

PAPER

Distributed Construction Protocols of Probabilistic Degree-Weighted Peer-to-Peer Overlays

Yu WU^{†a)}, *Nonmember*, Fukuhito OOSHITA[†], *Member*, Hirotsugu KAKUGAWA[†], *Nonmember*, and Toshimitsu MASUZAWA[†], *Member*

SUMMARY Unstructured overlay networks are widely adopted in large-scale and heterogeneous peer-to-peer (P2P) systems for their scalability and flexibility. A distinct feature of such systems is that they randomly route messages e.g., by flooding or random walk. In such systems, the number of messages and tasks carrying by those messages each peer receives is greatly affected by the number of the peer's incoming links. The objective of this paper is to build controllable degree-weighted networks in which the expected number of incoming links of each peer is proportional to its *weight* which is a local parameter. In such a network, a peer can control the number of those randomly disseminated messages and tasks it receives by adjust it weight. In addition, in order to bound the construction overhead for highly biased networks, we restrict all peers to have the same number of outgoing links. The objective network is constructed by local topology transformations that peers periodically exchange outgoing links with each other. A framework, which includes 81 different protocols by combination of exchange rules, is presented and evaluated by simulation. The simulation result shows that two of them can generate the networks having similar properties with the objective network. This work first achieves the weight-proportional degree control under the out-regular network model.

key words: *peer-to-peer, overlay, proportional degree, PWDN*

1. Introduction

1.1 Background

Peer-to-Peer (P2P) systems, with prominent advantages such as scalability, robustness and low development cost, have developed quickly in recent years [1]. A P2P network is an overlay network over the IP-based Internet. In such a network, a peer v_i can directly send messages to any other peer v_j as long as v_i knows the network address (e.g., IP address) of v_j . Thus a P2P network is free to form any topology regardless of the underlay structure of the physical network. P2P systems are usually large-scale and highly dynamic networks in which peers frequently leave and join [2], [3]. The problem of how to maintain an efficient and robust P2P overlay attracts many attentions.

Unstructured networks are widely adopted in large-scale and heterogeneous P2P systems for their scalability and flexibility [5]–[7]. These so called unstructured P2P systems usually adopt blind routing methods (e.g. flooding and random walk) that peers randomly disseminate messages to nearby peers [8], [9]. In such systems, the number

of messages a peer receives in many cases greatly depends on the number of its incoming links (in-degree). Then the tasks such as search queries and data requests carrying by those messages that each peer receives is also decided by its in-degree. With different workload allocation policies, there are two kinds of traditional unstructured networks: the uniform-random networks and the layered networks (super peer networks).

Uniform-random networks is utilized to uniformly distribute the system workload to all peers [10]–[13]. In such networks, any pair of peers are connected by the same probability so that all peers have nearly the same degree and thus the workload each peer takes on is also nearly the same [14]. However, it is not reasonable to allocate the same workload to peers of different capacities (i.e., communication speed, CPU power etc.) because some powerless peers may slow down the system operation while powerful peers are not fully utilized.

Layered networks concentrate the system workload to some powerful peers [15]. They are proved can improve the search performance for unstructured P2P systems [6], [7], [16]. In such networks, a small number of powerful peers (called super peers) have very high degrees and process all search queries. Other peers (called leaf peers) are only connected to several super peers. Leaf peers register the information of their shared contents in neighbouring super peers but never process any search queries. Then, one can find the target contents by searching only a few super peers. A restriction of the layered network is that there must be enough powerful peers to take on the workload of the whole system. Actually, this condition does not always hold.

Between the above two extreme network topologies, some compromise approaches are proposed in recent years [17]–[19]. These approaches aim to construct non-uniform overlays in which more powerful peers have higher degrees. Obviously, the most reasonable and straightforward approach is to adjust each peer's degree and thus the workload proportional to its capacity. However, there are still no practicable solutions which can construct such a network well. The most possible reason is that the traditional undirected network model is not suitable for constructing such a network in highly biased P2P environments. In undirected networks, the most powerless peers also need to establish enough links in order to keep the network connected. Therefore, powerful peers need to establish many links in order to keep the same capacity-degree ratio as those pow-

Manuscript received September 29, 2008.

Manuscript revised January 2, 2009.

[†]The authors are with the Graduate School of Information Science and Technology, Osaka University, Suita-shi, 565-0871 Japan.

a) E-mail: wu-yu@ist.osaka-u.ac.jp

DOI: 10.1587/transinf.E92.D.563

erless peers. Many studies show that, in real P2P systems, the most powerful peer's capacity may be up to 10000 times higher than the most powerless peer's [17]. That implies those powerful peers must maintain a huge number of links which causes the network construction overhead to become terribly high.

1.2 Our Contribution

We propose distributed protocols to construct degree-weighted networks in which the degree of each peer can be adjusted proportionally to its capacity. Clearly, the problem of the high construction overhead can not be overcome in undirected networks. To solve this problem, we adopt the out-regular directed network model. In this model, all peers have the same number of outgoing links (out-degree) but the number of incoming links (in-degree) of each peer is adjusted proportional to its capacity. In such a network, a powerless peer can have very low average in-degree (even less than 1) because it has enough outgoing links to connect to the network. Therefore, powerful peers need much less incoming links to keep the capacity-degree ratio and thus highly-biased networks can also be constructed with reasonable overhead.

Moreover, in most cases, an out-regular network can control the workload that each peer receives as well as undirected networks. In many P2P applications, peers passively receive tasks from others so that the number of tasks a peer passively receives is mainly decided by its in-degree but hardly affected by its out-degree [18]. Notice that the active workload generated by a peer itself can be locally controlled which need not be managed by the system. In addition, if an application must be executed in undirected networks, one can easily transform a directed overlay network to an undirected one by letting each peer disseminate its network address through its outgoing links. Of course, a peer should remember the original direction of each link to keep the in-degree distribution.

The objective network is defined by the *Probabilistic Weighted d -out-regular Directed Network* (PWDN), where d is the predefined out-degree for all peers. The PWDN is a probabilistic space on the universe of the d -out-regular directed networks. The network topology continuously transforms over time. A certain network randomly appears with the probability defined by the PWDN. Many studies show that lazy networks (i.e. existing links are fixed until the peers leave the system) will be distorted by peers' leave and join, e.g. the random growing network [20]. Such a dynamic network model is favorable for maintaining the network topology (e.g. desired degree distribution) in dynamic P2P system that peers frequently join and leave.

The objective network is constructed by periodically exchanging links between peers. A distributed framework, which includes 81 protocols by combination of link exchange rules, is presented and evaluated by simulation. The evaluation criterion includes the accuracy of the degree control, self-organization property, randomness and so on. The

simulation result shows that two protocols can generate the networks having similar statistical properties with the PWDN.

2. Preliminaries

2.1 System Model and Notations

A P2P network is a directed network $D(V, E)$ that consists of a set of n peers $V = \{v_1, v_2, \dots, v_n\}$ and a set of directed links $E \subseteq \{e_{i,j} \mid v_i, v_j \in V, i \neq j\}$ where $e_{i,j}$ indicates a directed link from v_i to v_j . The network is simple that contains no self-loops and multiple links.

A *weight vector* of V is defined by $\bar{w} = (w_1, w_2, \dots, w_n)$ where $w_i (\geq 0)$ is the *weight* of v_i . We assume that any single peer's weight is much smaller than the total weights of all peers. That is, for any $v_i \in V$, $w_i \ll \sum_{x=1}^n w_x$. The weight is a local parameter decided by each peer itself. It express the activeness of a peer to take on system workload. For example, by setting each peer's weight to its capacity, one can adjust the peer's in-degree proportional to its capacity. We use the term 'weight' instead of 'capacity' because a peer may not always contribute all its physical capacity for system use. In addition, by giving different rules to set the value of each peer's weight, one can apply the network in different types of usages besides the workload control. Some related examples will be introduced in Sect. 5.

Below, we introduce some notations which are used throughout this paper. If there exists a link $e_{i,j} \in E$, v_i is called the *in-neighbour* of v_j and v_j is called the *out-neighbour* of v_i . The *out-view*, denoted by $view_i^+$, of v_i is defined by the set of outgoing links of v_i . The *out-degree* of v_i is denoted by $\Delta_i^+ = |view_i^+|$. Similarly, we define the *in-view*, denoted by $view_i^-$, by the set of incoming links of v_i . The *in-degree* of v_i is denoted by $\Delta_i^- = |view_i^-|$. Obviously, the following equality holds:

$$|E| = \sum_{v_i \in V} \Delta_i^+ = \sum_{v_j \in V} \Delta_j^- . \quad (1)$$

We say a network is *strongly-connected* if for all pairs of peers, there exists a directed path in E and *weakly-connected* if there exists a path neglecting the direction of the links for all pairs of peers.

A P2P network is an overlay network in which a peer v_i can directly send messages to any other peer v_j as long as v_i holds the entry (e.g. network address) of v_j , which is denoted by r_j . Practically, a link $e_{i,j}$ is the entry of v_j stored in v_i . Notice that implies the in-view of a peer is a set of the peer's entries stored in other peers. Therefore, a peer knows neither the in-view nor the in-degree of itself.

2.2 Objective Network

In this subsection we firstly define the objective network by the *Probabilistic Weighted d -out-regular Directed Network* (PWDN) and then introduce some properties of it.

Table 1 Symbols' definition.

V	$= \{v_1, v_2, \dots, v_n\}$, a set of peers.
n	$= V $, the number of peers.
$e_{i,j}$	A directed link from v_i to v_j .
r_i	The entry (network address) of v_i .
E	$\subseteq \{e_{i,j} \mid v_i, v_j \in V, i \neq j\}$, a set of links.
$D(V, E)$	A directed network.
w_i	The weight of v_i .
\bar{w}	$= (w_1, w_2, \dots, w_n)$, a weight vector of V .
$view_i^+$	The set of outgoing links of v_i .
$view_i^-$	The set of incoming links of v_i .
Δ_i^+	$= view_i^+ $, the out-degree of v_i .
Δ_i^-	$= view_i^- $, the in-degree of v_i .
δ_i^-	$= E[\Delta_i^-]$, the expectation of Δ_i^- .
$D_d^+(V, E)$	A d -out regular directed network.
d	The predefined out-degree of all peers.
$\mathcal{G}(V)$	$= \{g_1, \dots, g_m\}$, the universe of $D_d^+(V, E)$.
m	$= \mathcal{G}(V) $, the size of $\mathcal{G}(V)$.
$\bar{\pi}$	$= (\pi_1, \dots, \pi_m)$, a probability vector of $\mathcal{G}(V)$.
$\langle \mathcal{G}(V), \bar{\pi} \rangle$	A probability space of $D_d^+(V, E)$.

Definition 2.1: (d -out-regular directed network)

The d -out-regular directed network $D_d^+(V, E)$ is defined by a set of peers $V = \{v_1, \dots, v_n\}$ and a set of directed links $E \subseteq \{e_{i,j} \mid v_i, v_j \in V, i \neq j\}$ that for any $v_i \in V$, $\Delta_i^+ = d$. \square

The networks to be discussed in the following of this paper are based on this model. For short, we call the d -out-regular directed network simply by 'network'.

Definition 2.2: (Universe of $D_d^+(V, E)$)

The universe $\mathcal{G}(V)$ of $D_d^+(V, E)$ is the set of all possible d -out-regular directed networks with the peer set V , that is $\mathcal{G}(V) = \{g_1, \dots, g_m\}$ where $m = |\mathcal{G}(V)|$ and g_x ($1 \leq x \leq m$) denotes each certain network. \square

Since a peer's out-view can be any d peers selected from the other $n - 1$ peers a combination of $\binom{n-1}{d}$ selections are available for each peer. Therefore, the size of $\mathcal{G}(V)$ is $m = \binom{n-1}{d}^n$.

Definition 2.3: (Probability vector)

A probability vector $\bar{\pi}$ of $\mathcal{G}(V) = \{g_1, \dots, g_m\}$ is defined by $\bar{\pi} = (\pi_1, \dots, \pi_m)$ such that $\sum_{x=1}^m \pi_x = 1$, where π_x ($1 \leq x \leq m$) indicates the occurrence probability of network g_x . \square

Definition 2.4: (PWDN)

A *Probabilistic Weighted d -out-regular Directed Network* with the weight vector \bar{w} is a probability space $\langle \mathcal{G}(V), \bar{\pi} \rangle$ iff the following condition is satisfied for each pair of peers v_i and v_j ($i \neq j$):

$$\sum_{x=1}^m \pi_x f(e_{i,j}, g_x) = \frac{dw_j}{\sum_{k=1}^n w_k - w_i}, \quad (2)$$

where,

$$f(e_{i,j}, g_x) = \begin{cases} 1, & \text{if } e_{i,j} \text{ exists in } g_x \\ 0, & \text{otherwise.} \end{cases}$$

\square

By the definition, we know that the probability of $e_{i,j}$ is

$$p_{i,j} = \sum_{x=1}^m \pi_x f(e_{i,j}, g_x). \quad (3)$$

Under the assumption that for any $v_i \in V$, $w_i \ll \sum_{x=1}^n w_x$, we have the following properties of the PWDN.

Property 1: The expected in-degree of each peer $v_i \in V$ is $\delta_i^- \approx ndw_i / \sum_{x=1}^n w_x$. \square

Proof: By letting $\Delta_i^-(g_x)$ be the in-degree of v_i in g_x , we have

$$\begin{aligned} \delta_i^- &= \sum_{x=1}^m \pi_x \Delta_i^-(g_x) \\ &= \sum_{x=1}^m \pi_x \sum_{j=1}^n f(e_{j,i}, g_x) \\ &= \sum_{j=1}^n \sum_{x=1}^m \pi_x f(e_{j,i}, g_x) \\ &= \sum_{j=1}^n \frac{dw_i}{\sum_{k=1}^n w_k - w_j} \\ &\approx \frac{ndw_i}{\sum_{k=1}^n w_k}. \end{aligned} \quad \square$$

Property 1 is the main purpose of this work. It implies, for any two peers, the ratio of their in-degrees equals to the ratio of their weights, i.e. $\forall v_i, v_j \in V, \delta_i^- / \delta_j^- \approx w_i / w_j$. Notice that the in-degree of a peer is automatically adjusted by the network adapting to all peers' weights, the peer needs not (can not) decide its in-degree manually. This property is reasonable because the ratio of different peers' in-degrees is much more important than their exact in-degrees especially from the view point of workload allocation.

Property 2: For each peer v_i , other peers $v_j \in V, i \neq j$ have nearly the same probability to be the in-neighbour of v_i . \square

Proof: By Equality 2 and 3, we have

$$\begin{aligned} p_{j,i} &= \sum_{x=1}^m \pi_x f(e_{j,i}, g_x) \\ &= \frac{dw_i}{\sum_{k=1}^n w_k - w_j} \\ &\approx \frac{dw_i}{\sum_{k=1}^n w_k}, \end{aligned}$$

which does not depend on v_j . \square

The Property 2 implies that the in-view of each peer holds randomness. That is, each peer is possible to be adjacent from other peers with a same probability unless its weight is 0. Therefore, it guarantees the expected number of the randomly disseminated messages a peer receives to be proportional to its in-degree. It also enables fair workload allocation even when the amounts of tasks generated by peers are not uniform. In addition, the randomness property is favorable for keeping the network topology compact [14].

3. A Framework

3.1 General Description

The objective network is constructed by local topology transformations that peers periodically exchange outgoing links (entries of out-neighbours) with each other. We divide the link exchange process to several independent operations and investigate the possible options of each operation. The proposed framework includes all of the possible combinations of those options. A wide range of traditional construction protocols, including both experimental and theoretical ones, can fit into this framework [21], [22].

The principle of the weight-based degree control can be roughly explained by two simple rules as follows. First, a link incident to a peer having a higher weight is replicated with a higher probability during the link exchange process. Second, if the number of links incident to a peer increases, the probability of replicating such links is decreased. The second rule gives negative feedback to the first one so that degree of each peer stabilizes. We introduce a new parameter, called heft, to indicate the priority (i.e. the probability to be replicated) of each link. The heft of a link $e_{i,j}$ is denoted by $h(e_{i,j})$, which is stored in v_i . Based on the hefts of links, we propose distributed link exchange protocols without any global knowledges of peers' weights and in-degrees. In short, the initial heft of a newly created link $e_{i,j}$ is set to the weight of v_j and once a link is replicated, its heft is decreased by a half.

3.2 Operation Details

The framework is a set of protocols of how peers exchange links with each other. Each peer, denoted by v_i , periodically executes a 4-operation processes including (1) Target Selection, (2) Seed Planting, (3) View Merging, and (4) View Selection operations. Each operation has three options that can be selected independently. All peers in the system use a same combination of the options of each operation. Detailed descriptions of the operations from the viewpoint of peer v_i are as follows. For simple description, we assume that the outgoing links stored in each peer are sorted by decreasing order of the heft of each link.

(1) Target Selection: Peer v_i selects a local outgoing link $e_{i,j}$ from $view_i^+$. The peer v_j is decided to be the target peer to exchange links with. There are three different options of how to select a link from its out-view based on the heft of each link.

- **Random:** A link is selected from $view_i^+$ uniformly at random regardless of its heft.
- **Head:** The first outgoing link in $view_i^+$ (with the highest heft) is selected.
- **Tail:** The last outgoing link in $view_i^+$ (with the lowest

heft) is selected.

(2) Seed Planting: A new link, called seed, incident to v_i (or v_j) is created and inserted to v_j 's (or v_i 's) out-view. The initial heft of the newly created link is set to the weight of the link's destination peer.

- **Push:** Peer v_i inserts its seed $e_{j,i}$, into $view_j^+$. The seed's heft $h(e_{j,i})$ is set to the v_i 's weight w_i , i.e. creating a link $e_{j,i}$, $h(e_{j,i}) = w_i$.
- **Pull:** Peer v_j inserts its seed $e_{i,j}$, into $view_i^+$. The seed's heft $h(e_{i,j})$ is set to the v_j 's weight w_j . i.e. creating a link $e_{i,j}$, $h(e_{i,j}) = w_j$.
- **Push&Pull:** Execute both Push and Pull.

(3) View Merging: A peer, either v_i or v_j , inserts a copy of its out-view into the other's out-view. The heft of the replicated links (both the original and the copy) are decreased by a half. If there are two links of the same heft, the newly inserted one is ordered behind the other.

- **Push:** Peer v_i decreases the heft of all links in $view_i^+$ by a half, i.e. for all $e_{i,x} \in view_i^+$, $h(e_{i,x})/2 \rightarrow h(e_{i,x})$ where v_x is an out-neighbour of v_i . Then, v_i insert $view_i^+$ into $view_j^+$, i.e. $view_i^+ \cup view_j^+ \rightarrow view_j^+$.
- **Pull:** Peer v_j decreases the heft of all links in $view_j^+$ by a half, i.e. for all $e_{j,x} \in view_j^+$, $h(e_{j,x})/2 \rightarrow h(e_{j,x})$ where v_x is an out-neighbour of v_j . Then, v_j insert $view_j^+$ into $view_i^+$, i.e. $view_j^+ \cup view_i^+ \rightarrow view_i^+$.
- **Push&Pull:** Executes both Push and Pull.

(4) View Selection: After the Seed planting and View merging operations, some peers' out-degrees may temporarily be more than d and the network may temporarily have self-loops and multiple links. The View Selection operation is a mechanism for keeping the network be d -out-regular, i.e. to select d links to be remain in the view based on the heft of each link and deleted others. It also deletes self-loops and multiple links to keep the network to be simple.

- **Random:** Firstly, the self-loops are deleted. Then, for each group of multiple links, one link is selected uniformly at random from each group to remain and others are deleted. Finally, the peer selects a set of d links from its out-view uniformly at random to remain, other links are deleted.
- **Head:** Firstly, the self-loops are deleted. Then, for each group of multiple links, the link with the highest heft in each group remains and others are deleted. Finally, the first d outgoing links (with higher hefts) in the peer's out-view are selected to remain and other links are deleted.
- **Tail:** Firstly, the self-loops are deleted. Then, for each group of multiple links, the link with the lowest heft in each group remains and others are deleted. Finally, the last d outgoing links (with lower hefts) in the peer's out-view are selected to remain and other links are

deleted.

The framework includes $3^4 = 81$ protocols which are the combinations of the 4 operations' options. We denote each protocol by a 4-tuple (ts, sp, vm, vs) where ts, sp, vm, vs indicates the options of the target selection, seed planting, view merging and view selection operations respectively. A wildcard is denoted by the symbol '*'. For example, $(Random, Push, *, Head)$ indicates three different protocols that adopt the Random Target Selection, the Push Seed Planting, any one of the three view merging options and the Head View Selection.

3.3 Peers' Join and Leave

When a peer v_i joins the system, we assume it can access at least one peer, called initiator, in the system. The initiator can be a login server or simply a peer which registers its entry in a published website [23]. From the initiator, the peer can obtain some links as the initial out-view. Then, by several link exchange processes, the out-view can be full filled. We need not any restrictions on the initial out-view of each peer, e.g. it can be only a link incident to the initiator or a set of links collected by a random walker starting from the initiator.

When a peer leaves the system, no additional procedures are required such as leaving announcing because its entires will be finally deleted during the exchanging operations and no fresh seeds will be planted. Therefore, the network has fault tolerance to peers' crash and disconnect (the most frequently occurred faults in P2P systems) because a crashed (or disconnected) peer can be considered as a peer normally left from the system.

The join and leave procedures is so simple that the network can adapt to different system environments. However, such a design requires the network to be well self-organizing because some undesirable join and leave patterns may distort the network topology. Therefore, the self-organization property is an important evaluation criterion of the proposed protocols.

4. Simulation

4.1 Evaluation Criteria

We evaluate the proposed protocols by simulation to find some protocols that can generate networks having similar properties with the PWDN. The evaluation criteria are decided from two standpoints. First, the network generated by a protocol should have similar properties with the PWDN. Second, a protocol should be applicable in P2P environments. Below, we introduce the details of the evaluation criterion.

We say a network is a good approximation of the PWDN if it satisfies the PWDN's Property 1 and Property 2. We say a network is applicable for P2P environments if it satisfies the following properties.

- Compact. A network should have small diameters to achieve efficient gossip-based communications in large-scale P2P systems.
- Self-organized. A network should be able to quickly converge to the expected topology from any initial weakly-connected topologies. The property is important for keeping the topology while peers frequently leave and join.
- Connected. A network should be at least weakly-connected with a high probability.

4.2 Simulation Environment

The simulation includes 10000 peers, each peer has at most 30 outgoing links. Each peer has the same executing interval. In the first execution cycle, they execute the protocol in a random order.

All of the 81 protocols are evaluated by a set of tests. For lacking of space, not all of the results can be shown in this paper. In the following of this section, if a protocol clearly generates undesirable networks in any test, it will be excluded from further evaluation.

By preliminary extermination, the protocols $(*, *, *, Random/Tail)$ are excluded from the candidate protocols because they are clearly not able to control each peer's in-degree proportion to its weight.

From the principle of the weight-based degree control, we know that links incident to a peer of a high weight often has high hefts. In the case of the Random View Selection, links remain with the same probability regardless of its heft. And in the case of the Tail View Selection, links with lower hefts can remain earlier. Both of them are inconsistent to our purpose. In fact, such networks converge to star-like networks. Similar arguments can be found in M.Jelasiy's work [22]. In the following of this paper, we focus on the rest 27 protocols $(*, *, *, Head)$.

4.3 Uniform Weight Setting

We firstly evaluate the protocols under the uniform weight setting that each peer has the same weight. By the definition of the PWDN, if all of the peers have a same weight, any pairs of peers should be connected by a directed link with a same probability. That is the d -out-regular random network [14]. We verify if the protocols can generate networks having similar statistical and graph properties with that of the d -out-regular random network.

The protocols is started from two kinds of initial network topologies, the d -out-regular random network and the star-network. After 1000 cycles, we compute some statistical and graph parameters from the snapshots of the candidate networks. We also show same parameters of the d -out-regular random network generated by the centralized protocol that each peer selects d out-neighbours from the system uniformly at random. Notice that the simulation time is long enough for the networks to stabilize. Latter, it will be shown

Table 2 Protocols (*, *, *, *Head*), uniform weight setting, $n = 10000$, $d = 30$, 1000 cycles executed, starting from uniform-random network.

Protocols	VAR	SIGHT	Connectedness	Max Diameter	Average Path Length
Uniform-random	30	-	Strong	4	2.97
Random, Push, Push	23	3652	Strong	4	3.06
Random, Push, Pull	44	4048	Strong	4	3.03
Random, Push, Push&Pull	45	6905	Strong	4	3.06
Random, Pull, Push	991	2219	Weak	-	-
Random, Pull, Pull	38631	514	Weak	-	-
Random, Pull, Push&Pull	19028	1890	Weak	-	-
Random, Push&Pull, Push	77	3528	Strong	4	3.09
Random, Push&Pull, Pull	76	3735	Strong	4	3.07
Random, Push&Pull, Push&Pull	129	6807	Strong	5	3.10
Head, Push, Push	28	36	Strong	4	3.00
Head, Push, Pull	41	38	Strong	5	2.98
Head, Push, Push&Pull	74	46	Strong	5	3.01
Head, Pull, Push	29	30	Strong	4	2.97
Head, Pull, Pull	9748	64	Weak	-	-
Head, Pull, Push&Pull	7977	93	Weak	-	-
Head, Push, Push	32	33	Strong	4	2.98
Head, Push, Pull	54	43	Strong	4	2.99
Head, Push, Push&Pull	76	48	Strong	5	3.02
Tail, Push, Push	19	3810	Strong	4	3.03
Tail, Push, Pull	42	4254	Strong	4	3.00
Tail, Push, Push&Pull	49	6887	Strong	4	3.06
Tail, Pull, Push	951	2337	Weak	-	-
Tail, Pull, Pull	25308	746	Weak	-	-
Tail, Pull, Push&Pull	68258	1655	Weak	-	-
Tail, Push&Pull, Push	76	3735	Strong	4	3.05
Tail, Push&Pull, Pull	73	4014	Strong	4	3.02
Tail, Push&Pull, Push&Pull	154	6926	Strong	5	3.07

Table 3 Protocols (*, *, *, *Head*), uniform weight setting, $n = 10000$, $d = 30$, 1000 cycles executed, starting from star network.

Protocols	VAR	SIGHT	Connectedness	Max Diameter	Average Path Length
Random, Push, Push	5344	2881	Weak	-	-
Random, Push, Pull	44	4035	Strong	4	3.03
Random, Push, Push&Pull	47	6886	Strong	4	3.06
Random, Push&Pull, Push	5494	339	Weak	-	-
Random, Push&Pull, Pull	76	3714	Strong	4	3.07
Random, Push&Pull, Push&Pull	132	6781	Strong	5	3.10
Tail, Push, Push	4328	508	Weak	-	-
Tail, Push, Pull	43	4260	Strong	4	3.00
Tail, Push, Push&Pull	48	6872	Strong	4	3.06
Tail, Push&Pull, Push	4110	541	Weak	-	-
Tail, Push&Pull, Pull	74	4016	Strong	4	3.02
Tail, Push&Pull, Push&Pull	150	6909	Strong	5	3.07

that most networks stabilize within 100 cycles.

The test items are as follows.

- **Variance Test** evaluate the necessary condition of the Property 1 that peers' in-degree should be nearly the same under the uniform weight setting. The test excludes some networks in which peers in-degrees vary widely.
- **Transformation Test** evaluate the necessary condition of the PWDN's Property 2 that the network topology should continuously transform over time. The test excludes the networks in which peers have fixed views.
- **Self-organization Test** verifies if the protocols can

generate similar networks starting from different initial topologies.

- **Scale Test** compares the diameter and the average path length of the protocols with the out-regular random network.

Notice that the above test items are not sufficient condition of the objective network. The network passed the test will be further evaluated by other tests. Below, we show the detailed evaluation contents of the test items. The simulation results are shown in Tables 2 and 3.

- (1) Variance Test. The variance of the 10000 peers' in-

degrees are shown by the item VAR in Table 2. In the d-regular random network, each peers' in-degrees follows a *Binomial distribution* that

$$\forall v_i \in V, \Pr[\Delta_i^- = k] = C_{n-1}^k p^k (1-p)^{n-1-k} \quad (4)$$

where p is the probability of any pair of peers being connected by a directed link. By Equality 3, we have $p_i = d/(n-1)$. Therefore, the variance of an out-regular random network is $(n-1)p(1-p) \approx 30$. Compare with it, 8 networks, generated by $(*, Pull, *, Head)$ except for $(Head, Push, Push, Head)$, have obviously higher variances. That implies peers' in-degrees are quite different. In addition, the variances of such protocols keep increasing during the whole simulation period while others' stabilize within 100 cycles. Therefore, these 8 protocols are excluded from further evaluation.

The undesired result is mainly caused by the Pull option in the Seed Planting operation that creates more seeds for popular peers (i.e. the peers having high in-degrees) than unpopular ones. Since more seeds a peer has, more links incident to the peer are replicated by the View Merging operation, the Pull option gives positive feedback to the in-degree of each peer that makes the in-degree distribution divergent. With the same reason, it can be seen from Table 2 that the protocols adopting the Push&Pull option in the Seed Planting operation also have higher variance than that of adopting the Push option. However, we remain them for the moment because the results are still acceptable in this test.

(2) Transformation Test. A necessary condition of the PWDN's Property 2 is that each peer must have a dynamic in-view in which all peers' entries may appear. For each peer v_i , a parameter called *sight* is defined by the number of the peers from which a link to v_i is generated at least once during the simulation period, i.e. the size of the union of the in-neighbours of v_i during 1000 cycles. In Table 2, the average sight of all peers are shown by the item SIGHT. From the results we can find that 9 protocols ($Head, *, *, *$) have very low sights. That implies the peers are only accessed by a small part of all peers that is clearly undesirable. Therefore, these 9 protocols (2 of them have already been excluded by the Variance Test) are excluded from the simulation.

By the Head option in the Target Selecting operation, peers select a link of the highest heft. That implies a peer often selects a target with which the latest link exchange is executed since a newly received seed often has the highest heft. As the result, each peer exchanges links with a fixed set of peers which depends on the initial network topology. In fact, such networks result in severe clustering [22].

(3) Self-Organization Test. A network should be well self-organizing to keep the network topology in dynamic P2P environments that peers frequently leave and join. That is, it should be able to quickly converge to the desired topology from any initial network topologies. In this test, we start the simulation from a star network and test if the protocols can

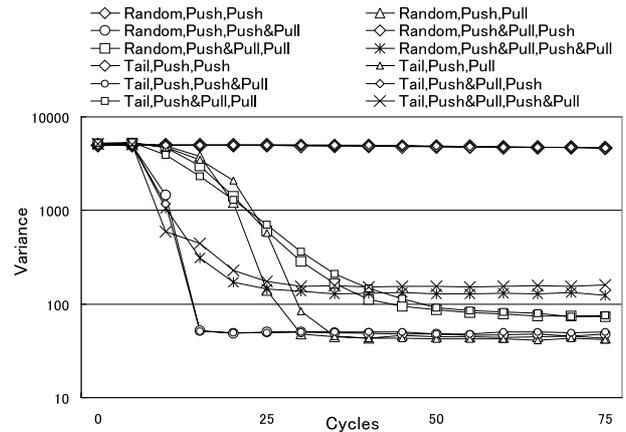


Fig. 1 Changing of in-degree variance.

generate similar networks as in previous tests. The results of remaining 12 protocols are shown in Table 3. Clearly, 4 networks, generated by $(Random/Tail, Push/ Push&Pull, Push, Head)$, have much higher variance than previous results. Other protocols' results are almost the same as those in Table 2. Figure 1 shows the change of each network's variance during the simulation period. We can see that, except the above 4 networks, the variance of other networks quickly decrease and converge in a short time. Therefore, they are excluded for lack of the self-organization ability.

These 4 excluded protocols adopt the Push option in the View Merging operation. In such a network, a peer must passively wait for other peers push outgoing links to it. That implies an unpopular peer (i.e. a peer of a low in-degree) can hardly establish enough outgoing links. Therefore, such networks can not recover from high-biased topologies quickly. This is also a critical drawback that new peers can not quickly join the system.

(4) Scale Test. Up to now, 8 protocols, $(Random/Tail, Push/ Push&Pull, Pull/ Push&Pull, Head)$, have passed all of the previous tests. We show some graph properties, such as the connectedness, the diameter and the average path length of them in Tables 2 and 3. It can be seen that these parameters of the 8 networks are similar to the out-regular random network. Therefore, all of the 8 protocols pass the Scale Test and will be evaluated by more strict tests for degree control.

Finally, we show degree distribution of the 8 remaining protocols under the uniform weight setting in Figs. 2 and 3. The curve 'Ideal' is the in-degree distribution of a out-regular random network that follows a binomial distribution (Equality 4) as we mentioned in the Variance Test. Among these protocols, only 2 protocols $(Random/Tail, Push&Pull, Push& Pull, Head)$ can also be generated by M. Jelasity's framework [22]. These results clearly show that, under the uniform weight setting, our new protocols generate better uniform networks (i.e. peers' in-degrees have less random variance) than any protocols in M. Jelasity's framework.

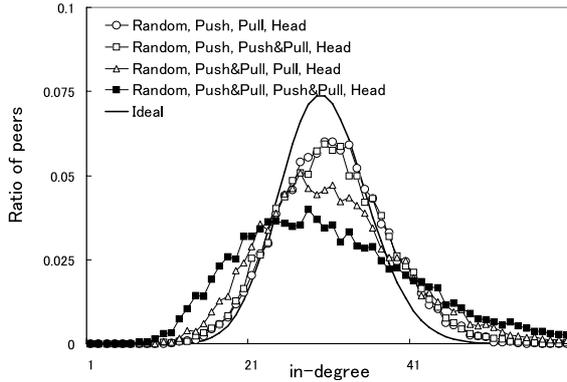


Fig. 2 In-degree distribution, Random Target Selection.

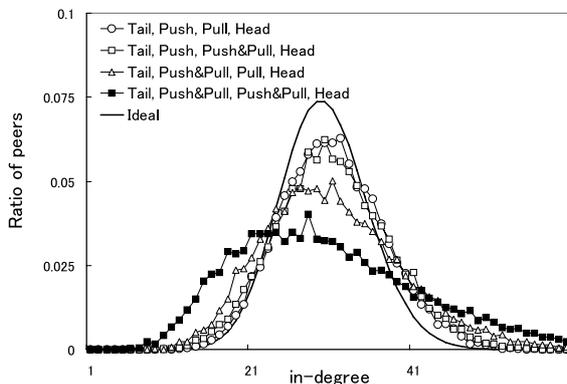


Fig. 3 In-degree distribution, Tail Target Selection.

4.4 Weighted Setting

In this subsection, the 8 protocols that passed all tests under the uniform weight setting are evaluated for their performance of degree control.

Firstly we test the accuracy of the weight-based in-degree control under a simple weight setting. We divide peers into two groups of $V_A = \{v_1, \dots, v_{9000}\}$ and $V_B = \{v_{9001}, \dots, v_{10000}\}$, called group A and B respectively. The weight of all peers in group A, denoted by W_A , is fixed by 1. The weight of all peers in group B, denoted by W_B , is set to $2 \sim 128$ in each experiment respectively. Notice that the grouping plan is imaged by the layered networks in which usually 10% of the peers are selected to be super peers [24]. The average in-degree of the two groups, denoted by $\bar{\Delta}_A^-$ and $\bar{\Delta}_B^-$ respectively, are computed after 1000 cycles starting from a d -out-regular random network. The ratio $R_{B,A} = \bar{\Delta}_B^- / \bar{\Delta}_A^-$ of the protocols is shown in Figs. 4 and 5. By Property 1, the ideal ratio is $R_{B,A} = W_B / W_A$ which is shown by the curve Ideal.

From the results, it can be clearly seen that the protocols (*Random/Tail, Push&Pull, Pull/Push&Pull*) and (*Random/Tail, Push, Pull*) can not control the peers' in-degree correctly. The results of protocols (*Random/Tail, Push, Push&Pull*) are accurate while W_B is lower than 32 but slightly higher than the expected value while W_B is higher

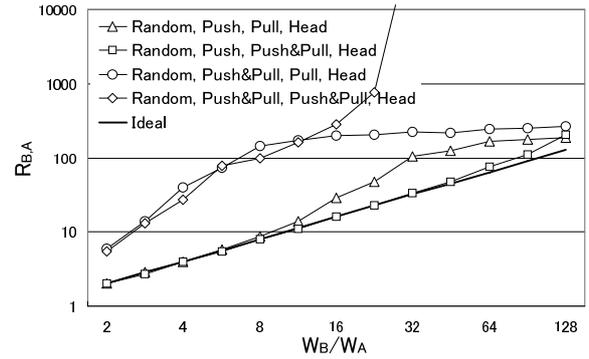


Fig. 4 Degree control results, Random Target Selection.

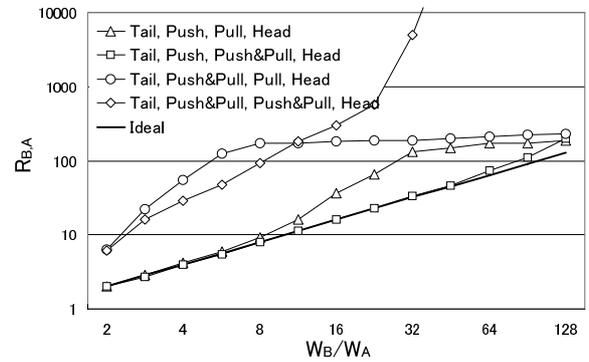


Fig. 5 Degree control results, Tail Target Selection.

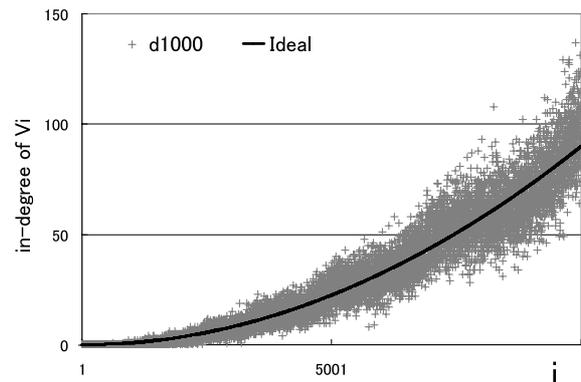


Fig. 6 Peers' in-degree under power-law weight distribution, (*Random, Push, Push&Pull, Head*).

than 64. Although the result implies they may have error when generate highly biased networks, they are still good approximations of the PWDN.

Then we test the two protocols under a power-law weight setting. For peer $v_i, 1 \leq i \leq n$, it's weight is set to $w_i = 1 + i^2 / 10000$. The simulation started from a d -out-regular random network and executed for 1000 cycles. The in-degree of each peer are shown in Figs. 6 and 7 for each protocol respectively. In these figures, the curve $d1000$ is the in-degree of each peer taken from the snap shot of the network at the end of the 1000-th cycle. By PWDN's Property 1, we know the expected in-degree of a peer is

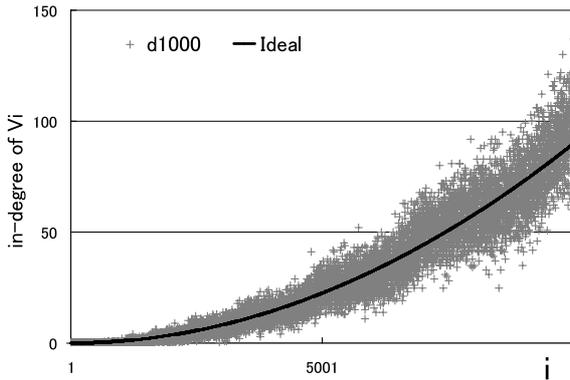


Fig. 7 Peers' in-degree under power-law weight distribution, (*Tail, Push, Push&Pull, Head*).

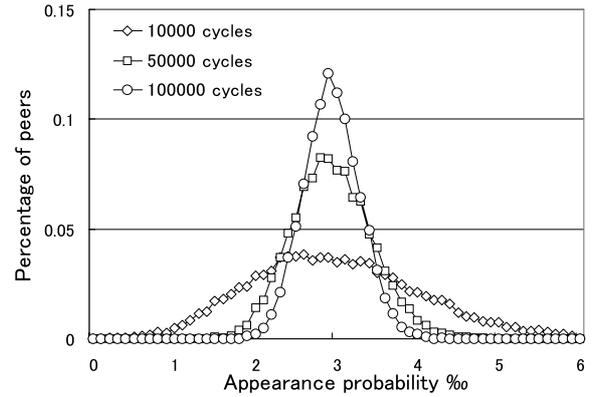


Fig. 9 Appearance probability distribution of in-neighbours, (*Tail, Push, Push&Pull, Head*).

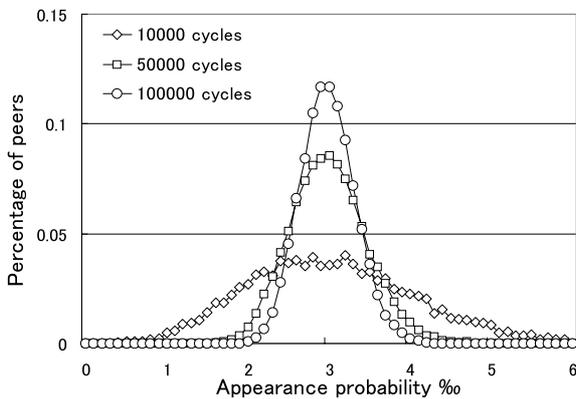


Fig. 8 Appearance probability distribution of in-neighbours, (*Random, Push, Push&Pull, Head*).

$$\delta_i^- = \frac{ndw_i}{\sum_{j=1}^n w_j} = \frac{30000 + 3i^2}{\sum_{j=1}^{10000} 1 + j^2/10000},$$

where $n = 10000, d = 30$. The ideal in-degree of each peer is shown by the curve Ideal. Clearly, both of the two protocols can generate networks having expected in-degree distributions. Notice in this test the network is highly biased that $w_1/w_{10000} = 5000$. The expected in-degrees of some peers having low weights are much less than 1. This is the reason why we can construct the network with only 30 links per peer. Clearly, such a highly biased network can not be realized with reasonable overheads using undirected networks.

4.5 Randomness

In this subsection, the two protocols (*Random/Tail, Push, Push&Pull*) are verified for the PWDN's Property 2. This test adopts the uniform-weight setting. We trace a randomly selected peer for a long time. At the end of each cycle, the entries in the in-view of the peer (the peer's in-neighbours) are recorded. The appearance probability of each peer's entry is computed from the accumulated records at the end of the 10000-th, 50000-th and 100000-th cycles. Figures 8 and 9 show the distribution of the appearance probabilities of each peer's entry for the two protocols respectively.

Clearly, the expected appearance probability of each peer's entry should be $d/(n - 1) \approx 3\%$. From both of the protocols' results we can see that if the simulation time is long enough, the appearance probability of each peer converges to the expected probability.

The term 'Randomness' also means a peer's view (either in-view or out-view) should be unpredictable. For the protocol (*Random, Push, Push&Pull*), the condition is clearly fulfilled because of the Random option in the Target Selection operation. For the protocol (*Tail, Push, Push&Pull*), if the executing order and the initial out-view (or in-view) of each peer is known, the network can be predicted. However, because peers randomly join and leaves, the network topology is unpredictable in practice.

4.6 Connectivity

During all of the above simulations, the two protocols always generate strongly-connected networks. It is mainly because the out-degree ($d = 30$) is large enough for a 10000 peers' network. B. Yang, et al. show that a 2-out-regular random directed graph is weakly-connected with high probability [25]. It has also been proved that a random-graph requires $O(\log n)$ average degree to be strongly-connected with a high probability [14]. We are interested in how much outgoing links are required for our protocols to generate weakly-connected networks as well as strongly-connected ones.

We execute the two protocols for 1000 cycles starting from the d -out-regular random network. The network size is set to $n = 1000$ or $n = 10000$. The out-degree d is a variable in this simulation. After each cycle, we take a snapshot of the network and check for the connectedness. The percentage of strongly-connected snapshots are shown in Table 4. The symbol '-' indicates the protocol can not keep the network connected in 1000 cycles. We also show the results of 1000 d -out-regular random networks generated by the centralized protocol.

From the Table 4 we can see that our protocols require more out-degree to generate weakly-connected networks than that of out-regular random networks. How-

Table 4 Percentage of strongly-connected networks. (*, *Push, Push&Pull, Head*), 1000 cycles executed.

n=1000:	d=6	d=7	d=8	d=9	d=10
Random, ...	-	-	97%	99%	99%
Tail, ...	-	71%	98%	99%	99%
random network	1%	9%	100%	100%	100%
n=10000:	d=6	d=7	d=8	d=9	d=10
Random, ...	-	-	-	94%	96%
Tail, ...	-	-	91%	95%	99%
random network	0	0	3%	10%	65%

ever the required out-degree only increased by 1 from a 1000 peers' network to a 10000 peers' one. It possibly implies that $O(\log n)$ outgoing links are necessary for constructing a connected network by our protocols but further certification are required. Interestingly, in some cases our protocols have higher probabilities to generate strongly-connected networks than the out-regular random network. It is unclear whether this feature is an advantage of better connectivity or just implies higher cluster coefficients.

5. Supplement

5.1 Applications

As we mentioned in Sect.1, the primal purpose of the PWDN is to adjust the workload a peer passively receives proportional to its capacity in unstructured P2P systems which randomly disseminate tasks by flooding or random walk etc. In different types of applications, the rules of deciding the value of weight (or say, the meaning of capacity) are different. The weight of a peer can be set to its communication speed, storage capacity and CPU power in file sharing systems, distributed storage systems and grid computing systems, respectively. One can also adopt the PWDN to construct Gnutella-like layered networks by setting the weight of super peers to their capacities and leaf peers to 0.

Besides the workload allocation, the PWDN can be applied for many other purpose. For example, Cooper proposed an optimized P2P search protocol based on a weighted overlay network [26]. The work shows that if a peer's in-degree is proportional to the square-root of the peers' popularity, the number of queries used to find target objects can be minimized. Here, a peer's popularity is the hit ratio of the queries which hit their targets in the peer among all queries received by it, that can be computed from the peer's historical record. However, the author did not provide a distributed method for constructing such a network. Obviously, such a network can be constructed by our protocols by setting each peer's weight to the square-root of its popularity.

5.2 Related Works

Structured networks.

The structured networks are another kind of popular P2P overlay networks. Most of such networks are constructed by the distributed hash table (DHT) technique [27],

[28]. Such networks have the advantage in efficient unicast, i.e. messages can be routed to any specified peer within a few hops while the average degree of peers is very small. However, they incur high maintenance overhead when peers leave and join the system [29]. Moreover, their tight structures are lack of flexibility in heterogeneous P2P environments.

Non-uniform networks.

The distributed non-uniform network construction is a mainstream topic of P2P technologies. Many works aim to construct degree-weighted networks but only a few of them can adjust each peer's degree proportional to its weight (capacity) [17], [19]. Vishnumurthy et al. proposed random-walk-based protocol to construct directed networks in which each peer's in-degree is proportional to its capacity. Their purpose is the same as ours but the approach is quite different [18]. In their networks, a peer must establish the same number of outgoing links as their expected in-degrees, i.e. both the out-degree and the in-degree are proportional to the peer's capacity. Therefore, the problem of high construction overhead for highly biased networks still remains in their approach. In addition,

Besides the degree-weighted networks, the location-aware network is another popular types of non-uniform networks [30]. In such networks, peers closer with each other in the physical network are connected with higher probability. Therefore, they can achieve more efficient routing in the physical network than random networks. The technique is in fact the same as that of the interest-based clustering [31]. Unfortunately, these networks' purposes are not consistent to the degree-weighted networks [19].

Gossip-based overlay construction.

Our protocol is a kind of gossip-based overlay construction protocols [21], [22]. A distinct feature of such protocols is the proactive link maintenance approach that incurs no additional overhead when peers join and leave. The contrary approach is the reactive link maintenance approach that the network topology is maintained only when peers join and leave [18], [19]. Since in real P2P systems peers frequently join and leave, it is considered there is no obvious difference in the construction overhead between the reactive approaches and the proactive approaches. In addition, reactively constructed networks usually do not have the 'randomness' property.

Jelasy et al. propose a gossip-based framework, which consists of 27 candidate protocols, to generate uniform-random networks under the out-regular directed network model [22]. The main design idea of our framework is inherited from it. However, their work can only construct networks with the uniform degree distribution which is a special case of the PWDN. In the case that all peers' weights are set to the same value, our framework can be regarded as an extension of theirs. A major difference of the two frameworks is that the Seed Planting and the View Merging operations are bounded in one operation in Jelasy's framework. So our framework has many new protocols including the two

eligible protocols of the PWDN. The simulation results also show these two protocols can generate better uniform networks than any protocols in their framework.

6. Concluding Remarks

In this paper, we study a fundamental and valuable problem for P2P overlay construction: given a set of peers with respective weights, adjust each peers' in-degree proportional to its weight. In order to bound the construction overhead in biased networks, we restrict all peers to have the same number of outgoing links. To the best of our knowledge, this problem is firstly solved under the out-regular directed network model.

The objective network is defined by the *Probabilistic Weighted d-out regular Directed Network (PWDN)*. By simulation, two protocols, the (*Random/Tail, Push, Push&Pull, Head*), are proved to be feasible for constructing the PWDN in P2P environments. The simulation result also shows that they can construct highly-biased networks with a reasonable number of total links. The result implies that we have overcome the problem of the high construction overhead for highly biased networks. This problem is considered impossible to be solved by traditional approaches which adopt the undirected network model.

This paper presents a powerful middleware for constructing heterogeneous overlay networks. It has only one interface parameter called 'weight'. By giving appropriate rules to decide the value of weight, one can apply the PWDN for different types of applications. The most representative application of the PWDN is to realize the capacity-proportional degree control in unstructured P2P system which randomly disseminate tasks. It can also be applied to solve other gossip-based problems such as distributed search and election. Its applications are worthy of further studies.

The future work includes the theoretical analysis under different churn models and improving the accuracy of degree control in highly-biased networks.

Acknowledgements

This work is supported in part by MEXT: "Global COE (Centers of Excellence) Program", JSPS: Grant-in-Aid for Scientific Research ((B)19300017, (B)17300020 and (B)20300012), MEXT: Grant-in-Aid for Scientific Research on Priority Areas (16092215), MEXT: Grant-in-Aid for Young Scientists ((B)18700059) and the Kayamori Foundation of Information Science Advancement.

References

- [1] D.S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, S. Rollins, and Z. Xu, "Peer-to-peer computing," Technical Report HPL-2002-57, HP Laboratories Palo Alto, March 2002.
- [2] M. Castro, M. Costa, and A. Rowstron, "Performance and dependability of structured peer-to-peer overlays," Proc. DSN '04, p.9, 2004.
- [3] B. Wilcox-O'Hearn, Experiences deploying a large-scale emergent network. citeseer.ist.psu.edu/734934.html
- [4] Freenet website: <http://freenet.sourceforge.net>
- [5] Gnutella website: <http://gnutella.wego.com>
- [6] KaZaA website: <http://www.kazaa.com>
- [7] Winny info.: <http://en.wikipedia.org/wiki/Winny>
- [8] R. Morselli, B. Bhattacharjee, M.A. Marsh, and A. Srinivasan, "Efficient lookup on unstructured topologies," IEEE J. Sel. Areas Commun., vol.25, no.1, pp.62–72, Jan. 2007.
- [9] M. Repeanu, "Peer-to-peer architecture case study: Gnutella network," Proc. P2P'01, p.99, 2001.
- [10] The Gnutella protocol specification v0.4.
- [11] <http://www.sun.com/software/jxta/>
- [12] K. Miura, T. Tagawa, and H. Kakugawa, "A quorum-based protocol for searching objects in peer-to-peer networks," IEEE Trans. Parallel Distrib. Syst., vol.17, no.1, pp.25–37, 2006.
- [13] Y. Wu, T. Izumi, F. Ooshita, H. Kakugawa, and T. Masuzawa, "An adaptive randomized search protocol in peer-to-peer systems," SAC '07: Proc. 2007 ACM Symposium on Applied Computing, pp.533–537, ACM Press, New York, NY, USA, 2007.
- [14] B. Bollobás, Random Graphs, Cambridge, 2001.
- [15] D. Stutzbach, R. Rejaie, and S. Sen, "Characterizing unstructured overlay topologies in modern p2p file-sharing systems," Proc. IMC'05, 2005.
- [16] S. Zhao, D. Stutzbach, and R. Rejaie, "Characterizing files in the modern gnutella network: A measurement study," Multimedia Computing and Networking 2006, vol.6071, no.1, p.60710M, SPIE, 2006.
- [17] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, "Making gnutella-like p2p systems scalable," Proc. SIGCOMM '03, pp.407–418, ACM Press, New York, NY, USA, 2003.
- [18] V. Vishnumurthy and P. Francis, "On heterogeneous overlay construction and random node selection in unstructured p2p networks," Proc. IEEE INFOCOM2006, 2006.
- [19] K.W. Kwong and H.K. Tsang, "Building heterogeneous peer-to-peer networks: Protocol and analysis," Proc. IEEE/ACM Trans. Netw., 2008.
- [20] D.S. Callaway, J.E. Hopcroft, J.M. Kleinberg, M.E.J. Newman, and S.H. Strogatz, "Are randomly grown graphs really random?," Phys. Rev. E, vol.64, no.4, 041902, 2001.
- [21] P. Mahlmann and C. Schindelhauer, "Peer-to-peer networks based on random transformations of connected regular undirected graphs," Proc. SPAA05, pp.155–164, 2005.
- [22] M. Jelasity, R. Guerraoui, A.-M. Kermarrec, and M. van Steen, "The peer sampling service: Experimental evaluation of unstructured gossip-based implementations," Proc. Middleware '04, 2004.
- [23] Winny initiating website: <http://winny.cool.ne.jp/index.html>
- [24] L. Xiao, Z. Zhuang, and Y. Liu, "Dynamic layer management in superpeer architectures," IEEE Trans. Parallel Distrib. Syst., vol.16, no.11, pp.1078–1091, 2005.
- [25] T.I. Fenner and A.M. Frieze, "On the connectivity of random m-orientable graphs and digraphs," AMS Subject Classification (1980), 05 C 40 - 60 C 05, 1981.
- [26] B. Cooper, "Quickly routing searches without having to move content," Proc. IPTPS2005, 2005.
- [27] S. Ratnasamy, P. Francis, M. Handley, M.F. Karp, and S. Shenker, "A scalable content-addressable network," Proc. SIGCOMM, pp.161–172, 2001.
- [28] I. Stoica, R. Morris, and D. Karger, "Chord: A scalable peer-to-peer lookup service for internet application," Proc. SIGCOMM, pp.149–160, 2001.
- [29] J. Li, J. Stribling, T. Gil, R. Morris, and F. Kaashoek, "Comparing the performance of distributed hash tables under churn," Proc. IPTPS2004, 2004.
- [30] M. Zhong, K. Shen, and J. Seiferas, "Non-uniform random membership management in peer-to-peer networks," Proc. IEEE INFOCOM2005, 2005.

- [31] M. Li, W.-C. Lee, and A. Sivasubramaniam, "Quickly routing searches without having to move content," Proc. ICNP'04, 2004.



Yu Wu received the B.E. degree in engineering in 2002 from Shanghai University and M.E. degree in computer science in 2006 from Osaka University. He is now a student of Graduate School of Information Science and Technology, Osaka University.



Fukuhito Ooshita received the M.E. and D.I. degrees in computer science from Osaka University in 2002 and 2006. Since 2003, he has been an Assistant Professor in the Graduate School of Information Science and Technology at Osaka University.



Hirotugu Kakugawa received the B.E. degree in engineering in 1990 from Yamaguchi University, and the M.E. and D.E. degrees in information engineering in 1992, 1995 respectively from Hiroshima University. He is currently an associate professor of Osaka University.



Toshimitsu Masuzawa received the B.E., M.E. and D.E. degrees in computer science from Osaka University in 1982, 1984 and 1987. He had worked at Osaka University during 1987–1994, and was an associate professor of Graduate School of Information Science, Nara Institute of Science and Technology (NAIST) during 1994–2000. He is now a professor of Graduate School of Information Science and Technology, Osaka University.