

An End-to-End Solution to Scalable Unstructured P2P Networking

Nima Sarshar^a, Vwani P. Roychowdhury^b

^a Faculty of Engineering, University of Regina, Regina SK, Canada S4S 0A4
Email: nima.sarshar@uregina.ca

^b Department of Electrical Engineering, University of California, Los Angeles, CA, USA 90029
Email: vwani@ee.ucla.edu

Abstract—Despite many improvements on original unstructured P2P networks, these systems still suffer from many problems, the most important of which are, (a) lack of guarantees on the integrity of the network topology in the face of churns, (b) excessive traffic cost, and (c) poor quality of search results. This paper introduces an end-to-end scalable unstructured P2P networking solution called SUPNET to address many of these issues. The solution consists of two sub-protocols, SUPNET-T and SUPNET-S, which are, respectively, responsible for network management and search. We investigate the end-to-end performance of our solution, both analytically and empirically. SUPNET-T is a scalable, highly robust protocol, capable of utilizing the heterogenous distribution of network resources. The high stability of SUPNET-T is the result of implementation of a novel distributed feedback mechanism. SUPNET-S, on the other hand, is capable of locating every item, even if a single copy of that item exists in the network. SUPNET-S does this while producing a traffic that scales provably sub-linear with the network size. The protocol also contains mechanisms for efficient search of popular items as well as distributed tuning algorithms. All this, along with a relative ease of implementation and a solid analytical foundation, make SUPNET a compelling solution for unstructured P2P networking.

I. INTRODUCTION

Unstructured P2P networks, that are the subject of this paper, are those that impose minimal constraints on the network topology or content distribution. Structured P2P networks, on the other hand, are those networks that strictly control the underlying network structure, content publication strategy and query routing. Many widely deployed and commercially successful P2P networks are unstructured. The lack of structure allows these networks to survive in the highly dynamic and often ad-hoc P2P environments. The main drawback of these networks, however, is the high traffic cost associated with unstructured search. By carefully guiding the queries in the network, on the other hand, structured P2P networks are able to drastically decrease the search traffic and increase the search efficiency. Several novel structured P2P systems, mainly based on Distributed Hash Tables (DHTs), have been proposed in the past few years.

It is known, however, that structured systems are not suitable for all applications. A number of potential problems with them limits their applicability in some scenarios. Firstly, these systems are sensitive to failures, errors and malicious peer behaviors that are frequent in a large scale unsupervised P2P environment [17]. This can incur a large overhead for

constantly monitoring the topology, locating the points of failure and triggering proper remedial procedures. Secondly, performing multi-attribute and complex queries in structured P2P networks is non-trivial. Newly proposed approaches have to go to great lengths to partially mitigate this problem. For complex queries, such as full text search, empirical studies show that the average search traffic in structured networks is comparable to (if not more than) their unstructured counterparts [4].

Nevertheless, structured P2P systems are perfect candidates for distributed storage and retrieval applications involving contents with unique identifiers. Hybrid networks (i.e., those consisting of both structured and unstructured networks) have proved successful in exploiting the points of strength of both systems. The structured Kademia [19], for instance, is used for tracking and torrent location in many, essentially unstructured, P2P systems such as Azureus [18]. More than ever, we believe, structured and unstructured networks are emerging as complementary rather than competing technologies.

Despite many improvements on original unstructured P2P networks such as Gnutella, several problems still exist. The most important of these problems are the lack of guarantees on the integrity of the network topology, excessive traffic caused by searching and even more importantly the poor quality of the returned search results. This paper introduces an end-to-end Scalable Unstructured P2P **NET**working solution called SUPNET to address many of these issues. The protocol consists of two functionally separate parts, a network formation and topology management protocol which we call SUPNET-T, and a distributed P2P search protocol, called SUPNET-S, based on percolation search algorithm (PSA) introduced by authors in [3]. SUPNET-T ensures the emergence of a network infrastructure with proper global statistical properties suitable for scalable operation of PSA (and hence SUPNET-S). The remainder of this introduction reviews these two major components.

A. Network Formation and Topology Management

Early P2P networking proposals, such as Gnutella are generic search protocols in the sense that they, deliberately, do not specify how the topological structure of the underlying network should evolve. Other proposals, such as the random-walk search of Lv et al. [5] or that of Adamic et al. [14] only

specify the preferred network topology, without prescribing a way to ensure the emergence of such preferred structures.

On the other hand, many generic network formation protocols have been developed that can lead to emergence of networks with some perceived desired properties. These works, a number of which are reviewed later on in this introduction, mainly follow this heuristic; the larger the degree of a node, the more search traffic and computational burden the node has to tolerate. The main goal, therefore, should be to ensure that nodes with higher degrees are those with more resources (usually bandwidth).

The performance of a search algorithm strongly depends on the underlying topology. As such, network formation protocols should be designed in close consideration of the search algorithms that will run on the emerging networks. Thus, designing algorithms which ensure the emergence of proper topologies to foster a successful search algorithm is as important as designing the search algorithm itself. This approach has been implemented, to various degrees, in most modern P2P proposal. Gnutella V2, the Fast-Track proposal (used in KaZaA) and GIA [15], for instance, all have some form of active network management provisions.

SUPNET-T, the topology management sub-protocol of SUPNET, is based on the above view point. SUPNET-T ensures that a network with proper statistical properties emerges even in a dynamic, heterogenous environment, where nodes frequently join and leave the network. SUPNET-T divides the nodes in the network into different classes (indexed by a parameter q) based on their available resources and willingness to participate in the P2P operation. The degree distribution of the network restricted to nodes of a given class will be shown to obey a power-law; i.e, the number of nodes of type q with degree k scales as $p_q(k) \sim k^{-\gamma_q}$ in steady state. For a smaller value of γ_q , a larger fraction of nodes of type q are expected to assume larger degrees. To distribute the load according to available resources, the protocol ensures that only the classes of nodes with high resources assume a small power-law exponent. To allow for scalable operation of the underlying search algorithm, SUPNET-T ensures that at least one of the classes of the nodes in the network assumes a power-law exponent less than 3. Such networks, often called networks with heavy-tailed power-law degree distributions, are particularly suitable for the operation of Percolation Search Algorithm (PSA), which is the search engine of SUPNET-S, discussed later in this introduction.

SUPNET-T protocol has the following properties. **(a) Locality:** The protocol is local, and stateless, in the sense that the interactions of a node with the network are only determined by the state of the node itself. **(b) Matching Roles and Resources:** The protocol recognizes the heterogeneity of the nodes in the network and is capable of automatically correlating, in a statistical sense, the degree of the nodes and their level of resources. **(c) Scalability:** Some P2P networks are expected to grow to hundreds of millions of nodes. The overhead in maintaining such huge networks can be high. Also, one needs to consider the scaling of topological properties of interest as a function of the network size. For a network of size N , SUPNET-T, on average,

requires $O(\log N)$ communications when a node enters or leaves the network. **(d) Robustness:** SUPNET-T has a built-in distributed feedback algorithm which makes it extremely robust to churns and transitions. **(e) Analytical Tractability:** Important topological properties of the networks emerging from SUPNET-T can be tracked analytically. Of particular interest is the region in the parameter space for which the emerging network topology is stable. On the converse side, given a set of desired topological features, these analytical relations enable a designer to provide guidelines for choosing protocol parameters.

B. Unstructured P2P Search Protocols

SUPNET-T is capable of shaping global topological network structure suitable for scalable operation of the search protocol SUPNET-S. SUPNET-S is a protocol that builds upon Percolation Search Algorithm (PSA) recently introduced by the authors in [3]. SUPNET-S is capable of solving many problems often associated with currently deployed P2P search algorithms, some of which are reviewed below.

Unstructured P2P searches are considered to be “open ended” [6] in the sense that it is not clear where to stop the search and conclude that the item does not exist in the network, neither when a reference to an item is found, the existence of the actual data is guaranteed. Also, one cannot guarantee that the set of hits returned are in fact *all* the results existing in the network. Finally, the quality of the search, including search time, highly depends on the number of hits returned; for rare items, the search might take an excessively long time to return a result, if any. The authors of [7], for instance, found that at least 40% of all queries for rare items on Gnutella do not find any reasonable number of hits, with 18% of them not returning any hits at all, even though the items actually existed in the network. This made a case for hybrid schemes to use structured P2P systems, in conjunction with deployed unstructured systems, merely for searching rare items.

SUPNET-S, in principal, can operate on any network topology. The performance of SUPNET-S, however, depends on the statistical properties of the underlying network. By borrowing the results in [3] (see Appendix) SUPNET-S can be shown to have scalable search traffic and guaranteed query result if the underlying network topology is random with a heavy-tailed power-law degree distribution. As stated earlier, these properties are satisfied by networks emerging from SUPNET-T. Apart from PSA as its main search engine, SUPNET-S contains several provisions for more efficient search of popular items, cache management and distributed parameter estimation.

SUPNET-S borrows the following properties from the PSA. **(1) Mathematical Guarantees on Search Quality:** Unlike many other known proposals, PSA does not rely on the frequency of contents in the network. It guarantees to locate *every* content even if there is only a single copy of it in the network. **(2) Scalable Overall Search Traffic:** Simple network flooding for search can produce guaranteed search results too. Flooding, however, generates a great deal of unnecessary traffic that will soon overwhelm the network as a

whole. With flooding, each query passes through every single link in the network, generating an overall traffic of the order of $\Theta(N)$. As reported in the Appendix, however, the overall traffic generated by PSA is sub-linear in N . **(3) Guaranteed Scalable Search Time:** PSA is massively parallel. Every single query is answered in time $O(\log N)$ with probability one, regardless of the number of hits returned. **(4) Scalable Cache Overhead:** In PSA, the number of redundant duplicates of each content in the network can be as low as $O(\log N)$ (see the Appendix). This should be compared with random-walk based search strategies proposed in [5]. There, to ensure that a copy of the content is found in time $O(\log N)$, one requires at least $\Theta(N)$ copies of the content to be randomly distributed in the network. **(5) Minimum Redundant Messaging:** In PSA, due to the critical random messaging, the effective topology a query message propagates on is very close to a tree, forcing a query to arrive at any node only through at most one connection. This would eliminate the need for some sophisticated provisions suggested for optimizing query propagation in flooding applications (see e.g., [8]).

C. Related Works

Many protocols for management of unstructured P2P networks have been proposed so far. The main ingredient of these protocols is the redirection of connections to nodes with higher capabilities, or those with less load.

SCAMP [9] uses random walks to build graphs suitable for gossiping. The main goal of SCAMP is to build graphs where the average node degree is proportional to the log of the number of nodes. SCAMP is interesting in that it is able to ensure a global feature (i.e., the average degree) through local interactions. SUPNET-T, however, goes far beyond the average degree; we are able to tune the whole degree distribution through local dynamical rules. Phenix [10], is a distributed algorithm that implements an indirect form of preferential attachment. As such, some form of power-law degree distribution is expected to emerge. Unfortunately, the conditions under which a stable, heavy-tailed, power-law network arises in a dynamic environment is unknown. This is particularly important given that preferential attachment alone is known to be inadequate for maintaining a power-law structure in the presence of heavy node deletion [1]. Unlike Phenix which encourages a skewed degree distribution, Araneola [11] builds almost regular graphs. So does the algorithm proposed in [12], which is a distributed mechanism to construct regular random graphs.

As for the search algorithms, we will consider the super-node proposal (used in KazaA) and that of Adamic et al. [14] based on random-walk in power-law networks. Both of these proposals have an excellent performance when used to search popular items, i.e., those items published by many nodes in the network. As our simulations in Section IV suggest, however, they are incapable of efficiently locating rare items (i.e., those items published by a single or very few nodes in the network) without generating an excessive amount of traffic.

We start from the SUPNET-T protocol in Section II, where we introduce the protocol and provide a brief analysis of

its properties. In Section III, we introduce the SUPNET-S protocol. Simulations are reported in Section IV to compare the performance with some competing proposals.

II. SUPNET-T PROTOCOL

Throughout this paper, $q \in \{1, 2, \dots, Q\}$ will represent the type or class or category (all three terms are used interchangeably) of a node in a network in which nodes can belong to one of Q different classes. These categories of nodes correspond to the heterogenous distribution of the network resources in the network; each class represents the set of nodes with close resources that employ the same protocol parameters.

A. SUPNET-T Protocol Modules

SUPNET-T consists of four major parts:

- **Target Selection Algorithm:** The target selection algorithm (TSA) chooses candidate nodes for new link connections. Starting from *any* bootstrap node, the algorithm performs a random walk of length L and returns the last node in the walk.
- **Linkage Algorithm (LA):** To add a link, a node S calls the TSA to obtain a candidate for connection and sends a “request to connect” message to the candidate node. If the target node accepts the request, a new link is added between the two nodes, otherwise, S will repeat the procedure, by selecting another candidate target node until it finds a node that accepts the link.
- **Link Acceptance Rule:** Each node T upon receiving a request for connection, will accept the connection with probability d_q and rejects it with probability $1 - d_q$, for some constant d_q depending only on q , the type of the node T .
- **Compensation Algorithm (CA):** If a node of type q loses a link due to any of its neighbors going down or logging off, with probability n_q , depending only on q , it will add a new link using the Linkage Algorithm.

The choice of the protocol parameters $n_q, d_q, \forall q = 1, 2, \dots, Q$ and L as discussed later in this section. Evidently, except for the requirement for a bootstrap node, all algorithm parts are completely local and do not require any form of global information. Moreover, there is no need for any node to reveal its type or any of the parameters during the interaction with other nodes. Obtaining a bootstrap node can be performed by, now standard, procedures (such as bootstrap servers).

B. Understanding SUPNET-T

Many “naturally occurring” networks are known to have power-law degree distributions, i.e., the number of nodes with degree k scales as $k^{-\tau}$ for some exponent $2 < \tau$, at least for large k . Networks with approximate power-law degree distributions include, WWW, the Internet and some early instances of Gnutella network. Many dynamical models have been proposed that give rise to networks with power-law degree distributions. Most of these dynamics are based on a variant of preferential attachment for linkage.

Preferential attachment is the linkage rule in which nodes with higher degrees are chosen more often for connection. In linear preferential attachment, a node with degree k is chosen for connection with probability proportional to k . TSA in SUPNET-T is in effect an approximation of linear preferential attachment.

The main ingredient of TSA is the choice of random walk of length L for node sampling. For large L , the random walk on any network will mix, i.e., the probability that a node of degree k is returned by TSA converges to ck for some constant c independent of k . It turns out that for most randomly generated networks, the value of L for the random walk to mix is only $O(\log N)$ [13]. Therefore, initiating connections by sampling nodes through a short random walk closely approximates linear preferential attachment.

Compensation Algorithm (CA) introduces a form of distributed feedback by locally reacting to the loss of any link in the network [1], [2]. SUPNET-T is therefore a reactive algorithm; nodes react to the changes in the network with varying reaction magnitudes, determined by n_q . One purpose of CA is to ensure the stability of SUPNET-T. Also, CA ensures that a power-law degree distribution emerges given that nodes constantly join and leave the network.

The role of Link Acceptance rule is to help direct new links to nodes that belong to a category with more resources. By setting d_q to a small value, a class q will redirect most of the new links to nodes in other classes.

The interaction of protocol modules in SUPNET-T is a very complex one. Unlike many other protocols, however, many properties of SUPNET-T can still be analytically traced. To our special interest is the degree distribution of the emerging subnetworks in steady states, which is the subject of the next subsection.

C. Modeling SUPNET-T

We can model the state of SUPNET-T networks in time as a continuous time Markov process. There are four distinct transitions in SUPNET-T; addition of a new node (log-in), deletion of an existing node (log-off), addition of new links due to node log-in, deletion of an existing link due to node log-off.

Assuming that log-in and log-off of nodes are memoryless processes, the underlying continuous time Markov process can be discretized into a discrete time Markov chain. In this case, a time step is defined as a single state transition in the chain. In other words, we can model the network dynamics by a sequence of events ordered by the time of their occurrence. Therefore, we can *index* time by defining time steps as the time of occurrence of a new event. Dynamical parameters of SUPNET-T can be modelled as follows.

- **Node Login Rates** To model this, we assume that at each time step, a new node is introduced into the network, which can belong to any of the Q different classes. The probability of the new node being of type q is assumed to be s_q , where $\sum_q s_q = 1$.
- **Node Departure Rates** Nodes of the network can log-off due to voluntary departures or failures. To model

this, we assume that at each time step, for each class $q = 1, 2, \dots, Q$ a randomly selected node of type q and all its links are deleted with probability c_q . Thus, the total deletion rate is $c = \sum_{i=1}^Q c_q < 1$.

The network parameters s_q, c_q as well as protocol parameters n_q, d_q represent the heterogeneity in the population, attraction/attachment, stability and responsiveness dynamics that characterize the different categories of nodes. Parameters s_q, c_q are empirical and out of the control of the protocol designer. n_q, d_q , however, are design parameters that can be tuned in the protocol implementation.

The degree distribution of each of the Q subnetworks of a SUPNET-T network is a power-law, the exponent of which (γ_q) depends on all the dynamical parameters s_q, c_q, n_q, d_q . We have been able to analytically derive γ_q as a function of these dynamical parameters for some interesting special cases. For instance, when all acceptance probabilities are equal to 1 (i.e., $d_q = 1, \forall q$), γ_q 's can be found as follows [2]:

$$\gamma_q = 1 + \frac{s_q}{\beta_q(s_q - c_q)} \quad (1)$$

where,

$$\beta_q = \frac{m}{B} - \frac{c(1 - n_q)}{s_q - c_q} + \frac{\sum_{q'} n_{q'} B_{q'}}{B} \times \sum_{q''} \frac{c_{q''} B_{q''}}{s_q - c_q} \quad (2)$$

and $B = \sum_q B_q$.

B_q 's can be found by solving the following non-linear set of equations:

$$\begin{aligned} B_q &= m \left(s_q + \frac{B_q}{\sum_{q'} B_{q'}} \right) - \frac{c_q}{s_q - c_q} (1 - n_q) B_q \\ &- \frac{B_q}{\sum_{q'} B_{q'}} \left(1 - \sum_{q''} n_{q''} \frac{B_{q''}}{\sum_{q'} B_{q'}} \right) \\ &\times \left(\sum_{q''} \frac{c_{q''}}{s_{q''} - c_{q''}} B_{q''} \right) \end{aligned} \quad (3)$$

D. Power-law Degree Distributions

The main structures in SUPNET-T networks are the power-law degree distributions within each of its network subclasses. But why are power-law exponents important? It turns out that the exponent γ_q of each class directly influences the connectivity structure of the nodes in that class. When γ_q is small, the nodes in class q are more probable to acquire a large number of connections and, thus, turn into "hubs". To see this, suppose for a moment that the network consisted of only two classes ($Q = 2$) with equal sizes. Consider two subclasses 1, 2 with power-law exponents γ_1, γ_2 . Now randomly pick a node that has a degree greater than some k . Let $\delta_q(k)$ be the probability that this node is of type q for $q = 1, 2$. It then follows that $\frac{\delta_1(k)}{\delta_2(k)} \approx k^{\gamma_2 - \gamma_1}$. Therefore, for $k \gg 1$, the majority of the nodes with degree greater than k would come from a class with smaller power-law exponent.

Conversely, one can ensure that the nodes of a given type q will not assume high degrees by ensuring that their corresponding power-law exponent γ_q is large. Likewise, if a class of nodes has more resources and capabilities that need to be utilized by assuming higher degrees, then it would be desirable to decrease the power-law exponent of the class to which the node belongs. Results similar to those reported in Subsection II-C, can be used to find the relation of the dynamical variables to the power-law exponents in each of the classes which allows the designer to tune these exponents by locally varying the protocol parameters. To ensure the stability of the system, one should ensure that $\gamma_q \geq 2$ for all q . Also, for scalable operation of the SUPNET-S protocol, introduced in the next section, it is desirable to have at least one of the exponents $2 \leq \gamma_q < 3$. To ensure that a class q' of nodes is effectively shielded from the search traffic, the designer might choose a value $\gamma_{q'} > 4$ for the power-law exponent of that class.

III. SUPNET-S SEARCH PROTOCOL

SUPNET-S is the sub-protocol of SUPNET responsible for P2P search. SUPNET-S builds upon the Percolation Search Algorithm in [3]. It also adds provisions for efficient search of rare items, cache management and distributed parameter estimation.

A. SUPNET-S Modules

SUPNET-S consists of 4 different modules, as follows:

(1) Publication: To publish a content c by a node v , a random walk of length L is started from v . A copy of c (i.e., its metadata) is cached at all the nodes visited. This is called content implantation or caching.

(2) Phase I Search: Any search starts at Phase I. To search for an item c , a random-walk of length L is initiated (the Time-to-Live or TTL of the message is set to L). The query specifies the minimum number of hits desired. If a node along the walk finds a match, it notifies the original requester and increments a counter in the query header. At each hop, the TTL is decremented by one. The walk ends when the TTL expires. The query in this state is called Forward Random-Walk Message (FRWM).

(3) Phase II Search: The last node who receives the FRWM (for which the TTL is zero) checks to see if enough hits have been returned by the nodes in the random-walk. If not, it sends a backward message, which travels back (in the opposite direction) along the same path established by the forward random-walk. The query in this stage is called Backward Random-Walk Message (BRWM). If a node receives a BRWM, it does the following: to each of its neighbors from which it has not received the current query, it passes a copy of the query with probability p (the query at this state is called a Percolation Message or PM). The process continues recursively by any node who received the PM.

(4) Parameter Tuning Algorithm: The random-walk length L and the percolation probability p are protocol parameters that change slowly as the network size evolves and thus should be tuned from time to time. The tuning works by

searching for a special item called a *void item*. All nodes are assumed to possess a void item and should respond to searches for it. The tuning algorithm uses a bisection procedure to search for the smallest value of p for which a search for void item returns a number of hits more than a threshold (we use $5 \times L$ for the threshold). It then sets the percolation probability to *twice* this minimum value. The value of L is then updated as $L = 8 \ln 1/p$.

(5) Cache Update Algorithm: Each node is required to maintain a list of nodes on which a given content has been cached. When logging off, or when un-publishing a certain content, a node notifies other nodes to remove a given content from their cache. Also, each cached item has a life-time. A content should be removed from the cache when its life-time expires. To continue to publish a content, a node should refresh the publication procedure.

B. Understanding SUPNET-S

1) Search in Phase II: The main ingredient of SUPNET-S is the Phase II search protocol which, essentially, is an implementation of the Percolation Search Algorithm (PSA). PSA is best suited for random networks with heavy-tailed and in particular, power-law degree distributions. SUPNET-T protocol introduced in the first part of this paper can ensure the emergence of a network with these properties. The work in [3] proves several scalability results of PSA for such networks. In particular, PSA can guarantee the quality of search results while generating a search traffic that (unlike flooding) is provably sub-linear with network size. We provide a simplified analysis of PSA on heavy-tailed random power-law networks. We only report the analysis that is essential to other developments in the paper. Please refer to [3] for a complete treatment of PSA.

Consider a random network with an approximate power-law degree distribution, $p_k \approx k^{-\tau}$ for $k \leq k_{max}$ where $k_{max} \sim N^{1/\tau}$ is the maximum degree. If $2 \leq \tau < 3$ the network is said to have a heavy-tailed degree distribution. When $2 < \tau < 3$, the average degree of the nodes is finite and independent of N . For $\tau = 2$, the average degree scales as $\langle k \rangle = O(\log N)$. The variance of the degree distribution, however, scales polynomially in N . It is easy to verify that $\langle k^2 \rangle = \Theta(k_{max}^{3-\tau}) = \Theta(N^{\frac{3-\tau}{\tau}})$.

Intuitively, a large variance implies that a small, but considerable, number of nodes in the network have very large degrees. These nodes, which we call them high-degree nodes, influence all topological properties of heavy-tailed networks. In particular, it can be shown that even if most of the links in the network are deleted randomly, a large connected component still remains in the network which contains most of these high-degree nodes. Let Γ be the set of all nodes with degree greater than $k_{max}/2$. We are interested in the probability that a random-walk of length L passes through a node in Γ . Assuming that the random-walk mixes, this probability is approximated as:

$$z \approx 1 - \left(1 - \frac{\sum_{k=k_{max}/2}^{k_{max}} k p_k}{\langle k \rangle} \right)^L \geq 1 - L k_{max}^{2-\tau} (1 - 2^{2-\tau})$$

Thus, by taking $L = O(k_{max}^{\tau-2}) = O(N^{\frac{\tau-2}{\tau}})$, one can ensure that a random walk of length L passes through a node with degree more than $k_{max}/2$ with high probability. PSA uses a random-walk of this length to cache contents. If $p = \kappa p_c = \kappa \frac{\binom{k}{k/2}}{\binom{k}{k/2}} \sim N^{-\frac{3-\tau}{\tau}}$, it can be shown that a percolation message will be received by all nodes in Γ almost surely [3]. This implies that, by using at most a vanishingly small fraction ($N^{-\frac{3-\tau}{\tau}}$) of all links in the network, all contents can be found with probability one.

2) *Search in Phase I and Rare vs. Popular Items:* While PSA provides a rigorous answer to efficient search of rare items, it can generate an unnecessary amount of traffic when the search is for popular items. In fact, empirical studies suggest that most P2P search queries are for popular items. A number of proposals exploit this property to enhance their search performance. In expanding ring flooding algorithms [5], flooding is done with a gradually increasing Time-to-Live (TTL) value, until enough hits are returned to a query. The idea is that if the item is popular enough, it can be found in a close neighborhood of the node itself without having to resort to total flooding. The random-walk based search algorithm in [14] (on which GIA protocol in [15] is based) uses a random-walk on a heavy-tailed network for search. The walk gravitates towards the high degree nodes, which are expected to have a larger number of contents cached in them, which ensures that popular items are efficiently found. To ensure that *all* contents (rare and popular) have been found, however, the search generates essentially the same traffic as total flooding.

The Phase I of the search is actually an attempt to find the popular items without initiating a percolation search. Only if enough number of contents are not found, does the search go into Phase II (which is percolation search). To see how Search in Phase I is capable of finding popular items, consider the following. As argued earlier, a random-walk of length L ensures that one copy of each content can be found in one of the nodes in Γ .

Now the total number of nodes in Γ is

$$|\Gamma| \sim N \sum_{k_{max}/2}^{k_{max}} p_k \sim N \cdot k_{max}^{-\tau+1} \sim N^{1/\tau}$$

Any query hits at least one of these nodes with high probability. Take a specific query q for a content c and let v be the high-degree node encountered through a random walk of length L . If a copy of c is cached at v , then the search is successful. Suppose that M nodes in the network have published the content c . The probability that none of these contents is cached at v is approximately $(1 - 1/N^{1/\tau})^M$. For this probability to be small, it suffices to have $M = O(N^{1/\tau})$. A more careful calculation leads to the following result: *If all contents are published through a random walk of length $L = O(N^{\frac{\tau-2}{\tau}} \log N)$, then any content published by a fraction $\alpha = O(N^{-\frac{\tau-1}{\tau}} \log N)$ of all nodes is almost surely found through a random-walk of length L starting from any node in the network.*

Note that the fraction α goes to zero as N goes to infinity. In other words, if a content c is published by an infinitesimally small fraction of the nodes in the network, then it can be

found through a random-walk of length $L = O(N^{\frac{\tau-2}{\tau}} \log N)$. Interestingly, for $\tau = 2$, the length of the walk is only $O(\log N)$. This result should be compared to some other schemes which require a walk length inversely proportional to the number of replicas of the content in the network.

3) *The Parameter Tuning Algorithm:* Our protocol uses two control parameters, the walk length L and the percolation threshold p . As stated before, the walk length should slowly increase as the network size grows, while the percolation probability should slowly decrease. The performance of PSA depends on the critical choice of the percolation probability. A smaller value of p directly results in a smaller traffic overhead. If p is reduced below a critical threshold (i.e., percolation threshold p_c), the search will be inefficient. The number of nodes that receive a percolation message undergoes a phase-transition as the percolation probability varies. This fact is used to fine tune the percolation probability p .

Lets assume that $p = \kappa p_c$ for some constant $\kappa > 0$. If $\kappa < 1$ (sub-threshold percolation) the percolation message will be received only by a small number of nodes. If $\kappa > 1$, on the other hand, the number of nodes that received the message jumps drastically (e.g., a phase transition at $\kappa = 1$). Ideally, one would like to have κ around 2 (see [3]). A bisection algorithm can be used to estimate the location of the phase transition.

The walk length can be estimated from the percolation probability as well. The initial value of the walk length is obtained from the bootstrap nodes. Once the percolation probability is updated, the walk length can be updated through the following formula $L \approx \beta \log(1/p)$, for some constant $\beta > 0$ in the order of one. This formula is based on the assumption of a random power-law network with exponent $\tau = 2$. We found the formula to work fine in all simulations reported for $\beta = 8$.

In the next section, we examine the performance of SUPNET using dynamical parameters that emulate the behavior of real-world unstructured P2P networks.

IV. END-TO-END PERFORMANCE OF SUPNET

This section provides detailed simulation results for an end-to-end P2P system that employs SUPNET-T for network management and SUPNET-S for search. The network simulated in our case consists of three categories of nodes that roughly indicate the three major types of P2P clients: (i) The category of low capacity nodes resembling dial-up, DSL and ADSL connections, which we call Modems. This comprises 60% of all nodes. (ii) The class of medium capacity nodes, representing broadband home users, which we call Cable nodes, that correspond to 27% of all nodes. (iii) The set of very high capability nodes, that represent university and office connections, which account to 13% of the nodes, and are called T3 users. Table IV summarizes the the parameters used in the SUPNET-T protocol for each of the classes. γ_{sim} is the approximate power-law exponent empirically observed in each class at steady state.

We compare our system with the the random-walk scheme of Adamic et al. (also used in GIA) as well as a Super-Node structure that employs expanding rings for search and

Class (q)	s_q	c_q	n_q	d_q	size	$\gamma_q(sim)$
0 (T3)	0.23	0.17	0.8	1	13%	1.98
1 (Cables)	0.27	0.21	1	0.7	27%	2.23
2 (Modems)	0.50	0.28	1	0.3	60%	5.16

TABLE I
SIMULATION PARAMETERS

resembles some of the widely deployed P2P applications like KaZaA. For the sake of comparison, to evaluate the proposal of Adamic et al., we have used a random power-law topology, for which the original search algorithm has been designed and analyzed in [14]. The power-law exponent is chosen to be $\tau = 2.2$. For the Super-Node structure, we have assumed a random graph with average degree 40 among the super nodes, where as the average number of ordinary nodes per super node is 100. These are meant to resemble the statistics of the KaZaA network suggested in [16].

While all three systems have scalable performance for searching popular items, only our system can guarantee search results when the traffic is for rare items while generating a scalable amount of traffic. In all comparisons, the total number of nodes and links for the three systems compared are kept roughly the same; the average degree of the network for our scheme is 1.7 and 1.4 for the super node structure and 2.6 for GIA.

A. Stability During Churns

We start by verifying the stability of our network management protocol, SUPNET-T. Our goal is to show that the average and variance of degree distribution of each subnetwork behaves graciously even in the face of churns that severely affect the network size. Fig. 1 shows the size of the network as a function of normalized time. The network size grows until time unit 25, where the deletion rates are suddenly increased, resulting in the size of the network to shrink until the time unit 55, where the network size reduces to roughly 1/3 of its original size. Then the deletion rates are reduced and the network size starts to grow. Fig. 1(b,c) show that even in the face of such extreme network fluctuations, network statistics remain fairly stable. Next, we compare the search traffic for both popular and rare items.

B. Scaling of Total Traffic

1) *Search for Popular Items:* Fig. 2 compares the normalized total traffic generated for searching a popular item. Each popular item is owned by 1% of all nodes. All systems have at least 99% hit-rate. All three systems have more or less the same performance, in particular, the fraction of links that need to be traversed goes to zero almost as $\sim N^{-1}$, indicating that roughly a constant number of nodes should be inspected until the items are found with high probability.

2) *Search for Rare Items:* The real challenge comes when the contents are rare. Fig. 3 depicts the guaranteed hit-rate vs. the fraction of links traversed when searching for random rare items. Each network has 100,000 nodes and roughly the same average degree (around 2). For the random-walk scheme of Adamic et. al, for different random-walk lengths we have

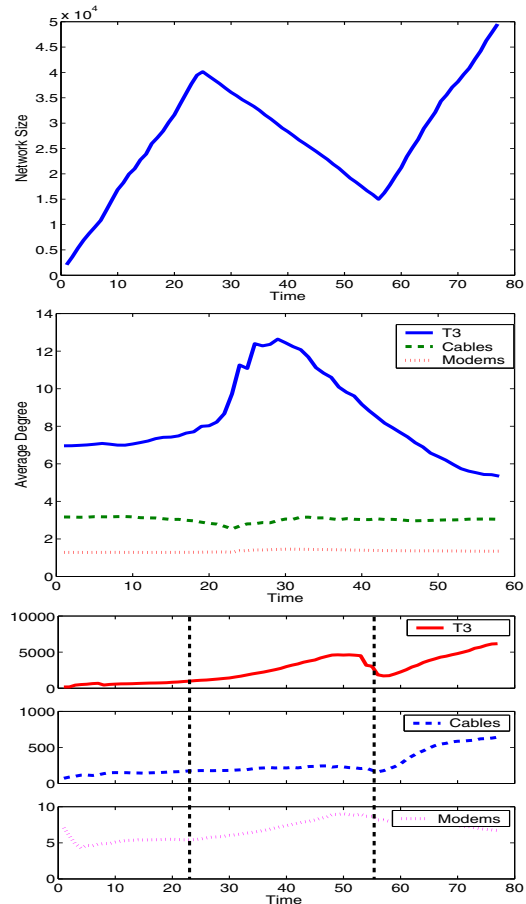


Fig. 1. (top) The size of the network as a function of normalized time. (bottom) Average degree of the nodes in various classes.

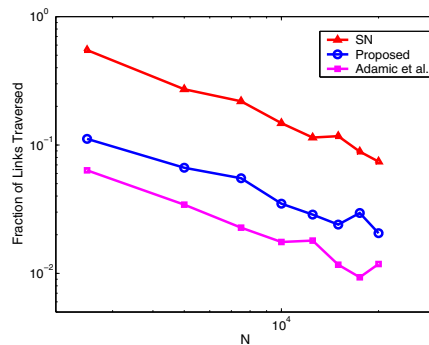


Fig. 2. The fraction of links traversed as a function of the network size N for random popular items.

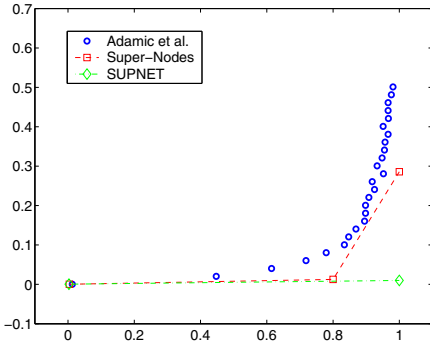


Fig. 3. The hit-rate vs. the fraction of links required for finding a rare item.

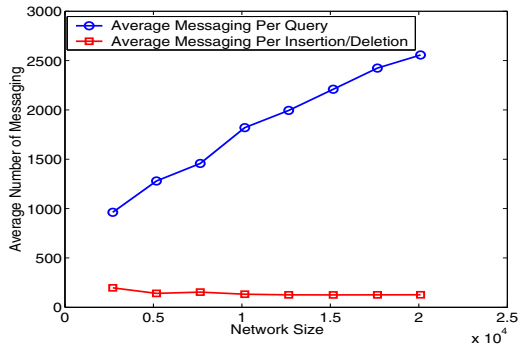


Fig. 4. The scaling of the traffic with the network size. (circles) The average number of messages required for a search of a rare item. (squares) The average number of messages passed for a random node insertion and deletion.

found the hit-rate, as depicted with circles in Fig. 3. For the Super-Nodes proposal, we have used the expanding ring search between the super nodes. Due to the large average degree of the super nodes (i.e., 40), the network diameter is only 3. Thus, there are only 3 different hit-rate values corresponding to TTL=1,2,3. As is evident from Fig. 3, a hit-rate of almost 80% can be guaranteed with a negligibly small traffic. To ensure a better hit-rate, TTL=3 should be used, which essentially is a total flooding within the super nodes. The situation should get worse as the network size grows. SUPNET, however, has a very negligible search traffic throughout. It can guarantee a hit-rate better than 99% by using less than 1% of all links in the network. Furthermore, this fraction will decrease as the network size grows.

C. Distribution of the Traffic

The goal of the network formation algorithm SUPNET-T is to ensure that different subclasses of nodes assume a degree distribution that fits the distribution of their resources. Classes of nodes with more resources will have smaller power-law exponents. This is required because most messages in PSA will be relayed on the nodes with higher degrees which in turn mostly belong to classes with smaller power-law exponents. Fig. 5 plots the distribution of the search traffic on nodes of different types for a network of size 68,000 nodes.

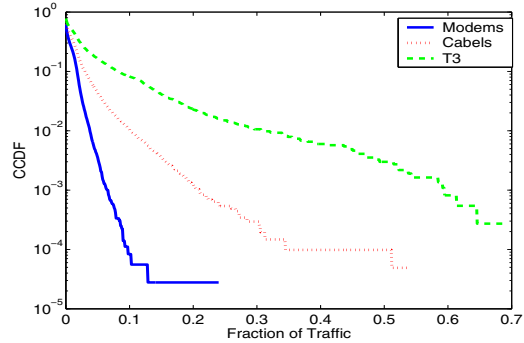


Fig. 5. The complementary cumulative distribution function (CCDF) of the search traffic, i.e., the probability that a node receives more than a certain fraction of all rare search queries: Modem node are least likely to receive a search query (the CCDF curve lies beneath others), followed by Cable and T3 nodes. The probability that a modem node receives more than 5% of all queries to rare items is less than 0.3%, and is 17% for a node of type T3.

APPENDIX

The following theorem has been proved in [3].

Theorem 1: For $\tau = 2$, PSA with high probability (w.h.p) can find any content (even if there is a single copy of the content in the network) within latency of only $O(\log N)$ requiring an average cache size of only $O(\log N)$. PSA achieves all this by requiring only $O(\log^2 N \sqrt{N})$ message passing per query. The percolation probability should be chosen as $\kappa \log(N)N^{-1/2}$, for some proper constant κ in the order of one and a walk length of $L = O(\log N)$. For $2 < \tau < 3$ on the other hand, the search time is still $O(\log N)$, while the average cache size is $O(N^{1-2/\tau})$, producing a traffic per query of only $O(N^{2-3/\tau})$. The percolation probability should be chosen as $\kappa' N^{1-3/\tau}$, for some proper $\kappa' > 0$ in the order of one and a walk length of $L = O(N^{2-2/\tau})$.

REFERENCES

- [1] Removed for anonymity
- [2] Removed for anonymity
- [3] Removed for anonymity
- [4] Yong Yang, Rocky Dunlap, Michael Rexroad and Brian F. Cooper. Performance of Full Text Search in Structured and Unstructured Peer-to-Peer Systems. IEEE INFOCOM'06
- [5] Qin Lv, Pei Cao, Edith Cohen, Kai Li and Scott Shenker. Search and replication in unstructured peer-to-peer networks, ICS '02.
- [6] D. Doval and Donal O'Mahony, Overlay Networks: A Scalable Alternative for P2P, IEEE Internet Computing, Vol. 7, No. 4, 2003.
- [7] Boon Thau Loo, Ryan Huebsch, Ion Stoica and Joseph M. Hellerstein, The Case for a Hybrid P2P Search Infrastructure, IPTPS' 04
- [8] R. A. Ferreira, M. K. Ramanathan, A. Grama, and S. Jagannathan. Randomized Protocols for Duplicate Elimination in Peer-to-Peer Storage Systems. IEEE P2P'05
- [9] A.J. Ganesh, A.M. Kermarrec, L. Massoulie. Peer-to-Peer Membership Management for Gossip-Based Protocols. IEEE Trans. Computers 52(2):2003
- [10] R. H. Wouhaybi and A. T. Campbell. Phenix: Supporting resilient low-diameter peer-to-peer topologies. INFOCOM '04
- [11] R. Melamed and I. Keidar, Araneola: A Scalable Reliable Multicast System for Dynamic Environments, NCA, 2004
- [12] J. Sum, S.C. Lau: A Novel Connection Algorithm for P2P Network. PDPTA 2004.
- [13] C. Gkantsidis, M. Mihail, and A. Saberi. Random walks in peer-to-peer networks. In Proc. Infocom, Mar. 2004
- [14] L.A. Adamic, R.M. Lukose, A.R. Puniyani, and B.A. Huberman. Search in Power-law Networks. Phys. Rev. E, 64 46135, 2001.
- [15] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, Making Gnutella-Like P2P Systems Scalable, SIGCOMM, 2003

- [16] J. Liang, Understanding KaZaA, (2004)
<http://cis.poly.edu/~ross/papers/UnderstandingKaZaA.pdf>
- [17] J.S. Kong, J.S.A. Bridgewater, V.P. Roychowdhury, A General Framework for Scalability and Performance Analysis of DHT Routing Systems, DNS'06
- [18] www.Azureus.com
- [19] P. Maymounkov and D. Mazieres, "Kademlia: A Peer-to-peer Information System Based on the XOR Metric", IPTPS'02