Enhancement of Stable Methods in Unstructured Network for Routing Queries

Kannam Siva[1], Vallabhaneni Pranav[2] , Vahibduddin shariff[3]

[1] *M. Tech Student, C. R. Reddy College of Engineering, Eluru, West Godavari Dt, AP, India*

[2] *Assistant Professor, C. R. Reddy College of Engineering, Eluru, West Godavari Dt, AP, India*

[3] *Assistant Professor, C. R. Reddy College of Engineering, Eluru, West Godavari Dt, AP, India*

Abstract - In an unstructured peer-to-peer network searching the document is very difficult problem because of more topologies and different processing capacity. In this system we propose a query routing approach which overcomes that problem. In this system the technique shown to be stabilize the query load subject to a grade of service constraint. In this system we introduce a query routing methods that works for arbitrary overlay topologies, nodes with heterogeneous processing capacity by using this technique. We get a result which further shows the proposed technique with decreasing complexity, estimation parameters and traffic load will be decreased. An explicit property of the region of system is given and mathematically compared to that associated with random walk based searches. One of the basic functions of this system is conveniently resolving queries or identifying files/resources.

Keywords- *Peer-to-Peer, Search, Stability, Backpressure, Simulation, Random walk, Heterogeneous.*

I. INTRODUCTION

In an unstructured peer-to-peer network searching the document is very difficult problem because of more topologies and different processing capacity. In this system we propose a query routing approach which overcomes that problem. This work proposes an approach where peers cache the outcomes of past queries as informed by reverse-path forwarding. The idea is to learn, from past experience, the best way to forward certain classes of queries, i.e., to intelligently "bias" their forwarding decisions by correlating classes of queries with neighbors who can best resolve them. This approach involves considerable overhead, is not load sensitive, and has not yet given guarantees on performance .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 paper. We assume that each super-peer contributes a possibly heterogeneous amount of processing resource for resolving queries for the network - incentives for doing so are outside of the scope of this paper. Super-peers serve their subordinates by resolving queries, or forwarding them to other super-peers. Super-peers can resolve queries by checking the files/resources they have, as well as those of their subordinate community. In our approach we also introduce a notion of query classes. These might, for example, represent types of content, such as music, films, animations, documents, or some other classification of files/resources relevant to the application at hand. The idea is that such a grouping of queries into classes can be used as a low overhead approach to make useful inferences on how to relay queries.

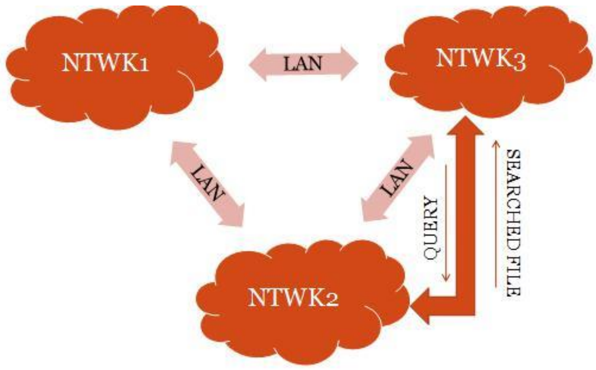

Fig.1. Architecture of proposed system

II. RELATED WORK

The Research on following on minimizing some function of average delay has unfortunately been an outstanding problem sometime. In this Dynamic programming represents a systematic approach for delay optimal control. Each node has to maintain a separate queue for each commodity in the network, and only one queue is served at a time. The back-pressure routing algorithm may route some packets along very long routes. So to improve the delay performance of the back-pressure algorithm. One of the Suggested solutions also decreases the complexity of the queuing data structures to be maintained each node.The Researcher on past it does not satisfy the end to end delay constraints for packets in queue on combining characteristics packets. In this paper construct the delay parameter p and d for satisfying end to end delay constraints. The Researchers on social network analysis has received renewed interest because of the significant increase in the number of users relying on applications based on them. An important criterion for the success of any social networking based application is the efficiency for search. So in this paper they overcome this problem and it reduces the search time by as much as 30 also indicate that the proposed method outperforms basic random walk even under considerable peer-churn. 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. T o summarize, our approach differs from standard work on backpressure in that we incorporate the following different issues that arise in P2P search: we model the uncertainty in the locations where a query may be resolved depending upon where the file/object of interest are placed, we guarantee a grade of service to each query under such uncertainties, we incorporate the information about a query's resolvability available through the knowledge of its history. In previously studied back-pressure based systems, the goal is to deliver packets to the corresponding destinations. By contrast, our aim is to provide a grade of service in resolving queries with no fixed destinations. 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. Further, in P2P systems, the probability of resolution of a query at a given node depends on the query's history, i.e., the path that led it to the current node. These characteristics of P2P systems are not captured in previous works on backpressure by Tassiulas and Ephremides and the subsequent enhancements.

III. LITERATURE SURVEY

Analyzing the three different approaches: Constructing System Model, Stable Query Forwarding Policy And Estimating Query Resolution Probabilities. All the three approaches a basis for substantially reducing complexity by approximations.

A. Constructing System model

We develop the system with the entities required to show the proposed 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.

B. Stable Query Forwarding Policy

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.

C. Query Resolution Probabilitie

We can estimation of query resolutionprobabilities. 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 at each node, suppose a small fraction of all queries is marked RW‗, forwarded via the random walk policy with a large TTL, and 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 timeaveraged 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 separation between content dynamics and search dynamics.

D. Reducing Complexity

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**.** 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. T he 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 c which have visited the same number of nodes k might be grouped together, reducing the number of queues to ⇥(|C||N|) or better. Alternatively we will show one can further reduce overheads by approximately grouping similar query types based on their classes c and the cumulative number of class c files/resources they have seen in nodes in H(☒), reducing the number of queues to ⇥(|C|L) where L is a set of quantization levels. Intuitively such queries have seen similar opportunities if files/resources are randomly made available in the network. Network with random file/resource placement. To better understand when such aggregation makes sense consider a network where files/resources are randomly and independently available at each node, i.e., at the super peers and/or their associated subordinate peers.

IV. PROBLEM STATEMENT

A. *Mathematical Model*

To create the system which is stabilize the query load subject to a grade of service constraints also this system decrease the complexity of traffic load and time complexity . It will make a easy searching a document in an unstructured peer-to-peer network. Peer-to-peer systems continue to find increasing and diverse uses as a distributed, scalable and robust framework to deliver services, e.g., document sharing, video streaming, expert/advice sharing, sensor networks, databases, etc. efficiently resolving queries or finding files/resources it is one of the basic function of such system. In peer to peer network all the node are connected to each other and they can communicate with each other directly without any data source (server). Peer- to- peer system has different topologies. In peer-to-peer networks all peers behave like data source and data sinks i.e. they can send or receive File/resource /documents from the source to destination. Peer- to- peer system has different topologies. They are various topologies like Star, bus, mesh, ring etc. So, finding document /file /resource in this network is difficult .In our system, we use query routing approach to find documents in unstructured peer to peer network.

B. *An Example*

In this system we use Dijkstra’s algorithm to overcome previous system drawback. For reduce time complexity, here we use shortest path algorithm Dijkstra's algorithm. This is asymptotically the fastest known single source shortest pat algorithm. For secure transmission of File/documents /resource in this network, we use asymmetric key encryption algorithm RSA. In RSA algorithm public and private key are used for encryption and decryption. Public key use for encryption and private key use for decryption. At user (N/W1) public key encrypt main file called as cipher text and send it to destination server(N/W2). At destination server private key use for convert cipher-text file into plain-text. If LAN network gets break then the transferring ocuments get stored in queue data structure (FIFO). So there is no data loss. Third party cannot access it because of security provided by RSA algorithm. The stored data in queue get forwarded into FIFO manner when network establishes. Input: Query as a input.Output: Successful searching of document Let S be the set of activitiesS={Q,A,R,F}Where,Q={Queries},A={Algorithms} R={Results},F ={Failures}

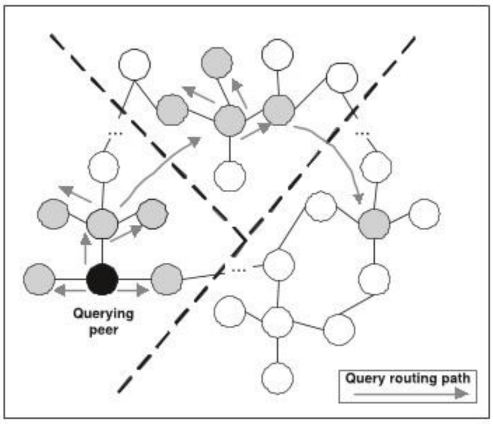

Fig -2: Query routing and forading

In this system we use Dijkstra’s for shortest path finding and for security RSA algorithm.

V. PROPOSED WORK

In the proposed work, 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. Specifically, we formally show that backpressure with aggregated queues, where aggregation is based on queries' histories, is stable for fully connected super-peer networks. This provides a basis for substantially reducing complexity by approximations.

a. BACKPRESSURE ALGORITHAM

The weights used in above algorithm for each link are different from those used in traditional multi- commodity backpressure algorithm Backpressure routing refers to an algorithm for dynamically routing traffic over a multi-hop network by using congestion gradients. It usually refers to a data network, but can apply to other types of networks as well. Below we focus on the data network application, where multiple data streams arrive to a network and must be delivered to appropriate destinations. The backpressure algorithm operates in slotted time, and every slot it seeks to route data in directions that maximize the differential backlog between neighboring nodes. This is similar to how water would flow through a network of pipes via pressure gradients. However, the backpressure algorithm can be applied to multi-commodity networks and to networks where transmission rates can be selected from different (possibly time-varying) options. Attractive features of the backpressure algorithm are: (i) it leads to maximum network throughput,(ii) it is provably robust to time-varying network conditions, (iii) it can be implemented without knowing traffic arrival rates or channel state probabilities. The original backpressure algorithm was developed by Tassiulas and Ephremides in [1]. There, they considered a multi-hop packet radio network with random packet arrivals and different link selection options. Their algorithm consisted of a max-weight link selection stage and a differential backlog routing stage. They did not call their algorithm "backpressure," and at the time of [1] the term "backpressure" had a different meaning in the area of data networks (it referred to a class of window-based flow control techniques, see [2]). The Tassiulas-Ephremides algorithm was first called "backpressure" in [3] and [4], where the algorithm was extended to treat networks with mobility, including ad-hoc mobile networks. An algorithm related to backpressure onside a multi-hop network with N nodes. The network operates in slotted time $t \in \{0,1,2,...\}$. On each slot, routing and transmission scheduling decisions are made in an effort to deliver all data to its proper destination. New data also arrives every slot. Let data that is destined for node $c \in \{1,...,N\}$ be labeled as commodity c data.Every node keeps internal queues that store data according to its destination. Let Q (c) n (t) be the current amount of commodity c data in node n , also called the queue backlog. The units of Q (c) n (t) depend on the context of the problem. For example, they can be integer units of packets or real valued units of bits. The queue backlogs from one slot to the next satisfy the following for all $n \in \{1,...,N\}$, $c \in \{1,...,N\}$ such that n 6= c.

VI. SIMULATION RESULT

In the baseline random walk policy, upon service, each node forwards an unresolved query to one of the neighbors chosen uniformly at random. Since, in a fully connected network, allowing queries to revisit nodes provides no advantages, queries are forwarded to only those nodes which are not previously visited. As with backpressure, whose achievable capacity region is given by Definition 2, we can characterize the achievable capacity region for the random walk policy. It is the set of arrival rates ? that satisfy the constraints (4)-(6),along with additional constraints that ensure that the outgoing flows of each type at each node are uniformly divided among unvisited nodes

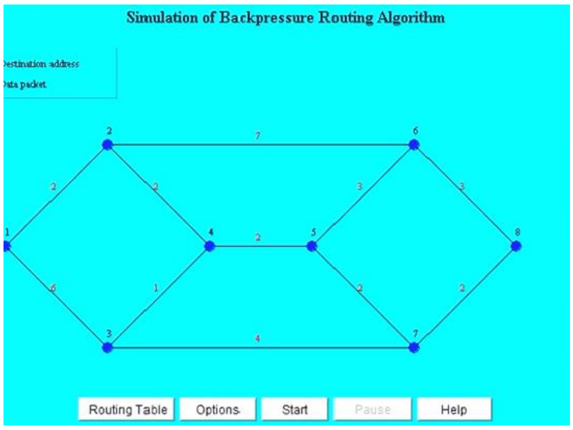

Fig.4: Simulation of back pressure algorithm

VII. CONCLUSIONS

In This Paper, We studied how get a result which further shows the proposed technique with decreasing complexity, estimation parameters and traffic load will be decreased. We studied how to transmit data over LAN network without service provider. The data security is maintained by using RSA algorithm and also we restrict the data loss by using data availability.

VIII. REFERENCES

[1]. Wikipedia, ―Peer-to-peer,‖ 2011 [Online]. Available: http://en.wikipedia.org/wiki/Peer-to-peer

[2]. I. Stoicaet al., ―Chord: A scalable peer-to-peer lookup protocol for internet applications,‖ IEEE/ACM Trans. Netw., vol. 11, no. 1, pp. 17–32, Feb. 2003.

[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. Boca Raton, FL, USA: CRC Press, 2004.

[4]. C. Gkantsidis, M. Mihail, and A. Saberi, ―Random walks in peer-topeer networks,‖ in Proc. IEEE INFOCOM, 2004, pp. 120–130.

[5]. C. Gkantsidis, M. Mihail, and A. Saberi, ―Hybrid search schemes for unstructured peer to peer networks, in Proc. IEEE INFOCOM, 2005, pp. 1526–1537.

[6]. S. Ioannidis and P. Marbach, ―On the design of hybrid peer-to-peer systems,‖ in Proc. ACM SIGMETRICS, Annapolis, MD, USA, Jun. 2008, pp. 157–168.

[7]. P. Patankar, G. Nam, G. Kesidis, T. Konstantopoulos, and C. Das, ―Peer-to-peer unstructured any casting using correlated swarms,‖ in Proc. ITC, Paris, France, Sep. 2009, pp. 1–8.

[8]. R. Gupta and A. Somani, ―An incentive driven lookup protocol for chord-based peer-to-peer (P2P) networks,‖ in Proc. Int. Conf. High Perform.Comput., Bangalore, India, Dec. 2004, pp. 8–18.

[9]. D. Menasche, L. Massoulie, and D. Towsley, Reciprocity and barter in peer-to-peer systems, in Proc. IEEE INFOCOM, 2010, pp. 1–9.

[10]. B. Mitra, A. K. Dubey, S. Ghose, and N. Ganguly, How do superpeer networks emerge?,‖ in Proc. IEEE INFOCOM, 2010, pp. 1–9.

[11]. D. Karger and M. Ruhl, ―Simple efficient load balancing algorithms for peer-to-peer systems,‖ in Proc. 16th ACMSPAA, 2004, pp. 36–43.

[12]. B. Yang and H. Garcia-Molina, ―Designing a super- peer network,‖ in Proc. IEEE ICDE, 2003, pp. 49–60.

[13]. L. Tassiulas and A. Ephremides, ―Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,‖ IEEE Trans. Autom. Control, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.

[14]. M. J. Neely, E. Modiano, and C. E. Rohrs, ―Dynamic power allocation and routing for time varying wireless networks,‖ in Proc. IEEE INFOCOM, 2003, pp. 745–755.