IOSR Journal of Engineering
Apr. 2012, Vol. 2(4) pp: 893-896

Best Information Search using Probabilistic Flooding in Unstructured Peer-to-Peer Networks

N. E. Pugalenthi[1], U. Pooranima[2]
[1]*Dept. of Information and Communication Engineering, Sri Venkateswara College of Engineering*
[2]*Faculty of Information and Communication Engineering, Sri Venkateswara College of Engineering*

ABSTRACT

The "peer-to-peer" (P2P) networks refer to a class of systems and applications that employ distributed resources to perform a function in a decentralized manner. Unstructured peer-to-peer networks do not impose any structure on the overlay networks. Peers in these networks connect in an ad-hoc fashion. Ideally, unstructured P2P systems would have absolutely no centralized system, but in practice there are several types of unstructured systems with various degrees of centralization. In heterogeneous decentralized unstructured P2P networks peers join and leave the application at their own will in an uncoordinated fashion and a central index for resource location is absent. Each peer is only responsible for maintaining a local index of the resources it owns and it is willing to provide to others. There are two main approaches for locating a resource in unstructured decentralized P2P networks; they are flooding and random walk. In random walk based strategies, peers forward a query message (termed as walker) to one randomly chosen neighbour whereas; in flooding based strategies peers forward a query message to all of its neighbours. The proposed work deals with the probabilistic approach which maintains shared table. The shared table consists of "terms" (query keywords) and the file name. When the peer receives the query, the shared table calculates the term frequency which is achieved by counting the number of occurrence of the term which the user has queried in each file. This process increases accuracy and also fetches the data very effectively.

***Keywords*: peer-to-peer, probabilistic search, unstructured, weighing algorithm**

1. Introduction

Peer-to-Peer is a term that typically refers to a file sharing network. P2P applications have been deployed in many different areas, such as distributed grid computing, storage, web cache, Internet telephony, streaming, conferencing, and content distribution and so on. But file sharing applications are perhaps the most popular P2P applications: many different file sharing systems, such as Gnutella, Kazaa, Edonkey, Emule, Bit Torrent, exist and collect millions of users. Peer-to-peer models are distinguished as structured P2P and unstructured P2P models. Structured P2P models are good for building systems where controlled resource placement is a high priority, such as distributed file storage. However, they are not good models for systems with highly dynamic membership. Unstructured models strive for complete decentralization of decision making and computation. They require only local maintenance procedures and are topologically robust in the face of system evolution. These models are good for building highly dynamic systems where anonymity and minimal administrative overhead are prized. Moreover, unstructured systems are characterized by a complete lack of constraints on resource distribution and minimal network growth policies. The systems focus on growing a network with the desirable low diameter of small world systems using only limited local information. Locating files based on their names is an essential mechanism for large-scale data sharing collaborations.

There are two main approaches for locating a file in unstructured decentralized P2P networks: flooding and random walk. In random walk based search strategies peers forward a query message (termed as walker) to one randomly chosen neighbour at each step although several walkers can be employed in parallel to increase the probability of successfully locating a resource (hit probability). In flooding based search strategies, when a peer requests a resource it sends queries to all its neighbours. This collection of neighbours may then forward the query to their neighbours (excluding, of course, the neighbour that sent the original request). These neighbours may then propagate the query to their neighbours and so on up to a certain predefined maximum level. In this paper, a mathematical model is developed to analyze the effect of these sources of heterogeneity in P2P networks on the number of messages required to discover a resource and on the hit probability. This paper propose and analyze a generalization of the flooding search strategies that exploits the advantages of heterogeneity to decrease the average amount of overhead traffic while increasing the hit probability for a resource. This paper also introduces the concepts to search the file that possess the best information content with that of the query message sent from the source.

In a mutual cast, a new delivery mechanism for content distribution is introduced in peer-to-peer (P2P) networks. Compared with prior one-to-many content distribution approaches, Mutual cast achieves full utilization of the upload bandwidths of the peer nodes, thereby maximizing the delivery throughput. Mutual cast splits the to-be-distributed content into many small blocks, so that the more resourceful nodes may redistribute more blocks, and the less resourceful nodes may redistribute fewer blocks. Each content block is assigned to a single node for distribution, which can be a content-requesting

peer node, a non-content-requesting peer node, or even the source node. The throughput of the distribution is controlled by redistribution queues between the source and the peer nodes. Furthermore, Mutual cast can be reliable and synchronous. Thus, it can be applied to file/software downloading, media streaming, real-time audio/video conferencing. [1] The use of peer-to-peer (P2P) applications is growing dramatically, particularly for sharing large video/audio files and software. The stunning growth and the bandwidth intensive nature of such applications suggests that P2P traffic can have significant impact on the underlying network. It is therefore important to understand and characterize this traffic in terms of end-system behavior and network impact in order to develop workload models and to provide insights into network traffic engineering and capacity planning. P2P traffic can be broadly classified into two categories: signaling and data transfer. Both types of traffic need to be measured in order to gain a solid understanding of P2P system behavior. [2] 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. [3] Peer to Peer networking has recently emerged as a new paradigm for building distributed networked applications. The peer-peer approach differs from the traditional client/server approach towards building networked applications in several crucial ways. Perhaps most importantly, a peer is both a producer and a consumer of the implemented service. In a peer to peer file-sharing application, for example, a peer both requests files from its peers, and stores and serves files to its peers. A peer thus generates workload for the peer-peer application, while also providing the capacity to process the workload requests of others. As a result, an increase in the number of peers results not just in an increase in workload, but also in a concomitant increase in the capacity to serve the workload. [4] Peer to Peer systems (P2P) have emerged as a significant social and technical phenomenon over the last year. They provide infrastructure for communities that share CPU cycles (e.g., SETI@Home, Entropia) and/or storage space (e.g., Napster, Free Net, Gnutella), or that support collaborative environments (Groove). Two factors have fostered the recent explosive growth of such systems: first, the low cost and high availability of large numbers of computing and storage resources, and second, increased network connectivity. As these trends continue, the P2P paradigm is bound to become more popular. [5] The simulation of a random walk or more generally a Markov chain is a fundamental algorithmic paradigm with highly sophisticated and profound impact in algorithms and complexity theory. Furthermore it has found a wide range of applications in such diverse fields as statistics, physics, artificial intelligence, vision, population dynamics, and bioinformatics, among others. Recently, random walks have been as primary algorithmic ingredients in protocols addressing searching and topology maintenance of unstructured P2P networks. [6]

2. SEARCHING STRATEGIES

In unstructured P2P systems, files are randomly distributed among peers and, consequently, there is no correlation between file placement and network topology. Therefore, file locating is generally concluded based on the flooding search mechanism: Each peer makes duplicate copies of a query it receives and broadcasts to all its directly connected neighbors except the one that delivered the incoming message in each forwarding step. The duplication process is terminated only when the TTL value of the query is reduced to zero, or a satisfying result has been found. This flooding mechanism is widely adopted in unstructured P2P systems due to its simplicity and robustness against node failure. The common existing search strategies used are

1) Random walk
2) Flooding

2.1 Random Walk

Random walk (RW) is a conservative search algorithm, which belongs to DFS-based. By RW, the query source just sends one query message (walker) to one of its neighbors. If this neighbor does not own the queried resource, it keeps on sending the walker to one of its neighbors, except for the one the query message comes from, and thus, the search cost is reduced.

Procedure RW

repeat
z:= an unsatisfied clause chosen at random
y:= a variable in z chosen at random
flip the value of y;
until TTL is alive.

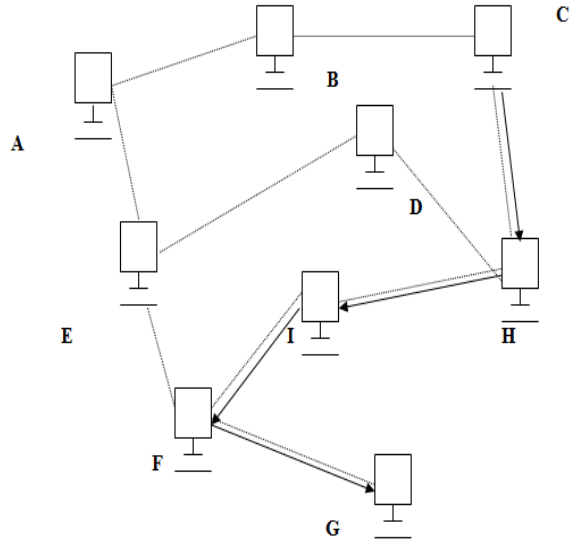

Fig. 2. 1 Random Walk Search Strategy

In Fig. 2.1, consider C as the initial node. The query is assigned to pass from C to the destination. The node C chooses a random path and passes the query to the node H.

Likewise the node H passes the query to its neighboring node I in random manner. This procedure is followed in all cases till it reaches the destination. Time to live concept is implemented in this process. If the TTL limit is set and the destination is found within the time then the query is passed successfully to destination else the packets will be discarded. For example if the TTL is set as 3 the query cannot reach the destination within the time, because the destination lies beyond the time period, hence the packets will be discarded. If suppose the time is set as 6 then the node C passes the query and the destination is found as G within the time, so the query is sent successfully.

The main drawback of RW is the long search time. Since RW only visits one node for each hop, the coverage of RW grows linearly with hop counts, which is slow compared with the exponential growth of the coverage of flooding. Moreover, the success rate of each query by RW is also low due to the same coverage issue. Increasing the number of walkers might help improve the search time and success rate, but the effect is limited due to the link degree and redundant path.

2.2 Flooding

Flooding, which belongs to BFS-based methods, is the default search algorithm for Gnutella network. By this method, the query source sends its query messages to all of its neighbors. When a node receives a query message, it first checks if it has the queried resource. If yes, it sends a response back to the query source to indicate a query hit. Otherwise, it sends the query messages to all of its neighbors, except for the one the query message comes from.

Procedure Flooding
repeat
z:= an unsatisfied clause chosen at random
if there exists a variable y in z with break value = 0
flip the value of y
else
y:= a variable in c chosen at random from z;
flip the value of y
until a satisfying assignment is found.

In flooding model designed in Fig. 2.2, the query is passed from initial node to all its neighboring nodes. It does not choose any particular path, it floods the packet to all the neighbors. For example assign D as the initial node. Then the source floods the packets in all direction. So the packets are passes to node I and E. From the nodes I and E the packets are transferred to their neighboring nodes and thus the process is repeated till the destination is reached.

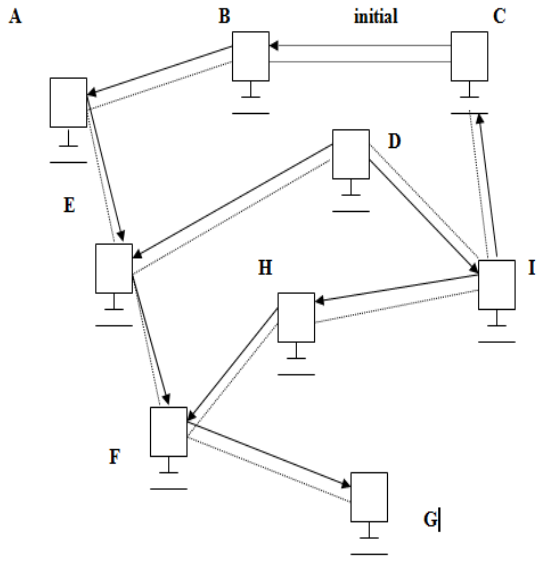

Fig. 2.2 Flooding Search Strategy

The limitation of flooding is the search cost. It produces considerable query messages even when the resource distribution is scarce. The search is especially inefficient when the target is far from the query source because the number of query messages would grow exponentially with the hop counts.

3. PROBABILISTIC FLOODING

Queries are originated by peers that set the time-to-live attribute to an integer value denoted as TTL. A peer that receives a query decreases the TTL by one. If the TTL reaches the value zero then the query is not forwarded. Processing a query involves decreasing its TTL, searching through the peer resources to look for a match, and forwarding the query (depending on the TTL) to its neighbours according to a particular search algorithm. A query is not forwarded back to the peer that sent it. A peer that finds a match for a query forwards it anyway to increase the hit probability. When a peer has to forward a query to one of its neighbours it does so with a probability that is a function of the degrees (the number of connections in the overlay) of both and of their distances from the query originator. Distances are expressed in number of hops and for the forwarding peer it must be less than TTL. In probabilistic flooding strategy where a peer decides to forward a query to its neighbours using a probability pf that is a function of its degree. A peer accepts an incoming query with a probability pr that is a function of its degree. To reduce the message overhead both probabilities may depend on the distance from the query originator, as well.

Probabilistic flooding search involves the dynamic search algorithm. The probabilistic flooding search generalizes the concepts of Random walk and flooding. The procedure for the probabilistic flooding is given below,

Procedure Probabilistic
repeat
z:= an unsatisfied clause chosen at random

if there exists a variable y in z with break value = 0
flip the value of y
else
with probability p
y:= a variable in z chosen at random;
flip the value of y
with probability (1-p)
y:= a variable in z with smallest break value
flip the value of y
until a satisfying assignment is found.

The operation of the dynamic search algorithm involves two phases, which is explained further. Each phase has a different searching strategy. The choice of search strategy at each phase depends on the relationship between the hop count h of query messages and the decision threshold n of DS.

1) Phase 1. When h ≤ n

At this phase, DS acts as flooding. The number of neighbors that a query source sends the query messages to depends on the predefined transmission probability p. If the link degree of this query source is d, it would only send the query messages to d.p neighbors. When p is equal to 1, DS resembles flooding.

2) Phase 2. When h > n

At this phase, the search strategy switches to RW. Each node that receives the query message would send the query message to one of its neighbours if it does not have the queried resource.

In the probabilistic flooding approach, the number of queries sent throughout the network starting from a peer that does not have a copy of a resource and that issues a request for it. From this generating function, the average number of query messages as well as the hit probability for a query is obtained. Finally the availability distribution is derived, i.e., the probability that a peer is not overloaded by query traffic.

4. BEST INFORMATION SEARCH USING PROBABILISTIC FLOODING

4.1 Shared Table

The information such as term (query keyword) and term frequency are stored as the Index file in the shared table, so that the data search is made to the shared table to find the best data relevant to the query. The performance can be increased by reducing number of query messages and time taken with the help of previous experience.

4.2 Probabilistic Flooding

Search is made to the shared table using probabilistic flooding search strategy in order to fetch the best data, which implicitly specifies the Node and the path to reach the node. This search identifies the node which has the data through searching with the query made by the user. This process avoids time loss, reduces band width and also fetches the data very effectively.

4.2 Finding Best File

Weighing algorithm is calculated to find the best file in the entire peer group. Once the query is provided to the shared table, the shared table will consider the index file stored in it & process the query to all the peers. Query key word is called as "**Term**", after the steaming algorithm is applied to fetch the key words in each & every file which is stored in all the peers. Once the peer receives the query, the shared table calculates the term frequency which is achieved by counting the number of occurrence of the term which the user has queried in each file. So the ratio is calculated by comparing term frequency with the total number of key words. This value is called as weight of term in a data of the particular peer. All the values are tabled in a shared table.

4.3 Ordering Top Values

After getting the corresponding weight of the term in a file of the particular peer, all the values are tabled for further processing. The scores are compared with each other, to fetch the top values and it is made in the ascending order. The top valued data are kept in the order.

5. CONCLUSION AND FUTURE WORK

The proposed work analyzes the impact of heterogeneity in P2P-based applications on the number of queries sent throughout the network by peers. Peers request resources and search the entire process. So the generalized random walk method is used. In random walk method, a peer forwards a query to one of its randomly chosen neighbor. Thus, query may or may not retrieve the contents. This serves as a drawback of random walk method. There by, flooding search method is adopted in which a peer forwards a query to all other peers in the network, and so the contents are retrieved at any cost. Flooding based search also has some drawbacks, which peers are forward query and disturbing the all neighbors. So peers fetch the appropriate data using the shared table. Shared table stores the each peer's information file name, which peer forward the query to the shared index table and fetch the relevant data. Our future work is to extend the model to avoid peers that have the requested resource to continue flooding the query.

REFERENCES

[1] Xinyan zhang and Jjianchaun Liu ; Bo Li ; Yum, Y.-S.P (2005) 'A data driven overlay network for peer to peer networks' on conrerence, vol. 3.
[2] S. Sen and J. Wang (2004) 'Analyzing peer-to-peer traffic across large networks' IEEE/ACM Transactions on Networking, vol. 12, no. 2, pp. 219–232.
[3] S. Tewari and L. Kleinrock (2005) 'Analysis of Search and Replication in Unstructured Peer-to-Peer Networks' in Proc. of the ACM SIGMETRICS Conference.
[4] Z. Ge, D. R. Figueiredo, S. Jaiswal, J. Kurose, and D. Towsley (2003) 'Modeling Peer-Peer File Sharing System' in Proc. of IEEE Infocom,.
[5] Matie Ripeanu and Ian Foster (2001) 'Large scale of peer to peer to system' IEEE Conference.
[6] Gkantsidis, M. Mihail, and A. Saberi (2004) 'Random Walks in Peer-to- Peer Networks,' in Proc. of IEEE Infocom.