Building Heterogeneous Peer-to-Peer Networks: Protocol and Analysis

Kin-Wah Kwong, *Student Member, IEEE*, and Danny H. K. Tsang, *Senior Member, IEEE*

***Abstract*—In this paper, we propose a simple protocol for building heterogeneous unstructured peer-to-peer (P2P) networks. The protocol consists of two parts—the joining process and the rebuilding process. The basic idea for the joining process is to use a random walk to assist new incoming peers in selecting their suitable neighbors in terms of capacity and connectivity to achieve load-balancing. The rebuilding process specifies how the nodes should react when they lose links. In particular, we examine two representative schemes, namely the probabilistic-rebuilding scheme and the adaptive-rebuilding scheme. Furthermore, we provide a detailed analysis to investigate our proposed protocol under any heterogenous P2P environment. We prove that the topology structure of the P2P network depends heavily on the node heterogeneity. The analytical results are validated by the simulations. Our framework provides a guideline to engineer and optimize a P2P network in different respects under a heterogeneous environment. The ultimate goal of this paper is to stimulate further research to explore the fundamental issues in heterogeneous P2P networks.**

***Index Terms*—Capacity, heterogeneity, random walk, topology, unstructured P2P network.**

I. Introduction

PEER-TO-PEER (P2P) networks have become an important part of the Internet and many P2P applications are emerging. Distributed, unstructured P2P networks have been proposed to replace Napster-like centralized P2P architecture. There are many successful applications using unstructured networks such as Gnutella [1], KaZaA [2] and Skype [3]. In general, unstructured P2P networks have wide applicability and file-sharing application is one of the successful examples.

Building a good P2P topology is not a trivial task because there are many important issues such as:

Network Heterogeneity: P2P networks are very heterogeneous. For example, users' access bandwidths are actually very diverse, from 56 Kbps connections to a few Mbps connections. Moreover, there are many other factors determining a P2P system's performance. Users' local resources such as CPU power, available memory and hard disk space are also an important consideration for a P2P system design. Therefore, we define the term "node capacity distribution" to represent the heterogeneity of the users. Next, our question is how to build the P2P network based on the "capacity" of each user which is a critical step in load-balancing and providing a stable service to the users.

Scalability: P2P networks are very large in size and have no central agent. The P2P topology may not be completely known to the users. Also, for the bandwidth-demanding applications such as large-scale application-level multicast, the nodes should connect to the suitable neighbors in order to prevent overloading. In this case, how to find such nodes in a large P2P network with low overhead is an important question. We would like to employ a scalable, lightweight, distributed algorithm for building a capacity-aware P2P topology.

Therefore, our first objective in this paper is to propose a protocol to build the heterogeneous, unstructured P2P network. The main criterion in topology formation is based on the node capacity and degree. Our protocol exploits a random walk idea which is completely decentralized and requires low overhead.

P2P networks are extremely complicated and totally different from the traditional server-client architecture. Typically, P2P networks involve millions of users simultaneously where they can continually join and leave the networks without a predictable pattern. On the other hand, the heterogeneity of the P2P network is changing from time to time. Due to the large complexity, it is difficult to evaluate our protocol solely based on large-scale simulations or Internet experiments which are time consuming and costly. Therefore, our second objective is to develop a mathematical model to analyze our proposed topology-formation algorithm. The main point in the analysis is to model the heterogeneity of the network. Hence, based on our mathematical framework, we can easily examine a large-scale P2P topology structure built by our protocol under any heterogeneous environment. The results also help us to further engineer and optimize P2P protocols running atop the topology.

The paper is organized as follows. Section II provides background information about different topology formation protocols and related work. Section III presents our protocol. We then provide a comprehensive analysis for the proposed protocol in Section IV. The simulation results are presented in Section V. We discuss some applications by using our protocol in Section VI. Section VII discusses the applicability of our analytical results and some open questions in the P2P protocol design and modeling. Finally, Section VIII concludes our paper.

II. Related Work

In Gnutella, the nodes join the P2P network by connecting to m live nodes (a node is live if it is currently connected to

Manuscript received August 30, 2005; revised July 6, 2006, and November 27, 2006; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor R. Srikant. The work of D. H. K. Tsang was supported by the RGC Earmarked Research Grant 620306.

K.-W. Kwong was with the Hong Kong University of Science and Technology. He is now with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: kkw@seas.upenn.edu).

D. H. K. Tsang is with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (e-mail: eetsang@ece.ust.hk).

Digital Object Identifier 10.1109/TNET.2007.899026

1063-6692/$25.00 © 2008 IEEE

the P2P network). m is typically 3–5. This method is very ad hoc without any mathematical support. KaZaA and Gnutella2 [4] employ a super-peer topology which is a kind of adaptation technique. In this case, powerful users (such as high bandwidth users) form the backbone of a P2P network and most of the query traffic is processed by them. Therefore, low capacity users can be shielded from massive query traffic, making the unstructured systems more scalable. The important question in constructing two-tier P2P networks is how to select "super" nodes. There are many ad hoc approaches such as selecting "super" nodes based on their lifetime, memory and bandwidth.

Recently, some algorithms have been proposed for building unstructured P2P networks. In [28], the authors suggest a protocol to build low-diameter P2P networks by using a bootstrap server to coordinate the connections between peers. In [29], an algorithm is proposed to produce a topology with $\mathcal{O}(\log N/\log\log N)$ diameter where N is the number of peers in the network. Their idea is based on a random graph theory to construct the topology with a certain number of edges. As a result, the targeted diameter is achieved. Phenix [34] was recently proposed for building resilient unstructured P2P networks. First, it utilizes the idea of scale-free network to shorten the topology diameter. Phenix also has a built-in mechanism to counter intentional attacks. However, all these algorithms assume the P2P network is homogeneous meaning that each node has the same capacity which is not true in reality. Therefore, it is necessary to develop a protocol and analytical models for heterogeneous P2P networks.

In [9], the authors proposed a scalable Gnutella-like system, called Gia, by employing different algorithms such as caching, adaptive overlay, flow control and biased random walk search. They carried out different simulations to investigate the system's performance. In our work, we propose a random walk algorithm to build the unstructured P2P network based on each node's capacity and degree, and also analyze two different link rebuilding schemes. Thus our contribution is completely different from [9].

Our joining process, inspired by the Metropolis–Hastings algorithm ([17], [26]), is to assist the peers in selecting their neighbors with a high capacity per connectivity. We also further analyze two different rebuilding schemes. Our idea is totally different from the recent paper by Zhong *et al.* [37] which suggested to use the Metropolis–Hastings algorithm to realize a distance-based random long-link connection. Their main contribution is to prove the mixing time of the random walk under some special networks such as ring topology and power-law graph.

In our analysis, we use a continuum approach (a.k.a. fluid model) to analyze the evolution of the P2P network structure under our proposed topology-formation algorithm. Recently, [30] and [24] proposed a fluid model to analyze the performance of the BitTorrent-like networks. Fundamentally, our paper investigates a completely different problem from them and one of our contributions is to model the P2P network as a heterogeneous environment meaning that every peer's capacity follows some probability distribution.

III. Our Proposed Protocol

Our protocol aims to construct the P2P network by exploiting node heterogeneity to realize load-balancing, and hence the P2P network is self-organized into a certain structure such that high capacity nodes form the hubs to reduce the network diameter. The protocol consists of two processes, namely the joining process and the rebuilding process. We first describe our network model.

A. P2P Network Model

Heterogeneity among P2P users are the main concern in forming the topology. Many factors affect user's heterogeneity. First, each user has certain access link bandwidth, ranged from dial-up modem to ADSL to LAN connections. Second, user's local scarce resources such as CPU power and available memory are also a dominant factor on a P2P system's performance. Therefore, to characterize the heterogeneity of the peers, we define the generic term "node capacity", $\eta_i > 0$, which is considered as a combination of the access link capacity and available scarce resources of node i. We assume that η_i is chosen from a probability density function (p.d.f.) $\rho(\eta)$ which is called "node capacity distribution" in this paper. This p.d.f. can be either continuous or discrete. Therefore the node capacity distribution characterizes the global heterogeneity of the P2P network. Determining the value of node capacity depends on applications. For example, in a P2P streaming application, a node capacity can be mainly regarded as a node's uploading bandwidth because it plays an important role on the performance of such system. For a network coding system like [16], the peers are required to encode and decode information received from their neighbors. This may be a CPU-consuming task. Therefore, a node capacity should be considered as a combination of CPU power and access link bandwidth.

In this paper, we assume that the network bottleneck in a P2P system happens only at the access links of the peers. In addition, we do not consider any correlation between the P2P logical links (i.e., shared congested links) in the Internet cloud since the physical routing path, congestion situation and traffic pattern are dynamically changing over time, and all these make the analytical model highly complicated and intractable. Unless a large-scale Internet experiment is being carried out, the actual environment is very difficult to realistically model and simulate due to the lack of publicly available information such as the Internet topology, routing information, link bandwidth and traffic pattern. This simplified but well-accepted network model has been previously employed by other researchers (e.g., [16], [19], [25], [11], and [35]).

B. Joining Process

To describe our protocol, we use k_i to represent the degree of node i. Generally, every node prefers to connect to a high capacity node in order to achieve a better service from its neighbors. However, it is not enough to purely consider the neighbor's capacity as a connection criterion. At the same time, each node should not connect to the neighbors with a high degree as well. This is because a high degree node would handle a large workload. Therefore, our joining process considers two important metrics, node's capacity and degree, to make the connection decision. Also, we would like to introduce some randomness for the joining process so that each node can connect to other different nodes. Mathematically, we can formulate our joining process as follows. The probability π_i that node i is connected

by a new incoming node is directly proportional to its capacity and inversely proportional to its current degree, i.e.,

$$\pi_i = \frac{\frac{\eta_i}{k_i}}{\sum_{j\in L(t)} \frac{\eta_j}{k_j}}, \qquad i \in L(t) \tag{1}$$

where $L(t)$ denotes the set of all live nodes in the network at time t. Equation (1) means that the nodes should connect, with a high probability, to some neighbors with a large capacity-to-degree ratio. This capacity-to-degree ratio can be considered as a "normalized" node capacity which is used to achieve a generalized load-balancing in the P2P network.

It is impractical to use bootstrap servers to maintain global network information to achieve (1). Therefore, we employ the Metropolis–Hastings algorithm ([17], [26]) to achieve (1) distributively as follows.

We model a P2P network as a connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node set $\mathcal{V} = \{1, \ldots, n\}$ and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, with $(i,j) \in \mathcal{E} \Leftrightarrow (j,i) \in \mathcal{E}$. We assign a transition probability p_{ij} to each edge $(i,j) \in \mathcal{E}$ as follows:

$$p_{ij} = \frac{1}{k_i + 1} \min\left\{1, \frac{\eta_j k_i (k_i+1)}{\eta_i k_j (k_j+1)}\right\} \tag{2}$$

and create a self-loop at each node i with $p_{ii} = 1 - \sum_{(i,j)\in\mathcal{E}} p_{ij}$ such that the total transition probability is 1 (k_i and k_j do not count the self-loop). The edge's transition probability is used for random walk which is discussed in the following. By using this algorithm, each node i has to broadcast its capacity η_i and current degree k_i values to its neighbors such that the neighbors can use this information to assign a transition probability to their edges.

When a new node joins the network, it contacts m live nodes in the network. These m live nodes can be retrieved from a bootstrap server or a cached node list stored in the new node. Then, the new node issues m different walkers to these m nodes. Each walker is assigned a time-to-live (TTL) value, τ. This TTL value is equivalent to the number of iterations in the Metropolis–Hastings algorithm. The walker is forwarded from the current node to a neighboring node based on the edge transition probability in (2) and the walker's TTL is decremented by one after each forwarding. The new node connects to a node at which the walker stops (i.e., TTL reaches 0). If a walker stops at the node which is already connected by that new node, then the walker moves additionally δ steps. In this paper, we assume $\delta = 1$. This process repeats until the walker can find a node for the new node to join. However, this situation is very rare to happen if the network is large. In addition, each walker should send a keep-alive message to the new node. If the new node does not receive the keep-alive message for a period of time, it assumes that a walker is lost and issues another new walker to compensate for the lost one. It is noted that each node should maintain at least three links in order to ensure network connectedness.

We can easily verify that the Markov chain defined by (2) is: 1) *reversible*, i.e., $\pi_i p_{ij} = \pi_j p_{ji}$ for all $i, j \in \mathcal{V}$; 2) ***aperiodic***; and 3) ***irreducible***. Therefore, the sampled probability by the random walk converges to a steady-state distribution that is exactly equal to (1) when $\tau \to \infty$. According to our previous experience, a small TTL value $(\tau \sim 10)$ can obtain a good random mixing for the network with 50 000 nodes [20].

C. Rebuilding Process

When a node leaves the network, all of its neighbors i lose a link. To prevent network breakdown and node isolation, those nodes which lose a link rebuild r_i new link(s) to compensate for the lost one. This is called the rebuilding process.

When a node i tries to rebuild a link, it issues a walker with a TTL value, τ, to one of its neighbors randomly.[1] Then the walker traverses the network, the same as the node joining process. Finally, a new link is created by connecting node i and the node at which the walker stops. We particularly examine two representative rebuilding schemes as follows. (In the following, k_i^- denotes the degree of node i just after losing a link.) In general, if $r_i \leq 1$, it can also be interpreted as the probability of building a link. For simplicity, we assume $r_i \leq 1$ for the rest of the paper.

Probabilistic-Rebuilding Scheme: The nodes rebuild a link based on a probability r. Mathematically,

$$r_i = \begin{cases} 1 & k_i^- = 2 \\ r & k_i^- \geqslant 3 \end{cases} \tag{3}$$

for every node i. The threshold on $k_i^- = 2$ means that each node has to maintain at least three links to ensure network connectedness.

Adaptive-Rebuilding Scheme: The nodes should gradually not rebuild links when their degrees are getting large in order to prevent overloading. At the same time, each node should maintain at least m links such that the overlay service performance and reachability are not degraded. Therefore, this rebuilding process allows the nodes to make rebuilding decision adaptive by considering their current degrees. Mathematically,

$$r_i = \frac{m-1}{k_i^-} \tag{4}$$

for each node i.

We believe that our two rebuilding schemes are very representative and cover many possibilities in P2P systems. As we show later, the main feature of the rebuilding schemes is that they allow the degrees of each node to grow very slowly which is a good property because it can prevent the nodes from consuming a lot of resources such as memory, CPU power and bandwidth when connecting to many neighbors. The detailed analysis of our proposed schemes is shown in the following sections.

IV. Protocol Analysis

Network model can be classified into two main categories, namely static random graph and evolving networks. As for static random graph, the number of nodes and links are kept constant. This model was first suggested by Erdős and Rényi in 1959 [13]. This model has been used in analyzing P2P networks. However, due to its static behavior which is totally opposite to dynamic P2P networks, we believe that this model is not suitable for analyzing P2P networks. Another graph model is so-called evolving networks. The generation method, which is quite different from static random graph, is that the new nodes and links are added to the network over time. Thus the number of nodes in the network keeps increasing.

[1]The node can also send a walker to a random node in its cache node list.

However, neither model can represent the actual properties of P2P networks. Basically, three phases happen in P2P networks—a growing phase, a stabilizing phase and a decaying phase. In a growing phase, the number of nodes keeps increasing (i.e., the node incoming rate is larger than the node leaving rate), which happens during the transition from non-busy hours to busy hours because many users go online. However, the size of the network cannot increase indefinitely. When the network reaches a certain size, the node incoming rate would be roughly the same as the node leaving rate. As a result, the number of nodes remaining in the network would be roughly constant. This is called a stabilizing phase. After the peak hours, the network enters a decaying phase in which the size decreases as many users go offline (i.e., the node incoming rate is smaller than the node leaving rate) and finally reaches another stabilizing phase. These three network evolution phases have been observed in PPLive [5] which is the most popular IPTV system today [18]. Their measurement study shows that the duration of the stabilizing phase is clearly dominant over the other two phases. Similar results are also observed in the Gnutella network [33]. Therefore, it is worth studying and analyzing the P2P networks under the stabilizing phase.

Stabilizing Network Model: This model is inspired by the growing network model (e.g., [7], [12] and [32]). There are two key differences. First, our model assumes the network size to be constant. We believe that our model is suitable to analyze the steady-state behavior of a topology meaning that the P2P network size remains roughly constant over time. This is a common observation in a P2P network as discussed above. Second, our load-balancing protocol for constructing the topology is orthogonal to the preferential attachment model (e.g., [7]). Our analysis focuses on a large population regime.

Let $k(s,t,\eta_s)$ be the degree of node s at time t and η_s be its capacity. Assume the initial network is connected with N nodes. At each time step $t = 1,2,3,\ldots$, a new node is added into the network by connecting to m different live nodes based on our protocol where m is a system design parameter. Each node s is labeled by the time of its arrival $s = 1,2,3,\ldots \leq t$. For example, node 1 arrives in the P2P network earlier than node 2 and etc. At the same time, a randomly chosen node is removed from the network. Then each neighbor s of the departing node should re-establish r_s new link(s) to compensate for the lost link. Thus, the node arrival rate is equal to the node leaving rate, and hence the network size remains N nodes.

We can establish the following differential equation to describe the evolution of a node degree $k(s,t,\eta_s)$ by using the ideas of a continuum approach (e.g., see [7] and [12]):

$$\frac{\partial k(s,t,\eta_s)}{\partial t} = m\frac{\frac{\eta_s}{k(s,t,\eta_s)}}{Z} + (r_s-1)\frac{k(s,t,\eta_s)}{N} + \frac{\frac{\eta_s}{k(s,t,\eta_s)}}{Z}\sum_{i\in A(t)} r_i \qquad (5)$$

where Z is the normalization factor

$$Z = \sum_{\varsigma\in L(t)} \frac{\eta_\varsigma}{k(\varsigma,t,\eta_\varsigma)} \qquad (6)$$

and the initial condition is $k(s,s,\eta_s) = m$ because node s connects to the P2P network at $t = s$ by establishing m links.

The rationale behind (5) is explained as follows. At each time step, a new incoming node connects to node s with probability $\pi_s \propto \frac{\eta_s}{k(s,t,\eta_s)}$ (i.e., (1)), the rate of change of the degree of node s for the new incoming node is taken into account by the first term. Additionally, since a randomly chosen node departs, node s loses a link with probability $k(s,t,\eta_s)/N$. After losing a link, node s triggers a rebuilding process to rebuild r_s link(s). This effect is modeled by the second term. Moreover, for each departure event at time t, a set $A(t)$ of nodes lose a link. Since a randomly chosen node is removed from the network, thus $|A(t)| = \langle k(t)\rangle$ where $\langle k(t)\rangle$ is the average degree of the P2P network at time t.[2] Then the rebuilt links may connect to node s and hence this event is taken into account by the third term. Remark that (5) is a generalized formula for any rebuilding process, not only limited to the probabilistic-rebuilding scheme and the adaptive-rebuilding scheme.

A. Scaling Form of Solutions

In order to analyze the topology structure, we have to solve the differential equation of the node degree, i.e., (5). We use the idea of mean-field approach to seek the scaling form of the solution to (5), i.e.,

$$\kappa(\zeta,\eta) := k(s,t,\eta) \qquad (7)$$

where $\zeta = (s-t)/N$. Before proceeding, we need the following lemma.

Lemma 1: In a stabilizing network, let $\theta(s,t)$ be the probability that node s is still in the network at time t. Then,

$$\theta(s,t) = e^{\frac{1}{N}(s-t)}. \qquad (8)$$

Proof: We can establish the recurrent equation for $\theta(s,t)$ as follows:

$$\theta(s,t+1) = \theta(s,t)\left(1-\frac{1}{N}\right) \qquad (9)$$

Assume the incremental step from t to $t+1$ is very small, then $\theta(s,t)$ can be represented by the differential form

$$\frac{d\theta(s,t)}{dt} = -\frac{\theta(s,t)}{N}. \qquad (10)$$

By solving this differential equation with the initial condition $\theta(s,s) = 1$ (because node s connects to the network at $t = s$), the result follows. ∎

Now, the first step in solving (5) is to determine Z which is in terms of the random variable η. By using the mean-field approach, we use the average value of Z over $\rho(\eta)$ to substitute Z. Thus,

$$\langle Z\rangle = \langle Z_0(t)\rangle + \left\langle \sum_{s\in L(t)} \frac{\eta_s}{k(s,t,\eta_s)} \right\rangle \qquad (11)$$

$$\approx \left\langle \int_0^t \theta(s,t)\frac{\eta_s}{k(s,t,\eta_s)}\,ds \right\rangle \qquad (12)$$

$$= \int_{\eta_{\min}}^{\eta_{\max}} \eta\rho(\eta)\int_0^t \frac{\theta(s,t)}{k(s,t,\eta_s)}\,ds\,d\eta \qquad (13)$$

[2]We use the symbol $\langle\cdot\rangle$ to denote expectation.

where $\langle Z_o(t)\rangle$ denotes the normalization constant at time t due to the original N nodes. The second term on the right side of (11) describes the effect due to the arrival nodes after $t = 0$. As the stabilizing network evolves and $t \to \infty$, $\langle Z_o(t)\rangle \to 0$ because these original N nodes gradually leave the network, and hence $\langle Z_o(t)\rangle$ can be ignored to write. From (11) to (12), we use integration to approximate the summation and set the lower limit of the integration to be 0 instead of 1 for easier calculation. Moreover, we multiply $\theta(s,t)$ inside the integration of (12) in order to take the effect of the node departure into account. Then by the change of variable $\zeta = (s-t)/N$, $\langle Z\rangle$ can be further expressed as follows:

$$\langle Z\rangle \approx N\int_{\eta_{\min}}^{\eta_{\max}} \eta\rho(\eta)\int_{\frac{-t}{N}}^{0}\frac{e^{\zeta}}{\kappa(\zeta,\eta)}d\zeta d\eta \tag{14}$$

$$\underset{=}{t\to\infty} N\int_{\eta_{\min}}^{\eta_{\max}} \eta\rho(\eta)\int_{-\infty}^{0}\frac{e^{\zeta}}{\kappa(\zeta,\eta)}d\zeta d\eta. \tag{15}$$

For convenience, we define G as

$$G := \int_{\eta_{\min}}^{\eta_{\max}} \eta\rho(\eta)\int_{-\infty}^{0}\frac{e^{\zeta}}{\kappa(\zeta,\eta)}d\zeta d\eta \tag{16}$$

and hence we replace Z by $\langle Z\rangle = NG$ in (5). To find the value of G, we have to obtain $k(s,t,\eta)$ and then put it back to (16) which becomes a self-consistency equation and is solvable by means of numerical methods. To analyze our protocol and understand the evolution of the node degree, we need to solve $k(s,t,\eta)$ for each rebuilding scheme, and the results are presented in the following sections.

B. Analysis of Probabilistic-Rebuilding Scheme

We first consider the mean degree of the P2P network by using the probabilistic-rebuilding scheme, i.e., $r_i = r$ for all $i \in L(t)$. Obviously, we cannot arbitrarily set the value of r because an improper value would generate excessive links in the P2P network. Thus, we would like to find a suitable range of r for the stabilizing P2P networks.

Theorem 1: In a stabilizing network, the following condition:

$$r \leqslant 1 - \frac{m}{N-1} \tag{17}$$

is required for the probabilistic-rebuilding process. The resulting asymptotic mean degree of the P2P network is

$$\langle k\rangle = \frac{m}{1-r}. \tag{18}$$

Proof: Let $l(t)$ be the number of links in the network at time t. We treat $l(t)$ as a continuous variable. We can establish the differential equation of $l(t)$ for the stabilizing network as follows:

$$\frac{dl(t)}{dt} = m - |A(t)| + \sum_{i\in A(t)} r_i \tag{19}$$

where $A(t)$ is the set of nodes which lose a link in a departure event at time t and $|A(t)|$ denotes the number of nodes in set $A(t)$. On the right side of (19), the first term represents the increasing rate of the number of links in the network due to the new node arrival. The second and third terms denote the change of links in the network because of the node departure and the rebuilding process respectively. Since a random node is removed in each departure event, thus $|A(t)| = \langle k(t)\rangle$, where $\langle k(t)\rangle = \frac{2l(t)}{N}$ is the mean degree at time t. Also $r_i = r$, $\forall i \in A(t)$. Therefore, (19) can be expressed as

$$\frac{dl(t)}{dt} = m - \frac{2l(t)}{N}(1-r). \tag{20}$$

Suppose $0 \leq r < 1$, the steady state of $l(t)$ can be obtained by letting $\frac{dl(t)}{dt} = 0$ and hence

$$l(\infty) := \lim_{t\to\infty} l(t) = \frac{Nm}{2(1-r)}. \tag{21}$$

Since the size of the P2P network is fixed, the number of overlay links must be less than or equal to $\frac{1}{2}(N^2-N)$ which represents a fully connected network. Therefore

$$l(\infty) = \frac{Nm}{2(1-r)} \leqslant \frac{1}{2}(N^2-N) \tag{22}$$

$$\Rightarrow r \leqslant 1 - \frac{m}{N-1}. \tag{23}$$

The asymptotic mean degree of the network is

$$\langle k\rangle = \frac{2l(\infty)}{N} = \frac{m}{(1-r)}. \tag{24}$$

■

Theorem 1 provides a foundation to choose the value of r for the probabilistic-rebuilding scheme. Note that from (17) the upper bound value of r is very close to 1 because $m \ll N$.

In order to analyze the probabilistic-rebuilding scheme, we need to solve $k(s,t,\eta_s)$. First, the asymptotic mean degree of the P2P network under this rebuilding scheme is given by (18), i.e., $\langle k\rangle = \frac{m}{1-r}$. It is noted that we neglect the rebuilding threshold in (3). This assumption greatly reduces the complexity of the solution but does not introduce a significant error as shown in the simulations. To solve for $k(s,t,\eta_s)$, we substitute $r_s = r_i = r$ and $|A(t)| = \langle k\rangle = \frac{m}{1-r}$ into (5) which can then be solved into the following solution:

$$k(s,t,\eta_s) = \sqrt{\frac{1}{G(1-r)}\left(\frac{m\eta_s}{1-r} - \Delta e^{\frac{2(1-r)}{N}(s-t)}\right)} \tag{25}$$

where $\Delta := \frac{m\eta_s}{1-r} - G(1-r)m^2$ and G is the non-zero positive constant satisfying the self-consistency equation in (16).

In the probabilistic-rebuilding scheme, every node degree tends to a bounded equilibrium depending on its capacity. More importantly, these equilibrium values are independent of network size N which is a very nice property. We can summarize this result into the following theorem.

Theorem 2: Suppose a node s always stays in the network, then the degree $k(s,t,\eta_s)$ of that node converges to the value $k^*(\eta_s)$ as $t \to \infty$. Mathematically,

$$k^*(\eta_s) = \max\left\{\frac{1}{1-r}\sqrt{\frac{m\eta_s}{G}}, 3\right\}. \tag{26}$$

Proof: The result can be proved by letting $(s-t) \to -\infty$ in (25). The lower limit is due to the threshold of (3). ■

Remark that this theorem can also be easily proved by showing that the rate equation of the node degree has the attracting equilibrium which is identical to (26).

From this theorem, we show that the equilibrium value of a node degree changes accordingly as a node's capacity changes. Thus our protocol can adapt to the fluctuation of a node's capacity and prevent the nodes from overloading.

It is noted that the convergence speed of the node degree depends on the network size. The bigger the network size, the slower the convergence speed. For example, if the network size is large, the probability of a node being connected is small due to the sampling nature of our algorithm. This property can allow the node degrees to grow slowly and prevent them from overloading.

Under this rebuilding scheme, the equilibrium degree of a node may be smaller than the initial number of connections (i.e., m) if its capacity is too small. This is because the rate of losing neighbors due to node departure is larger than "gaining" neighbors since the node capacity is too small. This is another property of our algorithm to prevent low-capacity nodes from overloading.

However, this behavior may not be suitable for some applications which require all nodes maintain at least a certain number of neighbors. Therefore, we need the adaptive-rebuilding scheme which is analyzed in the following section.

C. Analysis of Adaptive-Rebuilding Scheme

The adaptive-rebuilding scheme is different from the probabilistic-rebuilding scheme. Under this scheme, each node maintains at least m neighbors, and carries out the rebuilding process depending on its current degree. For example, the nodes with a high degree have a strong tendency not to rebuild the lost links to prevent node overloading. However, this rebuilding scheme makes the scenario more complicated to analyze than the previous scheme. Instead of exact analysis, we try to find an approximation to understand the P2P network behaviors under this scheme.

Theorem 3: The asymptotic mean degree of the P2P network under the adaptive rebuilding scheme is $\langle k \rangle \approx 2m$.

Proof: The rate of change of links $l(t)$ in the network at time t can be expressed as

$$\frac{dl(t)}{dt} \approx m - \langle k(t) \rangle + \sum_{i \in A(t)} \frac{m}{k(i, \eta_i, t)} \tag{27}$$

$$\approx m - \langle k(t) \rangle + \sum_{i \in A(t)} \frac{m}{\langle k(t) \rangle} \tag{28}$$

$$= 2m - \langle k(t) \rangle \tag{29}$$

$$= 2m - \frac{2l(t)}{N}. \tag{30}$$

It is noted that in (27), since we do not know $k(i, \eta_i, t)$ yet, it is approximated by $\langle k(t) \rangle$ which is the mean degree of the P2P network at time t. From (28) to (29), we use the fact that $|A(t)| = \langle k(t) \rangle$ because each departing node is randomly selected from the network. By using (30) and following the steps similar to Theorem 1, the result is obtained easily. ■

To analyze the degree evolution in the adaptive-rebuilding scheme, we approximately solve the differential equation of $k(s, t, \eta_s)$ (i.e., (5)). The main step of the approximation is to solve a simpler differential equation with a concave solution to approximate the original one. As before, we assume the scaling form solution, i.e., (7), holds for the adaptive rebuilding scheme. Thus we replace Z by $\langle Z \rangle = NG$. Then the rate equation of $k(s, t, \eta_s)$ can be written as

$$\frac{\partial k(s,t,\eta_s)}{\partial t} \approx \left\{ m + \sum_{i \in A(t)} \frac{m}{k(i,t,\eta_i)} \right\} \frac{\eta_s}{k(s,t,\eta_s)NG} + \frac{m - k(s,t,\eta_s)}{N} \tag{31}$$

$$\approx \frac{2m\eta_s}{k(s,t,\eta_s)NG} + \frac{m - k(s,t,\eta_s)}{N} \tag{32}$$

where we use $k(i, t, \eta_i) \approx \langle k \rangle \approx 2m$.

Now we can use (32) to find the equilibrium of a node degree approximately. The result is summarized in the following theorem.

Theorem 4: Suppose a node s always stays in the network, then the degree $k(s, t, \eta_s)$ of that node converges to the value $k^*(\eta_s)$ as $t \to \infty$. Mathematically,

$$k^*(\eta_s) \approx \frac{1}{2} \left\{ m + \sqrt{m^2 + \frac{8m\eta_s}{G}} \right\}. \tag{33}$$

Proof: We consider (32). For convenience, we define

$$g(k) := \frac{2m\eta_s}{k(s,t,\eta_s)NG} + \frac{m - k(s,t,\eta_s)}{N}. \tag{34}$$

The equilibrium value of $k(s, t, \eta_s)$ can be obtained by finding the root of (34) (i.e., $g(k) = 0$). Obviously, $g(k) = 0$ is a quadratic equation with two real roots. It is not too difficult to show that one root is positive and the other one is negative. The negative root has to be rejected since it is impossible in our protocol. The positive root, denoted by k^*, is

$$k^* = \frac{1}{2} \left\{ m + \sqrt{m^2 + \frac{8m\eta_s}{G}} \right\}. \tag{35}$$

Since $g'(k)$ is negative for all $k(s, t, \eta_s) > 0$. Therefore, $g'(k^*) < 0$ meaning that k^* is the attracting equilibrium. Hence $k^*(\eta_s) := \lim_{t \to \infty} k(s, t, \eta_s) \approx k^*$. ■

In this rebuilding scheme, each node maintains at least m neighbors and rebuilds links based on its current degree. This behavior is totally different from the probabilistic-rebuilding scheme. At the same time, each node still has equilibrium degree which is also independent of the network size.

In order to obtain the value of the degree equilibrium, we need to find G which depends on $k(s, t, \eta_s)$. However, the solution to (32) is very cumbersome and difficult to analyze. Thus we use the solution $\tilde{k}(s, t, \eta_s)$ of the following simpler differential equation to approximate the solution of (32):

$$\frac{\partial \tilde{k}(s,t,\eta_s)}{\partial t} = \frac{2m\eta_s\beta_1}{\tilde{k}(s,t,\eta_s)NG} - \frac{\tilde{k}(s,t,\eta_s)\beta_2}{N} \tag{36}$$

where β_1 and β_2 are non-zero positive constants independent of $\tilde{k}(s,t,\eta_s)$ and t. The initial condition of (36) is $\tilde{k}(s,s,\eta_s)=m$. We need to set up two conditions in order to find β_1 and β_2. It is obvious that (36) has an attracting equilibrium. Thus the first condition is that this equilibrium should be the same as (33), i.e.,

$$\frac{1}{2}\left\{m+\sqrt{m^2+\frac{8m\eta_s}{G}}\right\}=\sqrt{\frac{2m\eta_s\beta_1}{G\beta_2}}. \tag{37}$$

It is noted that $k(s,t,\eta_s)$ and $\tilde{k}(s,t,\eta_s)$ are both strictly concave (before reaching equilibrium) which can be shown by differentiating (32) and (36) with respect to t once and observing that $\frac{\partial k(s,t,\eta_s)}{\partial t}$ and $\frac{\partial \tilde{k}(s,t,\eta_s)}{\partial t}$ are both positive (before reaching equilibrium). Thus, the second condition to make the approximation accurate is to assume that the initial rates of change of node degree in (32) and (36) are the same, i.e.,

$$\left.\frac{\partial k(s,t,\eta_s)}{\partial t}\right|_{t=s}=\left.\frac{\partial \tilde{k}(s,t,\eta_s)}{\partial t}\right|_{t=s}. \tag{38}$$

Solving (37) and (38), we obtain

$$\beta_1=\frac{(k^*)^2}{(k^*)^2-m^2} \tag{39}$$

$$\beta_2=\frac{2m\eta_s}{G\{(k^*)^2-m^2\}} \tag{40}$$

where k^* means (35). Then the solution of (36) can be solved as

$$\tilde{k}(s,t,\eta_s)=\sqrt{\frac{1}{G\beta_2}\left(2m\eta_s\beta_1-\upsilon e^{\frac{2\beta_2}{N}(s-t)}\right)} \tag{41}$$

where $\upsilon := 2m\eta_s\beta_1 - G\beta_2 m^2$. Therefore, we use $\tilde{k}(s,t,\eta_s)$ to approximate $k(s,t,\eta_s)$, i.e., $k(s,t,\eta_s)\approx\tilde{k}(s,t,\eta_s)$. Hence, the value of G can be found by substituting $\tilde{k}(s,t,\eta_s)$ with the change of variable $\zeta=(s-t)/N$ into (16), resulting in a self-consistency equation in terms of unknown G only so that it can be solved by a numerical method.

Let us investigate the difference between G's in both rebuilding schemes. It is noted that in general

$$\langle Z\rangle=\left\langle\sum_{s\in L(t)}\frac{\eta_s}{k(s,t,\eta_s)}\right\rangle \tag{42}$$

$$\approx\sum_{s\in L(t)}\frac{\langle\eta\rangle}{\langle k\rangle} \tag{43}$$

$$=N\frac{\langle\eta\rangle}{\langle k\rangle}. \tag{44}$$

Therefore, by combining the above result with (15), we get $G\approx\langle\eta\rangle/\langle k\rangle$. To prevent ambiguity, G is rewritten as $G_p(r)$ and G_a for the probabilistic rebuilding scheme with parameter r and the adaptive rebuilding scheme respectively. Let $\langle k\rangle_a$ be the mean degree of the P2P network under the adaptive rebuilding scheme. Then,

$$G_a\approx\frac{\langle\eta\rangle}{\langle k\rangle_a}\approx\frac{\langle\eta\rangle}{2m}=\frac{\langle\eta\rangle}{\frac{m}{1-1/2}}\approx G_p\left(\frac{1}{2}\right). \tag{45}$$

As a result, the value of G_a is roughly equal to the scenario of the probabilistic rebuilding scheme with $r=0.5$.

D. Analysis of Diameter

Average node-to-node distance[3] is an important parameter in P2P networks. This information can be used to predict the network performance for the protocol design. We can use the following formula, derived in [14], to calculate the diameter D of the P2P network

$$D=\frac{\ln(\langle k^2\rangle-\langle k\rangle)-2\langle\ln k\rangle+\ln N-\varepsilon}{\ln\left[\frac{\langle k^2\rangle}{\langle k\rangle}-1\right]}+\frac{1}{2} \tag{46}$$

where $\varepsilon\simeq 0.5772$ is the Euler's constant. Based on our analysis result, for a given node capacity distribution and network size, we can easily estimate the suitable value of m (i.e., number of initial connections) such that the diameter is bounded by some specified value or system requirement.

Before using (46) to calculate the network diameter, we need to know $\langle k\rangle$, $\langle k^2\rangle$ and $\langle\ln k\rangle$. The mean degree, $\langle k\rangle$, is already known for each rebuilding scheme. $\langle k^2\rangle$ and $\langle\ln k\rangle$ can also be calculated easily. For example, $\langle f(k)\rangle$, where $f(\cdot)$ is some function, can be evaluated by using the following lemma.

Lemma 2: The value of $\langle f(k)\rangle$ under the stabilizing network can be evaluated as follows:

$$\langle f(k)\rangle:=\frac{1}{N}\left\langle\sum_{s\in L(t)}f(k(s,t,\eta_s))\right\rangle \tag{47}$$

$$\approx\int_{\eta_{\min}}^{\eta_{\max}}\rho(\eta)\int_{-\infty}^{0}e^{\zeta}f(\kappa(\zeta,\eta))d\zeta d\eta \tag{48}$$

where $\kappa(\zeta,\eta)$ is the form of (7).

Proof: We use the same technique as finding $\langle Z\rangle$ presented in Section IV-A for proving the above result. We do not show the proof again for brevity. ■

It is noted that we take the expectation with respect to the random variable η on the summation sign in (47).

In general, by using this lemma, we can easily calculate any degree moment for any given node capacity distribution. Since we can calculate any degree moment from the above lemma, thus it is equivalent to obtain the degree distribution of a topology.

V. Simulation

We simulate the P2P networks as follows. Each node i is assigned a non-zero capacity η_i randomly chosen from a p.d.f. $\rho(\eta)$. The nodes join the network at a Poisson arrival rate λ nodes per unit time. The inter-arrival time of the node departure events is exponentially distributed with mean $1/\mu$ ($\mu=0$ means no node departure). When a node departure event happens, a node is randomly removed from the network to model the node departure. Initially, the network grows from a small number of nodes $N_{\text{init}}=5$ by using $\lambda=1$ and $\mu=0$ until the network size reaches the value of N. Then we set $\lambda=\mu=1$ such that the network size maintains roughly at N in the stabilizing phase. During this phase, the simulation runs for 120 000 node arrivals, unless specified otherwise, to ensure the system reaches the steady state. Without loss of generality, we assume that the time unit is in seconds. The values of λ and μ determine

[3]In this paper, we use the terms, "network diameter" and "average node-to-node distance" interchangeably.

the rate of a node degree growth. For larger λ and μ, each node degree converges to equilibrium quicker. However, the actual values of λ and μ do not affect our analytical results.

In all the simulations, when a new node joins the network, it contacts a bootstrap server to randomly get m different nodes from the network. Then the new node issues one walker to each of these m nodes. The walkers traverse the network, with TTL τ, based on our proposed algorithm. We let $m = 5, \tau = 10$. We run each simulation 30 times, and report the average results. We have also tried the biased bootstrap server meaning that every new node randomly obtains m nodes from the last 100 incoming nodes only. The simulation results are roughly the same, so we do not repeat them here.

A. Node Capacity Distributions

In our simulations, we particularly explore two different node capacity distributions as follows.

Power-Law (PL) Capacity: This distribution is defined as follows:

$$\rho(\eta) = d\eta^{-3} \tag{49}$$

for $1 \leqslant \eta \leqslant 1000$, where d is the normalization constant. The node capacity is a continuous random variable in this case.

Discrete Gia Capacity: According to [31], the Internet access bandwidth of Napster's users is very heterogeneous. Measurement information has been used in [9] to model the heterogeneous P2P networks. The capacity distribution used in [9] can be modeled as

$$\begin{aligned}\rho(\eta) = {} & 0.2\delta(\eta - 1) + 0.45\delta(\eta - 10) \\ & + 0.3\delta(\eta - 100) + 0.049\delta(\eta - 1000) \\ & + 0.001\delta(\eta - 10\,000)\end{aligned} \tag{50}$$

where $\delta(\cdot)$ is the delta function.

In the following two sections, the power-law capacity distribution (i.e., (49)) and the discrete Gia capacity distribution (i.e., (50)) are labeled by "PL" and "Discrete" respectively.

B. P2P Network Diameter

The network diameter is an important metric for different overlay services. For example, Gnutella-like systems prefer a low diameter network because it can shorten the search time and discover more peers. It is well known that the diameter of random graphs scales as $\mathcal{O}(\log N)$ where N is the size of the graph [8]. However, this result does not tell us how peer heterogeneity (i.e., node capacity distribution) affects the network diameter. We are interested in how different node capacities influence the network diameter. We employ (46) to calculate the diameter of the P2P network based on our analytical results.

In Fig. 1(a), we show that the analytical results for the diameters of the networks with the probabilistic-rebuilding scheme $(r = 0.5)$ match very well with the simulation results. Our analytical framework can be used to predict the network diameter under any given node capacity distribution. In our examples, the network diameter under the discrete Gia capacity distribution is much shorter than the one in the PL capacity distribution. The main reason is that our protocol makes the highest capacity nodes (e.g., $\eta = 10\,000$) to connect to more neighbors. As a result, these highest capacity nodes "self-organize" into the

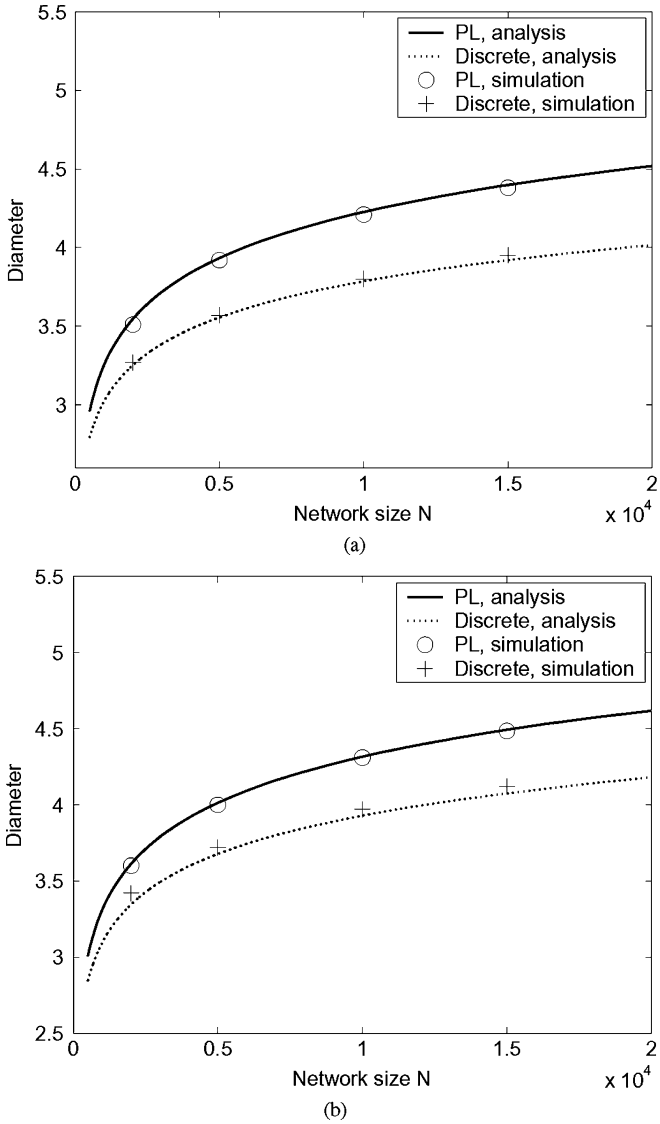


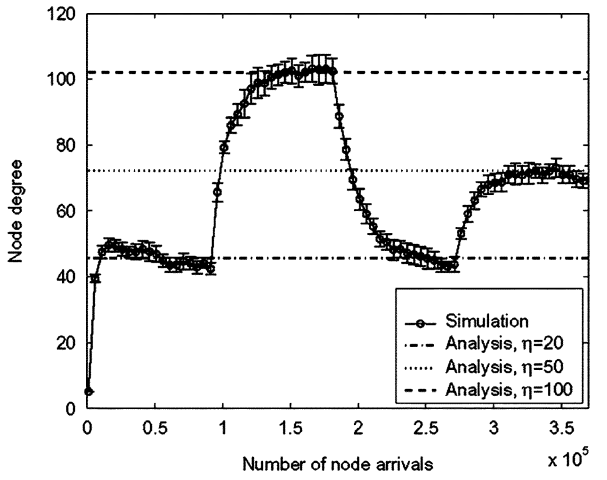
Fig. 1. Diameters of the stabilizing P2P networks: (a) Probabilistic-rebuilding scheme $(r = 0.5)$. (b) Adaptive-rebuilding scheme.

"hubs" of the networks, and hence the diameter can be greatly reduced through those "hubs". However, in the PL capacity, it is extremely rare to produce high capacity nodes (e.g., $\eta > 500$), because most of the nodes have capacity $\eta < 10$. In Fig. 1(b), we show the network diameters under the adaptive-rebuilding scheme. Our approximation expressed by (41) works very well and the analytical result matches with the simulations. It is noted that the variance of the measured diameters is very small, so we do not show the confidence interval for simplicity.

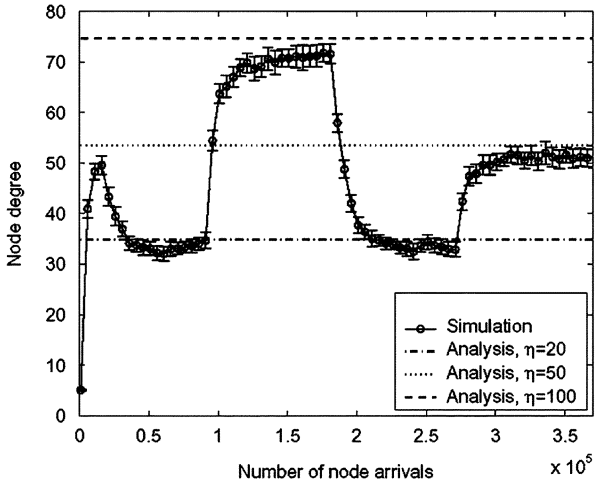
To conclude the simulations of measuring diameter, we suggest that the node capacity distribution should be considered when designing and simulating P2P protocols. This is because, as shown in our simulations, the diameter depends heavily on the node capacity and the results turn out to be very different in the "PL" and "Discrete" scenarios.

C. Node Degree Equilibrium

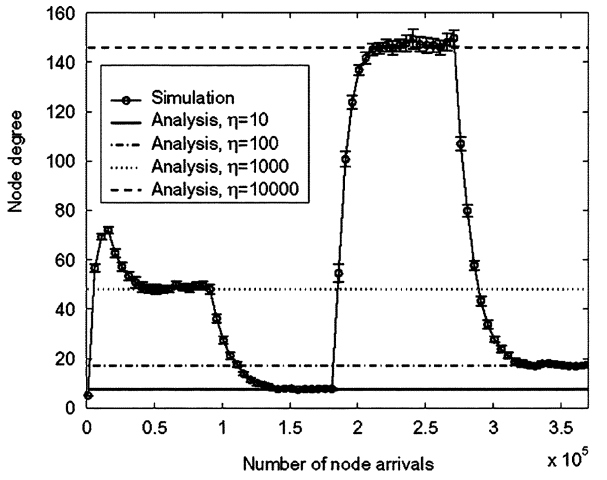
We have analytically shown that each node degree would eventually converge to a steady-state value, which depends on



(a)



(b)



(c)

Fig. 2. Node A's degree evolution in the stabilizing P2P network $N = 15\,000$: (a) Probabilistic-rebuilding scheme ($r = 0.5$) under the PL node capacity distribution. (b) Adaptive-rebuilding scheme under the PL node capacity distribution. (c) Adaptive-rebuilding scheme under the discrete Gia node capacity distribution. 95% confidence intervals are plotted in the simulations as well.

its capacity. In order to analyze the node evolution, we particularly monitor the 1000th incoming node (we call it node A) which is assumed to always stay in the network. Under the PL (resp. discrete Gia) node capacity distribution, the initial capacity η_A of node A is 20 (resp. 1000). Afterwards, node A changes its capacity to 100, 20 and 50 (resp. 10, 10 000 and 100) for each period of 90 000 node arrivals. This is used to model the capacity fluctuation of node A. In Fig. 2(a), we show the degree of node A under the probabilistic-rebuilding scheme ($r = 0.5$) and the analysis is predicted by (26). In Figs. 2(b) and (c), we plot the node A's degree under the PL and discrete Gia node capacity distributions respectively with the adaptive-rebuilding scheme and the analytical results are calculated from (33). In these simulations, we let the value of N to be 15 000.[4] In all these results, we show that our analysis matches very well with the simulations. The equilibrium degree changes accordingly as the node capacity varies. When node A's capacity is higher than other neighboring nodes, the random walkers have a larger chance to be "attracted" to node A. As a result, more links would be connected to it. On the contrary, if node A's capacity is low, the random walkers have a smaller chance to traverse node A, and hence the degree of node A is kept small. Therefore, from our analysis and simulations, we show that our algorithm can adapt to the fluctuation of node capacity.

VI. Applications

Our algorithm can be considered as a generic framework to construct P2P networks by taking capacity and load-balancing into account. We briefly outline two possible applications of our protocols in the following.

P2P Resource-Sharing System: Our protocol can be used to build the unstructured P2P systems such as KaZaA-like and Gnutella-like networks. By using our protocol, high-capacity peers would become a "hub" node, meaning that they connect to many neighbors. In this case, the network diameter is reduced due to those "hub" nodes. We can still employ our algorithm with a minor modification to construct KaZaA-like two-tier topology. For example, both new super-nodes and ordinary-nodes can follow our protocol to connect to the network. The starting point of the random walk should be a super-node. The only modification is that we set the transition probability of the links connecting to the ordinary nodes to be zero, because no node is allowed to connect to the ordinary nodes. Thus, in this case, our analysis results represent the super-peer topology.

P2P Live Video-Streaming System: The P2P approach is a promising way to deliver live video over the Internet without the support of IP multicast. Since video streaming is a bandwidth-demanding application, it is important to exploit the network heterogeneity to deliver the video. For example, high-capacity users should connect to and stream a video to more peers. As a result, the network diameter would be shorter meaning that the video streaming delay is improved. Thus, our protocol should be employed such that each peer is connected to the high capacity-to-degree-ratio neighbors in order to achieve load-balancing and receive a better streaming quality. Furthermore, the adaptive-rebuilding scheme can be used such that each peer maintains at least m neighbors where m is a pre-defined value. As a result, a video quality is assured. We believe that the ideas

[4]In Figs. 2(a), (b) and (c), there is a hump when the number of node arrivals equals 15 000. This is because, before the network size reaches 15 000, the network still behaves as a growing phase and hence the network behavior is totally different from a stabilizing phase when the network size reaches 15 000.

of CoolStreaming [36], network coding [16] and a BitTorrent-like algorithm can be used on top of the topology created by our algorithm to efficiently deliver live-video content.

VII. Discussion

In a heterogeneous environment consisting of millions of users, it is not possible to use a centralized approach to store the information of every node, such as node capacity and connectivity, for constructing the P2P topology due to the scalability problem. It is very natural to seek a distributed and lightweight approach to tackle this problem. Therefore, in this paper, we propose a very simple random walk protocol to form the P2P overlay network. Our idea is based on the Metropolis–Hastings algorithm which has been successfully employed in different fields such as statistics, physics and computer science. We believe that our protocol can be easily implemented in a large-scale P2P system because of its simplicity.

Furthermore, we propose a very detailed mathematical model to analyze our P2P topology-formation protocol. Our analytical model generalizes every possible heterogeneous environment by using the concept of "node capacity distribution" which is a probability density function characterizing how heterogeneous the P2P network is.

For example, the node capacity distribution, $\rho(\eta) = \delta(\eta - a)$ for some $a > 0$, represents a homogeneous network in which every node has the same capacity. Additionally, we can use our model to analyze some extreme P2P environments such as a star-like network in which there is a small portion of powerful nodes with a high capacity and degree, e.g., $\rho(\eta) = p_1\delta(\eta - a_1) + p_2\delta(\eta - a_2)$ where $p_1 \gg p_2 > 0$, $p_1 + p_2 = 1$ and $a_2 \gg a_1 > 0$. Due to the flexibility of our model, we can easily examine the structure of the P2P network under any heterogeneous environment.

Based on our analytical framework, we can formulate some optimization problems to further improve the P2P networks. Let us consider the following example. From our analysis, we know that the network diameter heavily depends on the peer heterogeneity. Therefore, it is important to design a suitable value of m (i.e., initial number of connections for a new node) such that the network diameter is bounded by some value larger than or equal to 1, say θ, under different heterogeneous environments in which the node capacity distributions are denoted by $\rho_1, \rho_2, \ldots, \rho_n$ (e.g., ρ_i represents the node capacity distribution for time period i and n denotes the number of time periods). Bounding the network diameter can ensure a better guarantee such as search and delivery delay in a P2P application. At the same time, we would like to keep m as small as possible so that the mean degree of the P2P network can be kept small. This is because if m is large, it may create a lot of traffic and signaling overhead in the network. Flooding search is an example. Thus, a simple optimization problem can be formulated as

Minimize m
Subject to
1) $\sum_{i=1}^{n} \alpha_i D(\rho_i, m) \le \theta$
2) $m \in \mathbb{Z}^+$

where $\alpha_i > 0$ is a weighting factor such that $\sum_{i=1}^{n} \alpha_i = 1$, $D(\rho_i, m)$ is the network diameter calculated by (46) given the node capacity distributions ρ_i $(i = 1, \ldots, n)$ and m. By solving this optimization problem, we can come up with the optimal value of m such that the network diameter requirement is satisfied.

We can also analyze other properties of a topology constructed by our protocol. For example, network robustness is a critical issue in network design. Recently, Cohen *et al.* [10] provided an analytical approach to calculate the critical threshold for network fragmentation based on the percolation theory. The critical threshold p_c representing the minimum fraction of the nodes to be removed randomly to fragment a network can be calculated in terms of $\langle k \rangle$ and $\langle k^2 \rangle$. From Lemma 2, any degree moment can be found and hence p_c can be obtained. From the engineering point of view, we can tune the value of m in order to achieve a system robustness requirement for a given set of node capacity distributions.

The rebuilding process is also very important for P2P system design. Our analysis can also be extended to cover more situations. Taking Gnutella as an example, there are many software implementations (e.g., Mutella and BearShare) and release versions. They may use their own rebuilding protocols which are different from others. Thus, based on our framework, it is possible and interesting to investigate the behavior of a topology under this "mixed-software" environment.

The spreading of viruses and polluted files over a P2P network has become a hot and important topic (e.g., [15], [22]). Traditionally, SIR and SIS models (e.g., [6]) have been used extensively to study many transmitting diseases. It is well known that the structure of a topology heavily interacts with the propagation pattern of a virus (e.g., [27]). Moreno *et al.* [27] analytically derived the epidemic threshold, λ_c, for a general topology based on the SIR model. Similarly, λ_c can also be analytically calculated by using our framework. Generally speaking, we believe that our analysis has a wider applicability than what we present in this paper.

Recently, we have extended the joining criterion of (1) to a more generic framework [21] as follows. The probability π_i that node i is connected by a new incoming node is proportional to its capacity and inversely proportional to its current degree in a nonlinear manner controlled by $\alpha > 0$ and $\beta > 0$, i.e.,

$$\pi_i = \frac{\frac{\eta_i^\alpha}{k_i^\beta}}{\sum_{j \in L(t)} \frac{\eta_j^\alpha}{k_j^\beta}}, \quad i \in L(t) \tag{51}$$

where $L(t)$ denotes the set of all live nodes in the network at time t. By tuning the values of α and β, they are analogy to transform the underlying node capacity distribution and control the growth of node degree respectively. As a result, our topology formation protocol becomes more flexible in terms of load-balancing or achieving other special requirements (e.g., link-level homogeneity [23]) for a spectrum of P2P applications running on top of the topology.

In this paper, we have not addressed the following two issues.

First, in our algorithm, each walker traverses the network with TTL steps. However, we have not considered how large TTL should be to achieve "good mixing". This topic has been discussed widely in the literature. In general, TTL should be large enough (e.g., three times of the network diameter) to mix the walker into the network. In our simulations, we set TTL equal to 10, and the results match very well with the analytical model.

In general, how the node capacity distribution would affect the walker mixing time is still unknown.

Second, node locality is another important factor in P2P networks. What is the effect of node locality on the P2P network structure such as connectedness and diameter? This question is related to the physical network and the locality distribution of the users. One obvious question is how to combine these two factors, locality and capacity, together. This is because a "close" neighbor (e.g., short ping delay) does not necessarily have enough bandwidth to serve our requests. Thus, the question becomes how to tradeoff between these two factors in order to maintain good properties for the topology such as low diameter and high robustness. More effort is required to develop an analytical model to study this phenomenon.

VIII. Conclusion

In this paper, we first propose a capacity-aware protocol to build the unstructured P2P networks to achieve load-balancing in a heterogeneous environment. Our protocol, inspired by the Metropolis–Hastings algorithm, is fully distributed. Only a node's neighboring information is used in searching out the nodes with a high capacity per connectivity in the P2P network. Our random walk algorithm is extremely simple and easy to implement. The protocol overhead is very low. Moreover, no specially-designed bootstrap server is required to support our algorithm because every new incoming node just needs some live nodes as a starting point of the random walk. Therefore, the workload, complexity and dependence of a bootstrap server can be greatly reduced.

Moreover, our protocol can utilize high-capacity nodes to produce a 'hub' effect so as to reduce network diameter which can provide advantages for different overlay applications such as P2P file-sharing systems and live video streaming. At the same time, our protocol also considers the connectivity of each node in order to effectively prevent node overloading.

We also introduce two representative rebuilding schemes namely the probabilistic-rebuilding scheme and the adaptively-rebuilding scheme. We believe that our two rebuilding schemes cover the main possibilities in the P2P system design. Furthermore, we provide a comprehensive analysis to study the performance of our proposed algorithm such as network diameter and degree evolution under any heterogeneous environment. The analytical results are validated by the simulations. Our analytical models are mathematically tractable and easy to extend to analyze other P2P topology-formation protocols. More importantly, our results point out that the structure of the P2P network is mainly influenced by the node capacity distribution. Different capacity distributions can result in different topology structures. We strongly believe that this point should be taken into account when designing and optimizing P2P protocols in order to exploit the overlay's properties.

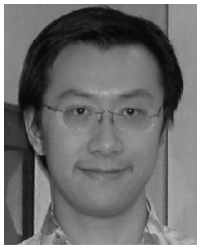
Acknowledgment

The authors would like to thank the anonymous reviewers for their constructive comments that improved the presentation of this paper.

References

[1] Gnutella. [Online]. Available: http://www.gnutella.com
[2] KaZaA. [Online]. Available: http://www.kazaa.com
[3] Skype. [Online]. Available: http://www.skype.com
[4] Gnutella2. [Online]. Available: http://en.wikipedia.org/wiki/Gnutella2
[5] PPLive. [Online]. Available: http://www.pplive.com
[6] F. Ball, T. Britton, and O. Lyne, "Stochastic multitype epidemics in a community of households: Estimation and form of optimal vaccination schemes," *Mathematical Biosciences*, vol. 191, no. 1, pp. 19–40, 2004.
[7] A.-L. Barabási, R. Albert, and H. Jeong, "Mean-field theory for scale-free random networks," *Physica A*, vol. 272, no. 1, pp. 173–187, 1999.
[8] B. Bollobás, *Random Graphs*, 2nd ed. Cambridge, U.K.: Cambridge University Press, 2001.
[9] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, "Making Gnutella-like P2P systems scalable," in *Proc. ACM SIGCOMM*, 2003.
[10] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin, "Resilience of the Internet to random breakdowns," *Phys. Rev. Lett.*, vol. 85, no. 21, pp. 4626–4628, Nov. 2000.
[11] Y. Cui, Y. Xue, and K. Nahrstedt, "Max-min overlay multicast: Rate allocation and tree construction," in *Proc. IWQoS*, 2004.
[12] S. N. Dorogovtsev and J. F. F. Mendes, "Scaling properties of scale-free evolving networks: Continuous approach," *Phys. Rev. E*, vol. 63, no. 5, p. 056125, Apr. 2001.
[13] P. Erdős and A. Rényi, "On random graphs," *Publicationes Mathematicae*, vol. 6, pp. 290–297, 1959.
[14] A. Fronczak, P. Fronczak, and J. A. Holyst, "Average path length in random networks," 2002 [Online]. Available: http://arxiv.org/abs/cond-mat/0212230
[15] A. Ganesh, L. Massoulié, and D. Towsley, "The effect of network topology on the spread of epidemics," in *Proc. IEEE INFOCOM*, 2005.
[16] C. Gkantsidis and P. R. Rodriguez, "Network coding for large scale content distribution," in *Proc. IEEE INFOCOM*, 2005.
[17] W. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.
[18] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, "A measurement study of a large-scale P2P IPTV system," *IEEE Trans. Multimedia*, vol. 9, no. 8, pp. 1672–1687, Dec. 2007.
[19] M. S. Kim, S. S. Lam, and D. Lee, "Optimal distribution tree for Internet streaming media," in *Proc. Int. Conf. Distributed Computing Systems*, 2003.
[20] K. W. Kwong and D. H. K. Tsang, "On the relationship of node capacity distribution and P2P topology formation," in *Proc. IEEE Workshop on High Performance Switching and Routing (HPSR)*, 2005.
[21] K. W. Kwong and D. H. K. Tsang, "Application-aware topology formation algorithm for peer-to-peer networks," in *Proc. IEEE Int. Conf. Communications (ICC)*, 2007.
[22] J. Liang, R. Kumar, Y. Xi, and K. W. Ross, "Pollution in P2P file sharing systems," in *Proc. IEEE INFOCOM*, 2005.
[23] H. Luan, D. H. K. Tsang, and K. W. Kwong, "Media overlay construction via a Markov chain Monte Carlo method," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 34, no. 3, pp. 9–11, Dec. 2006.
[24] L. Massoulie and M. Vojnovic, "Coupon replication systems," in *Proc. ACM SIGMETRICS*, 2005.
[25] M. Meo and F. Milan, "A rational model for service rate allocation in peer-to-peer networks," in *Proc. IEEE Global Internet Symp.*, 2005.
[26] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The J. Chemical Phys.*, vol. 21, no. 6, pp. 1087–1092, 1953.
[27] Y. Moreno, R. Pastor-Satorras, and A. Vespignani, "Epidemic outbreaks in complex heterogeneous networks," *Eur. Phys. J. B*, vol. 26, p. 521, 2002.
[28] G. Pandurangan, P. Raghavan, and E. Upfal, "Building low-diameter peer-to-peer networks," *IEEE J. Sel. Areas Commun.*, vol. 21, pp. 995–1002, 2003.
[29] Y. J. Pyun and D. S. Reeves, "Constructing a balanced, (log(n)/loglog(n))-diameter super-peer topology for scalable P2P systems," in *Proc. IEEE P2P Computing*, 2004.
[30] D. Qiu and R. Srikant, "Modeling and performance analysis of bittorrent-like peer-to-peer networks," in *Proc. ACM SIGCOMM*, 2004.
[31] S. Saroui, P. K. Gummadi, and S. D. Gribble, "Measurement study of peer-to-peer file sharing systems," in *Proc. Multimedia Computing and Networking*, 2002.

[32] N. Sarshar and V. Roychowdhury, "Scale-free and stable structures in complex ad hoc networks," *Phys. Rev. E*, vol. 69, no. 2, p. 026101, 2004.
[33] D. Stutzbach, S. Zhao, and R. Rejaie, "Characterizing files in the modern gnutella network," *Multimedia Syst. J.*, 2007.
[34] R. H. Wouhaybi and A. T. Campbell, "Phenix: Supporting resilient low-diameter peer-to-peer topologies," in *Proc. IEEE INFOCOM*, 2004.
[35] X. Yang and G. Veciana, "Service capacity of peer to peer networks," in *Proc. IEEE INFOCOM*, 2004.
[36] X. Zhang, J. Liu, B. Li, and T. P. Yum, "Coolstreaming/Donet: A data-driven overlay network for peer-to-peer live media streaming," in *Proc. IEEE INFOCOM*, 2005.
[37] M. Zhong, K. Shen, and J. Seiferas, "Non-uniform random membership management in peer-to-peer networks," in *Proc. IEEE INFOCOM*, 2005.



Kin-Wah Kwong (S'07) received the B.E. degree (first class honors) and the M.Phil. degree in electronic engineering from the Hong Kong University of Science and Technology, Kowloon, Hong Kong, in 2003 and 2005, respectively. He is now pursuing the Ph.D. degree at the University of Pennsylvania, Philadelphia.

His current research interests span the fields of network resiliency, routing algorithms and protocols, traffic engineering, P2P networks and distributed systems.

Danny H. K. Tsang (M'82–SM'00) received the B.Sc. degree in mathematics and physics from the University of Winnipeg, Canada, in 1979, and the B.Eng. and M.A.Sc. degrees both in electrical engineering from the Technical University of Nova Scotia, Canada, in 1982 and 1984, respectively. He also received the Ph.D. degree in electrical engineering from the Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, in 1989.

Upon the completion of his Ph.D. degree, he joined the Department of Mathematics, Statistics and Computing Science, Dalhousie University, Canada, where he was an Assistant Professor in the Computing Science Division. He has been with the Department of Electronic and Computer Engineering at the Hong Kong University of Science and Technology (HKUST), Kowloon, Hong Kong, since the summer of 1992. He is currently a Professor in the department. His current research interests include Internet quality of service, P2P video streaming over the Internet, and wireless multimedia networks. During his leave from HKUST in 2000–2001, he assumed the role of Principal Architect at Sycamore Networks in the United States. He was responsible for the network architecture design of Ethernet MAN/WAN over SONET/DWDM networks. He invented the 64B/65B encoding scheme (US Patent number 6 952 405) for Transparent GFP in the T1X1.5 standard which was later advanced to become the ITU G.GFP standard. The coding scheme has now been adopted by International Telecommunication Union (ITU)'s Generic Framing Procedure recommendation GFP-T (ITU-T G.7041/Y.1303)).

Dr. Tsang was the General Chair of the IFIP Broadband Communications'99 held in Hong Kong and received the Outstanding Paper from Academe Award at the IEEE ATM Workshop'99. He also served as Technical Program Committee (TPC) members of ITC18-20, ACM Multimedia 2002, and INFOCOM 1994–96 and 2006–2007. He is currently an Associate Editor for the *Journal of Optical Networking* published by the Optical Society of America and Associate Technical Editor for the *IEEE Communications Magazine*. He was also the Guest Editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS special issue on Advances in P2P Streaming Systems.