A Queueing Model for Evaluating the Transfer Latency of Peer to Peer Systems

Krishna Kumar Ramachandran and Biplab Sikdar *Member, IEEE*

Abstract—This paper presents a queueing model to evaluate the latency associated with file transfers or replications in peer to peer (P2P) computer systems. The main contribution of this paper is a modeling framework for the peers that accounts for the file size distribution, the search time, load distribution at peers and number of concurrent downloads allowed by a peer. We propose a queueing model that models the nodes or peers in such systems as M/G/1/K processor sharing queues. The model is extended to account for peers which alternate between online and offline states. The proposed queueing model for the peers is combined with a single class open queueing network for the routers interconnecting the peers to obtain the overall file transfer latency. We also show that in scenarios with multi-part downloads from different peers, a rate proportional allocation strategy minimizes the download times.

Index Terms—Peer-to-peer networks, Queueing Model, Performance Evaluation

1 INTRODUCTION

Peer to peer systems provide a paradigm shift from the traditional client server model of most networked computing 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 or over the Internet as a whole. These systems have grown in popularity in the recent past and the fraction of network traffic originating from these networks has consistently increased. Developing models to understand and quantify the impact of factors affecting their performance is of importance to facilitate the development of P2P systems and to ensure proper utilization of the networking infrastructure. In this paper we address this issue and develop a queueing model for evaluating the performance of peers in such systems in terms of the latencies associated with file replication while accounting for architectural, topological and user related factors.

The paradigm shifts associated with P2P systems and its inherent features necessitate the development of new models to account for their behavior. The presence of a node in the P2P system can be transitory with peers continually joining and leaving the network arbitrarily over any given period of time. Also, network and end user heterogeneities 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 [5], [13], [17], [20], [25], [29], [32], [1], [31], [27], [30], [43]. Analytic efforts to model the performance of P2P networks using fluid or branching process based Markov models are presented in [9], [10], [15], [19], [28], [34], [26], [39], [22], [42] and focus on the steady state behavior of the number of peers in the network. A closed queueing system model for P2P systems is presented in [16] that focuses on the saturation throughput of the system and ignores the effect of the network topology. The existing models in literature fail to capture the performance of a P2P system in terms of a user's viewpoint: "*How long does it take to replicate a file, if available, in the P2P system?*" while accounting for the various user and network level factors which are inherent to P2P systems. This paper addresses this issue.

The latency associated with a file replication in a P2P system consists of two components: the query search time and the time required by the peers to transmit the file. In order to model the peer level latency, we develop a queueing model to evaluate the time required at each peer to serve its replication requests. Each peer is modeled as M/G/1/K processor sharing queue with arbitrary constraints on the number of simultaneous downloads allowed by the peers and file size distributions. We also develop models to evaluate the search time associated with a query in both centralized and decentralized P2P systems. To evaluate the overall delay, these models are then combined with existing results for single class open queueing networks with arbitrary arrival and service patterns (which evaluate the router level delays by modeling each router as a GI/G/1 queue). The model is able to account for a number of factors of P2P systems and network heterogeneities like file popularity and size distribution, peer specific settings like the number of simultaneous downloads, different access rates, physical topologies, search strategies and

- *K. Ramachandran now works at India Science Labs, General Motors, Bangalore, Karnataka, India. The work was done while he was at Rensselaer Polytechnic Institute, Troy, NY, USA. E-mail:krishna.k.ramachandran@gmail.com*
- *B. Sikdar is with the Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY, USA. E-mail:sikdab@rpi.edu*

Manuscript received April 14, 2008; revised February 4, 2009.
This work supported in part by NSF grant 0347623.

transitions between online and offline states. 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 [11]. 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 2 we differentiate the work presented in this paper from existing literature. Section 3 presents the analytic framework for evaluating the peer level latencies and Section 4 discusses download strategies in multi-part scenarios. In Section 5 we present simulation results to verify our model and also analyze the impact of various factors on the network's performance. Finally, Section 6 presents the concluding remarks.

2 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, storage and grid computation. While the majority of the existing literature on the performance of P2P systems has focused on measurements, various analytic models have also been proposed. A measurement study of the Gnutella network's properties is presented in [29]. In [32], 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 [1]. The measurement study in [31] characterizes the behavior of users as well the network, for the Gnutella and Napster networks. In [17], [27] the authors present measurement studies of the BitTorrent P2P system. A measurement study of the nature and magnitude of file pollution in the KaZaA P2P system is undertaken in [20] while [43] presents a measurement study of the characteristics of available files in the Gnutella P2P system. A measurement based study of the eDonkey P2P system is presented in [30]. The effect of free-riding and free-identities on the performance of peer-to-peer systems is studied in [13]. A simulation based study of BitTorrent is presented in [5]. In contrast to these studies, this paper develops an analytic model for evaluating P2P networks.

In [28], [15] fluid models are used to characterize the performance of BitTorrent like networks in terms of the average number of downloads and download times. A fluid model capable of accounting for features such as peer churn, heterogeneous upload capacity and limited infrastructure capacity is presented in [19]. User churn is also modeled in [41] while [42] uses a fluid model to evaluate the impact of free-riding in BitTorrent like P2P systems. A fluid based model for eMule-like P2P systems is proposed in [26]. Stochastic fluid models are also used in [9], [10] to model the performance of P2P web caches and web clusters. Branching process based Markovian models to study BitTorrent like networks are used in [39], [34]. In [22] 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 in the network.

A simple model to study search strategies in large-scale decentralized unstructured P2P networks is presented in [14]. The fairness of resource allocation strategies in P2P networks is studied in [40]. A bound on the replication time of a file that is divided into multiple parts is obtained in [23] without considering the network and queueing delays. Similar bounds on the download time based on the service rates of neighboring peers are provided in [21]. The server capacity in P2P systems with streaming data is modeled in [35], [33], [38]. A game theoretic analysis of P2P systems that evaluates the impact of user altruism is presented in [36]. A Markov chain model for schemes for recovering lost data in P2P storage systems is presented in [2]. In contrast to the literature described above, this paper focuses on evaluating the download time associated with a given file while accounting for the search mechanism, node heterogeneity and the queueing effects.

The literature closest to our approach is that in [16] 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 [16] 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.

3 Analytic Framework

In this section, we present our analytic framework for modeling the performance of peer-to-peer networks. In general, a P2P system can be considered to consist of an

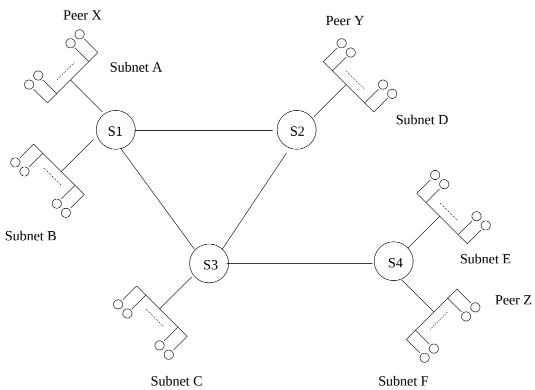


Fig. 1. An example topology of a peer-to-peer network in a campus environment.

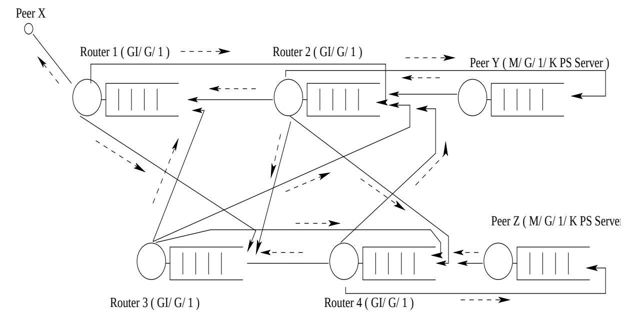


Fig. 2. Queuing model equivalent of Fig 1.

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 j^{th} router
c^2_{sj}	SCV of the service distribution of j^{th} router
W_{Q_j}	packet waiting time at j^{th} router before service
Q	routing matrix
p_{ij}	proportion of arrivals to router j from router i
N_C	number of packets in the network at any time
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 connections currently being served at the peer
W_p	service time for a request at the peer
C	link capacity of the peer
m	number of simultaneous requests allowed by a 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 i^{th} most popular file in the network
B	size of the file currently being downloaded
N_{p_i}	number of concurrent downloads at 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 file transfer from the peer(s)
T	Overall waiting time, sum of T_{QS} and T_D

TABLE 1
Notation and model parameters

interconnecting network of routers forming the network backbone which connects the various sub-networks in which the peers themselves reside. These subnets may be local area networks (LANs) in the case of campus networks and autonomous systems (ASes) with their own network of intra-domain routers and subnets in the case of large networks. A simple example scenario is shown in Fig. 1 with four routers and six subnets where user Y in Subnet D wishes to download a file from another peer, X, in Subnet A. From the 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 (1) query search time, (2) peer level delay i.e. the transmission time of the file being downloaded and (3) core network delay i.e. queuing delay at the intermediate routers. 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) The end peers which are analyzed as processor shared nodes with finite/infinite capacity and arbitrary service times.
2) 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.

3.1 Queueing Model for the End Peers

We now propose a queuing model for end peers and derive an expression for the expected time it takes to serve an user requesting a file download, starting with the arrival process. We approach the model from a per file request basis rather than a per packet basis. In the network model considered in this paper, 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 splitting into several streams, one for every active end peer, as shown in Figure 3.

Let there be $\mathcal{N}$ peers in the subnets of a given router and on an average, λ_d download requests arrive at the router per second with an arbitrary inter-arrival time distribution $F_I[\xi]$. It is also reasonable to assume that the fraction $q_i(\mathcal{N})$ of these request arrivals that are destined for particular peer i is inversely proportional to $\mathcal{N}$, with appropriate scaling, depending on the number of files hosted by peer i. Also, assume that the choice of the destination peer as well as the times between successive download requests arriving at a router are not correlated. The presence of cross-traffic, predominantly HTTP, also lends credibility to this assumption. The assumption also becomes more valid as the number of peers in the subnets of the router increases. Let $\xi_1, \xi_2, \cdots$ be the sequence of inter-arrival times of the download requests arriving at a given router. The inter-arrival time between

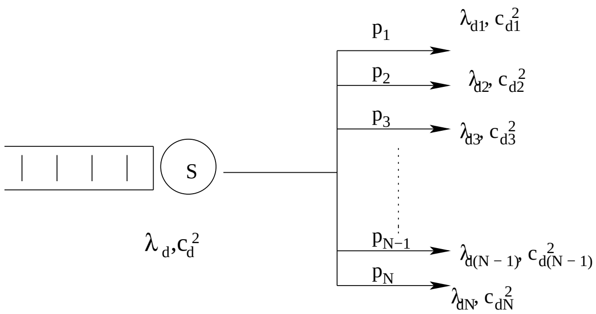


Fig. 3. Splitting of the output stream at the router.

to requests at peer i, T_i, is then given by the random variable

$$\Gamma_i = \sum_{j=1}^{\nu} \xi_j \tag{1}$$

where ν is a geometric random variable with parameter $q_i(\mathcal{N})$, i.e., $P[\nu = n] = q_i(\mathcal{N})(1 - q_i(\mathcal{N}))^n - 1$, for $n = 1, 2, \cdots$. Thus the inter-arrival time between two successive arrivals *at a peer* is a random, geometric sum of the inter-arrival times ξ. The parameter governing the geometric sum is inversely proportional to $\mathcal{N}$ at each peer and thus $\lim_{\mathcal{N}\to\infty} q_i(\mathcal{N}) = 0$ at each peer. Now, it is well known from Renyi's theorem [3], [18] that the geometric sum of any arbitrary distribution for independent, non-negative random variables converges to the exponential distribution as $q_i(\mathcal{N}) \to 0$. Thus the distribution of the time between two download requests at peer i is exponentially distributed with rate $\lambda_{d_i} = q_i(\mathcal{N})\lambda_d$.

The service time is dependent on the size of the file being downloaded. We allow for arbitrary distributions for the service times thereby accommodating generalized models for the file sizes. The rate at which a download request is served also depends on the number of files the peer is willing to let other peers simultaneously download from it at any given instant of time. A savvy peer may limit this number in order to gain download bandwidth leverage (for example in a LAN where upload and download bandwidths are coupled). *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 [1]. If a request for a file is made when the download limit has been reached, it is lost and no file transfer takes place. In other words, a peer allowing at most m simultaneous downloads functions as a node with m servers and no queue buffer. With C and N_r representing the total service capacity of the node and number of requests being served, respectively, the service rate of each download is C/N_r. When transfers are initiated or terminated, the service capacity of the peer is divided equally among the current transfers. Hence we model each peer as a $M/G/1/m$ Processor Sharing (PS) queue.

Insensitivity results for $M/G/1/m$ PS queues [7], 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_i are then given by

$$p_i = \frac{\rho^i(1-\rho)}{1-\rho^{m+1}} \tag{2}$$

$$P_l = \frac{\rho^m(1-\rho)}{1-\rho^{m+1}} \tag{3}$$

$$\rho = \lambda_{d_i}\hat{X} \tag{4}$$

for $i = 0, 1, \ldots, m$ 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. Due to the file transfer threshold imposed, all requests that are made do not get serviced, with probability P_l. Thus the effective rate of arrivals to the peer becomes $\lambda_P(1 - P_l)$ where $\lambda_P = \lambda_{d_i}$ is the overall rate of request arrivals. Throughput of the $M/G/1/m$ PS queue can then be written as $\lambda_P(1 - P_l)$. 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:

$$E[W_p] = \frac{E[N_p]}{\lambda_P(1-P_l)} \tag{5}$$

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

$$E[N_p] = \sum_{i=1}^{m} i p_i \tag{6}$$

Here, p_i is obtained from Eq. (2). 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.

3.1.1 Aggregate Peer Latency

Peer to peer systems like Kazaa and BitTorrent 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. While more copies result in smaller replication times, the degree of improvement depends on the loads at the individual peers with the copies of the file. We now 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 P2P systems like Napster and Gnutella is heavily skewed and may be modeled by a Zipf distribution [1]. We denote the total number of files currently shared in the entire network by V and the number of files shared by the peers that are currently online by V_{on}. Then, by Zipf's Law, the i^{th} most frequent object from a total of V files occurs

$$O(i) = \frac{V_{on}}{i^{\theta} H_{\theta}(V)} \tag{7}$$

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

$$H_{\theta}(V) = \sum_{i=1}^{V} \frac{1}{i^{\theta}} \tag{8}$$

We assume that the downloading peer requests an equal fraction of the file from each of the $O(i)$ peers available. Assuming that equal amounts are downloaded from each available peer starting at the same time, the overall download time is characterized by the time taken by the "worst" peer to serve its share, i.e. the peer with the maximum service time. Here we have used the fact that the download is complete only each of the peers has served its share of the file. Since each peer serves an equal fraction of the file, the file download time is determined by the peer that serves its share at the slowest rate and this peer is referred to as the "worst peer". Since the link capacity, C, at each peer is the same and the maximum number of simultaneous downloads, m, allowed by each peer is drawn from the same distribution, the expected performance of the worst peer is governed by the highest download request arrival rate seen the $O(i)$ peers. 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

$$f_{A_{[O(i)]}}(x) = \frac{[O(i)]!}{[O(i)-1]!}[F(x)]^{O(i)-1} f(x) \tag{9}$$

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}.

$$\lambda_{WP} = \int_0^{\infty} x f_{A_{[O(i)]}}(x) dx \tag{10}$$

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

$$E[N_{WP} \mid m] = \sum_{i=0}^{m} i p(i) \tag{11}$$

where the state probabilities p_i are given in Eq. (2). Unconditioning on m, we have

$$E[N_{WP}] = \sum_{j=0}^{\infty} \left[\sum_{i=0}^{j} i p(i) \right] P(m = j) \tag{12}$$

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

$$E[T_{WP}] = \frac{B/\{O(i)\}}{C/E[N_{WP}]}. \tag{13}$$

3.2 Online-Offline Transition of Peers

While the analysis in the previous subsection assumes that all peers involved in the file transfer stay online, typical peers alternate between online and offline states. We now extend our model to account for these scenarios. The offline (H) and online time (G) distributions of each peer are assumed to be identical and mutually independent with average lengths of $E[H]$ and $E[G]$ respectively.

The implication of the online-offline transitions is that now merely characterizing the performance of the "worst" peer is no longer sufficient. This is because the file allocation is no longer static and is dependent on the number of peers with a copy that are online at that instant of time. We first consider the case when only one copy of the file is available and then generalize the result for the case with multiple copies.

3.2.1 *Single Peer*

When a download request arrives at the peer hosting the file, including this new request, the number of download requests at the peer, i, is between 1 and $m+1$, i.e., $1 \leq i \leq m+1$. With C denoting the service capacity of the peer, the service rate seen by the user is then C/i when $1 \leq i \leq m$ and the service rate is 0 when $i = m+1$ since the limit on the number of simultaneous downloads has been reached and the new request is blocked. Since peers can be modeled as $M/G/1/m$ PS queues when they are online, the conditional probability mass function (pmf) of the number of download requests at the peer is given by

$$Pr\{\text{requests } = i \mid \text{requests } > 0\} = \frac{Pr\{\text{requests } = i\}}{Pr\{\text{requests } > 0\}} = \frac{\rho^{i-1}(1-\rho)}{1-\rho^m}$$

with $1 \leq i \leq m+1$. The pmf of the service rate χ is then

$$Pr\{\chi = x_i\} = \begin{cases} \frac{\rho^{i-1}(1-\rho)}{1-\rho^m} & \text{for } x_i = \frac{C}{i}, \ \ 1 \leq i \leq m \\ \frac{\rho^{m}(1-\rho)}{1-\rho^m} & \text{for } x_i = 0 \end{cases} \tag{14}$$

where $\rho = \lambda_{d_i} \hat{X}$ and λ_{d_i} is the request arrival rate at the peer and $\hat{X}$ is the average service time per request. With the size of the file to be downloaded (in bytes) denoted by B, the expected time to transmit the file is given by $E[B/\chi]$. However, before it completely serves the download request the peer may go offline several times, as shown in Figure 4. Since the average online

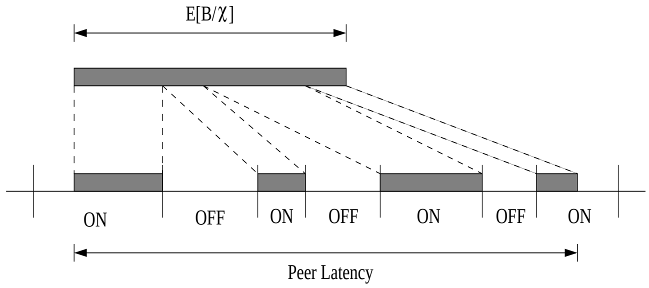


Fig. 4. Peer latency with online-offline transitions.

period is $E[G]$ seconds and online and offline states alternate, the expected number of the peer's transitions into the offline state while the file is being downloaded is $E[B/\chi]/E[G]$. Each of these transits adds $E[H]$ seconds to the peer latency. The total peer latency, $E[T_{DL}]$, is then given by

$$\begin{aligned} E[T_{DL}] &= E[B/\chi] + \frac{E[B/\chi]}{E[G]} E[H] \\ &= \left(1 + \frac{E[H]}{E[G]}\right) \sum_{i=1}^{m} \frac{Bi}{C} \frac{\rho^{i-1}(1-\rho)}{1-\rho^m} \end{aligned}$$

where $E[B/\chi]$ is given by the summation in the equation above.

3.2.2 *Multiple Peers*

Let the number of peers with the copies of the file be denoted by N. When copies of the file are available for download on multiple peers, the download rate seen by the user is the sum of all the download rates from all the available peers. It is also possible that the offline states of all the peers with the copies of the file overlap and the download falls to zero. We call the time when all the peers with the file are offline as the *off* time and the time when at least one of the peers is online as the *on* time and we denote the average time spent in the off and on states by $E[T_{OFF}]$ and $E[T_{ON}]$ respectively. Now the fraction of time an arbitrary peer is online, γ, is given by

$$\gamma = \frac{E[G]}{E[H] + E[G]} \tag{15}$$

The fraction of time the peers stay in the on state, or equivalently, the probability that at least one of the peers is online is given by

$$\frac{E[T_{ON}]}{E[T_{ON}] + E[T_{OFF}]} = 1 - (1-\gamma)^N \tag{16}$$

Also, given that the set of peers are in the on state, the probability that there are n peers online is a conditional binomial distribution with parameter γ and

$$Pr\{n \mid \mathrm{ON}\} = \frac{\binom{N}{n} \gamma^n (1-\gamma)^{N-n}}{1-(1-\gamma)^N} \tag{17}$$

When n peers with copies of the file are online, the pmf of the service rate is obtained by convolving the service rates of the individual peers, as given in Eq. (14). Consider the sequence $\varsigma_0 = 0$ and $\varsigma_i = \frac{C}{i}$ for $1 \le i \le m$. The expected peer latency is then given by

$$\begin{aligned} E[T_{PL}] &= \sum_{n=1}^{N} \Bigg[\sum_{i_1=0}^{m} \cdots \sum_{\substack{i_n=0 \\ i_1+\cdots+i_n>0}}^{m} \frac{B}{\varsigma_{i_1} + \cdots + \varsigma_{i_n}} Pr\{\chi = \varsigma_{i_1}\} \\ &\qquad \cdots Pr\{\chi = \varsigma_{i_n}\} \Bigg] Pr\{n \mid \mathrm{ON}\} \end{aligned} \tag{18}$$

Note that the summation above is done only for those cases where there is at least one peer that does not block the download request (by checking the condition $i_1 + i_2 + \cdots + i_n > 0$). Now, as in the single peer case, the expected number of times the set of peers transitions into the off state is given by $E[T_{PL}]/E[T_{ON}]$, each of which adds $E[T_{OFF}]$ time units to the download time. The expected download time is thus

$$\begin{aligned} E[T_{DL}] &= E[T_{PL}] + \frac{E[T_{PL}]}{E[T_{ON}]} E[T_{OFF}] \\ &= \frac{1}{1-(1-\gamma)^N} E[T_{PL}] \end{aligned} \tag{19}$$

where we have used Eq. (16) and $E[T_{PL}]$ is given in Eq. (18).

3.3 Query Search Time

Before a peer can start downloading a file, it has to first search for the desired content in the P2P system. 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, since knowledge of and selection from all available peers with the desired content results in the best download time. We now derive the expressions for the query search time in centralized and decentralized P2P architectures.

Centralized: In centralized architectures a central server contains an index of all the files that the nodes in the P2P system share. In such an architecture, the search time for a query is primarily the average lookup time to retrieve the information. Thus the expected query search time is given by

$$E[T_{QS}] = \frac{k}{\mu_{Cs}} \tag{20}$$

where k is a constant and μ_{Cs} is the mean service time of the central server.

Decentralized: In such architectures a peer forwards the query to it's immediate neighbors and this process is repeated until a specified threshold for the maximum number of hops (TTL_P) is reached. The query search time thus depends on the time to live (TTL) value associated with the queries and the number of routers the queries have to pass through. However, the number of routers is not equal to the TTL value since each hop on the logical, overlay network connecting the peers may comprise of a number of physical communication links and their routers. The average number of routers between two peers in the network is given by the expected length of the shortest path between two randomly chosen nodes on the *router graph*. For any random graph it has been shown in [24] that this distance is approximately:

$$\langle d \rangle = \frac{\ln[(N_R - 1)(\hat{z_2} - \hat{z_1}) + \hat{z_1}^2] - \ln(\hat{z_1}^2)}{\ln(\hat{z_2}/\hat{z_1})} \tag{21}$$

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 utilized to derive expressions for $\hat{z_1}$ and $\hat{z_2}$. It is not too difficult to see that

$$\hat{z_1} = \frac{1}{N_R} \sum_{i,j=1}^{N_R} \mathcal{A}_{ij} \tag{22}$$

$$\hat{z_2} = \frac{1}{N_R} \sum_{\substack{i,j=1 \\ i \neq j}}^{N_R} \mathcal{I}_{\hat{\mathcal{A}}}(i,j) \tag{23}$$

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

$$\mathcal{I}_{\hat{\mathcal{A}}}(i,j) = \begin{cases} 1 & \text{if } \hat{\mathcal{A}}_{ij} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

The query is is forwarded for TTL_P hops with an average of $\langle d \rangle$ routers on each hop and the return path has an equal number of routers. The expected time elapsed between the query generation and termination is then

$$E[T_{QS}] = [2TTL_P \langle d \rangle \sum_{i=1}^{N_R} (E[W_{Q_i}] + \tau_i)]/N_R \tag{25}$$

where $\sum_{i=1}^{N_R}(E[W_{Q_i}] + \tau_i)/N_R$ is the average queueing delay at a router where $E[W_{Q_i}]$ is the expected waiting time in the i^{th} router and is given in Eq. (36).

3.4 Router Network Model

For characterizing the delays at the core routers, we consider an interconnection network of N_R routers whose topology can be considered to be 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.

We model each router as a GI/G/1 queue to allow for arbitrary arrival patterns and packet size or service time distributions. The total arrival rate at the j^{th} router, λ_j, is a function of both the total external arrival rate 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, measured by the squared co-efficient of variance (SCV), of its external arrival process as well as that of the arrivals from its immediate neighbors. Existing results for single class open networks such as [37] may be used to characterize the delays in the routers. For completeness, the appendix lists the steps necessary to evaluate the average delay, $E[T_{N_R}]$, experienced by a packet as it traverses the network.

3.5 Expected File Replication Time

We conclude this section by presenting the final expression for the file replication time. The download time is determined by the time when the last packet of the file part being downloaded from 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 $E[T_{N_R}]$ seconds in the network. Adding the search time in the final expression, the overall waiting time, $E[T]$, is given by

$$E[T] = E[T_{WP}] + E[T_{N_R}] + E[T_{QS}] \tag{26}$$

where $E[T_{WP}]$ and $E[T_{N_R}]$ are given in Eqns. (13) and (39) respectively and $E[T_{QS}]$ given by Eq. (20) for a centralized architecture and by Eq. (25) otherwise. Note that the expression assumes that the query search process and the file transfer process are sequential and do not overlap in time. While there are P2P systems where an user may wait for all replies to the query to arrive before initiating downloads, in some systems, partial responses to a query may trigger downloads. In the latter case, the expression above serves as an upper bound on the expected overall waiting time. This bound will be very tight when the file being downloaded is large since the file transmission time will dwarf the query search time.

For peers with online and offline transitions, the overall file replication time is obtained by adding the expected peer latency $E[T_{DL}]$ from Eq. (19) with the query search time $E[T_{QS}]$ and network delay $E[T_{N_R}]$ so that

$$E[T] = E[T_{DL}] + E[T_{N_R}] + E[T_{QS}] \tag{27}$$

4 Multi-part Download

Due to replication of files, multiple peers may host copies of a file in the network. Splitting the replication into non-overlapping parts and downloading the respective part from each peer, instead of downloading the entire file from a single peer can reduce the download time. The question that naturally arises is: *How should the file be split among the peers so as to minimize the total download time?* We answer this question 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 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 the 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.,

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

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

$$(r_1 + r_2 + \cdots + r_n)t_a = F \tag{28}$$

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

$$(r_1 + r_2 + \cdots + r_n)t_b > F \tag{29}$$

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

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

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

$$\sum_{\substack{k=1 \\ k\neq i}}^{n} t_b r_k + t_i r_i = F \tag{30}$$

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

$$\begin{aligned} \sum_{\substack{k=1 \\ k\neq i}}^{n} t_b r_k + t_b r_i &> \sum_{\substack{k=1 \\ k\neq i}}^{n} t_b r_k + t_i r_i \\ \Rightarrow t_b \sum_{1}^{n} r_k &> F \ \ \text{(From Eq. (30))} \end{aligned}$$

Hence Eq. (29) holds. Clearly, the above proof holds if there exists more then one transfer time that differs from the maximum. The ratio of Eq. (29) and Eq. (28) gives

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

therefore we have $t_b > t_a$. □

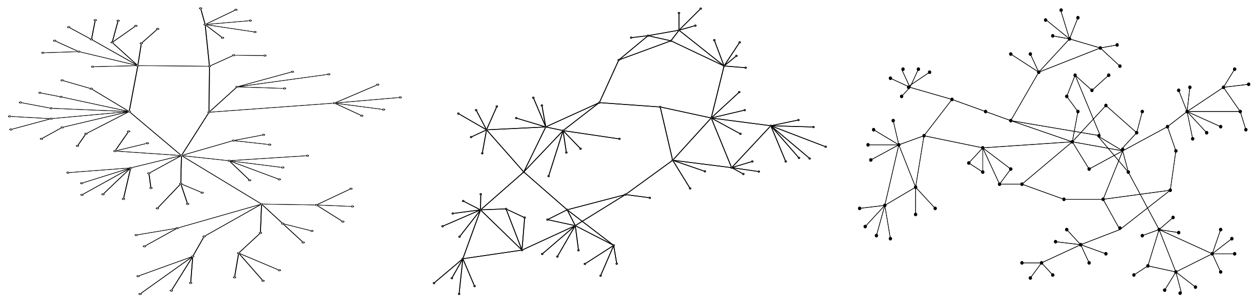
5 Simulation Results

In this section, we validate our analytical model by comparing the results with those obtained from simulations. To demonstrate 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. For each set of parameters, we repeated the simulation 200 times with various combinations of the source and destination peers and report the average of the 200 runs. The parameters that remain fixed across all simulations are: $\tau_i = 0.002$, and the peer service rate $\mu_p = 10$.

I. University Network: The simulated topology of the Columbia University network from [8] is shown in Fig. 5(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. 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 used for these simulations. In Fig. 6(a) we compare the simulation and the analysis results for the download time for a file size of 120 packets. The figure plots the download time as a function of the degree of replication of the requested file in the network. The simulation results match closely with the analysis and as expected, the download time decreases with increasing number of copies.

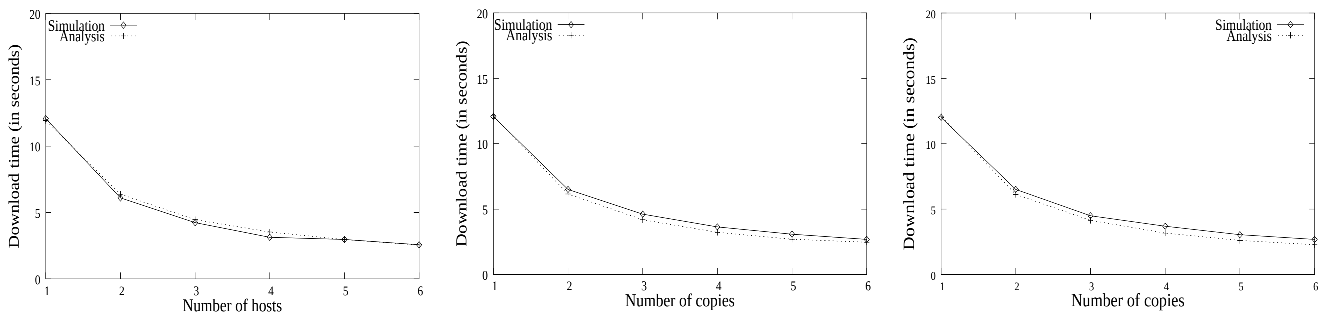
II. Power Law Topology: The power law topology generated using BRITE [6] was constructed as a two-tier hierarchical network with 25 routers and 50 peers. Peers are attached randomly to the network and the resulting topology is as shown in Fig. 5(b). A value of $c_{si} = 1.25$ was chosen for these simulations and the file size was again 120 packets. Fig. 6(b) compares the simulation and the analysis results for the download times for this topology. We again note the close match with the simulation results.

III. ISP Network: For an ISP level network we considered the topology of AT&T's backbone in the United States. The backbone layout obtained is from [4] and the network was extended by attaching a random number of Autonomous Systems (generated using BRITE [6])



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

Fig. 5. Simulation topologies.



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

Fig. 6. Download time vs. Number of copies for the three topologies.

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. 5(c). Again a value of $c_{si} = 1.25$ was used and the file size was 120 packets. Fig. 6(c) compares the simulation and the analysis results for the download time for this topology and we again note the close match with the simulation results.

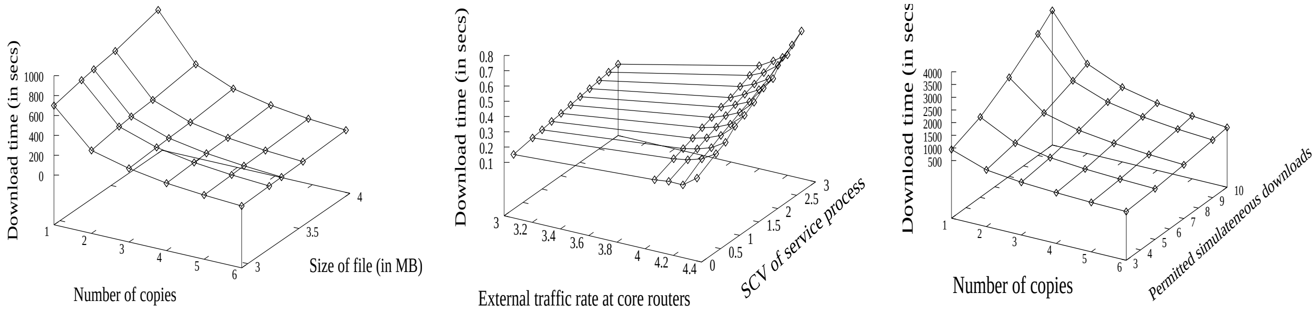
5.1 Sensitivity Analysis

We now evaluate the impact of the P2P system's features on the download times. Fig. 7(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. 7(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 across 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. 7(b) concurs with this observation. Finally, Fig. 7(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.

5.2 Impact of Online and Offline Times

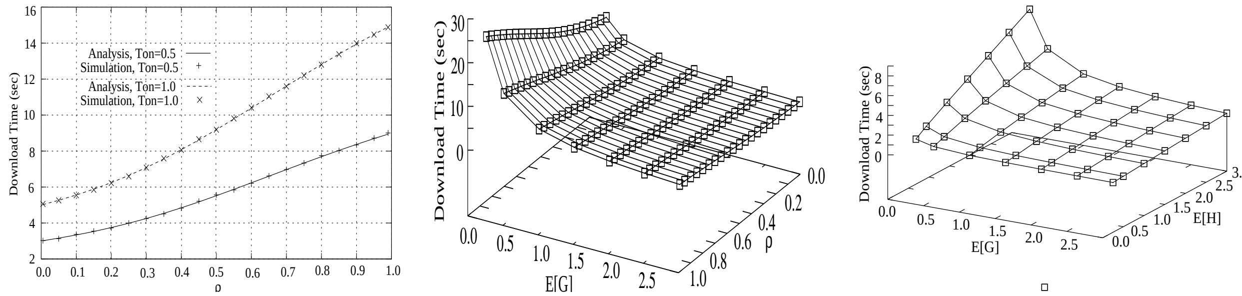
In Fig. 8(a) we compare the analytic and simulation results for the peer model with online and offline transitions developed in Section 3.2. All results in this section correspond to $C = 1$MBps and $B = 1$MB. The results are shown for the case where the online and offline times are drawn from a Pareto distribution with the parameters chosen such that the second moment was infinite, for both the online and offline times. The results are shown for the case with $m = 5$, $E[H] = 2.0$ and two values of $E[H]$: 0.5 and 1.0. We see that the analytic and simulation results match very closely. Similar results were also obtained for the case when the online and offline times are exponentially distributed.

In Fig. 8(b) we show the impact of expected duration of the online time on the download times. In these results, the expected offline time was kept fixed at 2.0. Thus, the curves for different values of $E[G]$ may also be interpreted to correspond to different values of γ. As expected, larger online times lead to smaller download times though the marginal decrease in the download times becomes smaller with increasing $E[G]$. Finally, in Fig. 8(c) we evaluate the impact of the expected durations of the online and offline times on the download times. The results were evaluated for $\rho = 0.75$, $N = 4$ and $m = 5$. We note that while the download time



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

Fig. 7. Parameter sensitivity analysis.



(a) Analysis vs simulation results (b) Online time vs ρ (c) Online vs offline times

Fig. 8. Impact of offline times on file download.

increases linearly with the offline times, they decrease non-linearly with increase in the online times.

5.3 Effect of File Allocation Strategies

In this section we use simulations to quantify the performance improvement obtained with the optimal download strategy of Section 4. In Fig. 9 we compare the replication time of the optimal rate proportional allocation mechanism with two other 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. 5(a) and 4 copies of the file were assumed to be available. We note that as expected, the proportional allocation leads to significantly lower delays.

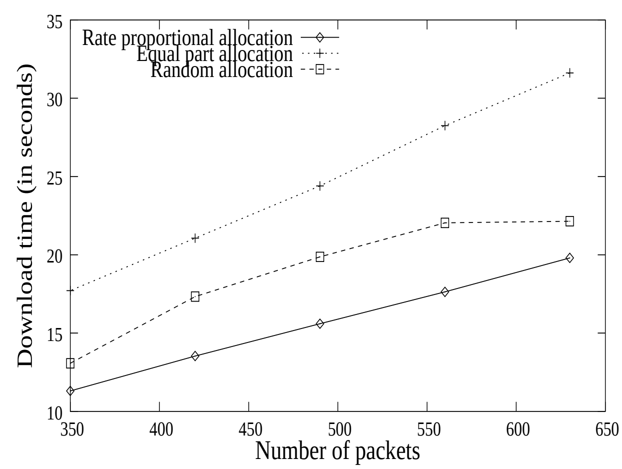


Fig. 9. Download times for different allocation strategies.

6 Conclusions

In this paper, we presented an analytic framework to evaluate the latencies associated with file replication in P2P systems. The main contribution of the paper is a queueing model to evaluate the file transfer delay at the peers. Our model accounts for the query search times and peer characteristics like the number of simultaneously allowed downloads at a peer, file popularity, number of copies of the file etc. The model has been validated using simulations. The paper also showed that a rate proportional allocation strategy is optimal for minimizing the file download time in scenarios with multi-part downloads.

Appendix

The appendix presents the methodology for obtaining the packet latency in the router network with each router modeled as a GI/G/1 queue. While the mean and SCV of the service time distribution at the routers are assumed to be known, parameters of the arrival process at each router are unknowns that first need to be determined.

6.0.1 Traffic Rate Equations

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

$$\lambda_j = \lambda_{0j} + \sum_{i=1}^{N_R} \lambda_i q_{ij} \quad j = 1, 2, \cdots, N_R. \tag{31}$$

In matrix notation, these equations can be written as

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

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

$$\alpha_i = \lambda_i \tau_i, \quad 1 \leq i \leq N_R$$

The rate of arrivals at 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 $\lambda_{ij} = \lambda_i q_{ij}$ and $p_{ij} = \lambda_{ij}/\lambda_j$. Equation (31) is essentially a rate balance equation since stable queues and infinite buffers imply that the incoming traffic rate equals the outgoing rate. The only unknowns in Eqn. (31) 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. (31), a system of N_R linear equations in N_R variables, will therefore yield the total arrival rate at each router.

6.0.2 Traffic Variability Equations

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 [37] and are enumerated below

$$c_{aj}^2 = a_j + \sum_{i=1}^{N_R} c_{ai}^2 b_{ij} \quad 1 \leq i \leq N_R, \tag{32}$$

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

$$a_j = 1 + w_j \left\{ (p_{0j}^2 c_{0j}^2 - 1) + \sum_{i=1}^{n} p_{ij} [(1 - q_{ij}) + (1 - \nu_{ij}) q_{ij} \rho_i^2 x_i] \right\} \tag{33}$$

and

$$b_{ij} = w_j p_{ij} q_{ij} [\nu_{ij} + (1 - \nu_{ij})(1 - \rho_i^2)], \tag{34}$$

where x_i, ν_{ij} and w_j are independent of the variability parameters c_{aj}^2 being calculated. In Eqns. (33) and (34) p_{0j} is the weight associated 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. x_i is given by

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

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

$$w_j = [1 + 4(1 - \rho_j)^2 (\nu_j - 1)]^{-1} \quad \text{with} \quad \nu_j = \left[\sum_{i=0}^{N_R} p_{ij}^2 \right]^{-1} \tag{35}$$

In our analysis 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 a combination of several arrival streams. Let κ_l and ζ_l^2 denote the rate and variability parameters for the l^{th} stream into router j. Thus, we have

$$\begin{aligned} \lambda_{0j} &= \sum_l \kappa_l \\ c_{0j}^2 &= w_j \sum_l \left(\frac{\kappa_l}{\sum_k \kappa_k} \right) \zeta_l^2 + 1 - w_j \end{aligned}$$

6.0.3 Network Latency

Using the results from [37], the expected waiting time at the i^{th} router, W_{Q_i}, can be shown to be

$$E[W_{Q_i}] = \tau_i \rho_i (c_{ai}^2 + c_{si}^2) g_i / 2(1 - \rho_i), \tag{36}$$

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

$$g_i(\rho_i, c_{ai}^2, c_{si}^2) = \begin{cases} exp\left[-\frac{2(1-\rho_i)}{3\rho_i} \frac{(1-c_{ai}^2)^2}{(c_{ai}^2 + c_{si}^2)} \right] & c_{ai}^2 < 1 \\ 1 & c_{ai}^2 \geq 1 \end{cases} \tag{37}$$

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: $E[N_{C_i}] = \rho_i + \lambda_i E[W_{Q_i}]$. 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 number of packets in the network N_C and therefore the sojourn time $E[T_{N_R}]$ or the router network delay per packet are given by

$$N_C = \sum_{i=1}^{N_R} N_{C_i} \tag{38}$$

$$E[T_{N_R}] = \frac{N_C}{\lambda_0} \tag{39}$$

REFERENCES

[1] E. Adar and B. Huberman, "Free Riding on Gnutella," *First Monday*, vol. 5, no. 10, October 2000.

[2] S. Alouf, A. Dandoush and P. Nain, " Performance Analysis of Peer-to-Peer Storage Systems," *Lecture Notes in Computer Science*, vol 4516, pp. 642-653, September 2007.

[3] B. Arnold, "Some Characterizations of the Exponential Distribution by Geometric Compounding," *SIAM Journal of Applied Mathematics*, vol. 24, no. 2, pp. 242-244, March 1973.

[4] ATT Backbone Topology, `http://www.jingleinc.com/html/backbone.html`.

[5] A. Bharambe, C. Herley and V. Padmanabhan, "Analyzing and improving bittorrent performance," *Proceedings of IEEE INFOCOM*, pp. 1-12, Barcelona, Spain, April 2006.

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

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

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

[9] F. Clevenot-Perronnin 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.

[10] F. Clevenot-Perronnin and P. Nain, "Stochastic fluid model for P2P caching evaluation," *Proceedings of Workshop on Web Content Caching and Distribution*, pp. 104-111, Sophia Antipolis, France, September 2005.

[11] 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.
[12] B. Fan, D.-M. Chiu and J. Lui, "Stochastic Differential Equation Approach to Model BitTorrent-like P2P Systems," *Proceedings of IEEE ICC,* Istanbul, Turkey, June 2006.
[13] M. Feldman, C. Papadimitriou, J. Chuang and I. Stoica, "Free-riding and whitewashing in peer-to-peer systems," *IEEE Journal on Selected Areas in Communications,* vol.24, no.5, pp. 1010-1019, May 2006.
[14] R. Gaeta, G. Balbo, S. Bruell, M. Gribaudo and M. Sereno, "A simple analytical framework to analyze search strategies in large-scale peer-to-peer networks," *Performance Evaluation,* vol. 62, no. 1-4, pp. 1-16, October 2005.
[15] R. Gaeta, M. Gribaudo, D. Manini and M. Sereno, "Analysis of resource transfers in peer-to-peer file sharing applications using fluid models," *Performance Evaluation,* vol. 63, no. 3, pp. 149-174, March 2006.
[16] 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.
[17] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding and X. Zhang, "A performance study of BitTorrent-like peer-to-peer systems," *IEEE Journal on Selected Areas in Communications,* vol.25, no.1, pp. 155-169, January 2007.
[18] V. Kalashnikov, *Geometric Sums: Bounds for Rare Events with Applications. Risk Analysis, Reliability, Queueing,* Kluwer Academic Publishers, London 1997.
[19] R. Kumar, Y. Liu and K. Ross, "Stochastic Fluid Theory for P2P Streaming Systems," *Proceedings of IEEE INFOCOM,* Anchorage, AK, April 2007.
[20] J. Liang, R. Kumar, Y. Xi and K. Ross, "Pollution in P2P File Sharing Systems," *Proceedings of IEEE INFOCOM,* Miami, FL, March 2005.
[21] W.-C. Liao, F. Papadopoulos and K. Psounis, "Performance analysis of BitTorrent-like systems with heterogeneous users," *Performance Evaluation,* vol. 64, no. 9-12, pp. 876-891, October 2007.
[22] 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.
[23] J. Mundinger, R. Weber and G. Weiss, "Analysis of Peer-to-Peer File Dissemination amongst Users of Different Upload Capacities," *Performance Evaluation Review,* 2006.
[24] M. Newman, S. Strogatz and D. Watts, "Random graphs with arbitrary degree distribution and their applications," *Physical Review E,* vol. 64, no. 026118, 2001.
[25] 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.
[26] S. Petrovic and P. Brown, "Fluid Model for eMule File Sharing System," *Proceedings of European Conference on Universal Multiservice Networks,* pp. 273-282, Toulouse, France, February 2007.
[27] J. Pouwelse, P. Garbacki, D. Epema and H. Sips, "The bittorrent p2p file-sharing system: Measurements and analysis," *Proceedings of the International Workshop on Peer-to-Peer Systems,* pp. 205-216, Ithaca, NY, February 2005.
[28] D. Qiu and R. Srikant, "Modeling and performance analysis of BitTorrent-like peer-to-peer networks," *Proc. of ACM SIGCOMM,* Portland, OR, August 2004.
[29] M. Ripeanu, A. Iamnitchi and I. Foster, "Mapping the Gnutella Network," *IEEE Internet Computing,* vol. 6, no. 1, pp. 50-57, January 2002.
[30] W. Saddi and F. Guillemin, " Measurement Based Modeling of eDonkey Peer-to-Peer File Sharing System," *Lecture Notes in Computer Science,* vol. 4516, pp. 974-985, September 2007.
[31] S. Saroiu, K. Gummadi, R. Dunn, S. Gribble and H. Levy, "A Measurement Study of Peer-to-Peer File Sharing Systems," *Proceedings of MCNC,* San Jose, CA, January 2002.
[32] S. Saroiu, K. Gummadi, R. Dunn, S. Gribble and H. Levy, "An Analysis of Internet Content Delivery Systems," *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation,,* Boston, MA, December 2002.
[33] S. Tewari and L. Kleinrock, "Analytical Model for BitTorrent-based Live Video Streaming," *Proceedings of IEEE NIME Workshop,* Las Vegas, NV, January 2007.
[34] D. Trang, R. Pereczes and S. Molnar, "Modeling the population of file-sharing peer-to-peer networks with branching processes," *Proceedings of IEEE ISCC,* pp. 809-815, Aveiro, Portugal, July 2007.
[35] Y. Tu, J. Sun, M. Hefeeda and S. Prabhakar, "An analytical study of peer-to-peer media streaming systems," *ACM Transactions on Multimedia Computing, Communications, and Applications,* vol. 1, no. 4, pp. 354-376, November 2005.
[36] D. Vassilakis and V. Vassalos, "Modelling Real P2P Networks: The Effect of Altruism," *Proceedings of IEEE Peer-to-Peer Computing,* pp.19-26, Galway, Ireland, September 2007.
[37] W. Whitt, "The Queuing Network Analyzer," *The Bell Systems Technical Journal*, 2779-2815, 1983.
[38] C. Wu, B. Li and S. Zhao, "Characterizing Peer-to-Peer Streaming Flows," *IEEE Journal on Selected Areas in Communications,* vol.25, no.9, pp.1612-1626, December 2007.
[39] X. Yang and G. de Veciana, "Service capacity in peer-to-peer networks," *Proc. of IEEE INFOCOM,* pp. 1-11, Hong Kong, China, March 2004.
[40] X. Yang and G. de Veciana, "Performance of peer-to-peer networks: Service capacity and role of resource sharing policies," *Performance Evaluation,* vol. 63, no. 3, pp. 175-194, March 2006.
[41] Z. Yao, D. Leonard, X. Wang and D. Loguinov, "Modeling Heterogeneous User Churn and Local Resilience of Unstructured P2P Networks," *Proceedings of IEEE ICNP,* Santa Barbara, CA, November 2006.
[42] J. Yu, M. Li and J. Wu, "Modeling Analysis and Improvement for Free-Riding on BitTorrent-like File Sharing Systems," *Proceedings of IEEE ICPPW,* p. 53, September 2007.
[43] S. Zhao, D. Stutzbach and R. Rejaie, "Characterizing Files in the Modern Gnutella Network: A Measurement Study," *Proceedings of SPIE/ACM Multimedia Computing and Networking,* San Jose, CA, January 2006.

PLACE PHOTO HERE

Krishna Ramachandran received his undergraduate degree in Mathematics and Computing from the Indian Institute of Technology, Kharagpur, India, in 2002 and a doctorate degree in Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY, USA in 2007. The focus of his dissertation was on mathematically quantifying P2P network performance by formulating relevant analytical frameworks. His broad inerests include application of stochastic calculus and optimization to analyze problems arising in the fields of networking, finance and the automotive industry.

PLACE PHOTO HERE

Biplab Sikdar (S'98, M'02) received the B. Tech degree in electronics and communication engineering from North Eastern Hill University, Shillong, India, the M. Tech degree in electrical engineering from Indian Institute of Technology, Kanpur and Ph.D in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, USA in 1996, 1998 and 2001, respectively. He is currently an Associate Professor in the Department of Electrical, Computer and Systems Engineering of Rensselaer Polytechnic Institute, Troy, NY, USA. His research interests include wireless MAC protocols, network routing and multicast protocols, network security and queueing theory. Dr. Sikdar is a member of IEEE, Eta Kappa Nu and Tau Beta Pi and is an Associate Editor of the IEEE Transactions on Communications.