The Case for Service Provider Deployment of Super-Peers in Peer-to-Peer Networks

Sumeet Singh, Sriram Ramabhadran, Florin Baboescu, and Alex C. Snoeren
University of California, San Diego
{susingh, sriram, baboescu, snoeren}@cs.ucsd.edu

***Abstract*— Peer-to-peer systems have recently introduced the notion of super-peers to improve search performance. While the benefit to end users is clear, it is not immediately evident who, if anyone, would be motivated to act as a super-peer. On the face of it, super-peers bear a much larger traffic burden than normal peers and receive negligible improvement in search performance when compared to any of the nodes they serve.**

In this paper we present incentives for several actors to deploy super-peers and propose a novel technical mechanism, topic-based search optimization, to increase the effectiveness of super-peers. By caching meta data—as opposed to content—at super-peers, topic-based search optimization has the potential to significantly improve the perceived search performance of a super-peer's clients with very modest storage and communication overhead at the super-peer itself.

I. Introduction

Unstructured peer-to-peer networks like Gnutella and KaZaA are characterized by the absence of specific mechanisms for enforcing a particular network topology or file placement. As a result, search proceeds by flooding queries to all nodes within a certain search horizon. Researchers have recently proposed extensions to the flooding mechanism, such as expanding ring search and random walks, that can improve search performance [1]. Unfortunately, these extensions require modifications to both the software and the protocols used at every node in the network. In contrast, systems like KaZaA and more recent versions of Gnutella improve both the efficiency and effectiveness of the search process by introducing a notion of network structure, elevating certain well-provisioned nodes to the role of super-peers. Super-peers serve as network hubs that index files belonging to other nodes.

While super-peers have proven effective in improving search performance, serving as a super-peer can incur significant communication costs for an individual node. Further, these communication costs grow with the number of client nodes a super-peer supports. Sadly, the effectiveness of super-peer-based searching is also directly tied to the out-degree of the super-peer. Hence, a tension exists: the more neighbors a super-peer has, the more effective it is in improving search performance for itself and its adjacent nodes but the greater the communication costs for the node serving as the super-peer. Since an individual peer-to-peer node can get similar performance by joining an existing, well-connected super-peer, rather than becoming a super-peer itself, it has no incentive to become a super-peer.

We observe, however, that several entities external to the peer-to-peer network, such as Internet Service Providers (ISPs) and content distributors or publishers, can leverage the unique capabilities of super-peers to affect their business goals. By enforcing policy at super-peers, service providers can transparently influence individual node behavior without the explicit knowledge or consent of the user. Unfortunately for the service providers, users (and their client software) are free to connect to the super-peer(s) of their choice. Hence, it is in the service providers' interest to provide an enhanced user experience to keep clients connected to their super-peer(s). If harnessed effectively, this confluence of incentives might result in externally sponsored super-peers that simultaneously improve users' perceived quality of peer-to-peer networks and service providers' ability to affect their business goals.

In this paper, we address two separate issues. First, we present incentives for several classes of service providers to deploy super-peers in unstructured peer-to-peer networks. Second, we address the technical issue of implementing a super-peer that simultaneously provides high-quality search results, admits a low-cost implementation (both in terms of communication and storage cost), and introduces a value-added service: topic-based search. Our key observation is that by caching meta data—as opposed to content—super-peers can often route queries locally (without flooding) but off-load content delivery to individual nodes. Further, by intelligently structuring the meta-data cache, super-peers can respond to topic-based queries, a service not currently available in most peer-to-peer networks.

II. Incentivizing service providers

We envision that at least two broad groups of service providers, ISP's and content distributors, could profit from deploying super-peers as an infrastructure service. This section discusses possible benefits to each group and commensurate economic incentives for them to maintain a super-peer infrastructure.

A. Peer-to-peer traffic engineering

Given the popularity of peer-to-peer applications and the large amounts of multimedia data downloaded through these applications, peer-to-peer traffic is growing to form a significant fraction of the traffic on ISP networks. Published analysis of the impact of peer-to-peer traffic on a particular ISP network concludes that "the high volume and good stability properties of peer-to-peer traffic indicates that the peer-to-peer workload is a good candidate for application-specific layer three traffic engineering in an ISP network." [2]

Further studies show that peer-to-peer traffic engineering could reap significant benefits for ISPs. In particular, Gerber *et. al.* analyze the current trends followed by cable-modem users of a Tier-1 ISP [3] and observe that "Peer-to-peer traffic does not show strong signs of geographic locality" and that the peer-to-peer applications do not exploit topological locality. Additionally Sen and Wang observe that 80% of the ASes communicate with multiple ASes, and the top 1% of the ASes communicate with at least 476 other ASes [2]. This inter-AS traffic is especially important to ISPs, as it typically affects their bottom line.

Unfortunately, there are several technical issues that make it difficult to traffic engineer peer-to-peer flows at present. The high penetration of the new generation of file sharing applications that use any available port on the host as opposed to utilizing a well-known port make it quite difficult to perform any kind of flow control or policy-based forwarding for file-sharing applications based upon static filtering routes [3]. Given the difficulties in applying conventional traffic engineering approaches to peer-to-peer systems, we observe that super-peers represent an ideal mechanism for application-level traffic engineering.

We envisage a peer-to-peer architecture in which ISPs deploy and maintain super-peers in their network that participate[1] in many of the peer-to-peer file-sharing networks. These super-peers act as an explicit point of entry to the peer-to-peer system for individual nodes. Following existing super-peer-based protocols, all queries from these nodes are routed through the super-peer(s) they are connected to. Thus enhanced connectivity to the peer-to-peer system is provided as a service by the ISP to its users (either with or without monetary compensation, perhaps depending on the level of service provided).

Since all queries are routed through its super-peers, an ISP can control which nodes these queries are forwarded to. This measure of control provides the ISP with a policy-driven framework that can maximize the economic value for the ISP with respect to neighboring ISPs. For example, one such policy could be to, whenever possible, forward a query to peers within the ISP in preference to peers outside. (This policy could be considered analogous to "inverse hot potato" routing but applied to peer-to-peer traffic.) Such a policy ensures that an ISP forwards as little peer-to-peer traffic as possible to peer ASes or upstream providers. Since inter-ISP settlements are often based on traffic volumes, such a policy might minimize the economic impact of peer-to-peer traffic that is flowing upstream. Another policy could determine, given a choice of upstream ISPs, which one to forward to. If both ISPs contain peers that can satisfy the query, it might make economic sense to forward the query only to the cheaper upstream provider.

Policies could also govern which peers to forward to within a single ISP's network. A super-peer could exploit detailed knowledge of the network topology to implement some form of load balancing or quality of service (QoS). If there are multiple peers within the network that can satisfy a query, the query could be forwarded depending on factors such as congestion and latency. Not only does this result in better utilization of the ISP network, it also translates into better performance for the end user in the form of faster downloads, possibly driving customers' selection of an ISP.

B. Selective content distribution

The current content flow model in peer-to-peer systems is pull-based, *i.e.*, users only receive content that they explicitly search for and request. We believe that a push-based model for content distribution is also desirable for peer-to-peer networks for a number of reasons. Current systems lack sophisticated meta-data-based search capabilities, which implies that users can only find content of which they have *a priori* knowledge. It is difficult for users to find content that is potentially interesting but of which they are not aware. When combined with user profiling techniques, the super-peer architecture enables content publishers to push content to potentially interested users.

[1] By participate we mean only that the super-peers implement the application-level protocol and route packets for other peers as stipulated by the protocol; our super-peers explicitly do not store or share any content.

To illustrate this, consider the following hypothetical scenario. A relatively small software publisher wishes to release a new computer game. The publisher has made demo versions of the complete game available on its website but does not have the adequate finances to either market the game (*i.e.,* the publisher does not have access to popular advertising mediums), or to concurrently distribute the game to a large user base. In such a situation, the super-peer architecture could come to the aid of the publisher by providing a platform to selectively target users who search for games or a similar set of applications. The same scenario could be extended to a musician wishing to release excerpts of an upcoming album through a peer-to-peer network.

The key aspect of the push-based super-peer model is that content is not pushed blindly. Instead, super-peers can provide links to the push content as part of a response to queries made for similar items. As this information is introduced transparently in the network, the clients do not need to upgrade their software or subscribe to an additional service. This type of a service is analogous to the colorful "sponsored links" section present in many search result pages returned by Google and other Web search engines. It is also reminiscent of the recommendation feature found on many popular web sites such as Amazon. Of course, in order to effectively target push content, super-nodes need to efficiently determine the nature of the query made by the requester; we present such a mechanism in Section III.

C. End-user perspective

After describing how service providers can use super-peers to manipulate the service received by client nodes in a peer-to-peer system, it is natural to ask why an end user would subscribe to the model in which a service provider provides and potentially monitors, controls, modifies, or even sells connectivity to a peer-to-peer network. We identify four potential reasons why a user would preferentially connect through a super-peer sponsored by a specialized service provider.

- A super-peer architecture can provide improved search capabilities when compared to existing unstructured peer-to-peer networks. Flooding-based systems suffer from the fundamental drawback that it is relatively difficult to find rare items in the system.
- Service providers may attract clients by providing an enhanced level of download performance by routing queries to peers that are likely to have the best connectivity to the client.
- Super-peers can provide users with novel aggregated views of the content available in the network, like the most popular downloaded objects, or the list of available 70s music objects.
- Super-peers can act as application-level bridges between different peer-to-peer protocols. By searching multiple peer-to-peer networks, the efficacy of individual searches could be increased.

We believe that the first of these reasons is the most important from an end user's perspective. The success of a peer-to-peer file-sharing application is historically dependent on how well the search mechanism works. Hence, we believe that service providers with incentives to deploy super-peers will be motivated to provide an enhanced search service, both in terms of efficiency and functionality.

III. Topic-based search optimization

Mere participation in a peer-to-peer protocol by deploying a super-peer infrastructure is likely not sufficient to exert any control over client behavior. To fully realize the benefits of being able to control and manipulate the peer-to-peer protocol, it is essential for the super-peers to ensure that a large fraction of the queries are routed through the super-peer infrastructure. It follows that some incentive mechanism is required to encourage individual peers to preferentially connect through these super-peers.

The previous section enumerated several benefits to the end user of using a service-provider-sponsored super-peer infrastructure. The remainder of the paper focuses on what we believe is the key incentive: improved search results. By giving end users a superior search experience, both in terms of the ability to find what is being searched for as well as faster downloads, super-peers provide a strong incentive for end users to remain connected to them. Newer versions of some peer-to-peer applications give the end user the ability to choose to which super-peer its client software connects [3]. Alternatively, the application logic itself may make this decision by choosing peers that return better results. In either case, the superior quality and performance of searches increases the likelihood that individual peers will connect to a provider's super-peer infrastructure in preference to other peers in the system.

Super-peers therefore require specific mechanisms to improve search performance for end-user queries. Accordingly, we introduce a new paradigm for query routing in our super-peers, which we call *topic-based search optimization*. Our strategy tries to exploit the following observations:

- There is likely to be significant locality in the type of content requested by individual peers, *i.e.*, peers

```
"OBJECT TYPE"   = "VD"
"Title"         = "[tmd]signs.(twciso).tc.(1of2)"
"Author"        = "TMD"
"Performers"    = "Mel Gibson"
"Description"   = "A Movie ’bout some aliens"
"Language"      = "-68EN"
"Category"      = "-76SF"
"Release Year"  = "2002"

"OBJECT TYPE"   = "AU"
"Title"         = "Call It What You Want"
"Artist"        = "Tesla"
"Album"         = "Psychotic Supper"
"Description"   = ""
"Category"      = "9RO"
"Release Year"  = "1991"
"Bitrate"       = "182"
"Length"        = "4:30"
```

Fig. 1. Meta information gathered from real Query Hit responses

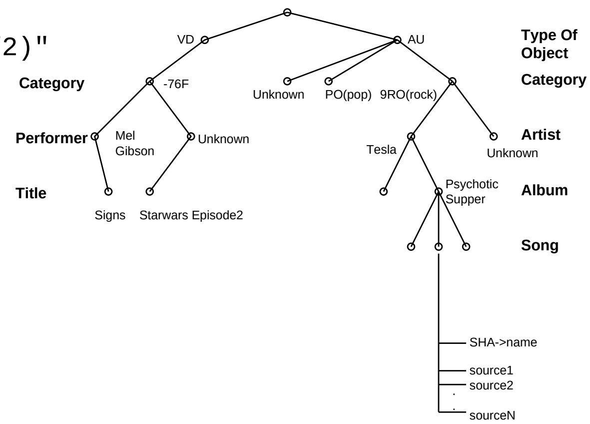


Fig. 2. A multi-dimensional name space search structure.

are likely to be able to respond to queries similar to queries they themselves have made in the past.

- Super-peers are in a unique position to generate aggregated views of the content in the network by observing the queries that are routed through them.

Topic-based search optimization involves generating a profile of the content currently available in the network by categorizing it into distinct topics. The topics are created dynamically by analyzing the meta data contained in Query Response messages routed via the super-peer. Since most queries are routed through a super-peer, it is in a unique position to be able to generate this topic-based view of the content and its location in the network. Meta data associated with an object typically contains a set of key-value pairs pertinent to the type of the object. For example, in the case of an audio file the meta data may contain one or more values in the following fields: *title, artist, album, category, release year, bit rate, length, description*, and *keywords*. In addition to meta data the Query Response message also provides a unique *signature* for every item returned in the form of a SHA digest.

Figure 1 shows meta-data information gathered from real Query Hit responses, and Figure 2 shows a view of this information in a multi-dimensional name space. Topics can be created by considering the type of objects, the category, the artist, or any representative combination of the fields that can be gathered from the meta data.

Associated with each topic is a list of peers who are likely to be interested in that particular topic based on their past query history. This list provides the super-peer with a wide selection of candidate peers who are likely to have related content. Topic-based search optimization improves upon two simpler potential strategies for improving search performance: The content itself could be cached at the super-peer, requiring large amounts of space and possibly incurring legal liability. Alternatively, the super-peer could maintain an index for each individual piece of content, rather than aggregating based on topics. We believe that, given the huge amounts of content available as well as the large number of peers we expect to connect to a single super-peer (a few thousand in our traces), these strategies are inefficient. Furthermore, we believe they might not even be necessary given sufficient effectiveness of topic-based search optimization. Also, unlike several previous schemes that exploit the locality of client interests, topic-based search optimization operates transparently to the end user.

IV. Evaluation

We evaluate our proposal through trace-based simulation. For the purposes of data collection we implemented a Gnutella client based on the publicly available Mutella [4] source code. When running, the client actively participates in the public Gnutella network as a super-peer. Our data-collection super-peers do not introduce any traffic into the network; they only gather the data that is being routed through them. Each super-peer allows between 20 and 200 leaf-nodes to connect to it. We have not yet compared the results of our study—which uses version 0.6 of the Gnutella protocol [5]—to those of a previous study using an earlier version [6].

We are interested in comparing the effectiveness of our scheme against a naive caching approach that simply stores a copy of all requested data. Figure 3 presents the query activity of a randomly selected two-hour period from our trace files. The absolute number of queries per minute is labeled QR. As a base line, we simulate the optimal caching strategy by checking the signatures

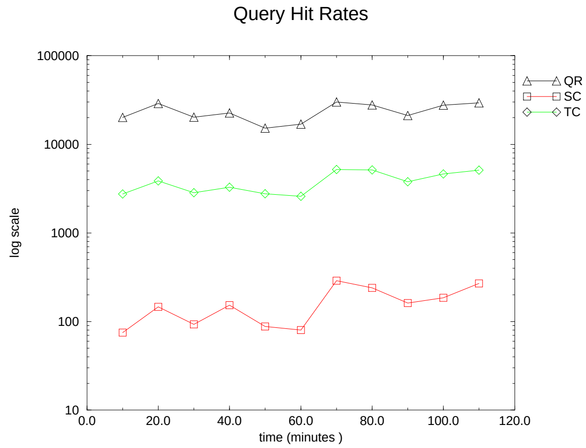


Fig. 3. The curve QR represents the number of queries, the curve SC represents the cache hit rate for objects matching exactly, and the curve TC represents the hit rate for the queries matching a topic of interest.

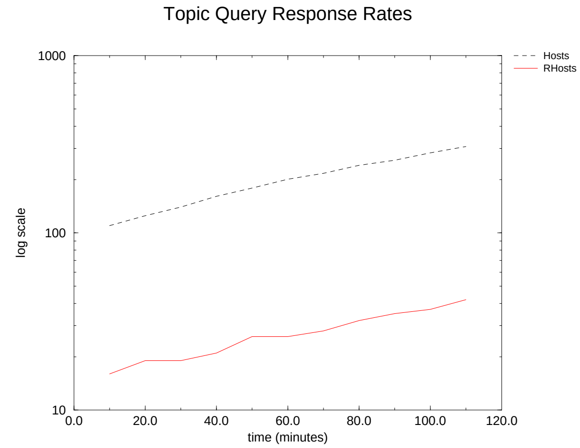


Fig. 4. This graphs compares the number of neighboring hosts making queries (Hosts), to the number of these hosts (RHosts) responding with useful results when sent a query with a topic extracted from their original query.

contained in all Query Responses against all previously received response signatures (recall that this is a two-hour section from the middle of our trace, so the cache is already warm). Hence, the line SC represents an optimistic upper bound for the performance of proposed caching based schemes [1], [7].

It is problematic to compare the performance of our scheme, as we return related search results that might not have been returned in the original network. Instead, we attempt to provide a rough estimate of the likelihood that topic-based search would return useful responses. The curve labeled TC represents the number of queries which had overlapping interest with at least one cached topic group. Although the number of queries that match this criteria is much higher than the number of queries that match the cached objects, we cannot directly compare the two. Recall the topic-based cache is built by aggregating the client's query requests and responses—thus we have no way to know what they actually store. Therefore there are no guarantees that clients will find the data they are looking for in any of the peers querying for content related to the topic. The likelihood of success is tied directly to the locality of clients' interest; that is, the chance that clients actually share data related to the topics of their queries.

We quantify the locality of client interest using a separate study. We wrote a second crawler based on the Gnutella protocol that connects to a large number of peers. Upon receiving a query from a neighboring peer, it extracts the first two words from the query, and creates a new query from these words and sends them to the neighbor the query had originated at. Even though this does not precisely capture the users interest topic, as many times the first two words signify nothing in particular, we believe they represent a crude notion of a search topic. The results of the crawler represented in Figure 4 seem promising: It is clear from the figure that approximately 15% of the peers do indeed respond to the queries that we are sending to them.

Combing the results of the previous two studies, we can deduce a very rough approximation of the effectiveness of topic-based search optimization. If we consider the 15% hit rate from above as representative of the response rate for peers in a given topic then we can infer from Figure 3 that topic-based search optimization is likely to out perform an infinitely large cache, since 15% of the TC curve is still substantially above the SC line. Furthermore, the storage requirements are substantially less. Figure 5 shows the number cached objects in our simulated cache as compared to the number of groups. Keeping in mind that the average size of each object is over 5 Megabytes, it's clear that topic-based search optimization is far more practical to implement.

A major concern with our approach, and, indeed any approach that only caches pointers to peers storing data as opposed to the data itself, is that the peers will not be available when subsequent queries arrive. Figure 6 attempts to quell that fear by plotting the uptime of all peers that connected to our super-peers during the duration of the trace. Note that a large number of peers remain connected for a substantial period of time.

Finally, we close by observing that topic-based content organization enables super-peers to provide clients with lists of popular content in each topic. Therefore, rather than the user searching for something specific, she can browse through the content currently available in the

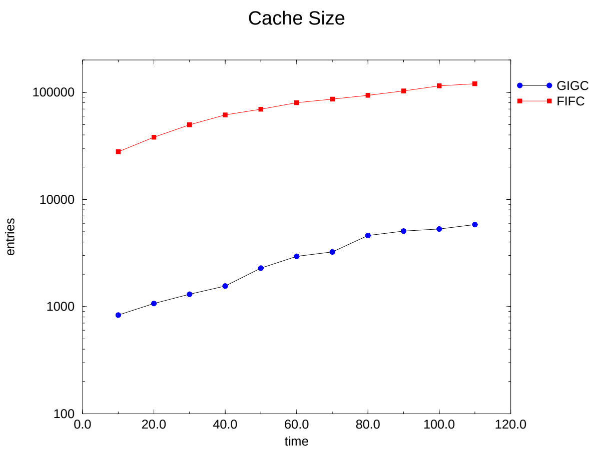


Fig. 5. This graph shows a comparison between the number of active groups in the topic-groups cache (GIGC), and the expected number of files in the file cache (FIFC).

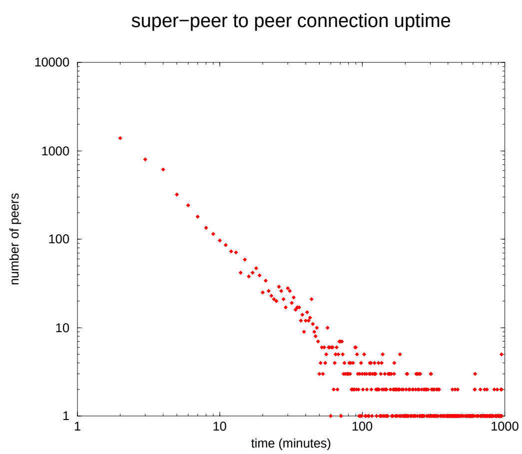
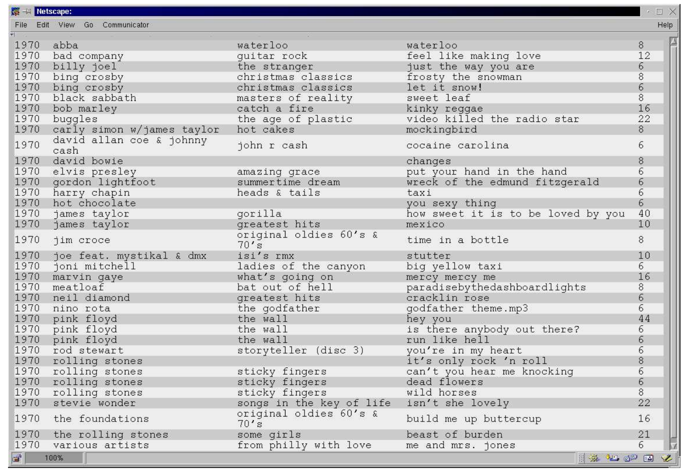


Fig. 6. The duration of peer relationships.



1970	abba	waterloo	waterloo	8
1970	bad company	guitar rock	feel like making love	12
1970	billy joel	the stranger	just the way you are	6
1970	bing crosby	christmas classics	frosty the snowman	8
1970	bing crosby	christmas classics	let it snow!	6
1970	black sabbath	masters of reality	sweet leaf	8
1970	bob marley	catch a fire	kinky reggae	16
1970	buggles	the age of plastic	video killed the radio star	22
1970	carly simon w/james taylor	hot cakes	mockingbird	8
1970	david allan coe & johnny cash	john r cash	cocaine carolina	6
1970	david bowie		changes	8
1970	elvis presley	amazing grace	put your hand in the hand	6
1970	gordon lightfoot	summertime dream	wreck of the edmund fitzgerald	6
1970	harry chapin	heads & tails	taxi	6
1970	hot chocolate		you sexy thing	6
1970	james taylor	gorilla	how sweet it is to be loved by you	40
1970	james taylor	greatest hits	mexico	10
1970	jim croce	original oldies 60's & 70's	time in a bottle	8
1970	joe feat. mystikal & dmx	isi's rmx	stutter	10
1970	joni mitchell	ladies of the canyon	big yellow taxi	6
1970	marvin gaye	what's going on	mercy mercy me	16
1970	meatloaf	bat out of hell	paradisebythedashboardlights	8
1970	neil diamond	greatest hits	cracklin rose	6
1970	nino rota	the godfather	godfather theme.mp3	6
1970	pink floyd	the wall	hey you	44
1970	pink floyd	the wall	is there anybody out there?	6
1970	pink floyd	the wall	run like hell	6
1970	rod stewart	storyteller (disc 3)	you're in my heart	6
1970	rolling stones		it's only rock 'n roll	8
1970	rolling stones	sticky fingers	can't you hear me knocking	6
1970	rolling stones	sticky fingers	dead flowers	6
1970	rolling stones	sticky fingers	wild horses	8
1970	stevie wonder	songs in the key of life	isn't she lovely	22
1970	the foundations	original oldies 60's & 70's	build me up buttercup	16
1970	the rolling stones	some girls	beast of burden	21
1970	various artists	from philly with love	me and mrs. jones	6

Fig. 7. A list of objects that match the filter for 70s music.

network, categorized by topics. For example, Figure 7 shows the simulated results of a query for 70s music produced by returning the results from a topic defined by an audio object type and a release year between 1970 and 1979. This type of search might prove popular with end users and is not currently supported by existing peer-to-peer protocols.

V. Conclusion and Related Work

In this paper, we argued that there are incentives for service providers to deploy a super-peer infrastructure. We are not alone in this observation, however. For example, Sandvine Inc. [8] is attempting to build a turn-key solution to perform tasks similar to traffic engineering for peer-to-peer networks. Unfortunately, they have not released any technical details of their system or provided case-studies that show the effectiveness of their product.

In order to attract users to their super-peers, we believe service providers will need to provide an enhanced level of service, including an improved search mechanism. To that end, we described a novel mechanism called topic-based search that improves search performance for client nodes by caching meta data at super-peers. Exploiting locality in user interests to improve search is a promising approach that has been pursued previously [9], [10]. Our approach is unique, however, in that information is collated at the super-nodes transparently to the end users. Therefore all the participating peers in the network can benefit from the optimizations without the need to upgrade the software at the client side.

References

[1] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in *Proc. 16th Annual ACM Int'l Conference on Supercomputing*, 2002.

[2] S. Sen and J. Wang, "Analyzing peer-to-peer traffic across large networks," in *Proc. Second Annual ACM Internet Measurement Workshop*, Nov. 2002, pp. 137–150.

[3] A. Gerber, J. Houle, H. Nguyen, M. Roughan, and S. Sen, "P2P: the gorilla in the cable," in *Proc. National Cable & Telecommunications Association (NCTA)*, June 2003, (to appear).

[4] "Mutella," 2003. [Online]. Available: mutella.sourceforge.net

[5] "Gnutella v0.6," 2002. [Online]. Available: groups.yahoo.com/groups/files/Development/GnutellaProtocol-v0.6-200206draft.txt

[6] M. Ripeanu, I. Foster, and A. Iamnitchi, "Mapping the Gnutella network: Properties of large-scale peer-to-peer systems and implications for system design," *IEEE Internet Computing Journal*, vol. 6, no. 1, 2002.

[7] E. Cohen and S. Shenker, "Replication strategies in unstructured peer-to-peer networks," in *Proc. ACM SIGCOMM*, Oct. 2002, pp. 177–190.

[8] "Sandvine." [Online]. Available: www.sandvine.com/

[9] E. Cohen, A. Fiat, and H. Kaplan, "Associative search in peer-to-peer networks: Harnessing latent semantics," in *Proc. IEEE INFOCOM*, Apr. 2003.

[10] K. Sripanidkulchai, B. Maggs, and H. Zhang, "Enabling efficient content location and retrieval in peer-to-peer systems by exploiting locality in interests," *ACM Computer Communication Review*, vol. 32, Jan. 2002.