An Analytic Framework for Modeling Peer to Peer Networks

Krishna K. Ramachandran and Biplab Sikdar
Department of Electrical, Computer and Systems Engineering
Rensselaer Polytechnic Institute, Troy NY 12180
{ramak,sikdab@rpi.edu}

***Abstract*—This paper presents an analytic framework to evaluate the performance of peer to peer (P2P) networks. Using the time to download or replicate an arbitrary file as the metric, we present a model which accurately captures the impact of various network and peer level characteristics on the performance of a P2P network. We propose a queueing model which evaluates the delays in the routers using a single class open queueing network and the peers as M/G/1/K processor sharing queues. The framework takes into account the underlying physical network topology and arbitrary file sizes, the search time, load distribution at peers and number of concurrent downloads allowed by a peer. The model has been validated using extensive simulations with campus level, power law AS level and ISP level topologies. The paper also describes the impact of various parameters associated with the network and peers incluing external traffic rates, service variability, file popularity etc. on the download times. We also show that in scenarios with multi-part downloads from different peers, a rate proportional allocation strategy minimizes the download times.**

I. Introduction

Peer to peer networks provide a paradigm shift from the traditional client server model of most networking applications by allowing all users to act as both clients and servers. The primary use of such networks so far has been to swap media files within a local network [25] or over the Internet as a whole [1], [3], [4]. These networks have grown in their popularity in the recent past and the fraction of network traffic originating from these networks has consistently increased [2]. Understanding the factors affecting their performance and developing models to quantify their impact is thus of critical importance to facilitate the development of peer-to-peer networks and to ensure the proper utilization of the networking infrastructure. In this paper we address this issue and develop an analytic framework for modeling and evaluating the performance of peer-to-peer networks while accounting for architectural, topological and user related factors.

The paradigm shifts associated with peer-to-peer networks and its inherent features necessitate the development of new models to account for their behavior. In addition to the blurring of the distinction between the roles of cilents and servers, the presence of a node in the network can be transitory with peers continually joining and leaving the network arbitrarily over any given period of time. Finally, network and end user heterogenieties like different access speeds at different peers, file popularity, number of simultaneously allowable downloads at a peer etc. need to be taken into account to get realistic results. Existing literature on the performance and modeling of P2P networks primarily focus on measurement and simulation based studies [14], [13], [11], [6], [10]. Analytic efforts to model the performance of P2P networks using fluid or branching process based Markov models are presented in [12], [15], [16], [17] and focus on the transient or steady state behavior of the number of peers in the network. A closed queueing system model for P2P networks is presented in [5]. The existing models fail to capture the performance of a P2P network in terms of a user's viewpoint: *How long does it take to download a file, if available, from the network?* while accounting for the various user and network level factors which are inherent to P2P networks. This paper addresses this issue and in addition to the to modeling the performance in terms of download times can also be used to develop peer selection policies and user level policies to improve the system performance.

In order to address the above concerns, our model decomposes the time required to replicate a file into two components: (1) network latencies and (2) peer level latencies. These components are then modeled using

1) Single class open network of the core routers with the first come, first serve (FCFS) discipline with arbitray arrival and service patterns to model the network latencies.
2) M/G/1/K processor sharing queues with arbitrary constraints on the number of allowable servers and file size distributions to model the peer level latencies.

The routers are analyzed as standard GI/G/1 queues, partially characterized by the first two moments of the arrival and service time distributions. The model is able to account for a number of factors of peer-to-peer networks and network heterogenieties like file popularity and size distribution, peer specific settings like the number of simultaneous downloads allowed, different access rates, physical topologies and search strategies. Using our model, we also show that the optimal workload division strategy in the presence of multiple sources is proportional to the service rates at each peer.

Extensive simulations are conducted to validate the results of the model. We show simulation results from three different scenarios (1) a real University network (Columbia University), (2) a national backbone (AT&T) with Internet service provider (ISP) level topologies and (3) power law topologies [18]. For each of these scenarios, our analytic results show a close match with the simulation results.

The rest of the paper is organized as follows. In Section II we differentiate the work presented in this paper from existing literature. Section III presents the analytic framework for evaluating peer to peer networks and Section IV discusses download strategies in multi-part scenarios. In Section V we present simulation results to verify our model and also analyze the impact of various factors on the network's performance. Finally, Section VI presents the concluding remarks.

II. Relationship to Prior Work

In addition to applications involving file sharing, P2P networks have also been proposed for use in web caching, distributed directory services, sotrage and grid computation. While the majority of the existing literature on the performance of P2P systems have focused on measurements, various analytic models have recently been proposed. A measurement study of the Gnutella network's properties is presented in [13]. In [11], the authors analyze four content delivery systems: HTTP web traffic, Akamai content delivery network, and Kazaa and Gnutella peer-to-peer file sharing traffic. An analysis of user traffic in Gnutella, specifically the performance in the presence of *freeloaders* can be found in [6]. The measurement study in [10] characterises the behavior of users as well the network, such as the degree of co-operation among peers, the time spent on-line, bottleneck bandwidths etc. for the Gnutella and Napster networks. In contrast to these studies, this paper develops an analytic model for evaluating P2P networks.

In [12] a fluid model is used to characterise the performance of BitTorrent like networks in terms of the average number of downloads and download times. Stochastic fluid models are also used in [15] to model the performance of P2P web caches and web clusters. Branching process based Markovian models to study BitTorrent like networks are used in [16]. In [17] P2P networks are studied in terms of the required rates at which nodes may enter and leave the network in order to maintain system state. In contrast to our approach, the focus of these studies is primarily on the evolutionary dynamics of the system. These studies also do not account for queueing effects and heterogeneities in hosts and the network.

The literature closest to our approach is that in [5] where a closed queuing model for peer to peer networks has been proposed. However, unlike our approach, this model does not capture the significance of the physical topology underlying the P2P network. Accounting the impact of the physical network topology is particularly important since a next hop peer in the P2P network is not necessarily the same as Internet Protocol (IP) next hop. The topology of the network governs the number of routers, and thus the queues a packet passes through before reaching the final destination. This model also does not capture the effect of the differences in the file sizes of different requests on the system performance. Another important abstraction unaddressed in [5] is the heterogeneities in the network and hosts. Although, the authors analyze the effect of *freeloaders* on the system, the behavior of peers allowing only limited number of simultaneous file transfers has not been modeled. Also, while different on and off times for different classes of users have been considered, different access rates and varying loads on different peers has not been modeled.

III. Analytic Framework

In this section, we present our analytic framework for modeling the performance of peer-to-peer networks. While the framework is applicable for large networks of the scale of the Internet, for illustrative purposes, we confine the example scenarios in this section to those of a campus or organization wide network. In general, a network can be viewed as that consisting of core routers, the interconnection of which form the network backbone. Each of these routers acts as a gateway to one or more subnets which may be local area networks (LANs) in the case of campus networks and autonomous systems (ASes) in the case of large networks. The autonomous systems can also be further broken down into a network of intra-domain routers which in turn act as gateways to subnets. The peers are the nodes residing in the various subnets. This scenario is shown in Fig. 1 where we consider a simple campus like scenario with four routers and six subnets. If a user, say Y in Subnet D were to download a file from another peer, say X in Subnet A, the delays encountered would be

1) queuing delay at the core routers,
2) delay at the end peer to service the request and
3) link propagation delay

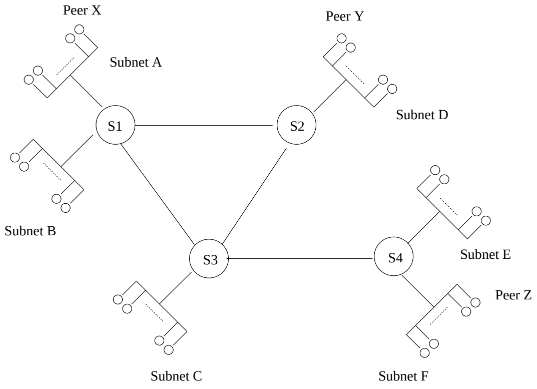


Fig. 1. An example topology of a peer-to-peer network in a campus environment.

Among the three, the propagation delay of each link is arguably the most predictable and can be treated as a constant. Propagation delays (of the order of few milliseconds in a LAN and a few hundreds of milliseconds in inter-continental links) are usually much lower the sum of the download delays due to the end peers and the network's queueing delays. In this paper we thus focus on the network queueing delays and the delays at the end peers. For our analysis we break up the system into two components:

1) Single class open network of the core routers with the first-come, first serve discipline, no capacity constraints and general arrival and service time distributions. The network model is described in Section III-A.
2) The end peers which are analyzed as processor shared nodes with finite/infinite capacity and arbitrary service times. The peer model is elaborated upon in Section III-B.

In other words, the peers residing in the various subnets connected to the routers are not part of the first component. From a perspective of queuing analysis, packets from node X to node Y would see the system in Fig. 1 as that in Fig. 2. The total file transfer delay is given by the sum of the core network delay component i.e. queuing delay at the intermediate routers and the end peer delay component i.e. which is nothing but the transmission time of the file being downloaded. We now present our models to evaluate these components.

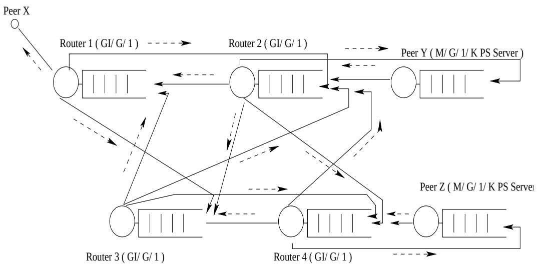


Fig. 2. Queuing model equivalent of Fig 1.

A. Router Network Model

In this section we present our model for characterizing the delays at the core routers. We first derive expressions for the traffic arrival statistics at the routers which are then used to derive the network delays. We consider an interconnection network of N_R routers whose topology can be considered a random graph and is specified using the routing matrix Q. Each element q_{ij} of Q specifies the fraction of traffic arriving at router i that is destined for router j. The information encapsulated in the routing matrix is leveraged to extract pertinent topological properties (Section III-C). Packets in each router are assumed to be served in a first come first served manner and no constraints are placed on the queue lengths at the routers.

We model each router as a GI/G/1 queue to allow for arbitrary arrival patterns and packet size or service time distributions. The choice of a generalized interarrival (GI) process to model the arrivals is motivated by the fact that traffic to a network router can be quite erratic and does not necessarily follow a Poisson distribution [21], [22] and may vary widely from network to network. The squared co-efficient of variance allows an extent of accountability for the variability that results from bursty traffic. In our system model, traffic to the core routers comprises of data from

- external sources which can either be the end hosts residing in the subnet(s) connected to it or cross-traffic extraneous to the network.
- from other routers that are directly connected to it.

Thus the total arrival rate at the j^{th} router, λ_j, is a function of both the total external arrival rate (assumed know) to it, denoted by λ_{0j}, as well as arrivals from each of the neighboring routers. Similarly, the variability of the arrival process at a given router is a function of the variability (SCV) of its external arrival process as well as those of the arrivals from its immediate neighbors. While the mean and SCV of the service time distribution at the routers are assumed as inputs, the parameters of the arrival process at each of the core routers are the unknowns that remain to be determined. The approximation method [7], [8] used in this paper to obtain the rate and variability parameters for the arrivals at the routers in the network is a parametric-decomposition method, since the nodes are analyzed separately after the parameters for the internal flows are determined.

λ_j	total arrival rate at the j^{th} router
τ_j	service rate of the j^{th} router
c^2_{aj}	SCV of the arrival process at the j^{th} router
c^2_{sj}	SCV of the service distribution of the j^{th} router
W_{Q_j}	waiting time at the j^{th} router before a packet gets service
Q	routing matrix
p_{ij}	proportion of arrivals to router j from router i
N_C	Total number of packets in the network at any given time instant
T_{N_R}	Time spent by a packet in the router network
P_l	blocking probability at the peer
μ_p	peer service rate
N_p	number of connnections currently being served at the peer
W_p	service time for a request at the peer
C	link capacity of the peer
m	maximum number of requests allowed by the peer
V	total number of files shared in the P2P network
V_{on}	number of files shared by peers that are currently online.
$O(i)$	number of copies of the i^{th} most popular file in the network
B	size of the file currently being downloaded
N_{p_i}	number ofconcurrent downloads at the i^{th} peer
$\hat{X}_i$	service time of the i^{th} peer
T_{QS}	Time elapsed between query generation and termination
T_D	Time required for the file transfer from the peer(s)
T	Overall waiting time, expressed as a summation of T_{QS} and T_D

TABLE I

NOTATION AND MODEL PARAMETERS

In Section III-A.1 we develop a system of linear equations involving the arrival rates and in Section III-A.2 those for the variability parameters of the arrival process. Finally, Section III-D derives the expressions for the mean waiting time and average number in the system as well as individual routers.

1) Traffic Rate Equations: We now calculate the traffic arrival rates at each router. With λ_j denoting the traffic arrival rate at router j, and τ_j denoting the average time taken by the router to process a packet, the fundamental traffic-rate equation at router j can then be formulated as

$$\lambda_j = \lambda_{0j} + \sum_{i=1}^{N_R} \lambda_i q_{ij} \quad j = 1, 2, \cdots, N_R. \tag{1}$$

In matrix notation, these equations can be written as

$$\Lambda = \Lambda_0 (I - Q)^{-1}, \tag{2}$$

where $\Lambda_0 \equiv (\lambda_{0j})$ is the external arrival-rate vector, i.e. traffic arriving from the subnets and $Q \equiv (q_{ij})$ is the routing matrix.

The assosciated offered load at node i, which also gives the probability that the queue is busy is given by

$$\alpha_i = \lambda_i \tau_i, \quad 1 \leq i \leq N_R \tag{3}$$

Two other relevant metrics, the rate of arrivals from router j from router i, λ_{ij}, and the proportion of arrivals at router j which originate at router i, p_{ij}, are given by

$$\begin{aligned} \lambda_{ij} &= \lambda_i q_{ij} \\ p_{ij} &= \lambda_{ij}/\lambda_j \end{aligned} \tag{4}$$

Equation (1) is essentially a rate balance equation, since our assumption of infinite buffers at the routers and stable queues implies that the incoming traffic rate equals the outgoing rate. The only unknowns in Eqn. (1) are the arrival rates $\lambda_i \quad i = 1, \cdots, N_R$ since both Λ_0 and Q are inputs to the model. The solution of Eqn. (1), a system of N_R linear equations in N_R variables, will therefore yield the total arrival rate at each router.

2) Traffic Variability Equations: Having calculated the rate parameters assosciated with the internal flows, we now proceed to obtain the system of equations yielding the corresponding variability parameters i.e., squared coefficients of variation for the arrival processes. We denote by c_{aj}^2 the SCV of the arrival process at router j. The expressions for c_{aj}^2 and the related parameters are as derived in [7] and are enumerated below

$$c_{aj}^2 = a_j + \sum_{i=1}^{N_R} c_{ai}^2 b_{ij} \ \ 1 \leq i \leq N_R, \tag{5}$$

where a_j and b_{ij} are constants, depending on the input data, and are given by

$$\begin{aligned} a_j &= 1 + w_j \Bigg\{ (p_{0j}^2 c_{0j}^2 - 1) \\ &\quad + \sum_{i=1}^{n} p_{ij}[(1 - q_{ij}) + (1 - \nu_{ij}) q_{ij} \rho_i^2 x_i] \Bigg\} \end{aligned} \tag{6}$$

and

$$b_{ij} = w_j p_{ij} q_{ij} [\nu_{ij} + (1 - \nu_{ij})(1 - \rho_i^2)], \tag{7}$$

where x_i, ν_{ij} and w_j are independent of the variability parameters c_{aj}^2 being calculated. In Eqns. (6) and (7) p_{0j} is the weight assosciated with the external traffic while c_{0j} denotes the SCV of the external arrival process into router j. The variables x_i and ν_{ij} are used to specify the departure operation from the router; the variable w_j characterizes the superposition of traffic streams at the router. In our simulations, we use the values for x_i, ν_{ij}, w_j and ν_j as specified in [7]. x_i is given by

$$x_i = 1 + (max\{c_{si}^2, 0.2\} - 1), \tag{8}$$

where c_{si}^2 is the SCV for the service time of the i^{th} router. Also, $\nu_{ij} = 0$ and

$$w_j = [1 + 4(1 - \rho_j)^2(\nu_j - 1)]^{-1} \tag{9}$$

with

$$\nu_j = \left[\sum_{i=0}^{N_R} p_{ij}^2 \right]^{-1} \tag{10}$$

where p_{ij} is given in Eq. (4).

In our analysis, while deriving expressions for the router network, the peers are decoupled from the system and traffic from them into the routers is equivalent to that generated by an external source. Hence, external traffic is often a combination of several arrival streams. Let κ_i and ζ_i^2 denote the rate and variability parameters for the i^{th} stream into node j. Thus, we have

$$\begin{aligned} \lambda_{0j} &= \sum_i \kappa_i \\ c_{0j}^2 &= w_j \sum_i \left(\kappa_i \Big/ \sum_k \kappa_k \right) \zeta_i^2 + 1 - w_j \end{aligned}$$

where w_j is as evaluated in Eq. (9).

3) Network Latency: We begin with the steady-state waiting time (before beginning service) at the routers, denoted by W_{Q_i}. Using the results from [7], the expected waiting time at the i^{th} router can be shown to be

$$E[W_{Q_i}] = \tau_i \rho_i (c_{ai}^2 + c_{si}^2) g_i / 2(1 - \rho_i), \tag{11}$$

where $g_i \equiv g_i(\rho_i, c_{ai}^2, c_{si}^2)$ is defined as

$$g_i(\rho_i, c_{ai}^2, c_{si}^2) = \begin{cases} exp\left[-\frac{2(1-\rho_i)}{3\rho_i} \frac{(1-c_{ai}^2)^2}{(c_{ai}^2 + c_{si}^2)} \right] & c_{ai}^2 < 1 \\ 1 & c_{ai}^2 \geq 1 \end{cases} \tag{12}$$

Let the number of packets in the i^{th} router, including the one in service, be denoted by N_{C_i}. Using Little's law, the expected number of packets, $E[N_{C_i}]$, is given by

$$E[N_{C_i}] = \rho_i + \lambda_i E[W_{Q_i}] \tag{13}$$

The average time spent by a packet among the routers, which is the network sojourn time is derived by applying Little's Law to the entire router network. Let λ_0 be the total external rate of traffic into the routers, i.e.

$$\lambda_0 = \sum_{i=1}^{N_R} \lambda_{0i}.$$

The total external arrival rate is also a measure of the *throughput* of the router network. The total number of packets in the network N_C and therefore the sojourn time $E[T_{N_R}]$ or the router network delay per packet is given by

$$N_C = \sum_{i=1}^{N_R} N_{C_i}, \qquad E[T_{N_R}] = \frac{N_C}{\lambda_0} \tag{14}$$

B. Modeling the end peer

We now propose a queuing model for the end peer and derive an expression for the expected time it takes to service an user requesting file download, starting with the arrival process. We approach this queueing model from a per file request basis rather than a per packet basis. In the network model that we have chosen, each router is attached to a number of subnets, which in turn harbor the end peers. In view of this, traffic or download requests from the edge router can be thought of as

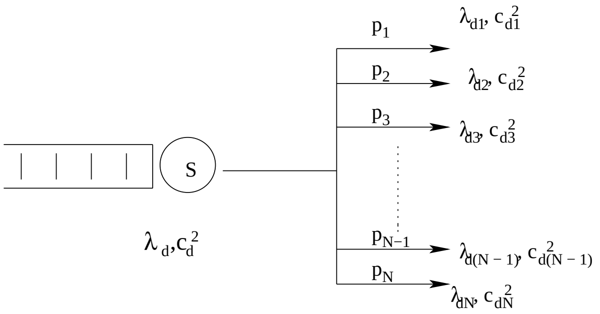


Fig. 3. Splitting of the output stream at the router.

splitting in to several streams, one for every active end peer as shown in Figure 3. It is also reasonable to assume that no single peer grabs a major chunk of the end bandwidth. The presence of cross-traffic, predominantly HTTP, also lends credibility to this assumption. Under these conditions we now show that the SCV of the arrival process at the end peers equals 1. If λ_d (c_d) denotes the departure rate (SCV) from the router and the probability that a request from that stream is destined for peer i is denoted by p_i, the incoming rate at peer i is

$$\lambda_{d_i} = p_i \lambda_d$$

and SCV of the this process is given by

$$c_{d_i}^2 = p_i c_d^2 + (1 - p_i).$$

As the number of streams increase we have

$$\lim_{i \to \infty} \frac{\lambda_{d_i}}{\lambda_d} = 0$$

resulting in $p_i \to 0$ and therefore $c_{d_i}^2 \to 1$. Though there could be many processes other than an exponential distribution that have an SCV of 1, we model the incoming process at the i^{th} end systems as Poisson with rate λ_{d_i}.

Since the service time is dependent on the file size being downloaded and file size distributions being typically heavy tailed [23], the service times at the queue cannot be modeled as an exponential process. We allow for arbitrary distributions for the service times thereby accomodating generalized models for the file sizes.

Another significant modeling parameter that needs to be addressed is the number of files that a peer is willing to let the other peers download from it at any given instant of time. A savvy peer may limit this number in order to gain download bandwidth leverage. *Freeloaders* form an extreme class of such peers and do not share any files but contribute to the network traffic by making frequent download requests [6]. If a request for a file is made when the download limit has been reached, it is lost and no file transfer takes place. Also note that there is no queuing of requests at the end peer. In other words, a peer allowing at most m simultaneous downloads functions as a node with m servers and no queue buffer. The peer does not distinguish among the various arrivals i.e., each request is served at the same rate as the others. When a new request arrives, the service rates of the current transfers changes since all request receive equal service. Specifically, each service rate now becomes $C/current$, with C and $current$ representing the total service capacity of the node and number being served, respectively. When a transfer terminates, the service rate for the others increase as the capacity utilized previously for the departed request can now be distributed among the current transfers. Hence we model each end peer as a $M/G/1/m$ Processor Sharing (PS) queue.

Insensitivity results for $M/G/1/m$ PS queues [9], reveal that the state probability distribution and blocking or loss probability results are identical to those obtained for the corresponding $M/M/1/m$ PS queue, whose state probabilities in turn are identical to a $M/M/1/m$ system. The state probabilities p_k are then given by

$$p_k = \frac{\rho^k (1-\rho)}{1-\rho^{m+1}} \qquad P_l = \tfrac{\rho^m (1-\rho)}{1-\rho^{m+1}} \qquad \rho = \lambda_{di} \hat{X} \tag{15}$$

for $k = 0, 1, \ldots, m$ where P_l is the blocking or loss probability i.e. the probability that the threshold limit for the file transfers has been reached, and $\hat{X}$ is the average service time per request. The results for the average number of connections and waiting time are the same for all the peers. Hence we omit the subscript in the equations i.e., the waiting time of the i^{th} peer $\hat{X}_i \equiv \hat{X}$ and so on.

Throughput of the $M/G/1/m$ PS queue can be written as $\lambda_P(1 - P_l)$, where $\lambda_P = \lambda_{di}$ is the overall rate of request arrival. Not all requests that are made get serviced, due to the file transfer threshold imposed, and hence only those arrivals that make it before the threshold is reached obtain service. This probability is $(1 - P_l)$. Thus the effective rate of arrival to the peer becomes $\lambda_P(1 - P_l)$. The throughput can be equated to the net arrival rate since no loss occurs at the peers, i.e. a file transfer is not terminated midway. Implicit in this derivation is that the end peer remains online throughout the period of the file transfer. Using Little's Law the expected service time that a user encounters can then be expressed as

$$\begin{aligned} E[N_p] &= \lambda_P (1 - P_l) E[W_p] \\ \Rightarrow \quad E[W_p] &= \frac{E[N_p]}{\lambda_P (1 - P_l)} \end{aligned} \tag{16}$$

where $E[N_p]$ denotes the expected number of file transfers in progress at the end peer at any given instant of time. $E[N_p]$ is given by

$$E[N_p] = \sum_{i=1}^{m} i p_i \tag{17}$$

where p_i is given in Eq. (15). Since the end peer is a Processor Sharing system, the arriving request does not spend any time waiting in the queue for service. Hence the total time spent at the peer, is equal to the service time.

As an aside, if the restriction on the number of simultaneously allowable file transfers is lifted, i.e. $m \to \infty$, the expected number in the system would then be

$$\begin{aligned} E[N_p] &= \sum_{i=1}^{\infty} i p_i = \frac{[\rho e^{\rho}]' - e^{\rho}}{e^{\rho}} \\ &= \rho \end{aligned} \tag{18}$$

The blocking or loss probability would become zero due the presence of the infinite server pool available.

C. Query Search Time

The time spent searching the network for a resource is an important benchmark to measure the performance of a peer-to-peer system. Although the potency of a query search hinges upon several factors such as file popularity, number of online peers, link bandwidth etc., we beleive that the queuing at the core routers of the physical network is also influential. We define the query search time as the time taken for the *entire* search process to terminate and *not* just the time for the first hit. We discuss the two most popular search strategies employed currently in commercial P2P networks, such as Napster, Gnutella and Kazaa, and derive an expression for the query search time.

The search technique employed to retrieve information in P2P networks can be broadly classified under two categories:

- An architecture wherein a central server contains an index of all the files that the nodes in the peer-to-peer network share. In such an architecture, the search time for a query is primarily the average lookup time to retrieve the information. Thus

  $$E[T_{QS}] = \frac{k}{\mu_{Cs}} \tag{19}$$

  where k is a constant and μ_{Cs} is the mean service time of the central server.
- A decentralized search architecture, wherein a peer forwards the query to it's immediate neighbors and this process is repeated until a specified threshold (TTL_P) is reached. Thus search proceeds by flooding the network. In order to limit the scope of flooding, a TTL value is associated with each query. Each time a peer receives a query, it decrements the TTL value and propagates the query further to it's peers only if the TTL value is greater than zero. Furthermore, there are two variants of the decentralized architecture
  - A hierarchical overlay network wherein certain peers, termed *group leaders*, are delegated with the responsibility of mapping the names of content to IP addresses, for all the peers that have been assigned to it's group. Kazaa and FastTrack employ this architecture.
  - A flat, unstructured distributed topology where all peers are equal; there is no hierarchical structure with *group leaders*. Such networks resort to *query flooding* when searching for data on the network. The Gnutella network is an example of this architecture.

The peers and their communication relationships form an abstract, logical network called an *overlay network*. Note that the edges in the overlay network are *not* physical communication links, but instead only virtual links between the peers. For example even though two peers span different ISPs on different continents, they can be directly connected by an edge in the overlay network, and therefore become one-hop neighbors. The communication between two neighbors can span several physical links and the response time for a query thus, can be significantly influenced by the number of routers, i.e. the total queuing delay, encountered in the path. The task then remains to find the average number of routers between two peers in the network. Equivalently, we require the typical length of the shortest path between two randomly chosen nodes on the *router graph*. For any random graph it has ben shown in [19] that this distance is approximately:

$$\langle d \rangle = \frac{ln[(N_R - 1)(\hat{z_2} - \hat{z_1}) + \hat{z_1}^2] - ln(\hat{z_1}^2)}{ln(\hat{z_2}/\hat{z_1})} \tag{20}$$

where $\hat{z_i}$ is the average number of i hop neighbors and N_R is the total number of nodes in the router graph. Since this is inherently a topological property, the information embedded in the router adjacency matrix, a known entity, can be utlized to derive expressions for $\hat{z_1}$ and $\hat{z_2}$. It is not too difficult to see that

$$\hat{z_1} = [\sum_{i,j=1}^{N_R} \mathcal{A}_{ij}]/N_R \qquad \hat{z_2} = [\sum_{\substack{i,j=1\\ i \neq j}}^{N_R} \mathcal{I}_{\hat{\mathcal{A}}}(i,j)]/N_R$$

where $\mathcal{A}$ is the router adjacency matrix, $\hat{\mathcal{A}} = \mathcal{A}^2$ and $\mathcal{I}_{\hat{\mathcal{A}}}(i,j)$ defined as:

$$\mathcal{I}_{\hat{\mathcal{A}}}(i,j) = \begin{cases} 1 & \text{if } \hat{\mathcal{A}}_{ij} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

The query propagates in the peer network for TTL_P hops, and the response traces back to the originating host along the same path in the overlay network through which it was forwarded. The query process terminates when the last of the responses finds it's way back to the source. The expected time elapsed between the query generation and termination is thus

$$E[T_{QS}] = [2TTL_P \langle d \rangle \sum_{i=1}^{N_R} (E[W_{Q_i}] + \tau_i)]/N_R \tag{22}$$

This is because the query packet encounters $\langle d \rangle$ routers on an average between two one hop peers, and since it is forwarded further for a total of TTL_P hops, the total number of routers encountered along the forward path is $TTL_P \langle d \rangle$. The factor of 2 comes in since the query response traces the same forward path back to the query originiator. Note that $\sum_{i=1}^{N_R}(E[W_{Q_i}] + \tau_i)/N_R$ is the average queueing delay at a router where $E[W_{Q_i}]$ is given in Eq. (11).

D. Expected Download Time

1) Aggregate Peer Latency: Peer to peer networks like Kazaa exploit the existence of multiple copies of a file to reduce the total download time by transferring different fragments of the file from different peers in parallel. Therefore a file that is highly replicated can be expected to have a smaller download time than a file with lesser number of copies. Note that the performance improvement also depends on the loads at the individual peers with the copies of the file, which in turn depends on factors like the number of files a peer allows for sharing, the maximum number of simultaneous downloads a peer allows etc. In this section we derive expressions to characterize the effect of splitting the download on the transfer time.

Measurement studies have shown that the number of replicas of files in Napster and Gnutella is heavily skewed [6]. A suitable and accepted candidate distribution to model such a phenomenon is the Zipf distribution. In order to find the number of occurrences of a certain file, we assume the knowledge of two modeling parameters, viz. the total number of files currently shared in the entire network V, and the number of files shared by the peers that are currently on-line V_{on}. Then, by Zipf's Law, the i^{th} most frequent object from a total of V files occurs

$$O(i) = \frac{V_{on}}{i^{\theta} H_{\theta}(V)} \tag{23}$$

times in a collection of V_{on} files, where $H_{\theta}(V)$ is the harmonic number of order θ of V and is defined as

$$H_{\theta}(V) = \sum_{i=1}^{V} \frac{1}{i^{\theta}} \tag{24}$$

We now present a generic analysis for the congestion measures at the peers. Among the $O(i)$ peers available, assuming that equal amounts are downloaded from each peer starting at the same time, the download time is characterized by the "worst" peer, i.e. the peer with the maximum service time. Let the arrival rates at the $O(i)$ peers, $A_1, A_2, \ldots, A_{O(i)}$, be independent and identically distributed, continuous random variables having a common density f and distribution function F. Define

$$A_{[O(i)]} = \max\{A_1, A_2, \ldots, A_{O(i)}\}$$

Using results from order statistics, the density function of $A_{[O(i)]}$ is given by

$$f_{A_{[O(i)]}}(x) = \frac{[O(i)]!}{[O(i)-1]!}[F(x)]^{O(i)-1} f(x) \tag{25}$$

Thus, having obtained the distribution of the largest arrival rate, we use the expected value of the distribution to characterize the arrival rate at the "worst" peer, λ_{WP}.

$$\lambda_{WP} = \int_0^{\infty} x f_{A_{[O(i)]}}(x) dx \tag{26}$$

Now each peer allows a random number (m) of simultaneous downloads and we assume that each peer choses this number independently from the same distribution. Then given that the worst peer allows m files to be downloaded concurrently at any instant of time, the expected number of files it is serving at any point in time, $E[N_{WP}]$, is given by

$$E[N_{WP} \mid m] = \sum_{i=0}^{m} i p(i) \tag{27}$$

where the state probabilities p_i are given in Eq. (15). Unconditioning on m, we have

$$E[N_{WP}] = \sum_{j=0}^{\infty} \left[\sum_{i=0}^{j} i p(i) \right] P(m = j) \tag{28}$$

When $O(i)$ copies of the file being requested are available, we schedule $B/O(i)$ bytes of data to be transfered from each peer where B is the total file size. The expected service time for the data transfer at the "worst peer" is then

$$E[T_{WP}] = \frac{B/\{O(i)\}}{C/E[N_{WP}]}. \tag{29}$$

2) Expected file transfer time: We conclude this section by presenting the final expression for the file download time, i.e. the time from when the query was generated until the the entire file is downloaded, with $O(i)$ copies of the file in the network. Note that the network delay derived in Section III-A is the delay encountered *per packet* and each packet spends a total time of $E[T_{N_R}]$ in the network, independent of the others. Thus, the download time is determined by the time when the last packet, that of the "worst" peer, reaches the destination. The time when the last packet reaches the edge of the network is when the "worst" peer is done transmitting it's allocated file part i.e. after $E[T_{WP}]$ seconds. The packet, then spends a further $E[T_{N_R}]$ in the network. Thus

$$E[T_D] = E[T_{WP}] + E[T_{N_R}] \tag{30}$$

where $E[T_D]$ denotes the total download time where $E[T_{WP}]$ and $E[T_{N_R}]$ are given in Eqns. (29) and (14) respectively. Incorporating the expression for the search time in the final expression for the overall waiting time, $E[T]$, gives

$$E[T] = E[T_D] + E[T_{QS}] \tag{31}$$

with $E[T_D]$ as in Eq. (30) and $E[T_{QS}]$ given by Eq. (19) for a centralized architecture and by Eq. (22) otherwise.

a) LAN: The expression for the file download time in Eqn (31) can be tailored to reflect more closely a LAN environment and thus campus or enterprise wide P2P networks. If all the $O(i)$ replicas are within the same subnet, the total download time would approximately be $E[T_{WP}]$, i.e. the network component can be ignored since no routers are involved. If there exists at least one host residing on a different subnet, the queuing delay at the routers comes into play. Using simple combinatorics, the number of ways to distribute $O(i)$ hosts among n_s subnets without any restriction is given by $\binom{O(i)+n_s-1}{O(i)}$. Then, the probability that all peers reside in the same subnet is given by

$$Pr\{\text{same}\} = \frac{\binom{n_s}{1}}{\binom{O(i)+n_s-1}{O(i)}} \tag{32}$$

where n_s is the total number of subnets in the LAN. Thus the probability that there is at least one peer on a different subnet having a copy of the file being downloaded is $Pr\{\text{diff}\} = 1 - Pr\{\text{same}\}$. Thus, Eq. (31) can be modified as

$$\begin{aligned} E[T_{LAN}] &= E[T_{QS}] + Pr\{\text{same}\}E[T_{WP}] \\ &\quad + Pr\{\text{diff}\}\big(E[T_{WP}] + E[T_{N_R}]\big) \end{aligned}$$

IV. Multi-part Download

Often, due to replication of files, multiple peers host a copy of the requested file in the network. Splitting the request into non-overlapping parts and downloading the respective part from each peer, instead of a single peer being responsible for the entire download, often reduces the load on the peers as well as saving on the total download time. The question that naturally arises is: *How should the file be split among the peers so as to minimise the total download time ?.* We claim that each peer should be allocated a part that is proportional to it's rate of service in order to minimize the file transfer time. The proof for our claim is elaborated below.

Claim 1 *In a multi-part download, an allocation strategy which downloads a part of the file from each peer proportional to it's service rate at the time of request minimizes the overall download time.*

Proof: The proof presented here assumes the service rate of each peer to be static and invariant during the course of download. This can easily be extended to a dynamic allocation by sampling the instantaneous rates and using the above scheme to determine the new assignments.

Let r_i, f_i and t_i denote the service rate of the i^{th} peer, size of the file F to be downloaded from i^{th} peer and the time taken to download f_i from the i^{th} peer respectively. Also, let t denote the total download time. Note that $t_i = f_i/r_i$. The download time for the entire file is determined by the time taken for the "worst" peer to finish it's service, i.e.,

$$\begin{aligned} t &= \max\{t_1, t_2, \cdots, t_n\} \\ &= \max\{\frac{f_1}{r_1}, \frac{f_2}{r_2}, \cdots, \frac{f_n}{r_n}\} \end{aligned}$$

If the file part allocation is done proportional to the rates then we have

$$\frac{f_1}{r_1} = \frac{f_2}{r_2} = \cdots = \frac{f_n}{r_n}$$

Therefore, $t_1 = t_2 = \cdots = t_n$ and all n peers take the same time to finish servicing their allocated quota, and we denote this time by t_a. Thus

$$t_a = \max\{t_1, t_2, \cdots, t_n\}$$

where $t_1 = t_2 = \cdots = t_n$. Since all hosts have equal download times we have

$$(r_1 + r_2 + \cdots + r_n)t_a = F \tag{33}$$

Now, consider an arbitrary allocation of the file parts where t_i denotes the transfer completion time of the i^{th} peer and and let t_b denote the maximum of these n times. Here not all the t_i, $i = 1, \cdots, n$ are equal, else it would equivalent to the previous case. Thus in this scenario there exists at least one peer i such that, $t_i < t_b$. Therefore

$$(r_1 + r_2 + \cdots + r_n)t_b > F \tag{34}$$

This can be explained as follows: in the case of arbitrary file-part transfers assignment we have

$$\sum_{k=1}^{n} t_k r_k = F \tag{35}$$

Since there exists at least one value distinct from t_b, consider the case where $t_k = t_b \quad \forall \quad k \neq i$. In this case, the previous equation can be written as

$$\sum_{\substack{k=1 \\ k\neq i}}^{n} t_b r_k + t_i r_i = F \tag{36}$$

Now $t_b r_i > t_i r_i$ since t_b is the maximum. Thus

$$\begin{aligned} \sum_{\substack{k=1 \\ k\neq i}}^{n} t_b r_k + t_b r_i \quad &> \quad \sum_{\substack{k=1 \\ k\neq i}}^{n} t_b r_k + t_i r_i \\ \Rightarrow t_b \sum_{1}^{n} r_k \quad &> \quad F \ \text{(From Eq. (36))} \end{aligned}$$

Hence Eq. (34) holds. Clearly, the above proof holds if there exists more then one transfer time that differs from the maximum. The ratio of Eq. (34) and Eq. (33) gives

$$\frac{t_b(r_1 + r_2 + \cdots + r_n)}{t_a(r_1 + r_2 + \cdots + r_n)} \quad > \quad 1$$

therefore we have $t_b > t_a$. ■

Futher results on the relative performance for three allocation strategies, namely rate proportional allocation, equal allocation and random allocation on the download times is presented in Section V.

V. Simulation Results

In this section, we validate our analytical model by comparing the results with those obtained from simulations. We also use simulations to capture the interplay among the various network and peer parameters and their contribution to the performance of the P2P system. To prove the robustness of the model, simulations are carried out for three structurally very different topologies: (1) a real University network (Columbia University), (2) power law AS level topologies and (3) a national backbone (AT&T) with Internet service provider (ISP) level topologies.

The location of the peers hosting the requested file, plays a central role in influencing the total file download period. In order to account for this, we repeat the simulation for a given set of parameters 200 times with various combinations of the source and destination peers so that the final time, calculated as the average of the 200 runs, is representative of the delay encountered by any source-destination tuple in the topology. The parameter values that remain fixed accross all simulations are: $\tau_i = .002$, and the peer service rate $\mu_p = 10$.

The rest of the section is structured as follows : Section V-A elaborates on the setup and results for the Columbia University network while the power law topology and the ISP network are discussed is Sections V-B and V-C respectively. Finally, in Section V-D, we reflect on the implications of the simulation/analysis plots obtained.

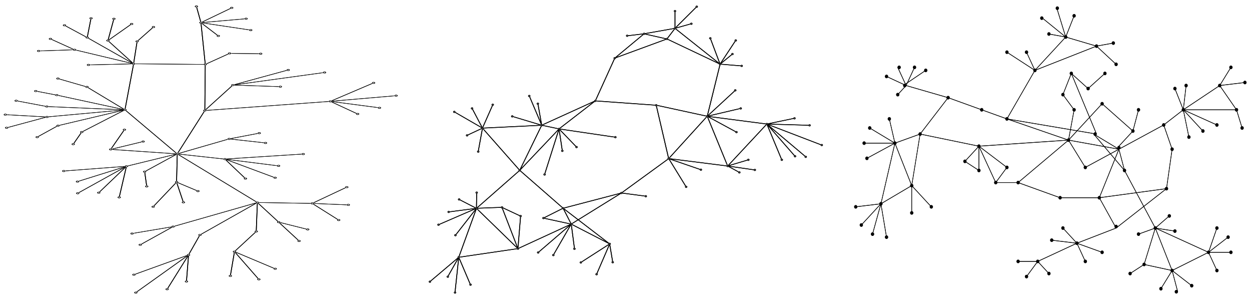

(a) Columbia University network (b) Two tier AS router network (c) AT&T network

Fig. 4. Simulation topologies.

A. University Network

To evaluate our model in a campus level network, we simulated the topology of Columbia University ([20]). The topology from [20] is shown in Fig. 4(a) and comprises of 92 nodes, 34 core routers and 58 peers. We assumed that most of the peers reside in the various dormitories of the University and that only a handful are active from within the various departments. For the simulations a random number (between 2 and 5) of peers were attached to each subnet while the department routers were assigned either one or two peers. Note that since our model groups all non P2P traffic together as external traffic, we do not have to explicitly place non-peer nodes in the simulation topologies. A value of $c_{si} = 1.0$ was chosen for these simulations.

To validate our model, in Fig. 5(a) we compare the simulation and the analysis results for the download time. The size of the file being transferred was chosen to be 120 packets and the figure plots the download time as a function of the degree of replication of the requested file in the network. We note that the simulation results match closely with the analysis and as expected, the download time decreases with increasing number of copies.

B. Power law Topology

The power law topology generated using BRITE [24] was constructed as a two-tier hierarchical network with 25 routers and 50 peers. Peers are attached randomly to the network, with the constraint that the chosen router be an Intra-AS node rather than an Inter-AS one. The resulting topology is as shown in Fig. 4(b). A value of $c_{si} = 1.25$ was chosen for these simulations and the file size was again 120 packets. Fig. 5(b) compares the simulation and the analysis results for the download time for this topology and the download time is plotted as a function of the degree of replication of the requested file in the network. We again note the close match with the simulation results.

C. ISP Network

The third topology considered is that of an ISP network, specifically, the topology of AT&T's backbone in the United States. The backbone layout obtained is from [26] and the network was extended by attaching a random number of Autonomous Systems (generated using BRITE [24]) to each core router. The peers were attached randomly to these AS routers. The final layout consisting of 44 routers, both backbone and AS, and 50 peers is shown in Fig. 4(c). Again a value of $c_{si} = 1.25$ was used and the file size was 120 packets. Fig. 5(c) compares the simulation and the analysis results for the download time for this topology as a function of the degree of file replication and we again note the close match with the simulation results.

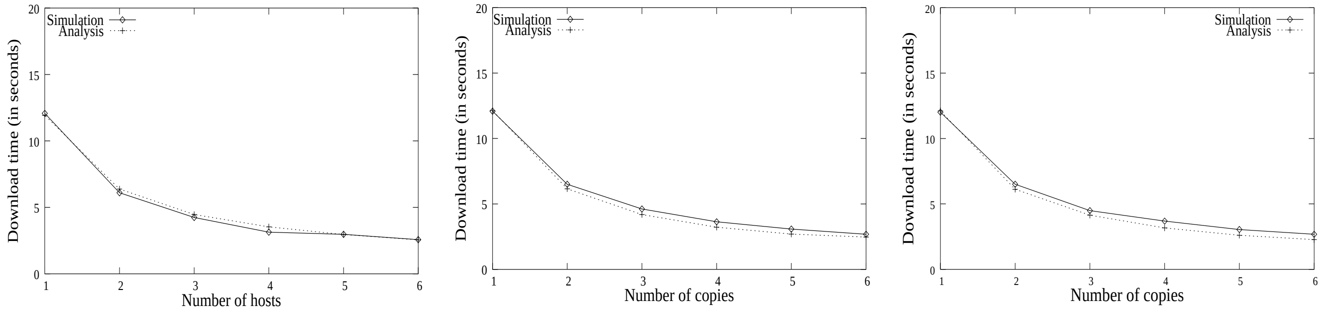
D. Sensitivity Analysis

We now evaluate the impact of P2P network features on the download times. Fig. 6(a), shows the download time as a function of the file size and number of copies. We note that the decrease in the file transfer time is not linear with the number of copies available in the network. This is because the network delay, which is small compared to the peer delays for small number of copies, now starts dominating the total download time. Fig. 6(b) shows the impact of the external traffic rate and its SCV at the core routers on the file download time. The external rate of traffic is uniformly increased accross all the routers in the network until the utilization of the busiest among them reaches 1. When this threshold is attained, the network becomes unstable, resulting in a steep increase in waiting time (theoretically infinity). The sharp upward curve in Fig. 6(b) concurs with this observation. Finally, Fig. 6(c) shows the effect of the file popularity and the number of simultaneous downloads allowed by a peer on the download times. We note that the number of allowed downloads has a more significant impact on the performance.

E. Effect of File Allocation Strategies

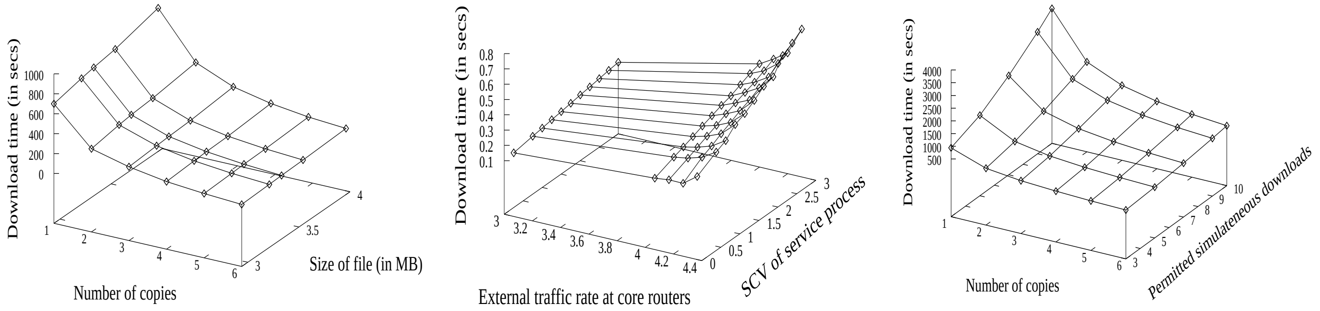
In Section IV we proved that an allocation strategy which divides the allocations proportional to the service rates is the optimal strategy. We now use simulations to compare this strategy against two others to evaluate the degree of performance improvement obtained with the optimal strategy. In addition to the optimal proportional allocation we consider strategies where (1) an equal amount is downloaded from each peer and (2) a randomly chosen amount is downloaded from each peer. The simulations were conducted on the Columbia university topology of Fig. 4(a).

In Fig. 7 we plot the download times associated with with the three strategies as a function of the file size. For these simulations, 4 copies of the file were assumed to be available. We



(a) Columbia University network (b) Two tier AS router network (c) AT&T network

Fig. 5. Download time vs. Number of copies for the three topologies.



(a) File size vs copies (b) External traffic rate vs SCV (c) Copies vs simultaneous downloads

Fig. 6. Parameter sensitivity analysis.

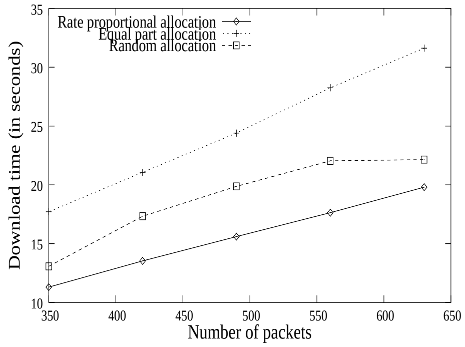


Fig. 7. Download times for different allocation strategies.

note that as expected, the proportional allocation leads to significantly lower delays. Also, the the download time is a linear function of the file size validating the intuition that the peer's component of the delay dominates the network delay in a LAN type environment.

VI. Conclusions

In this paper, we presented an analytic framework to model the performance of peer-to-peer networks. The model evaluates the expected time to download a file in the P2P network and accounts for a host of network and peer level characteristics. Our model accounts for the search times, the queueing delays at the routers of the network and peer characteristics like the number of simultaneously allowed downloads at a peer, the number of copies of the file etc. The model has been validated using simulations with campus level, power law AS level and ISP level topologies. Using the model and extensive simulations we also illustrated the interplay among various critical parameters such as external traffic rates, service variability, file popularity etc. and their influence on the download times. The paper also showed that a rate proportional allocation strategy is optimal for minimizing the file download time in scenarios with multi-part downloads. Our results also show that, the presence of multiple copies of a file beyond a certain number does not result in a proportional decrease in the transfer time.

Acknowledgements

This work was supported in part by NSF under award number 0347623.

References

[1] "Napster Protocol Specification," March 12 2001, http://opennap.sourceforge.net/napster.txt

[2] Characterization of Internet traffic loads, segregated by application, `http://www.caida.org/analysis/workload/byapplication/`.

[3] Kazaa. http://www.kazaa.com

[4] Clip2, "The Gnutella Protocol Specification v0.4," March 2001, http://www.clip2.com/GnutellaProtocol04.pdf.

[5] Z. Ge, D. Figueiredo, S. Jaiswal, J. Kurose and D. Towsley, "Modeling peer-peer file sharing systems," *Proceedings of IEEE INFOCOM,* San Francisco, CA, March 2003.

[6] Eytan Adar and Bernardo A. Huberman, "Free Riding on Gnutella," *First Monday*, vol. 5, no. 10, October 2000.

[7] W. Whitt, "The Queuing Network Analyzer," *The Bell Systems Technical Journal*, 2779-2815, 1983.

[8] W. Whitt, "Performance of the Queueing Network Analyzer", *Bell System Technical Journal*, vol. 62, no. 9, pp 2817-2843, Nov. 1983.

[9] D. Burman, "Insensitivity in queuing systems," *Advances in Applied Probability*, vol. 13, pp. 846-859, 1981.

[10] S. Saroiu, K.P. Gummadi, R.J. Dunn, S.D. Gribble and H.M Levy, "A Measurement Study of Peer-to-Peer File Sharing Systems," *Proceedings of Multimedia Computing and Networking 2002 (MMCN '02),* San Jose, CA, January 2002.

[11] S. Saroiu, K.P. Gummadi, R.J. Dunn, S.D. Gribble and H.M Levy, "An Analysis of Internet Content Delivery Systems," *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002),*, Boston, MA, December 2002

[12] D. Qiu and R. Srikant, "Modeling and performance analysis of BitTorrent-like peer-to-peer networks," *Proceedings of ACM SIGCOMM,* Portland, OR, August 2004.

[13] M. Ripeanu, A. Iamnitchi and I. Foster, "Mapping the Gnutella Network," *IEEE Internet Computing*, vol. 6, no. 1, pp. 50-57, January 2002.

[14] T. Ng, Y.-H. Chu, S. Rao, K. Sripanidkulchai and H. Zhang, "Measurement-based optimization techniques for bandwidth-demanding peer-to-peer systems", *Proceedings of IEEE INFOCOM,* San Francisco, CA, April 2003.

[15] F. Clevenot and P. Nain, "A simple model for the analysis of the Squirrel peer-to-peer caching system," *Proceedings of IEEE INFOCOM,* Hong Kong, China, March 2004.

[16] X. Yang and G. de Veciana, "Service capacity in peer-to-peer networks," *Proceedings of IEEE INFOCOM,* pp. 1-11, Hong Kong, China, March 2004.

[17] D. Liben-Nowell, H. Balakrishnan and D. Karger, "Observations on the dynamic evolution of peer-to-peer networks, *Proceedings of IPTPS,* pp. 22-33, Cambridge, MA, March 2002.

[18] M. Faloutsos, P. Faloutsos and C. Faloutsos, "On power-law relationships of the Internet topology," *Proceedings of ACM SIGCOMM,* pp. 251-262, Cambridge, MA, August 1999.

[19] M.E.J Newman, S.H. Strogatz, and D.J. Watts, "Random graphs with arbitrary degree distribution and their applications," *Physical Review E,* vol. 64, no. 026118, 2001

[20] Columbia University Network, `http://www.columbia.edu/acis/maps/fibermap.gif`

[21] W. E. Leland, M. S. Taqqu, W. Willinger and D. V. Wilson, "On the self-similar nature of Ethernet traffic (Extended Version)," *IEEE/ACM Trans. on Networking,* vol. 2, no. 1, pp. 1-15, Feb 1994.

[22] V. Paxson and S. Floyd, "Wide area traffic: The failure of Poisson modeling," *IEEE/ACM Trans. on Networking,* vol. 3, no. 3, pp. 226-244, June 1995.

[23] M. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: Evidence and possible causes," *IEEE/ACM Transactions on Networking,* vol. 5, no. 6, pp. 835-846, December 1997.

[24] BRITE, `http://www.cs.bu.edu/brite`

[25] Celery Search Engine, `http://site.n.ml.org/info/_celery/`

[26] `http://www.jingleinc.com/html/backbone.html`