

A Routing Technique for Query Scheduling and Forwarding in Unstructured Overlay Topology Networks

P. Deepa¹, A. Kanchana², V. Anitha³

Associate Professor, Department of CSE, Panimalar Engineering College, Chennai, India^{1,3}

Assistant Professor, Department of CSE, Panimalar Engineering College, Chennai, India²

Abstract: The peer-to-peer (P2P) paradigm has emerged as a viable alternative to overcome limitations of the client-server model namely, in terms of scalability, fault-tolerance, and even operational costs. This paradigm has gained significant popularity with its successful application in the context of file sharing applications. The success of these applications is illustrated by systems such as Napster, Emule, Gnutella, and recently, Bit Torrent. In order to ensure the scalability of these solutions many P2P services operate on top of unstructured overlay networks, which are logical networks deployed at the application level. Unstructured overlay networks build irregular neighboring acquaintanceships around members of the framework. In spite of the irregular nature for these overlays may be alluring Eventually Tom's perusing large portions P2P services, those coming about toponomy might display sub-optimal characteristics, for instance from the point of view of link latency. This may have a significant impact of the performance of P2P services executed over these overlays. In this paper we propose a query routing approach that accounts for arbitrary overlay topologies, nodes with heterogeneous processing capacity, e.g., reflecting their degree of altruism and heterogeneous class-based likelihoods of query resolution at nodes which may reflect query loads and the manner in which files/resources are distributed across the network. The approach is shown to be stabilizing the query load subject to a grade of service constraint, i.e., a guarantee that queries' routes meet pre-specified class-based bounds on their associated a priori probability of query resolution. Experimental results shows the performance benefits in terms of mean delay and reduced complexity. The query resolution probabilities and traffic loads are also studied.

Keywords: P2P networks, Overlay, Query resolution, Query Forwarding, Priori probability.

I. INTRODUCTION

Unstructured networks, by contrast, are easier to setup and maintain, but their mostly ad hoc overlay topologies make realizing efficient searches challenging. In a purely unstructured P2P network, a node only knows its overlay neighbors, with such are the limited information. The inefficiencies of purely unstructured networks can be partially addressed by hybrid P2P systems, but In structured networks, peers/files/resources are organized to form overlays with specific topologies and properties. Search mechanisms that perform name resolution based on distributed hash table. Although, as will be clear in the sequel, our results are not exclusive to hybrid P2P networks, these will serve as the focus of the project. The random nature of the location of query resolution in the network leads us to deal with expected queue backlog instead of current queue backlog Such systems use a simple two-level hierarchy where some peers serve as super-peers. These are high degree nodes which are well connected to other super-peers and to a set of subordinate nodes in a hub-and-spoke manner. Though such systems have advantages in terms of scalability, proposed search techniques are still based on variants of flooding and random walks.

II. EXISTING SYSTEM

In a purely unstructured P2P network, a node only knows its overlay neighbors. With such limited information, search techniques for unstructured networks have mostly been based on limited-scope flooding, simulated random walks, and their variants. Much research in this area has focused on evaluating these search techniques based on the contact time, i.e., number of hops required to find the target, using the spectral theory of Markov chains on (random) graphs, see e.g., Unfortunately in heterogenous settings where service capacity or resolution likelihoods vary across peers, such search techniques perform poorly under high query loads. The inefficiencies of purely unstructured networks can be partially addressed by hybrid P2P systems, e.g., FastTrack and Gnutella2.

2.1 Disadvantages of Existing System

In structured networks the difficulty of search/discovery is shifted to that of maintaining the structural invariants required to achieve efficient. In query resolution particularly in dynamic settings with peer/content churn or when



reactive load balancing is required. Standard backpressure-based routing our policies suffer from a major drawback: each node needs to share the state of its potentially large number of non-empty queues with its neighbors. Complexity problem will be also raised.

III. PROPOSED SYSTEM

Given a hybrid P2P topology and query classification, we propose a novel query resolution mechanism which stabilizes the system for all query loads within a 'capacity region', i.e., the set of loads for which stability is feasible. Essentially, our policy is a biased random walk where forwarding decision for each query is based on instantaneous query loads at super-peers. To balance the load across heterogeneous super-peers, the policy aims at reducing the differential backlog at neighboring super-peers, while taking into account the class and history information to improve the query's resolvability. Our policy draws upon standard backpressure routing algorithm, which is used to achieve stability in packet switching networks. We propose a query forwarding mechanism for unstructured (hybrid) P2P networks with the following properties. It dynamically accounts for heterogeneity in super-peer's 'service rate,' reflecting their altruism, and query loads across the network. To the best of our knowledge, this is the first work to rigorously account for such heterogeneity in devising a search mechanism for P2P networks. It is based on classifying queries into classes. This classification serves as a type of name aggregation, which enables nodes to infer the likelihoods of resolving class queries, which, in turn, are used in learning how to forward queries. Our approach is fully distributed in that it involves information sharing only amongst neighbors, and achieves stability subject to a Grade of Service (GoS) constraint on query resolution. The GoS constraint corresponds to guaranteeing that each query class follows a route over which it has a reasonable 'chance' of being resolved. We provide and evaluate several interesting variations on our stable mechanism that help significantly improve the delay performance, and further reduce the complexity making it amenable to implementation.

3.1 Advantages of Proposed System

Reducing complexity. Estimating parameters, and adaptation to class-based query resolution probabilities and traffic loads are studied. Stable Policies, Estimating Query Resolution Probabilities. Alternate Grades of Service Strategies. It is based on classifying queries into classes. The GoS constraint corresponds to guaranteeing that each query class follows a route. It has a reasonable 'chance' of being resolved. This provides a basis for substantially reducing complexity by approximations.

IV. SYSTEM DESIGN

4.1 System Architecture

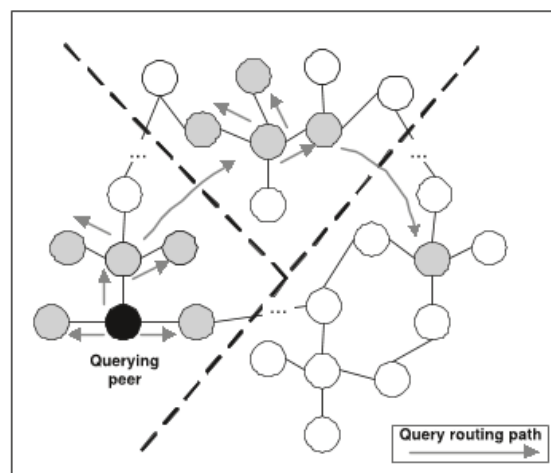


Fig 1 - System Architecture

The work presented in this chapter addresses the problem of dependable group creation and maintenance in large-scale P2P environment. In particular we propose a generic service that allows nodes to structure themselves in an overlay network of fault-tolerant process groups. The algorithm explores a topology management based on a control approach, where constraints are imposed over the neighboring associations among peers to achieve a unique topology. The solution presented here is fully decentralized and self-organizing. The topology resulting from our algorithm is composed of cliques of peers (a process group), where the members of each clique maintain connections to a large variety of other cliques, ensuring the overall connectivity of the overlay. This is achieved without relying on



neighboring constraints which take into consideration the identifiers of peers in the system. The global connectivity of the overlay is essential, not only to potentially ensure the correct operation of P2P services but also to ensure the correct operation of the algorithms employed to maintain the overlay topology despite failures, or even churn scenarios.

4.2 System Flow Diagram

A system flow diagram is a graphical representation of the "flow" of data through an information system, modeling its process aspects. It is often used as a preliminary step to create an overview of the system.

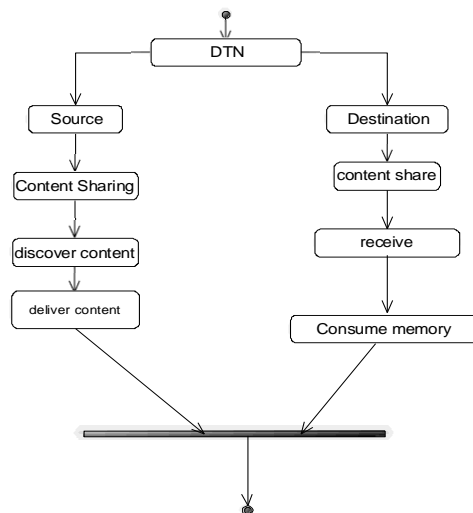


Fig 2 - System Flow Diagram

V. SYSTEM IMPLEMENTATION

The implementation of the proposed system consists of four modules. They are

1. Constructing System Model
2. Stable Query Forwarding Policy
3. Estimating Query Resolution Probabilities
4. Reducing Complexity

1. Constructing System Model

In the initial module, we develop the system with the entities required to show the proposed model with the evaluation proof of your novel contribution. So we develop the module with nodes. We assume that time is slotted, and each super-peer has an associated service rate, corresponding to positive integer number of queries it is willing to resolve/forward in each slot. We assume that super peers keep a record of files/resources available at subordinate peer. This information is communicated to super peers when a subordinate peer joins a super peer. Subordinate peers may initiate a query request at a super peer, but do not participate in forwarding or query resolution.

If a class query at node cannot be resolved it may be forwarded to one of its neighbors. The likelihood a node can resolve such a query depends not only on its class but also its history, i.e., the set of nodes it visited in the past. Note that the history is not ordered. For example, suppose 3 nodes in a network partition files/resources associated with class. If two of these nodes attempted and failed in resolving a given class query then it will for sure be resolved at the third node. In other contexts, if a search for a particular media file failed at many nodes, it is more likely that the file is rare, and the conditional likelihood that it is resolved at the next node might lower.

2. Stable Query Forwarding Policy

In this module, we will propose a query scheduling and forwarding policy that ensures the GoS for each class, is distributed, easy to implement, and is stable. We begin by defining the stability for such networks and the associated capacity region. The module is develop such that the following aspects arising in P2P search systems: (a) history dependent probability of query resolution at each node, (b) updates in 'types' of queries as they get forwarded to different nodes, (c) computing the quality of service received by query via its history and designing an appropriate exit strategy upon receiving enough service.

While the routing decisions are to be based on instantaneous queue loads at the neighbors, the decisions themselves affect the type/queue to which a query belongs. In this module, we develop a distributed dynamic algorithm where each node makes decisions based on its queue states and that of its neighbors and only needs to know.



3. Estimating Query Resolution Probabilities

In this module, we develop the estimation of query resolution probabilities. So far we have assumed that resolution probabilities for queries of different types are known. In practice they can be easily estimated. In order to ensure unbiased estimates can be obtained at each node, suppose a small fraction of all queries is marked ‘RW’, forwarded the random walk policy with large TTL, given scheduling priority over other queries.

With a sufficiently large TTL this ensures that each node will see a random sample of all query and types it could see and thus allow for unbiased estimates. All queries which are not marked ‘RW’ are treated according to our backpressure policy based on the estimated query resolution probabilities. A node receives ‘RW’ marked samples in time. Thus the error is small for large enough. If the contents are static, one may discontinue the estimation process after large enough time, in which case the time-averaged performance of the policy remains unchanged.

Alternatively, to allow persistent tracking of changes in resolution probabilities, we may estimate the query resolution probabilities via samples provided from a control algorithm, without using a separate unbiased random walk. The convergence of estimation and stability of the system can be jointly obtained via stochastic approximation framework under time scale separation between content dynamics and search dynamics.

4. Reducing Complexity

In this module, we develop the Reducing the complexities in the system. Not unlike standard backpressure-based routing our policies suffer from a major drawback: each node needs to share the state of its potentially large number of non-empty queues with its neighbors. For backpressure-based routing the number of queues per node corresponds to the number of flows (commodities) in the network. In our context, the number of queues per node corresponds to number of query types it could see worst case. In this module we propose simple modification and approximations that considerably reduce the overheads, albeit with some penalty in the performance. The key idea is to define equivalence classes of query types that share a ‘similar’ history, in the sense that they have similar conditional probabilities of resolution, and have them share a queue. For example, all query types of class which have visited the same number of nodes might be grouped together, reducing the number of queues to or better. Alternatively we will show one can further reduce overheads by approximately grouping similar query types.

Based on their classes and the cumulative number of class files/resources they have seen in nodes, reducing the number of queues to where is a set of quantization levels. Intuitively such queries have seen similar opportunities if files/resources are randomly made available in the network.

VI. RESULTS

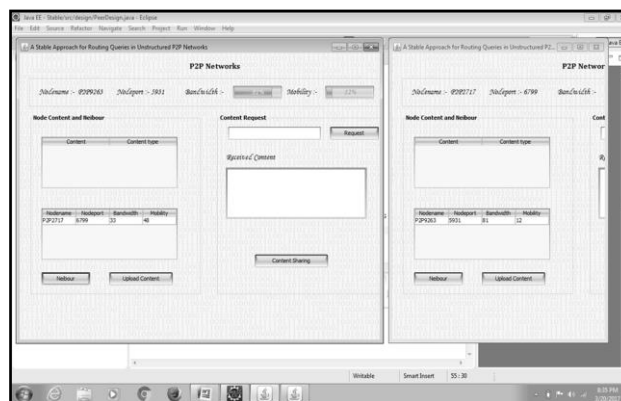


Fig. 3 Neighbor Node

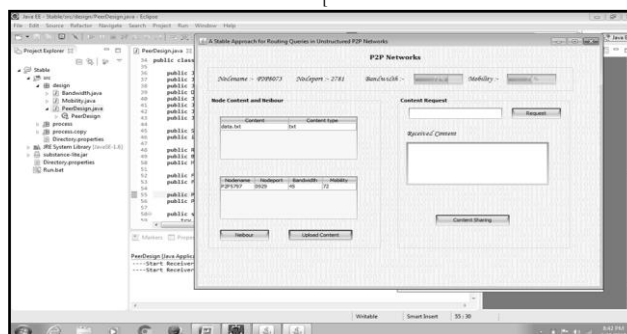


Fig.4 Upload Content

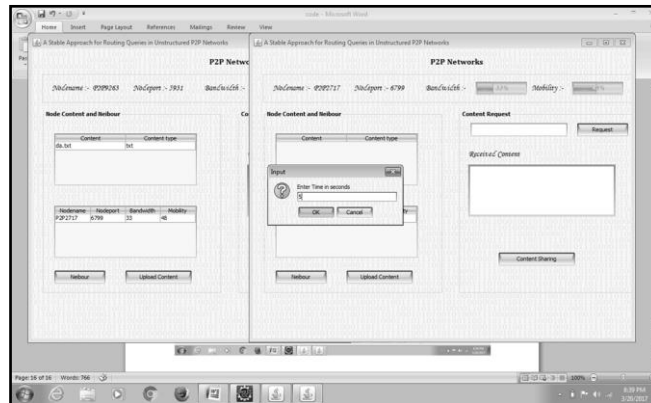


Fig.5 Sharing Content

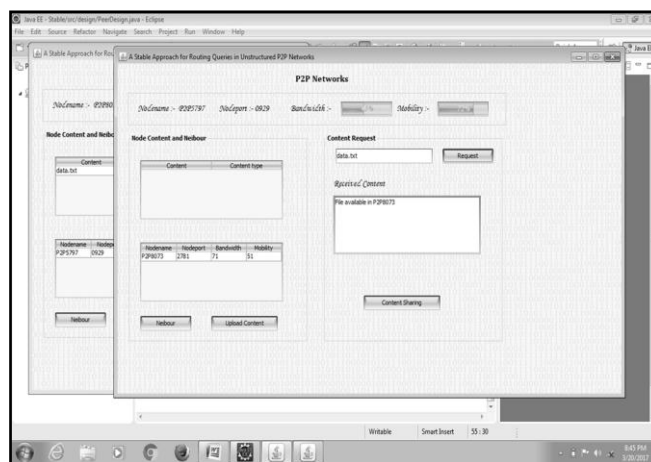


Fig.6 Received Content

VII. CONCLUSION

To summarize, we provided a novel, distributed, and reliable search policy for unstructured peer-to-peer networks with super-peers. Our backpressure based policy can provide capacity gains of as large as 68% over traditional random walk techniques. We also provide modifications to the algorithm that make it amenable to implement and the sharing content are also to be used to its successful sharing sources from one node to another node file the documents were available the certain file and its including data in the peer to peer networks with calculates the mobility and bandwidth rate of the each node in unstructured peer to peer networks.

REFERENCES

- [1] Virag Shah, Gustavo de Veciana, Fellow, IEEE, and George Kesidis, "A Stable approach for Routing Queries in Unstructured P2P Networks", IEEE/ACM TRANSACTIONS ON NETWORKING, 2016.
- [2] I. Stoica et al., "Chord: A scalable peer-to-peer lookup protocol for internet applications," IEEE/ACM Trans
- [3] X .Li and J. Wu, "Searching techniques in peer-to-peer networks," in Handbook of Theoretical and Algorithmic Aspects of Ad Hoc, Sensor, Peer-to-Peer Networks.
- [4] C. Gkantsidis, M. Mihail, and A. Saberi, "Random walks in peer-to peer networks," in Proc. IEEE INFOCOM.
- [5] C. Gkantsidis, M. Mihail, and A. Saberi, "Hybrid search schemes for unstructured peer to peer networks," in Proc. IEEE INFOCOM, 2005
- [6] S. Ioannidis and P. Marbach, "On the design of hybrid peer-to-peer systems," in Proc. ACM SIGMETRICS, Annapolis, MD, USA.
- [7] D. Menasche, L. Massoulie, and D. Towsley, "Reciprocity and barter in peer-to-peer systems," in Proc. IEEE INFOCOM, 2010.
- [8] B. Mitra, A. K. Dubey, S. Ghose, and N. Ganguly, "How do superpeer networks emerge?," in Proc. IEEE INFOCOM, 2010.