if $d_{u,v} = d_{u,w}$ for all $u \in N_{C_w}$, and the algorithm terminates.
3. Otherwise, v picks a group C_u such that $u = \arg\min_u\{d(u, v)|\forall u \in N_{C_w}\}$ and $d_{u,v} < d_{\min}$, where $d_{\min}$ is the smallest latency measured so far between v and every visited cluster, and $d_{\min} = \infty$, initially.

Notably, Step 3 is performed repeatedly until v identifies a group C_u satisfying the grouping criterion as specified in Step 2 and then joins C_u. Otherwise, v creates a new group comprising itself only and becomes the leader of the group interconnecting with C_u. Clearly, $d_{\min}$ reduces when Step 3 is performed every time. This allows v to approach a physically close cluster. However, mOverlay suggests to perform Step 3 up to $\mathcal{K}$ times, where $\mathcal{K} = \log_m k + 3$, m is the

[7] The PlanetLab topology specifies end-to-end routing delays of all participating nodes.

[8] The topology matching algorithm, THANCS [8], requires a number of rounds to improve and stabilize the performance of unstructured P2P networks. A simulation run taking 150 minutes suffices for THANCS to stabilize.



(a) (b) (c)

Fig. 6. The averaged communication latency. (a) $\gamma = 1,\ \Delta = 6$, and $\max_u^{(red)} = 5$, (b) $\gamma = 2,\ \Delta = 15$, and $\max_u^{(red)} = 10$, (c) $\gamma = 2,\ \Delta = 30$, and $\max_u^{(red)} = 15$.

default number of neighboring groups of any cluster, and k is the maximal number of clusters in the system. This is due to the minimization of overheads for locating a geographically close cluster.

5.3 Simulation Results

5.3.1 The Averaged Communication Latency

Fig. 6 depicts the simulation results for the averaged communication latency. Fig. 6a shows the averaged communication latency for networks in which each participating peer u has the maximal degree up to $\Delta = 6$ and $\max_u^{(red)} = 5$. In this experiment, we report the simulation results for our proposal with $\gamma = 1$. The simulation results in Fig. 6a present that our proposal outperforms THANCS and mOverlay remarkably.

Figs. 6b and 6c illustrate the experimental results, respectively, for $\Delta = 15, 30$ and $\max_u^{(red)} = 10, 15$. The variable γ is 2 in these experiments. Note that in Fig. 6, the results for random regular graphs without exploiting the physical network locality are also shown. As mentioned, random graphs approximate the real Gnutella topologies, and in our simulations, THANCS that we implement is based on the random regular graphs. Fig. 6 reveals that random graphs clearly outperform mOverlay mainly due to its small diameter.[9] Second, we can see that the performance of THANCS is sensitive and heavily depends on the base overlay topology. THANCS performs better in terms of the averaged communication delay if the overlay diameter is small (see Fig. 6c). However, given different Δ values, the performance of our proposal remains robust. This confirms our analytical results (i.e., Theorems 1 and 2) discussed in Section 4.

Interestingly, we also investigate whether THANCS can improve our proposed network. More precisely, we implement THANCS in our network to see if THANCS could further minimize the average communication latency between any two nodes in our network. Fig. 6 depicts the simulation results for such an optimization (see `THANCS+Ours`). The results conclude that our constructed network performs very well since THANCS obviously cannot improve our network.

As mentioned in Section 5.1, popular P2P network topologies (i.e., Gnutella) can be approximated by random graphs [38]. In Fig. 6, we experiment with random regular graphs in which any participating node contributes up to $\Delta =$ 6, 15, 30 connections to the networks. To further investigate the performance of THANCS and our proposal, we have also simulated the real P2P network topologies using the traces publicly available in [40]. In the traces, the maximal-connected network component has about 76,000 peers and these peers contribute various numbers of connections to the network. While the averaged number of connections contributed by a peer is 30 in the traces, the maximal number of connections contributed by a peer can be up to ≈ 150. Hence, for studying THANCS and our proposal, we let each simulated peer have an individual Δ value according to the traces and we simulate the system in size of 76,000 peers. Our simulation results show that the measured average communication latency between nodes is similar to that in Fig. 6c for $n = 70{,}000$ and $n = 80{,}000$. That is, in a modest size P2P network where nodes have relatively larger degrees, THANCS performs well due to the small diameter of the network. However, as discussed for Fig. 6a, we extrapolate that when the system size becomes large and when nodes join with a constant number of connections regardless of the system size, our proposal will considerably outperform THANCS.

5.3.2 The Averaged Broadcasting Scope

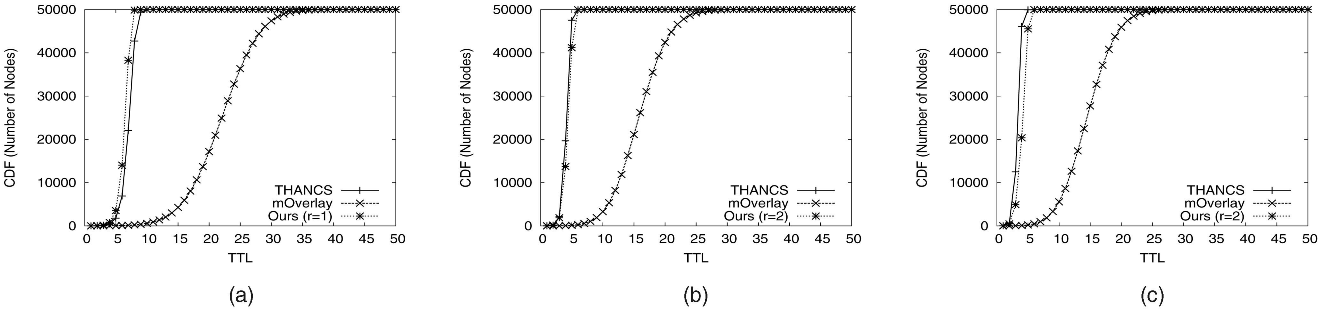
Fig. 7 presents the averaged broadcasting scopes for THANCS, mOverlay, and our proposal for $n = 50{,}000$. In Fig. 7, we vary the TTL value from 1 to 50. The simulation results indicate that THANCS and our proposal are comparable. A participating peer has the exponential broadcasting scope in THANCS. THANCS performs well since the base topologies are random regular graphs. Our design also allows a joining peer to have the exponential broadcasting scope due to the logarithmic diameter of our network (see Corollaries 1 and 4). On the contrary, mOverlay performs poorly in terms of the averaged broadcasting scope due to the high diameter of the network it constructs.

5.3.3 Overheads

Theorem 1 states that with high probability, a node samples $\gamma 2^{\alpha+\gamma} 3^{\gamma} \ln^{\alpha+\gamma+1} n$ of nodes to set up γ red neighbors. We note that this sampling quantity is bounded from above due to Lemma 5. We are interested to know whether we can rely on less samples to provide comparable performance results.

Given $\Delta =$ 6, 15, 30, Fig. 8 depicts the simulation results for the averaged communication latency using $s_0 = \gamma 2^{\alpha+\gamma} 3^{\gamma} \ln^{\alpha+\gamma+1} n$, $s_1 = \gamma \ln^{\alpha+\gamma+1} n$, and $s_2 = \gamma \ln^{\alpha+\gamma} n$

[9] By diameter, we mean the maximal shortest path, in terms of hopcount, between nodes in the system.



(a)

(b)

(c)

Fig. 7. The averaged broadcasting scope. (a) $\gamma = 1$, $\Delta = 6$, and $\max_u^{(red)} = 5$, (b) $\gamma = 2$, $\Delta = 15$, and $\max_u^{(red)} = 10$, (c) $\gamma = 2$, $\Delta = 30$, and $\max_u^{(red)} = 15$.

samples. The results conclude that it suffices to sample $s_2 = \gamma \ln^{\alpha+\gamma} n$ nodes by each participating node. We thus conclude that in our design, it suffices for a node to participate in the network by sampling $\gamma \ln^{\gamma+\Theta(1)} n$ nodes if α ($\alpha = 0.4$ in our study) is smaller than γ.

Regarding the averaged broadcasting scope, our simulation results with various numbers of samples are similar to those shown in Fig. 7 and we thus omit the results in this paper.

6 Summary and Future Work

In this paper, we have presented a locality-aware, flooding-based P2P network. In our design, a node joins the network by maintaining a constant number of neighbors. We show that our network has the provable performance metrics. That is, the expected communication latency between any two nodes in our network is a constant. Additionally, any joining node has the exponential broadcasting scope in expectation. Furthermore, nodes take polylogarithmic overheads to exploit the network locality and maintain the overlay network. Through extensive simulations, we have shown that our design significantly outperforms the recent proposals given by Liu [8] and Zhang et al. [14].

In this study, we place emphasis on the rigorous design and optimization for matching an overlay topology with its underlying network. To the best of our knowledge, our work is a first study presenting a locality-aware, unstructured P2P network with provable performance metrics. Our algorithm has not yet taken the content knowledge of participating peers into consideration. It would be interesting to further improve the search effectiveness and efficiency of our flooding-based P2P system by exploiting the peers' interests and/or the content locality.

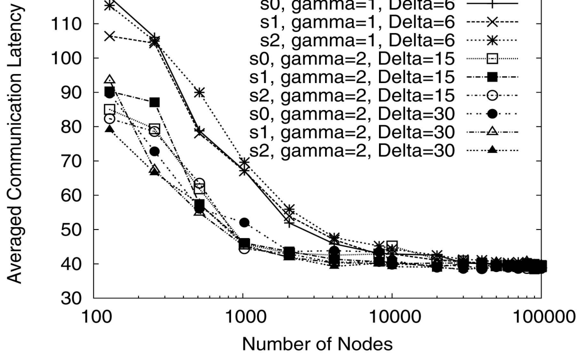


Fig. 8. The effect of the number of samples, where $s_0 = \gamma 2^{\alpha+\gamma} 3^{\gamma} \ln^{\alpha+\gamma+1} n$, $s_1 = \gamma \ln^{\alpha+\gamma+1} n$, and $s_2 = \gamma \ln^{\alpha+\gamma} n$.

Acknowledgments

The authors are grateful to the anonymous reviewers for providing them invaluable comments to improve the quality of their study. This work was partially supported by the National Science Council, Taiwan, under Grant 97-2221-E-006-132 and Grant 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,* 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,* 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 (LNCS),* vol. 2218, pp. 161-172, 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] Gnutella, http://rfc-gnutella.sourceforge.net/, 2009.

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

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

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

[11] C. Law and K.-Y. Siu, "Distributed Construction of Random Expander Networks," *Proc. IEEE INFOCOM,* pp. 2133-2143, Mar. 2003.

[12] 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.

[13] M.K. Reiter, A. Samar, and C. Wang, "Distributed Construction of a Fault-Tolerant Network from a Tree," *Proc. 24th IEEE Symp. Reliable Distributed Systems (SRDS '05),* pp. 155-165, Oct. 2005.

[14] X. Zhang, Q. Zhang, Z. Zhang, G. Song, and W. Zhu, "A Construction of Locality-Aware Overlay Network: mOverlay and Its Performance," *IEEE J. Selected Areas in Comm.,* vol. 18, no. 28, pp. 995-1002, Jan. 2004.

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

[16] B. Bollobás and W.F. de la Vega, "The Diameter of Random Regular Graphs," *Combinatorica,* vol. 2, no. 2, pp. 125-134, June 1982.

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

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

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

[20] 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.

[21] Y.K. Hui, C.S. Lui, and K.Y. Yau, "Small-World Overlay P2P Networks: Construction and Handling Dynamic Flash Crowd," *Computer Networks,* vol. 50, no. 15, pp. 2727-2746, Oct. 2006.

[22] S. Wang, D. Xuan, and W. Zhao, "Analyzing and Enhancing the Resilience of Structured Peer-to-Peer Systems," *J. Parallel Distributed Computing,* vol. 65, no. 2, pp. 207-219, Feb. 2005.

[23] L. Xiao, Y. Liu, and L.M. Ni, "Improving Unstructured Peer-to-Peer Systems by Adaptive Connection Establishment," *IEEE Trans. Computers,* vol. 54, no. 9, pp. 1091-1103, Sept. 2005.

[24] S. Jiang, L. Guo, X. Zhang, and H. Wang, "LightFlood: Minimizing Redundant Messages and Maximizing Scope of Peer-to-Peer Search," *IEEE Trans. Parallel Distributed Systems,* vol. 19, no. 5, pp. 601-614, May 2008.

[25] 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.

[26] L. Lovász, "Random Walks on Graphs: A Survey," *Combinatorics, Paul Erdős Is Eighty,* vol. 2, pp. 1-46, 1993.

[27] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis.* Cambridge Univ. Press, 2005.

[28] V. King and J. Saia, "Choosing a Random Peer," *Proc. 23rd ACM Symp. Principles Distributed Computing (PODC '03),* pp. 125-130, July 2004.

[29] M. Zhong, K. Shen, and J. Seiferas, "The Convergence-Guaranteed Random Walk and Its Applications in Peer-to-Peer Networks," *IEEE Trans. Computers,* vol. 57, no. 5, pp. 619-633, May 2008.

[30] 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.

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

[32] PlanetLab, http://www.planet-lab.org/, 2009.

[33] J.C. Chu, K.S. Labonte, and B.N. Levine, "Availability and Locality Measurements of Peer-to-Peer File Systems," *Proc. SPIE—ITCom Conf. Scalability and Traffic Control in IP Networks,* pp. 310-321, July 2002.

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

[35] Napster, http://www.napster.com/, 2009.

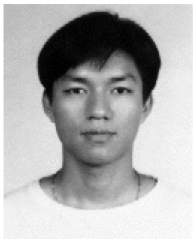
[36] K.P. Gummadi, R.J. Dunn, S. Saroiu, S.D. Gribble, H.M. Levy, and J. Zahorjan, "Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload," *Proc. 19th ACM Symp. Operating Systems Principles (SOSP '03),* pp. 314-329, Oct. 2003.

[37] E. Aarts and J.K. Lenstra, *Local Search in Combinatorial Optimization.* Princeton Univ. Press, 2003.

[38] 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.

[39] W. Aiello, F.R.K. Chung, and L. Lu, "A Random Graph Model for Massive Graphs," *Proc. 32nd ACM Ann. Symp. Theory of Computing (STOC '00),* pp. 171-180, May 2000.

[40] D. Stutzbach, "The Ion P2P Project: Empirical Characterizations of P2P Systems," http://mirage.cs.uoregon.edu/P2P/info.cgi, 2009.



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 the Department of Computer Science and Information Engineering, National Cheng-Kung University, Taiwan, since August 2005. From October 2000 to July 2005, he was a postdoctoral researcher in computer science, National Tsing-Hua University. 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 National Cheng-Kung University. His research interests include peer-to-peer computing, grid computing, and algorithm design and analysis.

Cheng-Chyun Huang received the BS and MS degrees in computer science and information engineering from the National Cheng-Kung University, Taiwan, in 2006 and 2008, respectively. His research interests include algorithm design and analysis for peer-to-peer networks.