This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2008 proceedings.

Balanced Relay Allocation on Heterogeneous Unstructured Overlays

Hung X. Nguyen
EPFL–Switzerland
hung.nguyen@epfl.ch

Daniel R. Figueiredo
UFRJ–Brazil
daniel@land.ufrj.br

Matthias Grossglauser
Nokia Research Center–Finland
matthias.grossglauser@nokia.com

Patrick Thiran
EPFL–Switzerland
patrick.thiran@epfl.ch

***Abstract*— Due to the increased usage of NAT boxes and firewalls, it has become harder for applications to establish direct connections seamlessly among two end-hosts. A recently adopted proposal to mitigate this problem is to use *relay nodes*, end-hosts that act as intermediary points to bridge connections. Efficiently selecting a relay node is not a trivial problem, specially in a large-scale unstructured overlay system where end-hosts are heterogeneous. In such environment, heterogeneity among the relay nodes comes from the inherent differences in their capacities and from the way overlay networks are constructed. Despite this fact, good relay selection algorithms should effectively balance the aggregate load across the set of relay nodes. In this paper, we address this problem using algorithms based on the *two random choices* method. We first prove that the classic load-based algorithm can effectively balance the load even when relays are heterogeneous, and that its performance depends directly on relay heterogeneity. Second, we propose an utilization-based random choice algorithm to distribute load in order to balance relay utilization. Numerical evaluations through simulations illustrate the effectiveness of this algorithm, indicating that it might also yield provable performance (which we conjecture). Finally, we support our theoretical findings through simulations of various large-scale scenarios, with realistic relay heterogeneity.**

I. Introduction

In today's Internet, it is often the case that two end-hosts cannot establish a direct connection between themselves. This occurs when both the end-hosts are located behind NAT boxes or firewalls, which for security reasons block connection establishment requests that arrive from outside end-hosts. This limitation is a serious concern to network application developers, as it limits the services the application can provide to some of its users. A recently adopted proposal to mitigate this problem is to use *relay nodes*, end-hosts that act as intermediary points, bridging the connection between two other end-hosts that want to communicate. In fact, such a solution is already adopted by a few large-scale peer-to-peer (P2P) applications, such as Skype [21] and Gnutella [1].

Within this framework, an open problem is to determine a *good* relay node for a given pair of end-hosts. As the Internet is mostly well connected, any candidate end-host could serve as a relay node. However, because relay nodes are simply other users' computers running the same application, the application developer would prefer to avoid overloading any particular relay node. Thus, the problem of selecting a good relay becomes one of balancing the load (i.e., connections) generated by the users across the set of relay nodes. This problem becomes non-trivial when we consider the conditions and assumptions under which these applications must operate.

Large-scale overlay systems are vastly decentralized, executed by a highly heterogenous end-host population, and crafted to operate in a very dynamic scenario, where nodes frequently join and leave the system. In such an environment, even requiring every end-host to have full knowledge of the set of available relay nodes can be too expensive, let alone knowing the instantaneous state (i.e., load) of each node. Therefore, any practical algorithm for selecting relay nodes cannot assume to have this knowledge and should be decentralized. Currently deployed systems such as Skype, attempt to balance the load by imposing a maximum number of connections a relay can handle [4], [10]. Clearly, such solution is suboptimal and measurement studies indicate that it can translate to longer delays when selecting a relay node [12].

Another important consideration when designing an efficient relay selection algorithm is the heterogeneity present in the set of relay nodes. In such systems, heterogeneity is caused by two different reasons: (i) application-level mechanisms introduce biases when sampling the relay nodes; (ii) relay nodes have inherent capacity differences (e.g., access bandwidth). The first case leads to a scenario where relay nodes have different "popularities", despite the fact that they may all have identical capacities. The popularity of a given relay node is related to the likelihood that other end-hosts have knowledge of its identity, which consequently, is related to the likelihood that the relay is used. Thus, a relay selection algorithm has a biased view of the set of relays present in the system.

In the presence of the first type of heterogeneity, where all relay nodes have identical capacities, an efficient relay selection algorithm should spread the load equally across the set of relays. However, in the presence of the second type of heterogeneity, an efficient algorithm should balance the *utilization* of the relay nodes, which is a metric proportional to the ratio between the load and the capacity of the relay. Note that balancing the absolute load in this case does not lead to a good relay allocation, as some relay nodes can be overloaded in terms of their utilization. In this paper we address these two problems as follows.

When considering the first type of heterogeneity, we adopt the simple and well-studied two random choices algorithm [3], which bases its choice solely on the current load on the relays. We then establish the performance of this algorithm by proving a tight upper bound and an almost matching lower bound on the maximum load on any relay at any time. Numerical evaluations through simulations support our theoretical findings, establishing the actual performance for

978-1-4244-2026-1/08/$25.00

different popularity models.

We also consider the more general case where both types of heterogeneity can be present in the system. We propose a simple modification to the previous algorithm to base its choice on the projected relay utilization. We evaluate the algorithm through simulations by considering realistic popularity and capacity models. Our results show that the proposed algorithm can effectively balance the utilization, and we conjecture that it also exhibits provably good performance in terms of balancing relay utilization.

Our contributions: The problem of relay selection when relay nodes have heterogenous popularity levels has been previously considered [6], [20]. However, these works focus on *structured* overlay systems, which by design impose limits on the popularity bias (i.e., relay nodes cannot have arbitrarily large popularity levels and only a small fraction them can have a large popularity level). Such assumptions are not reasonable for *unstructured* overlay systems, where relay nodes can have arbitrarily different popularity levels. To the best of our knowledge, we are the first to provide theoretical bounds for the two random choices algorithm under a general popularity model. By using a single parameter to measure the bias of the popularity model, we identify a scaling law on the performance of the algorithm that has not been captured nor predicted by any previous work. Our model relaxes some of the assumptions of [6] generalizing their model and our results are significantly different from theirs. Finally, to the best of our knowledge, we are also the first to apply a variation of the two random choices algorithm to balance relay utilization when relay nodes have heterogeneous capacities. Numerical evaluations indicate that this algorithm may yield provable good performance in terms of balancing utilizations.

The remainder of this paper is organized as follows. In Section II we review the related work. In Section III we formalize the problem considered. Section IV presents the load-based algorithm and our main theorerical result, while Section V presents the utilization-based algorithm and its numerical evaluation. Finally, Section VI concludes the paper.

II. Related work

A. Overlay networks

Overlay networks have been used in many applications such as file sharing [1], IP telephony [4], [10], and recently to improve network performance [9], [11]. Overlay systems can be structured or unstructured. In structured overlays, both data placement and overlay topology are tightly controlled using distributed hash tables, see for example [17]. In these topologies, (1) the maximum node degree is bounded from above by $\log n$ and (2) the number of nodes with degree $\log n$ is less than n^c for some constant $0 < c < 1$ with high probability (whp)[1] [6]. The main drawback of structured overlays is that they require significant maintenance overheads under high node churns. Unstructured overlays avoid this cost by allowing overlay nodes to connect largely unconstrained with each other. The resulting topology is robust to node churns, but has unbalanced node degrees [15], [18], [22]. Specifically, the two topological properties for structured overlays listed above do not carry over to their unstructured counterparts. It was shown in [15], [18], [22] that nodes in an unstructured overlay can have arbitrary degrees (i.e., property (1) of structured overlays in [6] is violated) and there is no guaranteed bound on the number of nodes with a given degree (i.e., property (2) of structured overlays in [6] is violated).

[1]We use the term with high probability (whp) to mean with probability at least $1 - 1/n^\gamma$ for some fixed constant γ.

B. Theoretical Load Balancing

The general problem of load balancing can usually be framed as a balls and bins problem, where the bins correspond to resources and the balls correspond to load. A load balancing algorithm will place balls into bins under some given conditions. Consider the simple case of n balls and n bins, but where the algorithm has no knowledge of (and does not remember) the state of the bins. A simple load balancing algorithm is to choose a bin independently and uniformly at random for each ball. This algorithm ensures that the number of balls in any bin is at most $(1 + o(1))\frac{\log n}{\log \log n}$ with high probability, see for example [14]. Azar et al. [3] suggest a better algorithm that selects $d \geq 2$ bins at random for each ball and places the ball in the least loaded bin, which is known as the d-random choices algorithm. Although at first sight this algorithm may look similar to the previous, the authors show that the maximum load in any bin is at most $\frac{\log \log n}{\log d} + \Theta(1)$, whp. The significance of this result is the exponential reduction in the maximum load when compared to the previous algorithm, which can be obtained by simply selecting two bins for each ball. Moreover, choosing more bins only yields a marginal improvement over the two choices. This phenomenon is called "the power of two random choices" [13].

Byers et al. [6] study the performance of the d-random choices algorithm when the bin sampling distribution is not uniform. Under the assumptions guaranteed by a *structured* overlay system (listed above), they prove that the 2-random choice algorithm yields a maximum load on any bin of at most $\log \log n$ whp. This is a strong result because the sampling distribution is far from uniform, with some bins sampled with probability $\frac{\log n}{n}$ and others with probability $\frac{1}{n \log n}$. More recently, however, Wieder [20] has shown that this double logarithmic bound ceases to hold when the number of choices d is small and the number of balls (m) is much larger than the number of bins (n). To achieve the maximum load of $\frac{m}{n} + \frac{\log \log n}{\log d} + O(1)$, i.e., the result obtained when all bins are chosen with equal probability $\frac{1}{n}$ [5], d has to be significantly large ($d = O(\log n)$) [20].

Despite the fact that the d-random choice algorithm and its variations have been applied to a wide range of load balancing problems, its performance on unstructured overlays is still an open issue. First, as we mentioned earlier the topological constraints of structured overlays do not apply to unstructured overlays hence the balancing results in [6] may not carry over. Indeed, we show that when these constraints are violated, the bound in [6] does not apply. Second, in a decentralized

system, it is unrealistic to adapt the number of choices d to $\log n$ as in [20], because this would require every node to know the instantaneous number of relays in the system. Third, in real overlay systems, connections arrive and depart the system continuously, but the approaches in [6], [20] apply only to static systems where m balls arrive in sequential order without any departure. In this paper, we remove all of the above restricted constraints to provide tight bounds on the performance of the d-random choices method on realistic unstructured overlays.

III. Problem Statement

In this paper we are interested in large-scale networked applications that use *relay* nodes. Relays are used as intermediary nodes to bridge a connection between two end-hosts that could not otherwise communicate. We assume here that users of this application are interested in communicating with one another and generate traffic in the form of connections. This is motivated by large-scale voice-over-IP (VoIP) (Skype [4], [10], [21]) and file sharing (Gnutella [1]) applications.

The relay selection algorithm is responsible for assigning a relay node for each new connection generated by the users. The algorithm is part of the large-scale networked application and runs on the end-host of the user initiating the connection, and it is therefore a distributed algorithm. We assume that the algorithm has no knowledge of the set of relays currently available. However, the algorithm can request the system to provide the identity of a small number of relay nodes. The answer provided by the system need not be uniformly distributed over the entire set of candidate relays. In particular, we assume that there can be bias based on the "popularity" of the relay node.

In practical applications [1], [21], special end-hosts, known as super-nodes or ultra-peers, play the role of the system by maintaining an overlay network. In particular, super-nodes provide the relay selection algorithm with a small subset of the candidate relays that they know [4], [10]. However, the super-nodes themselves need to learn the identity of candidate relays. One proposal is to sample the overlay network using a random walk algorithm. It is well known that this mechanism produces a biased view of the candidate relays, with bias proportional to the super-node degree. Furthermore, a relay that is present in the system for prolonged periods of time is more likely to have its identity known by more super-nodes than a relay node that has a high churn. For these and other reasons, it is therefore unlikely that relay nodes are used equally in unstructured overlay systems [18], even when they have identical capacities. Thus, an effective relay selection algorithm should cope with this fact.

Finally, relay nodes can also be inherently heterogeneous when we consider their capacities (e.g., access bandwidth or CPU speed) [16]. Note that although future networked systems can implement algorithms to remove the popularity bias discussed above, heterogeneity in terms of capacity will always be inherent in the system. We note that relay popularity and capacity need not be correlated, although in practice this is probably the case. Again, an efficient relay selection algorithm should also cope with this kind of heterogeneity, and distribute the load such that relay *utilization* is balanced.

The problem we investigate is the following: Given the scenario above, how should we design a "good" relay selection algorithm? A good relay selection algorithm is one that distributes the aggregate load over the set of relays such that either the absolute load or the utilization is effectively balanced. At the same time, the relay selection algorithm should be efficient, quickly finding a relay node without imposing a high communication overhead. In what follows, we describe more precisely the model we consider, followed by our proposed solutions and their evaluations.

Let $\mathcal{R}$ denote the set of relay nodes available in the system, where $n = |\mathcal{R}|$ denotes their total number. In order to model relay popularity, we will assign to each relay node $r_i \in \mathcal{R}$ a fixed popularity level $\alpha_i > 0$. The popularity of the relays will be used to determine the probability that its identity is revealed to the relay selection algorithm. In particular, let p_i denote this probability with $p_i = \alpha_i / \sum_{r_i \in \mathcal{R}} \alpha_i$. Moreover, we assume that each relay $r_i \in \mathcal{R}$ will have a fixed capacity, denoted by $\kappa_i > 0$.

Let $m_i(t)$ denote the number of connections traversing relay $r_i \in \mathcal{R}$ at time t. We call $m_i(t)$ the *load* of r_i at time t. Let $u_i(t)$ denote the utilization of relay $r_i \in \mathcal{R}$ at time t, which is simply given by $u_i(t) = m_i(t)/\kappa_i$. Each relay node keeps track of this information ($m_i(t)$ and $u_i(t)$) and responds with these values to probes generated by other nodes.

IV. Balancing Relay Load

In this section, we address the problem of balancing the load across the set of relays in the presence of only relay popularity heterogeneity. We assume that all relays have identical capacities, such that $\kappa_i = 1$, for $i = 1, \ldots, n$ (an assumption we relax in the next section). In this scenario, the ultimate goal of a distributed relay selection algorithm is to spread the load over the relays as equally as possible, even though some relays can be much more popular than others.

In order to measure the quality of load balancing achieved by the algorithm, we will consider the maximum load (i.e., number of connections) imposed on any relay node. In particular, let $m(t) = \max_{i \in \mathcal{R}} m_i(t)$ denote the maximum number of connections across any given relay node at time t. A good load balancing algorithm minimizes $m(t)$ at all times, where the optimal value at time t is given by $\lceil \sum_{i \in \mathcal{R}} m_i(t)/n \rceil$.

In this scenario, we propose to use the load-based d-random choices algorithm. This algorithm is very simple and efficient in selecting a relay node, as we next describe. Moreover, its performance in terms of load balancing ($m(t)$) can be established theoretically, as a function of system parameters.

A. The load-based d-random choices algorithm

The load-based d-random choices algorithm works as follows. At the time of arrival of a new connection, t, the algorithm requests d relay nodes from the system. Let $\mathcal{R}_d$ be the set of relay nodes provided by the system. The algorithm then probes all relay nodes in $\mathcal{R}_d$ to obtain their current load. The least loaded relay node is selected for this new

connection. Thus, the chosen relay $r(t)$ is given by $r(t) = \arg\min_{i \in \mathcal{R}_d} m_i(t)$. This algorithm will be referred to as the "load-based d-relay" algorithm or "d-relay" in short.

In the d-relay algorithm, the number of relay nodes d requested from the system and probed can vary from 1 to $n = |\mathcal{R}|$. We advocate the 2-relay algorithm for the relay selection problem, which is very efficient as it only requires two probes per new connection. Moreover, as we demonstrate next, the 2-relay algorithm yields provable load balancing performance on the relays.

B. Theoretical Evaluation

In this section we show that the performance of the load-based d-relay algorithm can be captured using a simple metric to model heterogeneity in the relay popularity. In particular, recall that p_i, the probability that the system reveals relay i to the algorithm, depends on i. Thus, we introduce a metric we call *popularity deviation*, $\alpha(n)$, defined as follows:

$$\alpha(n) = \frac{\max_{r_i \in \mathcal{R}} \alpha_i}{\frac{1}{n}\sum_{r_i \in \mathcal{R}} \alpha_i} = n \max_{r_i \in \mathcal{R}} p_i. \quad (1)$$

Note that the popularity deviation measures the ratio between the maximum and the average popularity levels of the relays. More importantly, we have the following bound:

$$p_i \leq \frac{\alpha(n)}{n} \text{ for all } r_i \in \mathcal{R} \quad (2)$$

This model is very parsimonious because it summarizes the heterogeneity among the relay nodes in a single parameter $\alpha(n)$. Moreover, as we will see shortly the performance of the load-based 2-relay algorithm is strictly a function of $\alpha(n)$. In the rest of this paper, we use α instead of $\alpha(n)$ for notational convenience.

Consider the maximum relay load in the system at any point in time $m(t)$. We first provide an upper-bound of $\log\log n + O(\alpha)$ on $m(t)$ in Theorem 1 for any overlay system. We then prove an almost matching lower bound of $\Omega(\log\log n + \alpha)$ on the expected maximum relay load in Theorem 2 by constructing overlay systems that achieve this bound. Note that if $\alpha = \Theta(\log n)$, the expected load on the most loaded relay is $\Theta(\log n)$. A simple application of the Chernoff bound yields that the maximum load is also $\Theta(\log n)$ whp. This result is in stark contrast with the rather optimistic bound of $\log\log n$ in [6]. The difference between our result and those in [6] stems from the fact that we relax the constraints needed in [6] on (1) the maximum relay sampling probability and on (2) the number of relays with a given sampling probability. This relaxation is necessary to study unstructured overlays as explained earlier.

Theorem 1: If the number of connections in the system is always smaller than or equal to n and the popularity deviation is α, then at any time t, with probability at least $1 - 1/n^{\Omega(1)}$, the maximum load is no more than $\log\log n + 8e\alpha + O(1)$.

We will use the witness tree method in [2] to prove the theorem. A special case of the theorem where $\alpha = 1$ was proven in [8]. Our proof develops further the techniques used in [8], but differs in the way we construct the witness tree. This modification is needed to handle $\alpha > 1$. As in the classical proof of the theorem with $\alpha = 1$ [8], the main idea of the witness tree method is that if at time t there is a relay r with more than $l+w+q$ connections where $l = \log\log n, w = 8e\alpha$, and $q > 2$ is a constant, then a certain sequence of "bad" events must have happened before time t in order to build up the connections in r. The witness tree is used to capture this sequence of events. This tree reports only the events that we know for sure have happened between time 0 and t. By proving that the probability of the occurrence of such a tree is upper-bounded by $1/n^{\Omega(1)}$, we can then deduce that the probability that there is a relay with at least $l + w + q$ connections is also upper-bounded by the same number.

TABLE I
NOTATION USED ONLY IN THE PROOF

Symbol	Definition
c_m	The m-th connection in the system
v_j	The jth node of the witness tree
$e = (v_j, v_o)$	An edge between node v_j and v_o
$r(u)$	The relay represented by node u
$c(e)$	The connection represented by edge e
$r_1^{c_m}$ and $r_2^{c_m}$	The two relays choices of connection c_m where $r_1^{c_m}$ is the finally chosen relay
t_m	The time c_m arrives in the system
$\mathcal{T}$	The set of all unlabeled pruned witness trees
$T \in \mathcal{T}$	An unlabeled pruned witness tree
$s = \|T\|$	Number of nodes in a tree T
$\mathcal{Q}_T$ and $z = \|\mathcal{Q}_T\|$	The set of pruning edges in T and its cardinality
N_T	Number of ways to label the tree T
l	$\log\log n$
w	$8e\alpha$
q	a constant, $q \geq 2$

Throughout the proof, we will use the notations in Table I.

Proof: When $\alpha > \log n$, the proof is straightforward. Indeed, the probability that a connection c chooses a specific relay r in at least one of its two choices is upper-bounded by $2\alpha/n$. The expected number of connections that goes to r at any time t is therefore at most $n2\alpha/n = 2\alpha$. A simple application of the Chernoff bound yields that the number of connections in any relay is at most 4α with probability at least $1 - 1/n^{\Omega(1)}$.

When $\alpha \leq \log n$, the proof consists of three steps. In the first step, we show that if there is a relay $r \in \mathcal{R}$ at time t with $l+w+q$ connections then we can construct an irregular, labeled *full witness tree* of depth $l + 1$ that captures a set of events that must have happened before time t. In the second step, we prune the full witness tree to obtain a *pruned witness tree* to remove stochastic dependencies between nodes and edges of the tree. The probability that the pruned witness tree occurs is an upper-bound on the probability that the corresponding full witness tree occurs. In the last step, we enumerate all the possible pruned witness trees to obtain an upper-bound on the probability that such a tree occurs.

Step 1: Constructing the full witness tree

A witness tree is a *labeled* tree in which each node v_j is labeled by a relay $r(v_j) \in \mathcal{R}$ together with a set of w connections, and each edge e is labeled by a connection $c(e)$. The detailed construction of the witness is as follows.

Assume that at time t there is a relay r with more than $l +$

$w+q$ connections. We place r at the root v_0 of the witness tree, i.e., $r(v_0) = r$. We denote the $l+w+q$ connections in r at time t by $c_1, ..., c_{l+w+q}$ and assume that these connections have arrived at node r at times $t_1 < t_2 < ... < t_{l+w+q} \leq t$. From v_0 we draw q edges, each connecting v_0 with a child node that represents the other relay choice $r_2^{c_m}$ of the connection c_m, for all $l+w+1 \leq m \leq l+w+q$. Note that when c_m arrives to r at time t_m there are already at least $m-1$ connections in r (connections $c_1, ..., c_{m-1}$). As c_m chose r as its relay, the number of connections in $r_2^{c_m}$ must be larger than or equal to the number of connections in r. Therefore the number of connections in $r_2^{c_m}$ at time t_m must be at least $m-1$. Hence, each child of the root v_0 is labeled by a relay that has at least $l+w$ connections at some given time (specified by the corresponding t_m).

Using the previous observation, we then recursively grow the witness tree by constructing a subtree from each child of the root v_0. Take a child v_i, v_i is labeled by a relay $r(v_i)$ that has at least $l+w$ connections at some given time. We call any of the last l connections that arrive to $r(v_i)$ among the $l+w$ connections in $r(v_i)$ a "top" connection. The other w connections are called "bottom" connections. For each of the top l connections in this relay there must be an alternate relay choice. These alternate relay choices are set to be the children of v_i and the edges are labeled by the corresponding connections. We recurse similarly for each of the l children. For a parent node with load $x+w$, the $(x-i+1)$th top connection must have had an alternate relay choice with load at least $x-i+w$; this is the ith child of the parent node. The ith child, which has load at least $x-i+w$ will be expanded down further to $x-i$ children corresponding to the alternate relay choices for its top $x-i$ connections. We continue this recursion as long as a relay has a load greater than w and stop when it equals w, which is the load of the leaf nodes. We also label each node (relay) with the set of w bottom connections. These connections are distinct from the connections that label the edges incident on this node because by construction only the top connections label edges.

The resulting witness tree has a depth of $l+1$, with the root having q children. We call this tree the *full witness tree*. It is easy to check inductively that the total number of nodes in each sub-tree stemming out of each child of the root is exactly 2^l. Hence the total number of nodes in the full witness tree is $q2^l+1$. Different nodes in this tree can represent the same relay and different edges can represent the same connection. Note however that adjacent edges represent distinct connections. An illustration of the full witness tree is given in Figure 1.

Step 2: Pruning the full witness tree

If the nodes of the witness tree are all distinct, then it is relatively easy to upper-bound the probability that such a tree exists. However, many nodes in the full witness tree can be labeled by the same relay. It is therefore necessary to trim the tree so that all nodes in the tree are labeled by distinct relays, which will then eliminate stochastic dependencies between the nodes and edges. We can then upper-bound the probability of the occurrence of the trimmed tree. The pruning procedure works as follows. We perform a breadth first search of the full witness tree starting at the root. As we move along the tree, we keep a set $\mathcal{U}$ of the labeled nodes that we have already visited and a set $\mathcal{Q}$ of labeled edges that cause repetition of relays in the full witness tree. At the beginning, $\mathcal{U} = \emptyset$ and $\mathcal{Q} = \emptyset$. During the trimming procedure, whenever we visit a node v_j there are two possibilities:

- If v_j is labeled by a relay that is different from the labels of nodes in $\mathcal{U}$, we add it to $\mathcal{U}$ and move to the next node.
- Otherwise, we remove the subtree rooted at v_j from the witness tree. We call the edge from v_j to its parent a *pruning edge* and add this pruning edge to $\mathcal{Q}$.

The procedure continues until one of the following two conditions is met:

- All nodes of the full witness tree have been visited.
- There are already q pruning edges. In this case, we prune all the nodes of the full witness tree that have not been visited.

The tree that results from this process is called the *pruned witness tree*. Each pruned witness tree T_p is a labeled tree that consists of (i) all the labeled nodes and edges that remain in the tree after the pruning process, and (ii) the set of pruning edges $\mathcal{Q}_{T_p}$ (with their labels). Let $z = |\mathcal{Q}_{T_p}|$ be the number of pruning edges in a pruned witness tree T_p. Figure 2 provides an illustration for the pruning procedure that obtains a pruned witness tree from the full witness tree of Figure 1.

Step 3: Bounding the probability of occurrence of a pruned witness tree

Define T to be an unlabeled, rooted pruned witness tree. That is, T provides a detailed description of the topology of a pruned witness tree and the corresponding set of pruning

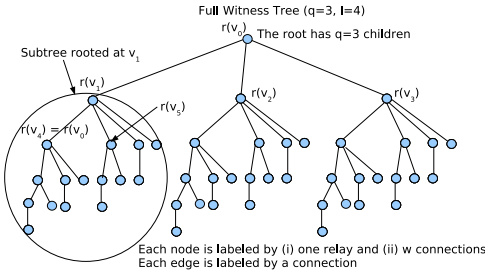

Fig. 1. Illustration of the construction of the full witness tree, with $l = 4, q = 3$. Each node v_i in the tree is labeled with a relay $r(v_i)$ and a set of w connections. Each edge in the tree is labeled with a connection. In this example, the load of $r(v_0)$ is $4+3+w$, each of the relays $r(v_1)$, $r(v_2)$ and $r(v_3)$ has load at least $4+w$. In the subtree rooted at v_1, $r(v_4)$, which is the same relay as $r(v_0)$, has load at least $3+w$ and $r(v_5)$ has load at least $2+w$ and so on.

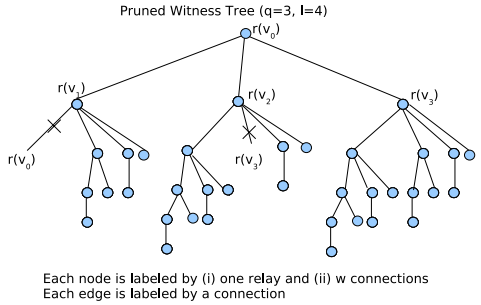

Fig. 2. Illustration of the pruned witness tree. There are $z = 2$ pruning edges, marked by a cross. Here, $z = 2 < q = 3$, observe that the subtree rooted at v_3 is the same as in the full witness tree.

edges $\mathcal{Q}_T$. However, each node of T is not associated with a relay and a set of w connections. The edges of T and the pruning edges in $\mathcal{Q}_T$ are also not associated with connections. Let $\mathcal{T}$ be the set of all possible unlabeled pruned witness trees.

We upper-bound the probability that a pruned witness tree exists by upper-bounding both the number of possible unlabeled trees T, the number of ways to label a tree T, and the probability that each such labeled tree could arise.

Bounding the number of unlabeled trees

The pruning process stops if there are q pruning edges, i.e., $z \leq q$. Hence there are at most q different values for the number of pruning edges z. For each value of z, because the pruned witness tree has at most $q2^l$ edges, the number of ways we can choose z pruning edges is at most $\left(\begin{array}{c} q2^l \\ z \end{array} \right) \leq q^z 2^{lz}$.

Thus the total number of unlabeled trees is at most

$$|\mathcal{T}| \leq q^{q+1} 2^{lq}. \tag{3}$$

Number of ways to label an unlabeled tree T:

We now focus on an unlabeled tree T. Let $s = |T|$ be the total number of nodes in T. We start by counting the number of ways to label T and $\mathcal{Q}_T$, that is, the number of ways to assign (i) a relay to every node of T , (ii) one connection to each edge of T, (iii) w connections to each relay associated with a node in T, and (iv) one connection to each of the pruning edges in $\mathcal{Q}_T$. We now enumerate each of the above individually.

(i) There are at most n ways of assigning a relay to a node. Therefore, there are at most n^s ways to label s nodes in the unlabeled tree T.

(ii) The total number of ways of assigning connections to all edges in the unlabeled tree T is at most n^{s-1} (there are $s-1$ edges) because at any time there are at most n connections in the system.

(iii) Similarly, there are at most $\left(\begin{array}{c} n \\ w \end{array} \right)$ ways to assign w connections to the relay associated with one node in T, and thus at most $\left(\begin{array}{c} n \\ w \end{array} \right)^s$ ways to assign w connections to each of the s nodes.

(iv) There are at most n^z ways of assigning connections to the z pruning edges in $\mathcal{Q}_T$.

Hence the total number of different ways to label T and $\mathcal{Q}_T$ (i.e, the number of ways of assigning relays to nodes, w connections to each relay, connections to edges of T, and connections to pruning edges of $\mathcal{Q}_T$), denoted by N_T, is bounded from above by

$$N_T \leq n^s n^{s-1} \left(\begin{array}{c} n \\ w \end{array} \right)^s n^z. \tag{4}$$

Probability that a labeled pruned witness tree T occurs:

We say that a given (labeled) pruned witness tree is activated if the choices of relays for the connections occur in such a way that (i) all the chosen relays associated with the nodes are connected by the chosen connections associated with the edges of the tree, (ii) all w connections associated with a relay do finally choose that relay, (iii) and all the chosen connections associated with the z pruning edges in $\mathcal{Q}_T$ have their two relay choices represented by two nodes in the pruned witness tree.

We now bound the probability that a given pruned witness tree is activated by bounding each of the above items.

(i) Recall here that the probability that a connection chooses a specific relay is bounded from above by α/n because of (2). Therefore, the probability that a given connection picks two specific relays is bounded from above by $\frac{2\alpha^2}{n^2} = 2\frac{\alpha}{n}\frac{\alpha}{n}$, where the factor of 2 comes from the 2 possible assignments of 2 relays to a connection. For the sake of simplicity, we assume that the system chooses relays with replacement. Hence, the probability that $s-1$ edges of T are labeled by $s-1$ specific labels is at most $\left(\frac{2\alpha^2}{n^2}\right)^{s-1}$.

(ii) We now consider the probability that a relay r_i has w connections. The probability that a connection picks a particular relay in one of its two choices is bounded by $2\alpha/n$. Hence, the probability that w specific connections go to a specific relay is at most $(2\alpha/n)^w$. Thus the probability that all s nodes in T are labels by ws specific connections is at most $\left(\frac{2\alpha}{n}\right)^{ws}$.

(iii) A pruning connection has both its relay choices represented by nodes in the pruned witness tree. Hence, the probability that a specific connection is a pruning connection is upper-bounded by $\frac{2\alpha^2}{n^2}\left(\begin{array}{c} q2^l+1 \\ 2 \end{array} \right)$, because the total number of nodes in the pruned witness tree is no more than that in the full witness tree, i.e., $q2^l+1$.

Hence, the probability that a specific pruned witness tree is activated is no more than

$$\left(\frac{2\alpha^2}{n^2}\right)^{s-1} \left(\frac{2\alpha}{n}\right)^{ws} \left[\frac{2\alpha^2}{n^2}\left(\begin{array}{c} q2^l+1 \\ 2 \end{array} \right)\right]^z. \tag{5}$$

Note that we can multiply the probabilities because the w connections in each relay, the connections represented by the edges of the pruned witness tree and the pruning connections are all distinct.

Combining the bounds

Combining (4) and (5), an application of the union bound yields an upper bound on the probability $\mathbb{P}(T)$ that an unlabeled tree T occurs

$$\mathbb{P}(T) \quad \leq \quad \frac{n}{2\alpha^2}\left[2\alpha^2\left(\frac{2e\alpha}{w}\right)^w\right]^s \left(\frac{2\alpha^2 q^2 2^{2l}}{n}\right)^z \stackrel{\text{def}}{=} B(n).$$

In the above inequality, we use $\left(\begin{array}{c} q2^l+1 \\ 2 \end{array} \right) \leq q^2 2^{2l}$ and $\left(\begin{array}{c} n \\ w \end{array} \right) \leq \left(\frac{ne}{w}\right)^w$ for all $w > 0$.

Observe that either the number of pruning connections, z, equals q or the number of nodes $s \geq 2^l = \log n$ (because in the latter case, at least one subtree rooted at a child of the root in the full witness tree will remain untouched in the pruning process and this subtree has 2^l nodes). Note also that when $w = 8e\alpha$, $2\alpha^2\left(\frac{2e\alpha}{w}\right)^w \leq 1/4$ for all $\alpha > 0$. In both cases, replacing $l = \log\log n$ yields $B(n) \leq 1/n^{\Omega(1)}$. From (3),

$|\mathcal{T}| = O(\log^q n)$. Hence the probability that at time t there exists a relay with more than $l+w+q$ connections is at most $|\mathcal{T}|/n^{\Omega(1)} = 1/n^{\Omega(1)}$. This completes the proof. ■

Theorem 1 can be easily extended to the case where there are at most kn ($k \geq 1$) connections in the system at any time by repeating the same arguments in the proof but using $w = 8ek\alpha$.

Corollary 1: If the number of connections in the system at any time is at most kn and the popularity deviation is α, with probability at least $1 - 1/n^{\Omega(1)}$ the maximum load on any relay is at most $\log\log n + 8ek\alpha + O(1)$.

We now provide an almost matching lower bound of $\Omega(\log\log n + \alpha)$ on the maximum number of connections on any relay by constructing a relay system $\mathcal{R}^*$ that satisfies the condition in (2) and achieves this lower-bound.

Theorem 2: If the number of connections in the system is always n and the popularity deviation is α, then there is a set of relays such that at any time t the expected load on the heaviest relay is at least $\Omega(\log\log n + \alpha)$.

We follow similar strategies as those in [19] and [20] to prove this lower bound.

Proof: The $\Omega(\log\log n)$ term in the lower bound follows from Vöcking's lower bound [19] for placement of n balls into n bins when each ball picks two bins at random using any arbitrary but fixed distribution.

We now prove the second term $\Omega(\alpha)$ using a strategy similar to [20]. Consider a set of relays $\mathcal{R}^*$ where $\frac{n(\beta-1)}{\beta\alpha-1}$ relays have sampling probability $\frac{\alpha}{n}$, and $\frac{n(\beta\alpha-\beta)}{\beta\alpha-1}$ relays have sampling probability $\frac{1}{\beta n}$, for any constant $\beta \geq 2$. Denote by $\mathcal{H}$ the set of $\frac{n(\beta-1)}{\beta\alpha-1}$ relays in $\mathcal{R}^*$ with high sampling probability α/n. The probability that a connection chooses a relay in $\mathcal{H}$ is

$$\frac{n(\beta-1)}{\beta\alpha-1}\frac{\alpha}{n} = \frac{\alpha(\beta-1)}{\beta\alpha-1}.$$

Hence the probability that a connection goes to a relay in $\mathcal{H}$ is at least $\left(\frac{\alpha(\beta-1)}{\beta\alpha-1}\right)^2$, i.e., the probability that the connection chooses two relays in $\mathcal{H}$ for both of its choices. Therefore, after inserting n balls the expected total number of connections in $\mathcal{H}$ is at least $n\left(\frac{\alpha(\beta-1)}{\beta\alpha-1}\right)^2$. Hence, the expected load in the most loaded relay in $\mathcal{H}$ (with $\frac{n(\beta-1)}{\beta\alpha-1}$ relays) is at least $\frac{\alpha^2(\beta-1)}{\beta\alpha-1} > \frac{\alpha}{\beta} = \Omega(\alpha)$ for any constant $\beta \geq 2$. This completes the proof. ■

C. Numerical evaluation

We design and implement a simple simulator of the model described in Section III. For the simulations, we assume that the aggregate connection arrival rate to the system follows a Poisson process with rate λ and that connections have exponentially distributed durations with mean 1. Notice that since each new connection is treated independently from any other event by the relay selection algorithm, it does not matter which end-host generates it. The number of relay nodes in the system is $n = 1000$, unless otherwise specified.

As the maximum load on any relay, $m(t)$, varies with time, we will consider its time average, namely $\overline{m}(t)$, in order to characterize the maximally loaded relay in the system. In particular, we define $\overline{m}(t_e) = 1/t_e \int_0^{t_e} m(t)\ dt$, where t_e is the simulation end time.

We begin by considering the performance of the algorithm as a function of the connection arrival rate (which is also the average number of connections in the system). Figure 3.(a) and 3.(b) present the results for a wide range of arrival rates under two different popularity models. In the linear popularity model of Figure 3.(a) the popularity level of relay r_i is given by i, thus, $\alpha_i = i$. This model yields a popularity deviation $\alpha = 2n/(n+1)$, which is close to 2. In the Zipf popularity model of Figure 3.(b) the popularity of relay r_i is $\alpha_i = i^{-1.5}$. The corresponding popularity deviation is $\alpha = 392$. As predicted by Theorem 1, the 2-relay and 4-relay algorithms balance the load well in both moderately (linear model) and extremely (Zipf distribution) biased popularity distributions, when compared to the performance of the 1-relay. Moreover, as predicted by Theorem 1 and 2, the maximum load depends directly on α. Notice that the maximum load in the linear model ($\alpha = 2$) is significantly less than that of the Zipf model ($\alpha = 392$). Indeed, as we soon discuss, the maximum load in these two cases follow two different scaling laws. Finally, we also observe that the average maximum load on the relays grows linearly with the arrival rate, as predicted by Corollary 1. Note that the arrival rate here is directly related to the maximum number of connections in the system, denoted by k, in the corollary statement.

We now consider a popularity model motivated by the Gnutella network. In Figure 3.(c) we consider the case where relays have popularities that are equal to the node degrees in the Gnutella network. Thus, according to the measurement studies of [18], [22], we assume that 5% of the relay nodes have popularity uniformly distributed between 1 and 27; 90% have popularity uniformly distributed between 28 and 32; and 5% have popularity uniformly distributed between 33 and 60. In the figure, we vary the number of relays n to study the effect of n on the performance of the d-relay algorithm. The arrival rate is $10 * n$, hence the maximum number of connections in the system remains constant. Thus, both k and the popularity deviation α remain unchanged. Again, we observe that our bounds predict very well the performance of the d-random choices algorithm where the maximum load increases very slowly with n (double logarithmic) when both α and k do not grow.

Finally, we consider the impact of the popularity deviation α on the maximum load using the Gnutella popularity model. In order to scale the popularity deviation we increase the maximum node degree in the system and keep all other parameters unchanged (5% of the relays have popularity between 33 and the maximum degree). Figure 4 shows a regime change in the 2-relay algorithm as α increases. As predicted in Theorems 1 and 2, when α is small ($\alpha = \Theta(\log\log n)$), the maximum load in the system increases very slowly, as a double logarithmic function of n. However, when α is large ($\alpha = \Theta(\log n)$), the maximum load increases much faster, as a logarithmic function

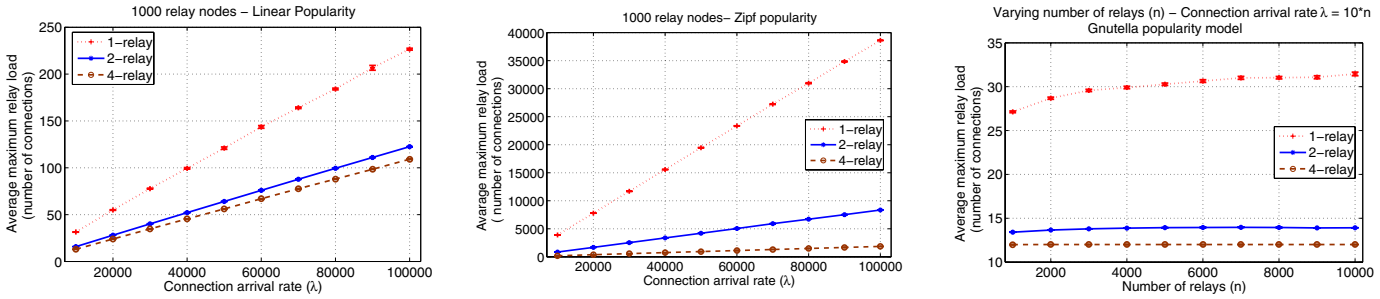

Fig. 3. Average maximum relay load as a function of the connection arrival rate: (a) relay nodes have linear popularity, $\alpha_i = i$; (b) relay nodes follow Zipf popularity, $\alpha_i = i^{-1.5}$; (c) relay nodes follow Gnutella popularity model. Note that in the last figure the number of relays n also changes.

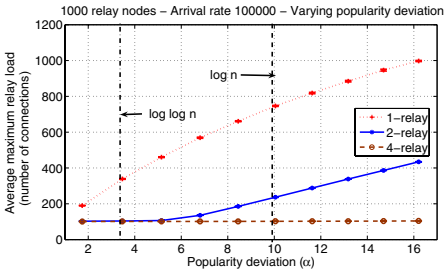

Fig. 4. Average maximum relay load as a function of the popularity deviation α. Gnutella popularity model with maximum node degree varying from 50 to 500.

of n. We note that this behavior of the maximum load has not been captured nor predicted by any previous work.

V. Balancing Relay Utilization

As previously discussed, relay nodes can be heterogeneous with respect to their inherent capacities. In this scenario, effectively balancing the load across the set of relays is not necessarily a good strategy. In fact, this may lead to a scenario where the *utilization* of the relay nodes are very unbalanced, possibly overloading some relay nodes. This is clearly undesirable since both the relay node (i.e., the user behind that end-host) and the connections traversing that relay will experience performance degradation. Thus, a good relay selection algorithm should distribute the load such that relay utilization is well balanced.

Notice that we treat heterogeneity in relay popularity orthogonally to heterogeneity in relay capacity. Although in practice these two types of heterogeneity are likely to be correlated (i.e., the most popular relay is the one with the highest access bandwidth), the algorithm we propose makes no assumption on their relationship.

A. The utilization-based d-random choices algorithm

We modify the load-based d-random choice algorithm to consider relay utilization instead of load as the performance metric for choosing the best candidate relay. In particular, the chosen relay r, is given by $r = \arg\min_{i \in \mathcal{R}_d}(m_i(t) + 1)/\kappa_i$. Note that we consider the utilization of the relay node as if it would be chosen to relay the new connection. This algorithm will be referred to as the "utilization-based d-relay algorithm". Finally, if all relays have identical capacities, then this algorithm is equivalent to the load-based d-relay algorithm introduced earlier.

Given the similarity between the load-based and the utilization-based algorithms, we expect the two to have similar performances in terms of balancing their respective metrics. This observation is well supported by the results obtained through numerical evaluation of the utilization-based algorithm (presented below). We therefore conjecture that the maximum relay utilization in any relay is at most $\log\log n + O(\alpha\overline{u})$ whp, where $\overline{u}$ is the average utilization in the system (i.e., the ratio between the total load on the system and the total system capacity). Extending the proof of Theorem 1 to the case of utilization is not trivial, because a ball brings different (non-integer) utilization amounts to different relays.

B. Numerical evaluation

We adapt our simulator to consider heterogeneity on relay capacities and implemented the utilization-based d-relay algorithm. Notice that each relay node $r_i \in \mathcal{R}$ is now associated with both a popularity level α_i and a capacity κ_i. These two values can be correlated (and in practice they most likely are) and we consider two scenarios in the following evaluation: fully-correlated and uncorrelated. In the former, we assume a perfect match between relay popularity and capacity (i.e., the relay with the highest popularity is also the one with the highest capacity, and so on). In the latter, we assume a random matching between relay popularity and capacity (i.e., each relay is randomly assigned a popularity level and a capacity from their respective distributions).

As for the relay popularity, we consider the Gnutella popularity model introduced in Section IV-C, which is based on actual measurements on the Gnutella network. To model the differences in relay capacities, we use a model proposed in [7], known as the Gia capacity model. This model is also based on measurements conducted on the Gnutella network. In summary, the model associates capacity 1 to 20% of the relays, capacity 10 to 45% of the relays, capacity 100 to 30% of the relays, capacity 1000 to 4.9% of the relays, and capacity 10000 to 0.1% of the relays. These capacities are given in terms of unit capacity, and we will assume that 1 unit of capacity is equivalent to the load of 25 connections.

We first consider the limitations of using the load-based d-relay algorithm when relay nodes have different capacities. We assume a fully correlated system and evaluate the maximum relay utilization as a function of the arrival rate. The results are shown in Figure 5.(a), which clearly indicate that the

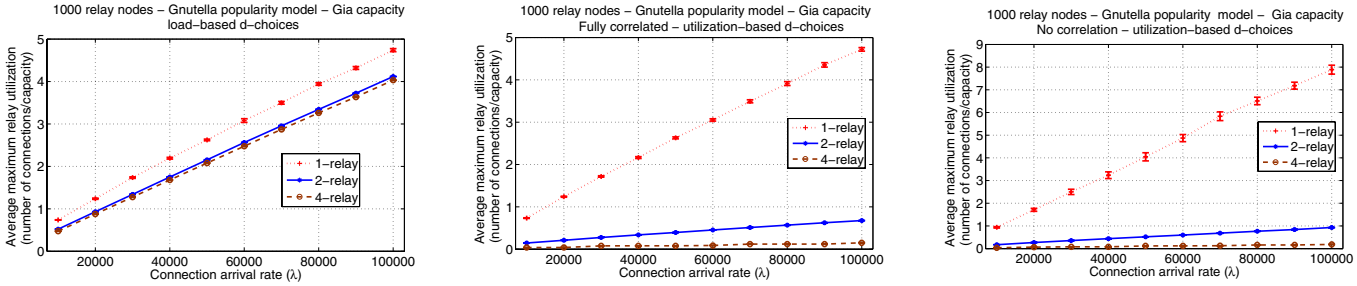

Fig. 5. Average maximum relay utilization as a function of the connection arrival rate: (a) Gnutella popularity model coupled with Gia capacity model using the load-based d-relay algorithm when both metrics are fully correlated; (b) Same scenario but using the utilization-based d-relay algorithm; (c) Same scenario but with both metrics fully uncorrelated.

load-based d-relay algorithm cannot balance the utilization effectively. In fact, the maximum utilization is much above 1, indicating that some relays are overloaded. This motivates the need for a utilization-based d-relay algorithm.

The identical scenario above is considered again but changing only the relay selection algorithm to the proposed utilization-based d-relay algorithm. Figure 5.(b) presents the results showing that utilization is much better balanced in this case. Notice, as expected, that there is no change in the performance of the 1-choice algorithm.

Finally, we consider the same scenario above, but when relay popularity and capacity are uncorrelated. In the previous case, one could argue that the relay selection algorithm receives help from the correlations, as high capacity nodes will be selected more often due to their high popularities. Figure 5.(c) shows the result for the uncorrelated scenario when using the utilization-based d-relay algorithm. Notice that the 2- and 4-relay algorithm perform only slightly worse than in the fully correlated case, thus showing that the algorithm indeed spreads the load to effectively balance utilization. Finally, notice that the performance of the 1-relay is much worse, due to the lack of correlation between popularity and capacity.

VI. Conclusion

This paper addressed the problem of efficiently and effectively distributing load over a set of heterogeneous relays in unstructured overlay systems. We prove a novel result for the maximum load on any relay under a very general relay popularity model when the well-known two random choice algorithm is used for relay selection. Our theoretical result characterizes the performance of the algorithm through a single parameter (α) used to measure the heterogeneity of the relay popularity.

Under the presence of heterogeneous relay capacities, balancing the load does not lead to a balanced utilization of the relays. We propose a simple modification to the two random choices algorithm to consider relay utilization directly. Numerical evaluations through simulations show that the proposed algorithm can effectively distribute the load in order to balance relay utilization, even when relay popularity and capacity are uncorrelated. Although we have not established a theoretical bound on its performance, we conjecture that the proposed algorithm can indeed deliver provable levels of performance. We leave this subject for future investigation.

References

[1] Gnutella development forum. Gnutella for users, 2007. http://basis.gnufu.net/gnufu/index.php/.
[2] F. M. auf der Heide, C. Scheideler, and V. Stemann. Exploiting storage redundancy to speed up randomized shared memory simulations. *Theoretical Computer Science*, 162:245–281, 1996.
[3] Y. Azar, A. Z. Border, A. R. Karlin, and E. Upfal. Balanced allocations. *SIAM Journal on Computing*, 29:180–200, 1999.
[4] S. A. Baset and H. G. Schulzrinne. An analysis of the skype peer-to-peer internet telephony protocol. In *Proc. of IEEE Infocom*, 2006.
[5] P. Berenbrink, A. Czumaj, A. Steger, and B. Vocking. Balanced allocations: the heaviliy loaded case. *Siam Journal Computing*, 35(6).
[6] J. Byers, J. Considine, and M. Mitzenmacher. Geometrics generalizations of the power of two choices. In *Proc. of ACM SPAA*, 2004.
[7] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. Making gnutella-like P2P systems scalable. In *Proc. of ACM SIGCOMM*, 2003.
[8] R. Cole, A. Frieze, B. M. Maggs, M. Mitzenmacher, A. W. Richa, R. Sitaraman, and E. Upfal. On balls and bins with deletions. In *Proc. of RANDOM'98*, 1998.
[9] T. Fei, S. Tao, L. Gao, and R. Guerin. How to select a good alternate path in large peer-to-peer systems? In *Proc. of INFOCOM*, 2006.
[10] S. Guha, N. Daswani, and R. Jain. An experimental study of the skype peer-to-peer voip system. In *Proceedings of IPTPS*, 2006.
[11] J. Han, D. Watson, and F. Jahanian. Topology aware overlay networks. In *Proc. of INFOCOM*, 2005.
[12] P. Karbhari, M. Ammar, A. Dhamdhere, H. Raj, G. Riley, and E. Zegura. Bootstrapping in gnutella: A measurement study. In *Proc. of PAM*, 2004.
[13] M. Mitzenmacher, A. Richa, and R. Sitaraman. The power of two random choices: A survey of the techniques and results. *Handbook of Randomized Computing*, 2000.
[14] M. Raab and A. Stege. Balls into bins - a simple and tight analysis. In *Proc. of RANDOM*, 1998.
[15] R. Ripeanu, I. Foster, and A. Iamnitchi. Mapping the gnutella network: properties of large scale peer-to-peer systems and implications for system design. *IEEE Internet Computing Journal*, 6(1), 2002.
[16] S. Saroiu, P. K. Gummadi, and S. D. Gribble. A measurement study of peer-to-peer file sharing systems. In *Proc. of Multimedia Computing and Networking.*, 2002.
[17] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proc. of Sigcomm 2001*, 2001.
[18] D. Stutzbach, R. Rejiae, and S. Sen. Characterizing unstructured overlay topologies in modern P2P file-sharing systems. In *Proc. of IMC*, 2005.
[19] B. Vocking. How asymmetry helps load balancing. *Journal of the ACM*, 50:568–589, 2003.
[20] U. Wieder. Balanced allocations with heterogeneous bins. In *Proc. of the ACM SPAA*, 2007.
[21] www.skype.com.
[22] C. Xie, S. Gou, R. Rajaie, and Y. Pan. Examinging graph properties of unstructured peer-to-peer overlay topology. In *Proc. of IEEE Global Internet Symposium 07*, 2007.