Optimal Search Performance in Unstructured Peer-to-Peer Networks With Clustered Demands

Saurabh Tewari, Leonard Kleinrock
Computer Science Department
University of California at Los Angeles
Los Angeles, CA 90095, U.S.A.
{stewari,lk}@cs.ucla.edu

Abstract—**This paper derives the optimal search time and the optimal search cost that can be achieved in unstructured peer-to-peer networks when the demand pattern exhibits clustering (i.e. file popularities vary from region to region in the network). Previous work in this area had assumed a uniform distribution of file replicas throughout the network with an implicit or explicit assumption of uniform file popularity distribution whereas in reality, there is clear evidence of clustering in file popularity patterns. The potential performance benefit that the clustering in demand patterns affords is captured by our results. Interestingly, the performance gains are shown to be independent of whether the search network topology reflects the clustering in file popularity. We also provide the relation between the query-processing load and the number of replicas of each file for the clustered demands case showing that flooding searches may have lower query-processing load than random walk searches in the clustered demands case.**

Keywords- Flooding, Peer-to-Peer Networks, Random Walk, Optimal Search Time, Optimal Search Cost, Clustered Demands

I. INTRODUCTION

Peer-to-peer networks are loosely organized networks of autonomous entities (user nodes or "peers") which make their resources available to other peers. Since each new peer brings additional resources, these networks are fully scalable provided that the resources one offers can be found by the peers who need those resources. Thus, finding the desired resource is a critical issue in peer-to-peer networks. Keeping a centralized index of the resources each peer is offering is an approach that has scalability issues and a single point of failure. Alternatively, a direct approach for finding the desired resource is to have the peer wanting a resource to query other nodes to find a node that has that resource. Since a node cannot realistically keep the addresses of all other peers, an overlay network is constructed where each node keeps addresses of a few other peers (called its *neighbors*) through whom it reaches the rest of the peers. Peer-to-peer networks following this approach are referred to as *unstructured* peer-to-peer networks to distinguish them from structured networks (e.g. [6]) which map each unique resource to a particular node in the network, an approach that can be more efficient but whose lack of flexibility introduces other issues [5]. In this paper we focus on unstructured peer-to-peer networks and address two major concerns in these networks: the time to find a peer who is offering a particular resource (the *search time*), and the amount of additional traffic introduced in the network in the process of locating the peer that is offering that resource (the *search cost*). The reference example is of peer-to-peer file sharing networks and we refer to resources as files throughout the rest of the paper.

As in our earlier related works [7, 8], we approximate the search time for a file in the network by the average number of hops it takes for a query to reach a node that has that file, and use *average search time*, i.e., the average time it takes to find a peer that is sharing the desired file, as our first metric for search performance. Our second metric is the search cost. Since a search for a file is done via peers sending query messages to other peers, the number of query messages each peer processes equals the additional traffic introduced in the network by a query. Therefore, we approximate the search cost by the *query-processing load*, i.e., the average number of nodes that are queried per file request. One expects that if many peers are sharing a file, in any reasonable search method, the search time and the search cost for the file will be smaller than if very few peers were sharing that file. In the extreme case, if all nodes could store all files, no search would be required. Since each peer has finite storage space, a system designer seeks to get the optimum search performance possible given the per-node storage constraint. The optimal average search time, the optimal query-processing load and the *file replica distribution* (number of replicas of each file as a function of that file's popularity) at the respective optima have been derived in [7] under the assumption of a uniform distribution of the file replicas. However, measurements on deployed peer-to-peer file sharing networks show a significant amount of clustering in interests [4], i.e., the popularity of a set of files in (geographical) regions differs from region to region. Further, more replicas of a file are found in those regions where that file is more popular.

The main contributions of this paper, given in Section 5, are the aforementioned optimal search performance expressions for the clustered demands case using the network model in [8] that allows for incorporating clustering in demand and file replica distribution. Section 3 gives the network model and the search time results for the model from [8]. We derive the query-processing load as a function of the file replica distribution for the network model in [8] in Section 4 for use in our optimization. Related work, including the results in [7], is discussed in Section 2. Our conclusions are given in Section 5.

II. BACKGROUND AND RELATED WORK

Flooding and *random walking* are the two main alternatives in how the search is conducted over the search network when no information is available about which nodes may have the file. In flooding, the node that wants the file sends a query to all its neighbors and they, in turn, forward the query to all their neighbors (except the one which sent the query) until a copy of the file is found. In random walking, the query is sent to one randomly selected neighbor and if that neighbor does not have the file, it forwards the query to one of its neighbors (selected randomly) other than the neighbor that sent it the query.

When nodes are similar in capacities and file interests (i.e. when files and file popularities are uniformly distributed), the Erdos-Renyi random graph [1] is a good topology model[1] for

TABLE I. NOTATION USED

M	Number of nodes
L	Number of clusters
N	Number of unique files
K	Per-node storage size (in number of files)
d	Average degree of the search overlay topology
q	Probability of any given pair of inter-cluster nodes having a direct link
n_i	Number of replicas of file i in the entire network
n_{ia}	Number of replicas of file i in the “high-density” cluster
n_{ib}	Number of replicas of file i in a “low-density” cluster
λ_i	Request rate of file i per node (averaged over the network)
λ_{ia}	Request rate of file i per node in the “high-density” cluster
λ_{ib}	Request rate of file i per node in a “low-density” cluster
λ	$= \sum_{i=1}^{N} \lambda_i$
τ_{ix}	Average search time for file i with search method x [a]
Q_{ix}	Query-processing load for file i with search method x [a]
τ_{ixa}	Average search time for file i from the high-density cluster with search method x [a]
Q_{ixa}	Query-processing load for file i from the high-density cluster with search method x [a]
τ_{ixb}	Average search time for file i from a low-density cluster with search method x [a]
Q_{ixb}	Query-processing load for file i from a low-density cluster with search method x [a]
τ_x^{opt}	Optimal average search time with search method x [a]
Q_x^{opt}	Optimal query-processing load with search method x [a]
$Q_x^{\tau opt}$	Query-processing load with the replica distribution that minimizes the average search time with search method x [a]

[a] For flooding search: $x=F$, For random walk search: $x=R$ e.g. τ_{iFb}=Average search time for file i from a low-density cluster with flooding search

TABLE II. RESULTS FOR UNIFORM DISTRIBUTION OF REPLICAS ([7])

Replica Distribution	Equation	
Valid for arbitrary replica distributions	$\tau_{iF}(n_i) = \log_d(M/n_i)$	(1)
	$Q_{iF}(n_i) = Q_{iR}(n_i) = \tau_{iR}(n_i) = M/n_i$	(2)
$n_i \propto \lambda_i$	$\tau_F^{opt} = -\sum_{i=1}^{N} \frac{\lambda_i}{\lambda} \log_d \frac{\lambda_i}{\lambda} - \log_d K$	(3)
	$Q_F^{\tau opt} = \frac{N}{K}$	(4)
$n_i \propto \sqrt{\lambda_i}$	$Q_F^{opt} = Q_R^{opt} = \tau_R^{opt} = \frac{(\sum_{i=1}^{N}\sqrt{\lambda_i})^2}{\lambda K}$	(5)

the overlay search network. The optimal search performance under the constraint of finite per-node storage is covered well by [3, 7] with the assumption of uniform distribution of file replicas. We summarize these results in Table 2. Table 1 gives the notation used in the paper. In addition to the results in Table 2, [7] compare random walking and controlled flooding at their respective optimal replica distributions and show the benefits that controlled flooding provides over random walking. In this paper, we seek to obtain results analogous to those in Table 2 when the file replica distribution and the demand patterns are not uniform.

Since each link is equiprobable in an Erdos-Renyi random graph, it is not suited for modeling clustering in file interests. Reference [8] provides a model of peer-to-peer networks that allows for incorporating varying degrees of clustering in demand and file replica distribution and derives the search times for these networks. Our work in this paper uses the model and the search time results from [8]. We list the relevant material from [8] in the next section.

III. A MODEL FOR CLUSTERED DEMANDS (FROM [8])

Let us assume that our peer-to-peer network has M nodes and that these M nodes are clustered in, say, L clusters. For ease of discussion, we make the following assumptions. Each cluster is of the same size (thus, each cluster has M/L nodes). There are only two levels of popularity of each file and there is only one cluster in which a file is more popular. Thus, for all files $i = 1$ to N, file i has request rate λ_{ia} per node in one cluster and λ_{ib} per node in each of the remaining $L-1$ clusters where $\lambda_{ia} > \lambda_{ib}$ and $M\lambda_i = \frac{M}{L}\lambda_{ia} + (L-1)\frac{M}{L}\lambda_{ib}$. where λ_i is the average node request rate for file i across the entire network. Let us further assume that the n_i replicas of file i are split as n_{ia} replicas in the cluster where the file is more popular and n_{ib} replicas in each of the remaining clusters where $n_{ia}>n_{ib}$, $n_i= n_{ia}+(L-1)n_{ib}$ and $n_{ia}<M/L$. One may then say that the cluster where file i is more popular has a *higher density* of file i replicas whereas a cluster where the file is not as popular has a *lower density*. Since clustering has already been accounted for, we assume that within each cluster the files are uniformly distributed over all the nodes in that cluster.

One possible model for the search network is to assume that the clusters are totally disconnected (i.e. there are no inter-cluster links) and within each cluster, the network follows the Erdos-Renyi random graph topology. For this model of

[1] When node capacities are very skewed, a power-law random graph is a topology choice which distributes the query-processing load unevenly among the peers but yields faster search methods (e.g. [2]).

TABLE III. SEARCH PERFORMANCE WITH DISCONNECTED CLUSTERS

Derived from	Equation	
(1)	$\tau_{iFa}(n_{ia}, n_{ib}) = \log_d(M/n_{ia}L)$	(6)
	$\tau_{iFb}(n_{ia}, n_{ib}) = \log_d(M/n_{ib}L)$	(7)
(2)	$Q_{iFa}(n_{ia}, n_{ib}) = Q_{iRa}(n_{ia}, n_{ib}) = \tau_{iRa}(n_{ia}, n_{ib}) = M/n_{ia}L$	(8)
	$Q_{iFb}(n_{ia}, n_{ib}) = Q_{iRb}(n_{ia}, n_{ib}) = \tau_{iRb}(n_{ia}, n_{ib}) = M/n_{ib}L$	(9)

clustering, the search time and the query-processing load expressions can be obtained from the analogous expressions for the uniform distribution case in Table 2 with (1) and (2) yielding (6), (7) and (8), (9) respectively as shown in Table 3.

While assuming disconnected clusters makes for an easy first-order analysis, actual peer-to-peer networks do not typically have such fully disconnected clusters. There is evidence of strong clustering but intercluster links do exist in real networks so neither an Erdos-Renyi random graph over the entire network nor the fully disconnected clusters model is an appropriate topology. A topology model that gives us a continuum of topologies with the Erdos-Renyi random graph at one extreme and the fully disconnected clusters at the other extreme is the following random graph variant. Consider a network in which the probability of including an intra-cluster link is p and the probability of including an inter-cluster link is q and the average per-node degree is d as before, i.e., assuming L clusters of equal sizes, the nodes are partitioned into L clusters and the probability that any given pair of intra-cluster nodes is connected is p and the probability that any given pair of inter- cluster nodes is connected is q. Thus, each node has an average of $(M/L)p$ links to nodes within its cluster and $(M-M/L)q$ links to nodes outside its cluster. Hence, the average degree $d = (M–M/L)q + (M/L)p$ and if one were to hold the average degree constant, defining one of p or q defines the other. Varying q provides a continuum of topologies from the completely disjoint clusters ($q=0$) to the Erdos-Renyi random graph ($p=q$). A flooding search in these topologies expands to d other nodes (in the higher-density or a lower-density cluster) in the next hop independent of whether the search process is at a node in the higher-density cluster or a lower-density cluster. Thus, the average number of nodes queried per search expands exponentially and the d^τ expression for the number of nodes queried given the average search distance of τ [7] still holds.

Reference [8] gives analytical bounds on the search time for

TABLE IV. SEARCH PERFORMANCE IN THE GENERAL CASE

Equation in [8]	Equation	
(11)	$\tau_{iFa}(n_{ia}, n_{ib}) \sim -\log_d[\frac{n_{ia}L}{M} - \frac{q(L-1)(n_{ia}-n_{ib})}{d}]$	(10)
(12)	$\tau_{iFb}(n_{ia}, n_{ib}) \sim -\log_d[\frac{n_{ib}L}{M} + \frac{q(n_{ia}-n_{ib})}{d}]$	(11)
(7)	$\tau_{iRa}(n_{ia}, n_{ib}) = Q_{iRa}(n_{ia}, n_{ib}) = [\frac{n_{ia}L}{M} - \frac{q(L-1)(n_{ia}-n_{ib})}{(n_{ib}L/M)(d-Mq)+Mq}]^{-1}$	(12)
(8)	$\tau_{iRb}(n_{ia}, n_{ib}) = Q_{iRb}(n_{ia}, n_{ib}) = [\frac{n_{ib}L}{M} + \frac{q(n_{ia}-n_{ib})}{(n_{ia}L/M(d-Mq)+Mq}]^{-1}$	(13)

flooding search in the aforementioned network model and indicates (via simulations and analysis) that the search time can be approximated well by the lower bound shown in (10) when searching from the high density cluster and by the upper bound shown in (11) when searching from the low-density clusters. Since the query-processing load is same as the average search time for random walking, we get (12) and (13) directly from [8]. One can verify that for disconnected clusters i.e. q=0, (10)-(13) revert to (6)-(9) and for a uniform distribution of file replicas i.e. $n_{ia}=n_{ib}=n_i/L$, (10)-(13) revert to (1) and (2). The query-processing loads as a function of the number of file replicas of each file are derived in the next section.

IV. QUERY-PROCESSING LOAD WITH CLUSTERED DEMANDS

As discussed earlier, for the network model described in Section 3, the query-processing load in the network can be estimated by d^τ when the average search distance is τ. Hence:

Theorem 1: The query-processing load for a flooding search in the clustered peer-to-peer network defined in Section 3 is

$$Q_{iFa}(n_{ia}, n_{ib}) \sim [\frac{n_{ia}L}{M} - \frac{q(L-1)(n_{ia}-n_{ib})}{d}]^{-1} \quad (14)$$

for searches initiated in the high-density cluster, and is

$$Q_{iFb}(n_{ia}, n_{ib}) \sim [\frac{n_{ib}L}{M} + \frac{q(n_{ia}-n_{ib})}{d}]^{-1} \quad (15)$$

for searches initiated in a low-density cluster. ■

Notice that unlike the uniform distribution case, the query-processing load for the flooding search and the random walk search are different now. In fact, we can show that:

Corollary 1: For the clustered peer-to-peer network defined in Section 3, (a) From the high-density cluster, a flooding search has a lower query-processing load than a random walk search whereas (b) From a low-density cluster, a flooding search has a higher query-processing load than a random walk search i.e. for searches for file i,

$$Q_{iRa}(n_{ia}, n_{ib}) > Q_{iFa}(n_{ia}, n_{ib})$$
$$Q_{iRb}(n_{ia}, n_{ib}) < Q_{iFb}(n_{ia}, n_{ib})$$

Proof:
Let $a = n_{ia}L/M$, $b = n_{ib}L/M$, $c = q(n_{ia}–n_{ib})/d$, $e = Mq/d$. Then $Q_{iFa} = [a–c(L–1)]^{-1}$, $Q_{iRa} = [a–c(L–1)/[b(1–e)+e]]^{-1}$ and $Q_{iFb} = [b+c]^{-1}$, $Q_{iRb} = [b+c(L–1)/[a(1–e)+e]]^{-1}$. Since a < 1, b < 1 and 1–e > 0, we get $b(1–e)+e < 1$ and $a(1–e)+e < 1$. $b(1–e)+e < 1 \Rightarrow c(L–1)/[b(1–e)+e] > c(L–1) \Rightarrow a–c(L–1)/[b(1–e)+e] < a–c(L–1) \Rightarrow Q_{iRa}(n_{ia}, n_{ib}) > Q_{iFa}(n_{ia}, n_{ib})$. Similarly, $a(1–e)+e < 1 \Rightarrow c/[a(1–e)+e] > c \Rightarrow b+c/[a(1–e)+e] > b+c \Rightarrow Q_{iRb}(n_{ia}, n_{ib}) < Q_{iFb}(n_{ia}, n_{ib})$. ■

We observe that, while the request rate in the high-density cluster, λ_{ia}, should be larger than the request rate in a low-density cluster, λ_{ib}, it is not clear whether, for arbitrary replica distributions, the lower query-processing load offered by a flooding search in the high-density cluster offsets the higher query-processing load incurred by the flooding search in the low-density cluster after weighting the query-processing costs by λ_{ia} and λ_{ib} respectively with $\lambda_{ia} > \lambda_{ib}$.

Corollary 1 also suggests that, for arbitrary replica distributions, it may be better for query processing to use flooding searches in the high-density cluster and random walk searches in the low-density clusters (at the cost of significantly larger search times for searches from the low-density clusters) if it were known that the item being searched has "low-density" in the local cluster (a simple approach may be to use flooding for a short hop-limit which would allow flooding searches from the high-density cluster to complete and assume the incomplete searches to be searches in a low-density cluster).

V. Search Performance Optimization

The optimal average search time τ_x^{opt} and the optimal query-processing load Q_x^{opt} over all file requests in the entire network are:

$$\tau_x^{opt} = \sum_{i=1}^{N}\left[\frac{1}{L}\frac{\lambda_{ia}}{\lambda}\tau_{ixa} + (1-\frac{1}{L})\frac{\lambda_{ib}}{\lambda}\tau_{ixb}\right] \qquad (16)$$

$$Q_x^{opt} = \sum_{i=1}^{N}\left[\frac{1}{L}\frac{\lambda_{ia}}{\lambda}Q_{ixa} + (1-\frac{1}{L})\frac{\lambda_{ib}}{\lambda}Q_{ixb}\right] \qquad (17)$$

where x=F for flooding search and =R for random walk search.

For the case of disconnected clusters, where (6)-(9) are the relevant expressions, the same steps as in [7] can be followed for the search performance optimization to yield the optimal results summarized in Table 5.

The optimization procedure in the general clustering case is the same as in [7] but the solution is harder to obtain. We summarize the optimization results for flooding search in Theorems 3 and 4. The results for random walk search optimization are not provided.

A. Average Search Time Optimization for Flooding Search

The Lagrangian for the average search time optimization is

$$H = -\sum_{i=1}^{N}\left[\frac{1}{L}\frac{\lambda_{ia}}{\lambda}\log_d(\frac{n_{ia}L}{M} - \frac{q(L-1)(n_{ia}-n_{ib})}{d}) + (1-\frac{1}{L})\frac{\lambda_{ib}}{\lambda}\log_d(\frac{n_{ib}L}{M} + \frac{q(n_{ia}-n_{ib})}{d})\right] + \gamma(\sum_{i=1}^{N}[n_{ia}+(L-1)n_{ib}] - KM)$$

TABLE V. Optimal Search Performance: Disconnected Clusters

Optimal Replica Distribution	Equation[a]
$\{n_{ia} \propto \lambda_{ia}, n_{ib} \propto \lambda_{ib}\}$	$\tau_F^{opt} = -\sum_{i=1}^{N}\frac{1}{L}\frac{\lambda_{ia}}{\lambda}\log_d\frac{\lambda_{ia}}{\lambda} - \sum_{i=1}^{N}\left(1-\frac{1}{L}\right)\frac{\lambda_{ib}}{\lambda}\log_d\frac{\lambda_{ib}}{\lambda} - \log_d K$ (18)
	$Q_F^{\tau opt} = \frac{N}{K}$ (19)
$\{n_{ia} \propto \sqrt{\lambda_{ia}}, n_{ib} \propto \sqrt{\lambda_{ib}}\}$	$Q_F^{opt} = Q_R^{opt} = \tau_R^{opt} = \frac{1}{\lambda K}\left[\sum_{i=1}^{N}\left(\frac{1}{L}\sqrt{\lambda_{ia}} + \left(1-\frac{1}{L}\right)\sqrt{\lambda_{ib}}\right)\right]^2$ (20)

[a] As in [7], we still have the constraints that $n_{ia} \geq 1, n_{ib} \geq 1, n_{ia} \leq \frac{M}{L}, n_{ib} \leq \frac{M}{L}$ and, hence, (18), (19) hold if $\frac{L}{KM} \leq \frac{\lambda_{ia}}{\lambda} \leq \frac{L}{K}$ and $\frac{L}{KM} \leq \frac{\lambda_{ib}}{\lambda} \leq \frac{L}{K}$ $\forall i$ (see reference 4 in [7] for conditions under which (20) holds).

Differentiating w.r.t n_{ia} and n_{ib} respectively, we obtain:

$$-\frac{1}{\lambda \ln d}\left[\frac{\frac{\lambda_{ia}}{L}[\frac{L}{M} - \frac{q(L-1)}{d}]}{\frac{n_{ia}L}{M} - \frac{q(L-1)(n_{ia}-n_{ib})}{d}} + \frac{\lambda_{ib}(1-\frac{1}{L})\frac{q}{d}}{\frac{n_{ib}L}{M} + \frac{q(n_{ia}-n_{ib})}{d}}\right] + \gamma = 0$$

$$-\frac{1}{\lambda \ln d}\left[\frac{\frac{\lambda_{ia}}{L}\frac{q(L-1)}{d}}{\frac{n_{ia}L}{M} - \frac{q(L-1)(n_{ia}-n_{ib})}{d}} + \frac{\lambda_{ib}(1-\frac{1}{L})(\frac{L}{M} - \frac{q}{d})}{\frac{n_{ib}L}{M} + \frac{q(n_{ia}-n_{ib})}{d}}\right] + \gamma(L-1) = 0$$

These equations are satisfied[2] by

$$n_{ia}L/M - q(L-1)(n_{ia}-n_{ib})/d = k\lambda_{ia} \qquad (21)$$

$$n_{ib}L/M + q(n_{ia}-n_{ib})/d = k\lambda_{ib} \qquad (22)$$

where k is a constant whose value is to be determined. Thus, at the optimal replica distribution, $\lambda_i = \lambda_{ia}/L + (1-1/L)\lambda_{ib} = [n_{ia}/M - q(1-1/L)(n_{ia}-n_{ib})/d + (L-1)n_{ib}/M + q(1-1/L)(n_{ia}-n_{ib})/d]/k = [(L-1)n_{ib}+n_{ia}]/Mk = n_i/Mk$, i.e. $n_i \propto \lambda_i$. $KM = \sum_{i=1}^{N} n_i = Mk\sum_{i=1}^{N}\lambda_i = M\lambda k \Rightarrow k = K/\lambda$. The following theorem summarizes these results.

Theorem 3: The average search time for a flooding search in the clustered peer-to-peer network defined in Section 3 is minimized when

$$n_{ia} = [KM(d\lambda_{ia} - Mq\lambda_i)]/[\lambda L(d - Mq)] \qquad (23)$$

$$n_{ib} = [KM(d\lambda_{ib} - Mq\lambda_i)]/[\lambda L(d - Mq)] \qquad (24)$$

if $L/KM \leq \lambda_{ia}/\lambda \leq L/K$ and $L/KM \leq \lambda_{ib}/\lambda \leq L/K$ $\forall i$, and at the replica distribution defined by these equations,

$$n_i = K\lambda_i/\lambda$$

$$\tau_F^{opt} = -\sum_{i=1}^{N}[\frac{1}{L}\frac{\lambda_{ia}}{\lambda}\log_d(\frac{\lambda_{ia}}{\lambda}) + (1-\frac{1}{L})\frac{\lambda_{ib}}{\lambda}\log_d(\frac{\lambda_{ib}}{\lambda})] - \log_d K$$

$$Q_F^{\tau opt} = N/K$$

i.e. the optimal average search time is independent of q (the level of clustering in search network topology) while the query-processing load when the average search time is minimized is independent of the skew in file popularity and the level of clustering in both the search network and the file popularity. ■

B. Query-Processing Load Optimization for Flooding Search

The Lagrangian for the query-processing load optimization is $\mathrm{H} = \sum_{i=1}^{N}\left[\frac{1}{L}\frac{\lambda_{ia}}{\lambda}[\frac{n_{ia}L}{M} - \frac{q(L-1)(n_{ia}-n_{ib})}{d}]^{-1} + (1-\frac{1}{L})\frac{\lambda_{ib}}{\lambda}[\frac{n_{ib}L}{M} + \frac{q(n_{ia}-n_{ib})}{d}]^{-1}\right] + \gamma(\sum_{i=1}^{N}[n_{ia}+(L-1)n_{ib}] - KM)$

---

2 $\frac{1}{k}[\frac{1}{M} - \frac{q}{d}(1-\frac{1}{L})] + \frac{q}{dk}(1-\frac{1}{L}) = \gamma\lambda \ln d$

$\frac{q}{dk}(1-\frac{1}{L}) + \frac{1}{k}(1-\frac{1}{L})[\frac{L}{M} - \frac{q}{d}] = \gamma(L-1)\lambda \ln d$

Differentiating w.r.t n_{ia} and n_{ib} respectively, we obtain:

$$\frac{-\frac{\lambda_{ia}}{\lambda L}[\frac{L}{M}-\frac{q(L-1)}{d}]}{[\frac{n_{ia}L}{M}-\frac{q(L-1)(n_{ia}-n_{ib})}{d}]^2}+\frac{-\frac{\lambda_{ib}}{\lambda}(1-\frac{1}{L})\frac{q}{d}}{[\frac{n_{ib}L}{M}+\frac{q(n_{ia}-n_{ib})}{d}]^2}+\gamma=0$$

$$\frac{-\frac{\lambda_{ia}}{\lambda L}\frac{q(L-1)}{d}}{[\frac{n_{ia}L}{M}-\frac{q(L-1)(n_{ia}-n_{ib})}{d}]^2}+\frac{-\frac{\lambda_{ib}}{\lambda}(1-\frac{1}{L})(\frac{L}{M}-\frac{q}{d})}{[\frac{n_{ib}L}{M}+\frac{q(n_{ia}-n_{ib})}{d}]^2}+\gamma(L-1)=0$$

Comparing these equations to those yielding (21) and (22), we can see that these equations will be satisfied by

$$n_{ia}L/M-q(L-1)(n_{ia}-n_{ib})/d=k'\sqrt{\lambda_{ia}} \tag{25}$$

$$n_{ib}L/M+q(n_{ia}-n_{ib})/d=k'\sqrt{\lambda_{ib}} \tag{26}$$

where k' is a constant whose value is to be determined. Thus, at the optimal replica distribution, $n_i = n_{ia}+(L-1)n_{ib} = (M/L)(n_{ia}L/M-q(L-1)(n_{ia}-n_{ib})/d + (L-1)[n_{ib}L/M+q(L-1)(n_{ia}-n_{ib})/d]) = (k'M/L)[\sqrt{\lambda_{ia}}+(L-1)\sqrt{\lambda_{ib}}]$. Using $n_i= (k'M/L)(\sqrt{\lambda_{ia}}+(L-1)\sqrt{\lambda_{ib}})$ in $\sum_{i=1}^{N} n_i = KM$, we get $k' = K\Big/\sum_{i=1}^{N}[\frac{1}{L}\sqrt{\lambda_{ia}}+(1-\frac{1}{L})\sqrt{\lambda_{ib}}]$ yielding the following theorem.

Theorem 4: The query-processing load for a flooding search in the clustered peer-to-peer network defined in Section 3 is minimized when

$$n_{ia}=\frac{(dL-Mq)\sqrt{\lambda_{ia}}-Mq(L-1)\sqrt{\lambda_{ib}}}{L(d-Mq)\sum_{i=1}^{N}[\sqrt{\lambda_{ia}}+(L-1)\sqrt{\lambda_{ib}}]}KM \tag{27}$$

$$n_{ib}=\frac{[dL-Mq(L-1)]\sqrt{\lambda_{ib}}-Mq\sqrt{\lambda_{ia}}}{L(d-Mq)\sum_{i=1}^{N}[\sqrt{\lambda_{ia}}+(L-1)\sqrt{\lambda_{ib}}]}KM \tag{28}$$

assuming λ_{ia} and λ_{ib} are such that $1 \le n_{ia} \le M/L$, $1 \le n_{ib} \le M/L$ $\forall i$ in these equations, and at this replica distribution

$$Q_F^{opt} = \frac{1}{\lambda K}\left(\sum_{i=1}^{N}[\frac{1}{L}\sqrt{\lambda_{ia}}+(1-\frac{1}{L})\sqrt{\lambda_{ib}}]\right)^2$$

i.e. the optimal query-processing load is independent of q (the level of clustering in search network topology). ■

As noted in the theorems, the optimal search performance is independent of the level of clustering in search network topology. Thus, the results for the optimal search performance provided in Table 5 for disconnected clusters hold for the general clustered demands network model *except* for the file replica distributions needed for the optimal search performance which are now defined by (23) and (24) for the optimal average search time i.e. (18) and (19) and by (27) and (28) for the optimal query-processing load i.e. (20).

C. Interpretation of Optimal Search Performance Results

We find it very interesting that the optimal search performance does not depend on the underlying search network topology. In fact, it is rather intriguing that the optimal average search time expression for the uniform distribution case seems to be related to the entropy in the file request probabilities $\{\lambda_i/\lambda\}$, and that the only change in the optimal average search time expression in the case of clustered demands is that the entropy now includes the spatial distribution of file requests (λ_{ia}/L is the probability that file i is requested by a node in the high-density cluster and $(1-1/L)\lambda_{ib}$ is the probability that file i is requested by a node in a low-density cluster). Similarly, the optimal query-processing load also changed only in that the expression includes the spatial distribution of file requests in clustered demands case.

Another interesting observation in comparing (21), (22) and (25), (26) to [3, 7] is that while the expressions for the optimal replica distribution are complex in the case of clustered demands, we still have the invariant from the uniform distribution case that the probability of finding the file over a random outgoing link from a node is proportional to the file request rate at that node when optimizing the average search time and is proportional to the square-root of the file request rate at that node when optimizing the query-processing load. We summarize this result in the following theorem.

Theorem 5: For flooding search in the clustered peer-to-peer network defined in Section 3, we have the following invariants independent of the level of clustering in demands and the level of clustering in the search network topology

1) The average search time τ is minimized when $p_{ij} \propto \lambda_{ij}$, and

2) The query-processing load Q is minimized when $p_{ij} \propto \sqrt{\lambda_{ij}}$

where p_{ij} is the probability of finding file i over a random outgoing link from a node in cluster j and λ_{ij} is the per-node request rate for file i in cluster j. ■

To evaluate the potential benefits of clustering in demands over the uniform distribution case, we plot the interesting part[3] of (18) and (20) in Figs. 1 and 2 respectively for a peer-to-peer network of 10 equal-sized clusters and 100 files with zipf-distributed request rates. Perfect clustering is defined as the case when the entire demand for a file is from its own cluster i.e. λ_{ib}=0 and $\lambda_{ia}= L\lambda_i$. Figs. 1 and 2 clearly demonstrate the potential advantage of clustering. The advantage in search performance afforded by perfect clustering can be summarized the following theorem.

Theorem 6: When the entire demand for a file is from its own cluster i.e. λ_{ib}=0 and $\lambda_{ia}= L\lambda_i$, the optimal average search time τ_F^{opt} decreases by $\log_d L$ and the optimal query-processing load Q_F^{opt} decreases by a factor of L, the number of clusters.

Proof:

Substituting λ_{ib}=0 and $\lambda_{ia}=L\lambda_i$ in (18) and (20), we get $\tau_F^{opt} = -\sum_{i=1}^{N}\frac{\lambda_i}{\lambda}\log_d(\frac{\lambda_i}{\lambda})-\log_d K-\log_d L$ and $Q_F^{opt}=\frac{1}{L}\left(\sum_{i=1}^{N}\sqrt{\lambda_i}\right)^2\Big/\lambda K$.

The theorem follows on comparing these to (3), (5). ■

[3] To eliminate the dependence on d and K, in Figs. 1 and 2, we plot $\tau_{opt}' = -\sum_{i=1}^{N}[\frac{1}{L}\frac{\lambda_{ia}}{\lambda}\ln(\frac{\lambda_{ia}}{\lambda})+(1-\frac{1}{L})\frac{\lambda_{ib}}{\lambda}\ln(\frac{\lambda_{ib}}{\lambda})]$ and $Q_{opt}' = \left(\sum_{i=1}^{N}[\frac{1}{L}\sqrt{\frac{\lambda_{ia}}{\lambda}}+(1-\frac{1}{L})\sqrt{\frac{\lambda_{ib}}{\lambda}}]\right)^2$ instead of (18) and (20) respectively.

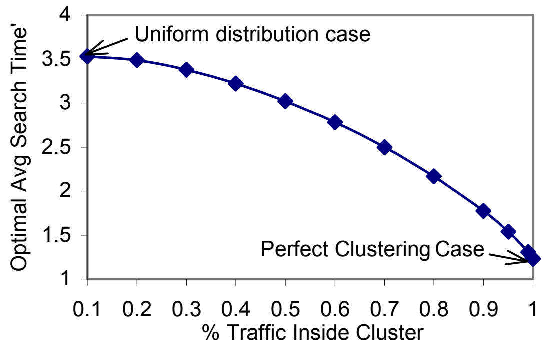

Figure 1. Benefit of Clustered Demands: Optimal Search Time

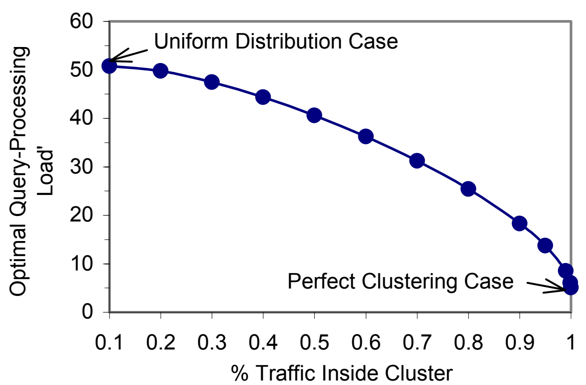

Figure 2. Benefit of Clustered Demands: Optimal Query-Processing Load

Finally, we note that the penalty over the optimal query-processing load incurred upon optimizing the average search time increases in the case of clustered demands. For example, for the peer-to-peer network shown in Fig. 2, $Q_F^{\tau opt}$ = 100 independent of the fraction of traffic inside the cluster while Q_F^{opt} ~50 in the uniform distribution case but goes down to ~8.5 when 99% of the file requests are from inside the cluster.

All of the above discussion assumes that the optimal replica distribution can be achieved. In the uniform distribution case, LRU storage management gave near-optimal replica distribution [10] but for the clustered demands case, as we can see in (21) and (22) for the optimal average search time and (23) and (24) for the optimal query-processing load, the desired replica distribution depends on the degree of clustering in the underlying search network topology. However, rather than being a hindrance, this dependence of the optimal replica distribution on the underlying search network topology offers us a very powerful tool to achieve the optimal search performance. In a preliminary study, we were able to achieve the optimal query-processing performance with LRU storage management algorithm by tuning the underlying search network topology. This suggests that it may be possible to achieve the optimal replica distribution with any local storage management algorithm by appropriately tuning the underlying topology. If this approach of tuning the search network topology to reach the optimal replica distribution works in most cases, we may be able to obtain the optimal download performance [9, 10] (by using an LRU-like approach that populates file replicas in near-linear proportionality to the file request rates) and, at the same time, obtain the optimal query-processing load as well by tuning the underlying search network topology appropriately.

VI. CONCLUSION

In this paper, we derived results on optimal search time and optimal search cost in an unstructured peer-to-peer network when the demand exhibits clustering. The previous work in this area assumed uniformity in replica and demand distribution. Since real networks show clustering in demands, our results provide a more accurate estimate of the search performance achievable in unstructured peer-to-peer networks. Interestingly, we found that the gains in the optimal search performance afforded by clustering in demand patterns are independent of whether the search network topology matches the clustering in file popularity. The optimal replica distribution, however, does depend on clustering in the search network topology. Since the replica distribution is driven by peer requests, we believe that tuning the search network topology to match the replica distribution generated by peer requests is more practical than matching the replica distribution to the topology. In our simulations, we were able to operate a peer-to-peer network at the optimal search cost by tuning the clustering in the search network topology depending on the clustering in demands while using LRU cache management at each peer. In the process of deriving the optimal search performance results, we derived the relation between the query-processing load and the number of replicas of each file for the clustered demands case showing that flooding searches may have a lower query-processing load than random walk searches in the clustered demands case.

REFERENCES

[1] B. Bollobas, Random Graphs, Academic Press, London, 1985.

[2] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham and S. Shenker, "Making Gnutella-like P2P Systems Scalable," in Proc. of ACM SIGCOMM, August 2003.

[3] E. Cohen and S. Shenker, "Replication Strategies in Unstructured Peer-to-Peer Networks," in Proc. of ACM SIGCOMM, August 2002.

[4] F. Le Fessant, S. Handurukande, A. M. Kermarrec and L. Massouli, "Clustering in Peer-to-Peer File Sharing Workloads," in Proc. of IPTPS, February 2004.

[5] D. Liben-Nowell, H. Balakrishnan and D. Karger, "Analysis of the evolution of peer-to-peer systems," in Proc. of PODC, July 2002.

[6] I. Stoica, R. Morris, D. Karger, M. Kaashoek and H. Balakrishnan, "Chord: A Scalable Peer-To-Peer Lookup Service For Internet Applications," in Proc. of ACM SIGCOMM, August 2001.

[7] S. Tewari and L. Kleinrock, "Analysis of Search and Replication in Unstructured Peer-to-Peer Networks," in Proc. of ACM SIGMETRICS, June 2005.

[8] S. Tewari and L. Kleinrock, "Search Time in Unstructured Peer-to-Peer Networks with Clustered Demands," in Proc. of IEEE GLOBECOM, November 2005.

[9] S. Tewari and L. Kleinrock, "On Fairness, Optimal Download Performance and Proportional Replication in Peer-to-Peer Networks," in Proc. of IFIP Networking, May 2005.

[10] S. Tewari and L. Kleinrock, "Proportional Replication in Peer-to-Peer Networks," to appear in Proc. of IEEE INFOCOM 2006, April 2006.