Simulating a File-Sharing P2P Network

Mario T. Schlosser, Tyson E. Condie, and Sepandar D. Kamvar

Department of Computer Science
Stanford University, Stanford, CA 94305, USA

Abstract. Assessing the performance of peer-to-peer algorithms is impossible without simulations since testing new algorithms by deploying them in an existing P2P network is prohibitively expensive. However, some P2P algorithms are sensitive to the network and traffic models that are used in the simulations. In order to produce realistic results, we therefore require simulations that resemble real-world P2P networks as closely as possible. We describe the *Query-Cycle Simulator*, a simulator for file-sharing P2P networks networks. We link the Query-Cycle Simulator to measurements on existing P2P networks and discuss some open issues in simulating these networks.

1 Introduction

Peer-to-peer research has encompassed promising work on algorithms in a variety of directions, including distributed protocols to construct efficient P2P network topologies, search algorithms for unstructured P2P networks, incentives to combat freeriding on P2P networks, and algorithms to determine reputation of peers in a network, among others. Due to the decentralized nature and fast growth of today's P2P networks, testing such algorithms in a real-world environment by simply deploying them on an existing P2P network and collecting data on their performance is a daunting task. In some cases, measurements are easier to carry out due to some easily accessible central control entity in the network that manages node joins and departures [14]. Also, some algorithms may be tested by deploying them on one or a few controlled nodes in the network (as in [15]). However, for a wide range of P2P-related algorithms and protocols, simply deploying and testing them on existing P2P networks is not possible. For example, most algorithms require each peer in the network to implement the algorithm. Today's popular peer-to-peer networks [5] have over 20,000 nodes. Performing a software update for each of these nodes in order to test each novel P2P algorithm is impractical. As another example, security protocols require testing under different threat scenarios such as an attack on the network by a coordinated group of malicious peers. Testing such protocols would require introducing malicious peers into the network, which is also not practical. Thus, P2P algorithms and protocols are tested by simulation, under network models that attempt to mimic typical node interconnections, traffic patterns etc. Since algorithms and protocols are often sensitive to the traffic and network behavior, there is a clear need for accurate P2P network models.

Work in this area has been mainly done on the fly to test novel algorithms. Because of this, most P2P simulators have used simple models. For example, [1] assumes entirely random interactions among peers in a P2P network to test a P2P reputation management protocol. In simulating a distributed search algorithm, [4] simply use a uniformly random location of files at and generation of queries by peers.

In this paper, we present the Query-Cycle Simulator, a P2P file-sharing network simulator based on the query-cycle model described in Section 2, and discuss the issues that arise in the accurate modeling of a P2P network. We focus on modeling a file-sharing network such as Gnutella [5].

2 The Query-Cycle Model

We consider a typical P2P network: Interconnected, file-sharing peers are able to issue queries for files, peers can respond to queries, and files can be transferred between two peers to conclude a search process. When a query is issued by a peer, it is propagated by broadcast with hop-count horizon throughout the network (in the usual Gnutella way), peers which receive the query forward it and check if they are able to respond to it.

We suggest a simulation process that proceeds in query cycles. In each query cycle, a peer i in the network may be actively issuing a query, inactive, or even down and not responding to queries passing by. Upon issuing a query, a peer waits for incoming responses, selects a download source among those nodes that responded and starts downloading the file. The query cycle finishes when all peers who have issued queries download a satisfactory response. Statistics may be collected at each peer, such as the number of downloads and uploads of the peer.

3 Peer-Level Properties

The system-level dynamics of a P2P network are highly dependent on local, peer-dependent properties, such as the activity level or file-sharing behavior of each peer. In [8], different convergence behavior and different characteristic path lengths are observed in simulating a novel P2P network topology construction algorithm under two different models, one assigning bandwidth capacities to nodes based on a Zipf distribution, the other one based on a real-world distribution measured in [13].

Since the system-level dynamics of a P2P network – and hence the system-level impact of a P2P algorithm – is dependent on local, peer-level parameters, it's essential to accurately model these parameters. We may classify these parameters into two types: content distribution parameters, and peer behavior parameters.

Content Distribution. We must accurately model the volume and type of content each peer carries. P2P networks are far from homogeneous in terms of type and volume of data shared, hence a model reflecting real-world P2P networks is required.

Peer Behavior. We must also accurately model peer behavior, including how a peer submits and responds to queries, how it chooses which query response to download, and its uptime and session duration.

In the next two sections, we discuss how to accurately model the content distribution and peer behavior parameters, and we discuss open questions and empirical studies that would be useful to the accurate modeling of these parameters.

4 Content Distribution Model

The dynamics of a P2P networks are highly dependent on the volume and variety of files each peer chooses to share. If few peers choose to share files, then queries are likely to be routed via many peers, and the load on the network referring to file uploads is likely to be highly imbalanced. If many peers choose to each share a wide variety of files, the network of peers who interact with one another is likely to be dense and unclustered, and query response times are likely to be quick.

Accurate assessment of the impact of intelligent query routing algorithms and content-based topologies depends on the accurate modeling of the volume and variety each peer shares. Furthermore, accurate modeling of the content shared by peers in the network gives us greater insight into the file-sharing and communication patterns in the network, which is useful in many areas of P2P research.

4.1 Data Volume

In our model, each peer in the network shares a certain number of files.

Real-world observations. [13] has measured the probability distribution over the number of files shared by peers in Gnutella.

Model. We use this distribution to assign a number of shared files, F^i, to each peer i in the network. Currently, we use the absolute values from [13].

4.2 Content Type

In this section, we describe how we model the individual files each peer chooses to share. It is important to accurately model this because this will determine patterns of peers who interact with one other. A model in which the files peers share are chosen randomly is insufficient, as it will fail to produce clusters of peers that interact with on another, as has been observed [3]. Such properties affect the performance of many algorithms, including search algorithms [2] and reputation algorithms [6].

Real-world observations. In [3], it is observed that peers in a P2P network are in general interested in a subset of the total available content on the network. Furthermore, it is also observed in [3] that peers are often interested only in files from a few content categories. For example, in the domain of educational resources [10], users have a certain affinity towards learning materials related to the course of study they undertake.

It also has been observed in [7] that many document storage systems, including the WWW, exhibit Zipf distributions on the popularity of documents. This reflects the fact that some popular documents are very widely copied and held, while most documents are held by far fewer peers. The same can be said of content categories: there are some content categories (such as "Top 40 Hits" in the music domain") which are very popular and widely held, while most other categories (such as "Acid Jazz") are less widely held.

Model. We model the properties described above as follows. Briefly, peers are assumed to be interested in a subset of the total available content in the network, i.e., each peer initially picks a number of content categories and shares files only in these categories. Furthermore, we assume that files with different popularities exist within each content category, governed by a Zipf distribution. Files are assigned to peers at initialization in the following manner. According to the probabilistic model described below, each peer i is assigned some content categories C^i. Then, peer i is given an interest level for each content category $c \in C^i$. Finally, peer i is assigned files F according to its content categories and interest levels in those categories. In this model, each distinct file $f_{c,r}$ may be uniquely identified by the content category c to which it belongs and its popularity ranking r within that category. The probabilistic model is based on empirical observations of file distributions in [13] and [7]

Assigning content categories. We assume n content categories $C = \{c_1, \ldots, c_n\}$. Some content categories are more popular than others. That is, the files in some content categories are more widely held than the files in other categories. We characterize a content category completely by its popularity rank. That is, $c_1 = 1, c_2 = 2, \ldots$. We model this popularity by a Zipf distribution: when a peer is initialized, it is set to be interested in content category $c \in C$ with probability

$p(c)$ given by $p(c) = \frac{\frac{1}{c}}{\sum_{i=0}^{n} \frac{1}{i}}$. We require a peer to be interested in at least C_{min} content categories, repeating the peer's interest test until C_{min} categories have been chosen. The set C^i is the set of content categories that interest peer i.

Modeling interest level. A peer i interested in content categories C^i is probably not equally interested in all categories $c \in C_i$. Rather, peer i is more likely to be more interested in some categories than others. We model this by assigning an interest value w_c^i to each content category $c \in C_i$ of interest to peer i. This interest value is determined uniformly at random for each content category for each peer i. The fraction of files shared by peer i that are in category c is given by $p(c|i) = \frac{w_c^i}{\sum_{c' \in C^i} w_{c'}^i}$. The number of files shared by peer i that are in category c is given by $F_c^i = p(c|i)F^i$.

Note that the interest value is not correlated with the general popularity of content category c. This reflects the fact that, while a certain category may be of interest to many peers (i.e., Top 40 hits), that category is not necessarily the main interest of those peers. Also note that since we assume a steady-state network, we assume that the interests of peers do not change over time.

Modeling Files We now wish to model the individual files held by each peer. Each distinct file may be uniquely identified by the tuple $\{c, r\}$, where c represents the content category to which the file belongs, and r represents its popularity rank within content category c. We denote this file $f_{c,r}$. Within each content category there are some files that are very popular, and some that are held by few people. We model this by a Zipf distribution as well. The fraction of files in content category c that are copies of file $f_{c,r}$ is given by:

$$p(f_{c,r}|c) = \frac{\frac{1}{r}}{\sum_{i=1}^{F_c} \frac{1}{i}} \tag{1}$$

where F_c is the number of distinct files in category c. Notice that in order to evaluate $p(f_{c,r}|c)$, we need to model the number of distinct files in each content category (see below). The probability that a file f shared by peer i is a copy of file $f_{c,r}$ is given by the level of interest $p(c|i)$ that peer i has in category c times the popularity $p(f_{c,r}|c)$ of file $f_{c,r}$ within category c $p(f_{c,r}|i) = p(c|i)p(f_{c,r}|c)$. At initialization, we assign files to each peer i based on this distribution and the number of files F_c^i shared by peer i in each category. Each peer stores the $\{c, r\}$ values for the files that it shares.

Modeling the number of distinct files per category. If there is maximum replication going on in the network, there are at maximum F_c^a files of content category c in the network, where F_c^a represents the number of files in category c shared by peer a, the peer who shares the most files in category c. On the other hand, if every single file on the network is distinct, then there are $p(c)F$ distinct files, where F is the total number of files on the network, and $p(c)$ is the fraction of files that are in category c. The truth probably lies somewhere in between, and we set $F_c = dF_c^a + (1-d)p(c)F$ where d is some number between 0 and 1. In our implementation, we set $d = .25$. Empirical evidence would be useful to determine an accurate choice of d.

4.3 More Complex Semantics

The current content model assumes a simple and flat list of content categories as basic taxonomy to categorize content in a P2P network. While this assumption may hold for today's simple P2P file sharing networks, more advanced P2P networks will use more sophisticated categorizations of content. For example, [3] also uses a concept hierarchy to describe content. More generally, ontologies may be used to mark up content in a P2P network. For example, in the Edutella [10] network, peers exchange learning objects described by ontologies. Since algorithms designed to run on such networks require an accurate modeling of the network, it will be necessary to adapt the content distribution model used in the simulations.

For example, if an ontology is used to mark up resources in a P2P network, the content distribution model should contain distributions over the popularity of certain concept combinations and attribute values. These distributions may be measured in the domain from which the content in the network is drawn.

5 Peer Behavior Model

In addition to content distribution, another primary factor in the system-wide dynamics of a P2P network is peer behavior, including peer uptime and session duration, peer activity levels, and how peers issue and respond to queries. These parameters affect the network in many ways. For example, frequently changing network participation (i.e., very short session durations, yet high uptimes of nodes) increases the administration overhead of topology construction protocols, which generally require communication among a number of peers to repair the network topology once a peer has left or joined, an important cost factor to consider in the design of such protocols. Second, node uptime represents the availability of storage space and computational power in the network. The pattern of node uptime in P2P networks is of interest for applications that wish to take advantage of these networks for large-scale computations, as in [6]. If the network consists of a large pool of nodes that participate only infrequently and are down most of the time, those nodes that remain in the network have to sustain a higher work and storage load.

Query activity level is another peer behavior that is of particular interest to P2P research, as the query behavior of peers (in conjunction with the network's content distribution patterns) determines which peers interact with each other. These interaction patterns are of importance to the effective design of P2P algorithms ranging from search algorithms to file indexing protocols.

5.1 Uptime and Session Duration

Participating nodes frequently leave and re-join a P2P network, and we define a peer's uptime to be the fraction of an observation period that a peer is participating in the P2P network, i.e., issuing, responding to and forwarding queries.

Real-world observations. Uptime and session duration of peers have been set in [13]. Observations on the MojoNation P2P network [14] have revealed that up to 84% enter the network one time, and for less than one hour. At the moment, we do not consider these peers in our simulations. They probably do not contribute to the shared data much, and they probably do not issue too many queries.

Model. We assume a pool of N peers, and each peer has a certain probability of being online, assigned based on the uptime distribution in [13]. At each query cycle, it is determined for each peer based on its probability of being up if it enters the network and stays there for a certain period of time which is drawn from the session duration distribution in [13].

5.2 Query Activity

Peers in a P2P network issue queries to search for downloadable files that match their interests. A peer's query activity determines the rate at which it issues queries when it is up.

Real-world observations. So far, we are not aware of measurements on query rates of peers in a P2P network. An empirical study on the distribution of query rates of real-world P2P networks would be straightforward and very useful to the accurate modeling of the network.

Model. In our model, nodes generate queries based on a Poisson process. The query rate of each node is set upon initialization and is picked uniformly at random from an interval $\{rate_{min}, rate_{max}\}$. In each query cycle, equation $p(\#queries == x) = \frac{\exp^{-\lambda} \cdot \lambda^x}{x!}$ gives the probability that a node issues x queries, where λ is the node's query rate.

5.3 Queries

In the query cycle model, each active peer issues a query at each query cycle. The specific query that peer i issues is given by the model described below.

Real-World observations. Peers in general query for files that exist on the network and are in the content category of their interest. The first is true in large and diverse P2P networks, the latter we claim to be true for the majority of queries a peer issues, albeit it is yet to be shown by empirical studies on the query behavior in P2P networks.

Model. In our model, a query $q_{c,r}$ represents a query for the file $f_{c,r}$. We say that a peer only issues queries in in the content categories in which it is interested. The probability that a peer i generates a query $q_{c,r}$ is given by it's interest level in category c times the popularity of file r in c:

$$p(q_{c,r}|i) = p(c|i)p(f_{c,r}|c) \tag{2}$$

(We say here that the popularity $p(q_{c,r}|c)$ of a query $q_{c,r}$ is equal to the popularity $p(f_{c,r}|c)$ of its corresponding file $f_{c,r}$.) We also suggest that a peer will not issue a query for a file that it already owns.

5.4 Query Responses

In this framework, modeling query responses is straightforward: If peer i receives a query $q_{c,r}$, and it owns a copy of the corresponding file $f_{c,r}$, it responds to the peer that has issued the query and offers to upload the file.

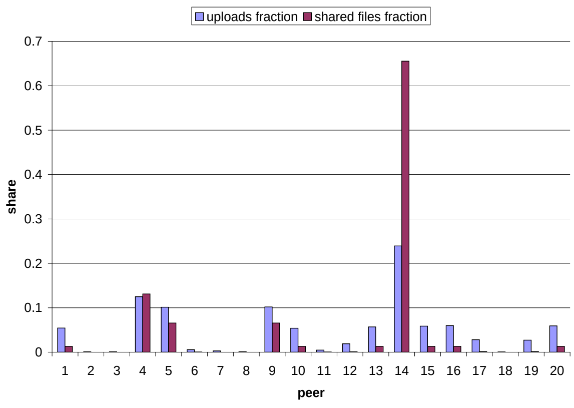

Fig. 1. Share of uploads and files in a P2P network.

5.5 Downloads

In the query cycle model, a cycle consists of each active peer issuing a query, waiting for the list of incoming responses, and downloading one of the responses. In our model, a peer randomly chooses a response to download. This may not accurately reflect reality. In fact, for a file-sharing system such as Gnutella, users tend to select peers with high bandwidth, hoping to being able to download a file fast. However, the Query-Cycle Simulator can easily be extended to bias source selection based on peer bandwidth.

6 Network Parameters

Network parameters characterize the underlying transport network of a P2P network and the transport network related properties of peers.

6.1 Topology

Peers form an overlay network on top of a transport network. Upon joining the network, peers establish links to a number of peers in the network. When leaving, peers disband these links. Query and control messages are passed along the interconnection links between peers.

Real-world observations. Freely evolving P2P networks have been shown to exhibit power-law network characteristics [11]. Hence we organize peers into a power-law network. Upon joining the network, peers connect to a node i with probability $\frac{d_i}{\sum_{j \in N} d_j}$, where N is the set of nodes currently in the network and d_i is the node degree of peer i. Thus, joining nodes have a higher probability of connecting to nodes which maintain a higher number of network connections, which yields a network with power-law characteristics [9].

Model. The topology of a P2P network is in several ways to be considered when designing P2P algorithms. First, the pattern of where new peers usually join the network can be important for designing topology construction protocols. For example, the fact that P2P networks exhibit a power-law topology shows that some peers in the network have a higher probability of being contacted when a new peer joins the network, challenging topology construction protocols with an inherent imbalance. Second, depending on the search method deployed in the network, the topology may determine the scope of a peer in the network (Gnutella uses a broadcast search with a hop-count horizon of 7 hops). Peers in the 'center' of the network, e.g., highly connected peers in a power-law network, will be able to see a larger fraction of the query traffic in the network. Malicious peers trying to attack the network by responding to queries with decoy files may try to locate and to connect to highly connected peers in the network to increase their chances of responding to many queries, a threat scenario to be considered for reputation algorithms.

6.2 Bandwidth

We currently have a simple understanding of a peer's bandwidth in our simulations: Bandwidth at a peer is consumed only while uploading or downloading files. Bandwidth is assigned to a peer upon the creation of the peer based on measurements in [13]. Upon up- or downloading a file, peers always try to use their full bandwidth (the actual transfer rate is limited by the peer with less bandwidth). If a peer has several up- and downloads going on, the available bandwidth is split up equally.

7 Discussion and Conclusion

Figure 1 depicts the load share in a sample network of 20 peers that was simulated based on the considerations above. One graph shows the number of uploads at a particular peer versus the total number of uploads in the system after 300 query cycles, the other graph shows the number of files shared by a peer versus the total number of files shared by all peers. Although the distribution of files is highly imbalanced – a property observed real-world P2P networks [13] – all peers participate in responding to queries, since even peers with only a few files have a fair likelihood of responding to queries for very popular files. This is a property that can also be observed on real-world P2P networks and provides a first indication that our model is somewhat accurate.

We have described first ideas and approaches for a P2P network simulator. The efficiency of algorithms can only be compared if they can be run on commonly accepted problem sets or simulated on widely accepted models, an insight accepted in many other research domains such as Internet research [12]. We believe the same to be true for P2P algorithms, and we believe it is important for the community to engage in a discussion of P2P modeling in order to develop some standards by which to simulate P2P networks.

References

1. K. Aberer and Z. Despotovic. Managing trust in a peer-2-peer information system. In *Proceedings of the 10th International Conference on Information and Knowledge Management (ACM CIKM)*, New York, USA, 2001.
2. A. Crespo and H. Garcia-Molina. Routing Indices for P2P Systems. In *Proceedings of the 28th Conference on Distributed Computing Systems*, July 2002.

3. A. Crespo and H. Garcia-Molina. Semantic Overlay Networks. In *Submitted for publication*, October 2002.
4. M. Freedman and R. Vingralek. Efficient P2P Lookup for a Distributed Trie. In *First International Workshop on P2P Systems*, 2002.
5. Gnutella website. www.gnutella.com. 2002.
6. S. Kamvar, M. Schlosser, and H. Garcia-Molina. The EigenTrust Algorithm for Reputation Management in P2P Networks. In *WWW 2003*, 2003.
7. R. Korfhage. *Information Storage and Retrieval*. John Wiley, 1997.
8. Q. Lv, S. Ratsnasamy, and S. Shenker. Can Heterogeneity Make Gnutella Scalable? In *First International Workshop on P2P Systems*, 2002.
9. A. Medina, I. Matta, and J. Byers. On the origin of power laws in internet topologies. Technical report, Boston University Computer Science Department, April 2000.
10. W. Nejdl and et al. EDUTELLA: A P2P Networking Infrastructure based on RDF. In *Proceedings of the 11th World Wide Web Conference*, May 2002.
11. M. Ripeanu and I. Foster. Mapping the Gnutella Network - Macroscopic Properties of Large-scale P2P Networks. *IEEE Internet Computing Journal*, 6(1), 2002.
12. S. Floyed and E. Kohler. Internet Research Needs Better Models. In *Proceedings of HotNest-I*, October 2002.
13. S. Saroiu, P. K. Gummadi, and S. D. Gribble. A measurement study of peer-to-peer file sharing systems. In *Proceedings of Multimedia Computing and Networking 2002 (MMCN '02)*, San Jose, CA, USA, January 2002.
14. B. Wilcox-O'Hearn. Experiences deploying a large-scale emergent network. In *1st International Workshop on P2P Systems*, 2002.
15. B. Yang and H. Garcia-Molina. Improving efficiency of p2p search. In *Proceedings of the 28th Conference on Distributed Computing Systems*, July 2002.