Using Proximity Measure to Improve Locality in Structured P2P Networks

Rofide Hadighi
Department of Computer Engineering
Mazandaran University of Science and Technology
Sheikh Tabarsi Avenue
Babol,Iran

Mohammed Gharib
Department of Computer Engineering & Information Technology
Sharif University
Azadi Avenue
Tehran, Iran

ABSTRACT
Peer-to-peer networks are generally characterized in terms of sharing computer resources without the intermediation of a centralized server. Interconnected nodes in peer-to-peer networks are able to communicate through a self organizing topology which runs as an overlay on top of the physical network. The mismatch problem between underlay and overlay network in such systems, known as locality problem, creates extra traffic in the network. Knowledge about peers in the underlay network can be used to find the solution of locality problem by defining a proximity measure. This paper proposes an algorithm to measure proximity of nodes in peer-to-peer networks. In this algorithm, we measure proximity among pairs of nodes in the overlay network. The main advantages of our algorithm are making use of two metric for proximity evaluation, and comparing our simulation results with a well known and a structured peer-to-peer network for a better assessment. Also using real data is proper for algorithm performance verification. Results on real data indicate a good performance for the algorithm with low overhead in time and traffic by.

General Terms
Proximity Measure, Locality Algorithms, Traffic, Time.

Keyword
Peer-to-peer networks; Locality Problem; Binary Tree; Diamond Tree; Hypercube topology.

1. INTRODUCTION
Peer-to-peer (P2P) networks are a combination of self-organized autonomous entities, named peers, with common interest. They are as a large-scale distributed resource sharing paradigm .The most important property of P2P network is distribution of references and data management. Based on these properties there are three types of P2P systems. The first type is unstructured File-sharing systems like Napster[1], with a central server, the second like Gnutella[2], is also responsible for sharing of unstructured data but file placement is done randomly; and the third type is structured P2P network, this system is based on a DHT[1][3].

P2P networks have initially been developed to allow sharing of unstructured data, with no restriction on data placement in the overlay topology. Since up to 70% of internet traffic is due to P2P users in unstructured model [4], recently P2P systems provide support for sharing of structured data [5].

P2P systems usually exist on top of TCP/IP networks as an overlay network. There is no dependency between the geographical position of any nodes in the underlay and the position of the nodes in P2P network. Due to lack of information about the underlay network, there is no guaranty to find the requested data in near nodes in the overlay network. This inefficiency creates locality problem that is not often considered in some of the structured P2P networks[6]. Locality can be described as the proximity between nodes in P2P networks. The proximity can be measured by using different criteria, such as physical location, the number of hops, link latency and etc. For the case that hosts are directly connected via an optic fiber, in link latency measure, the nodes are logically very close, even if they are physically placed far away. Hence, depending on the criteria for measuring proximity, different locality algorithms are needed[7]. In existing locality measuring algorithms, the researchers only used one criterion. For example, the authors in[8, 9] have used the round trip time[2], and have employed the hop count in[10,11] as a criterion for measuring the proximity. In this paper, we have simultaneously considered two proximity measuring criteria, the round trip time and the hop count. According to proposed algorithm, the nearest node is chosen by these two criteria as a neighbor for the new node attending to the network. For providing locality, it is desirable to use a structured P2P network, because of its good performance in using DHT function systems.

The existing algorithms for providing locality are categorized into three groups. The first group of the algorithm is relied on Landmark server, where server is a basement for node placement. But the server is at the risk of single failure and hotspots[12]. In the second group of the algorithm, peers allocation is related to their IP Prefix, peers with same PI prefix are close to each other in the geographic zone, but because of some limitations in IP mapping it is not definite and permanent[13]. ehT third group of the algorithm, uses dimensional coordinate system. This system assigns a coordinates to each node and then the nodes will improve their coordinates with respect of each others. The most important disadvantage of this type of algorithms is the probability of not leading the network to the stable state [14]. Also there is a new algorithm for improving the locality by using hypercube and cube connected cycle graph which has many advantages, it supports a low overhead for traffic and time, but it uses just one metric for measuring the locality[9]. Here we are interested in proposing a new algorithm that improves the network performance and provides locality by eliminating the existing problems. In the last proposed algorithm[8] for providing locality, hypercube topology was selected as an overlay network which has many advantages but the degree of nodes increases as network size increases

[1]Distributed Hash Table

[2] RTT

thus it generates much traffic. According to the similarity between tree and hypercube in address assigning of nodes, and similarity in structure, first we select the binary tree in our implementation. Simulation results showed the high traffic and RTT, thus we introduce the diamond tree, a symmetric and regular tree that is evolved of tree family topology for implementing the proposed algorithm. For a better evaluation of our work we implement the algorithm with hypercube too and compare the simulation results with Chord network. The results showed the similarities between hypercube and diamond tree, and high difference with chord network.

The remainder of our work is organized as follows; Section 2 presents an overview of the related work inP2P systems. Section 3 explains the proposed algorithm. Section 4 shows the experimental results and analyses. Finally, we conclude in Section 5 and discuss open issues.

2. RELATED WORK

P2P networks are distributed systems consisting of underlay and overlay networks. The physical connections between nodes construct the underlay network that has a virtual connection in overlay network. Overlay networks create a structured virtual topology on top of the transport protocol layer that facilitates deterministic search and guarantees convergence. Overlay networks have four common goals: guaranteed data retrieval, provable lookup-time(typically $O(log\ N)$ with N being the number of network nodes), automatic load balancing, and self-organization [15].

2.1 Types of Graphs

Structured P2P overlays use a number of different graphs to show the structure of nodes in P2P overlays. The term graph is referred to a structure to organize nodes in a P2P overlay. With this structure the network is able to a deterministic lookup. The performance of lookups in a structured P2P overlay is directly related to how nodes are arranged and how the graph is maintained when new nodes arrive and when old nodes leave. Further, these graphs have an important impact on the performance of the P2P overlay[16].The graph structures show the network topology. Topology is an important factor in designing the appropriate structure of a network[17]. Three major factors that influence the choice of a specific graph for the topology of overlay are the degree of a node, the diameter of the network and a recursive, symmetric and regular structure. Degree is the maximum number of outgoing edges from each node and diameter is the maximum distance between two nodes. If the network seems the same as the point of any node, it is named a symmetric network. Symmetric networks are desirable, since they simplify the algorithm design. A good topology should preferably be symmetric, regular with low degree and also low diameter[18]. The network topologies can be divided to two categories. The first one is basic topologies like Hypercube, Mesh, Tree and etc. The other category is compound topologies like Diamond tree.

Figure 1 shows the basic topologies. The first topology, Figure1.a, is a complete graph, which all nodes are connected to each other directly, so there is a direct path between every two nodes and all nodes know each other. The problem of this type of topology is that by inserting only one node to the network, degree of all other nodes will increase. The second topology, Figure1.b, swohs the mesh topology; degree of nodes in a 2 dimensional one is at least 2 up to 4. In these graphs, the diameter is relative to the number of network size and the graph does not have a symmetric structure because the node degree is not constant. The third topology, Figure1.c, is ring with long diameter. Nodes are in a ring structure. The message should traverse the entire node in the path to get the destination node and it causes much traffic and time to get the destination node. The forth topology, Figure1.d, is Torus. Torus is the same as mesh but in every row and every column the tsrif node and the last node is connected to each other[17, 19]. The fifth topology, Figure1.e, is hypercube. The advantage of this kind of graph is that the growing factor always is in order two. The number of nodes is duplicated with increasing the number of dimension and this is a good point of this graph. The diameter and the number of dimension are equal. Also, the number of edge is the same as the number of dimensions. The problem is that the most degree leads network to the more neighbor nodes which make easy access to other nodes, but it must be sent more messages for searching ,and as a result, the traffic of the network increases more and more but the joining time will reduce[8].

Ffinally the sixth topology, Figure1.f, is tree topology. It is a simple topology and the node address assigning is easy. The degree of root node is two, the degree of child nodes is three and the degree of leaf nodes is one. The diameter is equal to twice of the tree levels. Tree does not have some good properties of other types of topologies. It is not symmetric, not regular and not recursive, thus providing locality will be so difficult. There is just one path between two nodes, thus if one node fails or the path between two nodes fails the connection between nodes will drop[18, 19]. Because of these disadvantages, we have introduced the Diamond tree.

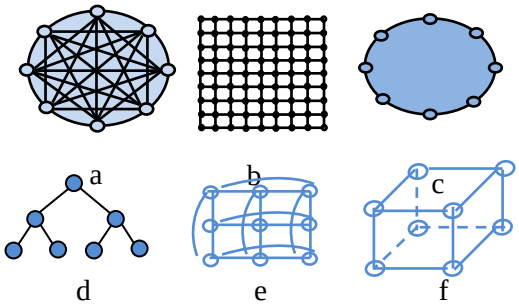

Figure 1.Basic Topologies

Figure2 shows Diamond tree as a compound topology.tI is based on tree family and compound of four trees.

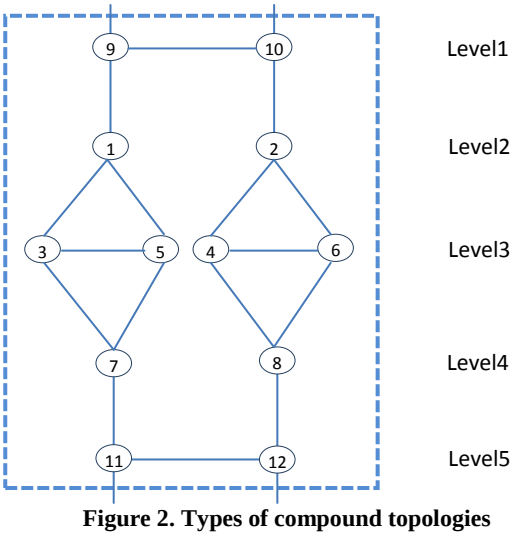

Figure 2. Types of compound topologies

Diamond tree has good properties against binary tree; it is a symmetric and regular tree. Nodes have a constant degree; the diameter is equal to number of levels of the diamond tree and shorter than diameter of the simple tree. Also it has a recursive structure.

2.2 P2P networks generations

There are three main noitarenegs for P2P systems: Centralized, Decentralized Unstructured and Decentralized structure. In the first generation, such as Napster, there is a central server, for storing the indexes of shared files. The new node sends its request to the server to assign an address for locating in the network. Every node shares the data with its neighbor through this server. Thus the server manages the network. So the server is vulnerable against DOS[1]attack .In the second noitareneg like Gnutella and KaZaA, file placement is random and osla is not related to the network topology. High traffic krowten ot gnidaelcollapse is the common problem in this class. In the third noitareneg, the server will be omitted and the network is based on DHT function. By this function given a key (e.g. a file name), return the location (addresses) of the nodes that currently have references for corresponding data objects (the files). All files are distributed between all peers. CAN[12], Chord[20], Pastry[10] and Tapestry[11] are examples of this noitareneg[21]. The good feature in this type of systems is scalable routing because each node maintainsa scalable number of routing state which means that the expected number of forwarding hops between any two nodes is small. Each message is routed to the nearest node whose address is specified in the destination field of message. All these overlay systems support deterministic routing of messages to a live node which is responsible to the destination key. There is a guarantee for queries to find an existing objects under non failure conditions[8].Recent works include systems such as Kademlia[22] , which overlay routing is based on XOR function, and Viceroy[23], which provides logarithmic hops through nodes.

Today traffic generated by the P2Psystems is a challenging issue, which has motivated many methods to reduce the impact of the increasing P2P traffic in networks. There are many solutions for reducing the traffic of P2P systems but Locality algorithm is one of the best strategies among all. Locality algorithms use information knowledge on the peer's network positioning, for selecting an optimized set of peers. With this algorithm, the sets of cooperative peers will be chosen in a logic manner as comparison with random selection of peers thus guaranteeing a better utilization of the available resources in the network and also optimizing the performance of the P2P network. Also locality selects the nearest node in the underlay network[7].Without locality, logical neighbor selection will be done randomly without any knowledge about underlying physical topology. It results in the same message may traverse the same physical link multiple times therefore the same query message may traverse the same logical link twice and then it causes much traffic in the network[21].

Locality can be defined as the proximity of the hosts in the network. Proximity can be measured byemos different criteria. Locality criteria are defined as physical location of node, number of hop count between two hosts and the link delay that is mentioned as RTT delay. Our measurement is based on two proximity criteria, time distance and hop count distance. For example, if the proximity only depends on time delay between network nodes, hosts directly connected by an optic fiber are most likely very near, even if they are physically placed far away from each other. One of the most important goals of locality algorithms is to find an optimized set of hosts that provide lower network latency, less traffic usage, cost reduction, depending on the optimal proximity criteria. So, measuring locality with erom criteria will be more accurate. There is a relation between hop count and RTT delay, and it is of importantt criteria in the network processing. Also estimating the RTT is easier and takes less cost than other criteria[24].

2.3 Classification of Locality algorithms

P2P networks refer to their highly distributed characteristics emaceb very popular between internet users. Each peer consumes and provides some resources, simultaneously; this characteristic makes the systems scalable. With this popularity, several studies show that the P2P technology is also responsible for a high volume of traffic in the network core links and it causes many costs for ISPs. Thus, this is a challenging issue for researchers to motivate the development of many proposals aiming to reduce the traffic of P2P systems in different networks. Among all solutions resulting in this research effort, one of the most prominent strategy is the use of locality algorithms. Due to their potential to provide useful information knowledge about the situation of the network, thus, leading to a higher performance and better usage of network resources. Locality algorithms use information knowledge about the peers' network positioning in determining an optimized set of peers. This causes a good properties opposed to the random selection of peers, there is a certain guaranty for a better utilization of the available resources in the network and also optimizing the performance of the P2P network[7].

The solutions for locality problems can be categorized in 3 classes. The first class of algorithm is based on servers which are distributed in the entire network and they are Landmark servers. The property of this server is its reliability. The server is a base for measuring the RTT and hop count number between peers and server. Peers that are near to the server are located close to each other in the overlay network. The problem of this class of algorithm is single of failure, Hotspots and delay in RTT and hop count measurement [12].The second class of algorithm is using IP Prefix. Nodes are located based on their prefix in a same geographical location. The IP of peers can be identified with information about the geographic location of hosts, but there are some limitations in mapping IP to locations. It uses IP to geographic mapping techniques to examine the geographic properties of multiple destinations within a single prefix. The disadvantage of this algorithm is that sometimes similar prefixes do not exist in the same geographic zone. In addition, IPs is usually hashed in order to place in distributed hash table, so it is difficult to identify near or far nodes[13]. In the third classes there is a virtual dimensional coordinate space such as calculating the distance between the coordinates of two hosts and then predicting the round trip latency between the hosts accurately. According to the first algorithm, a synthesis coordinate assigning to each host rely on minimizing the communication latency between two hosts with the knowledge of RTTs in the network. The algorithms use a Euclidean function. This procedure runs until the network achieves a stable state and this is a concern to be significant in this algorithms[14].There is another new algorithm for improving locality in P2P network with using cube connected cycle, in this algorithm the time is as a metric for measuring locality. The new node finds the nearest node in the overlay and then connects to this node. This algorithm has some

[1]Denial Service of attack

important advantages such as: it can be applied on the most of networks regardless of its search and other algorithm. This algorithm is completely distributed[9].

3. PROPOSED ALGORITHM

The Diamond tree is a combination of four binary tree structures, pairs of which are connected by their leaf nodes. Two leaf nodes with the same parent node are connected together. There are four interface nodes; each interface node is connected to the root node of each binary tree. The good properties of the Diamond network is that it is planar, top-down and left to right symmetric, the degree of all nodes is four and it is constant by increasing the size of the network. In addition, there are more than one path between two nodes in the Diamond tree, which are as alternative paths used when a link or a node fails. The value *L* is equal to the rebmun of the tree levels, and is an odd integer value greater than five. There is a relation between the total number of nodes,N and eulav eht fo L, as shown bellow[25].

$$N = 3 * 2^{\frac{L-1}{2}} \quad (1)$$

[25]

The numbering of the Diamond network is an odd-even schema[25]. In Figure 3, a diamond tree with 12 nodes is shown. Because of its good properties, we have chosen Diamond tree as overlay topology for implementing the locality algorithm.

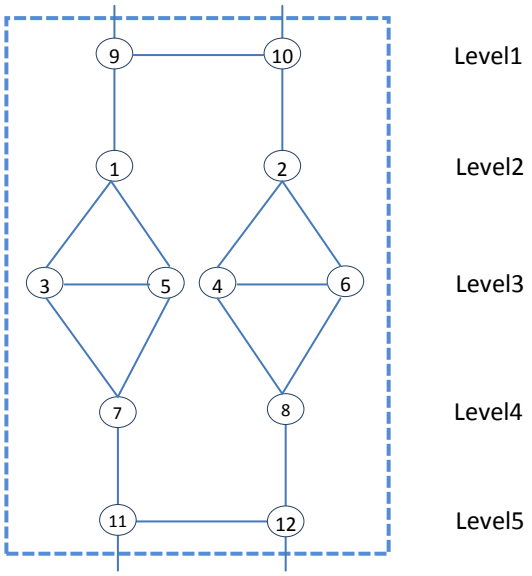

Figure 3. Diamond Graph With 12 Nodes [24]

As we mentioned earlier, all of the three types of exiting locality algorithms have problems such as single point of failure in the landmark servers, the lack of identifying a precise coordinate for a node in the network and possibility of not achieving a stable state for the network. Thus the new algorithm needs to be distributed without any server, has a certain address assigning method, and has a property to attain a stable state. We also use Diamond tree as a main graph in the overlay network. With respect to these properties we show that locality provision is simple and guaranteed.

In this algorithm, we want to improve locality by using RTT and hop count as locality measure. For better evaluation of locality, we introduce function *S* related to RTT and hop count. We use SPSS[26] software to identify function *S*. In this paper, we use the dataset provided by Washington University that contain the RTT delay and the hop count number of 72000 websites[27]. These are the datasets generated as part of the measurements gathered by iPlane[27]. Each dataset is accompanied by the methodology used to generate it and a description of its format. iPlane performs trace routes from several vantage points daily to map the Internet's topology. Our dataset is pointer to the trace route logs gathered over the past days [25]. With these dataset, we calculate the function *S*; to do this we should select the parameter of the function and enter the data in SSPS. Then some types of models in SSPS should be fitted on the data. The factor of the model determination will be obtained from the figures and tables of models that were fitted to the model in the SSPS. The factor of determination is a criterion for identifying the significance of the model. The results for the factor of the model determination showed that it is necessary to fit combination of models to the data. With reputation of the fitness process, the final model will be obtained, which has the highest factor determination. Our algorithm models the distance (S) as follow:

$$S(rtt,hop)=e\text{^}((a * rtt\text{^}2)) + e\text{^}((b/hope)) + e * rtt + f * rtt\text{^}2 + i/rtt + e\text{^}((j/rtt)) \quad (2)$$

Where we estimated good values for required parameters such as *a,b,I,e,j,f*. For this estimation, we used data set of 100 websites and we count the time and hop count by using a trace routes service.

In this algorithm, when a new node is joining to the network, it sends a broadcast message with TTL value equal to one, and waits for first three nearest node responses. If the responses are lower than three then the broadcast message will be repeated with the TTL increased by one. This operation will be continued until three nearest nodes reply new node's request or the TTL reach 11 which is computed from the proposed data set. We select the number 11 as a limitation for maximum hop count because without this limitation the procedure of finding the new node makes a loop. So the new node should be placed near these three nodes. After placing the new node, it should update the neighbor nodes. We will update the smallest diamond that contains all three found nodes. For updating reason, firstly the *S* parameter is calculated, by the new node, between itself and all other nodes that are in a one hope away. The node that has smaller *S* will be swapped with near node. This procedure will continue until each node replaces 3 times. Above procedure will occur whenever each node gets new address. We select state of each node equal to 3, with try and error, for guarantying the locality. We repeat this replacement 3 times for each node to guaranty the locality.

4. SIMULATION AND EXPERIMENTAL RESULTS

We perform experiments through simulation to test our algorithm on a wide range of network. Our selected simulator is PlanetSim[28]. It is a good overlay network simulation written in Java. It aims at providing easy transition from simulation code to prototypes by providing wrapper scripts in their simulator. PlanetSim does include a trivial peer-to-peer simulation. This Platform is an event-based simulator, which uses a time stepped method. This means there is a central clock that controls the events in a simulation. Planetsim has a 3 layered design; there are application, overlay and network layers. The overlay services are built in the application Layer with using the standard Common API. The overlay layer collects proximity information for other nodes that send the

request for getting information from the network layer. An overlay can run on top of the different networks using different underlying protocols. In the Application layer there are Application, Endpoint and Messages. The most important object in the overlay network is node; each node contains incoming and outgoing message queues and methods for sending, receiving and processing messages. The network layer acts as the main layer[28, 29].Simulations are expected to scale from 200 to 10,000 nodes.

We analyze the results obtained from our simulation on a wide range of network, and assess the topologies like binary Tree, diamond tree and hypercube and make a comparison with chord network for better evaluation. Simulations were executed on a computer with windows Xp operating system, a 2.6GHZ processor, 3GB of Ram. We have calculated two parameters in our simulation; the first one is the total time. The total time is the sum of creation time and the simulation time of the network. The second parameter is the traffic. The traffic is introduced as message numbers which are transmitted between nodes. Table 1 shows the traffic results for 10,000 nodes. Binary tree, Diamond tree, Hypercube and Chord are as the overlay topology in our simulation. Traffic is calculated before and after providing the proposed algorithm. The traffic before providing the locality contains the traffic for broadcasting a message to the network for finding a close empty place for new node, but after providing the locality traffic containing the traffic caused in broadcast and whom that caused from the operation of updating the positions after sitting the new node in the overlay network. As the results show, traffic is considerably high in Chord network and also in Binary tree. In chord network, the message travels a ring path to get the destination, the message passes all the nodes in the path to get the destination node while the diameter is so long. So this causes much traffic. Chord network is a structured network without locality. In Binary tree there is just one path between nodes, thus the path is long and the root node should send message to all nodes in the network to find a place for the new node. In comparison fo diamond tree and hypercube results, we found that the traffic has seulav ralimis resulting some similarities in those topologies, but the node adjacent to the hypercube network will increase by increasing the network size. Thus, the traffic in hypercube is much more than the traffic in diamond tree.

Table 1, Network traffic before and after providing locality algorithms

	Number of packets for network size 10000			
	Binary Tree	**Diamond Tree**	**Hypercube**	**Chord**
Without Locality	72,888,177	58,765,278	59,090,668	279,281,108
With Locality	73,406,783	60,058,034	60,679,389	

In regard to Table1, there is a significant difference between chord traffic and other overlays due to their difference in routing algorithm. In chord, the routing path is a ring and the message should traverse the entire nodes in the path to get the destination node, so the traffic and time will increase. Therefore, chord cannot support the best routing and it must send too many messages. In addition to the chord, every node can only communicate with its successor and predecessor. All messages may travel arbitrarily long distances in the internet in each routing hop.

These analyses are presented in Figure 4. The results show increased traffic for bigger network size. Also the figure has a non-uniform shape because of real dataset.

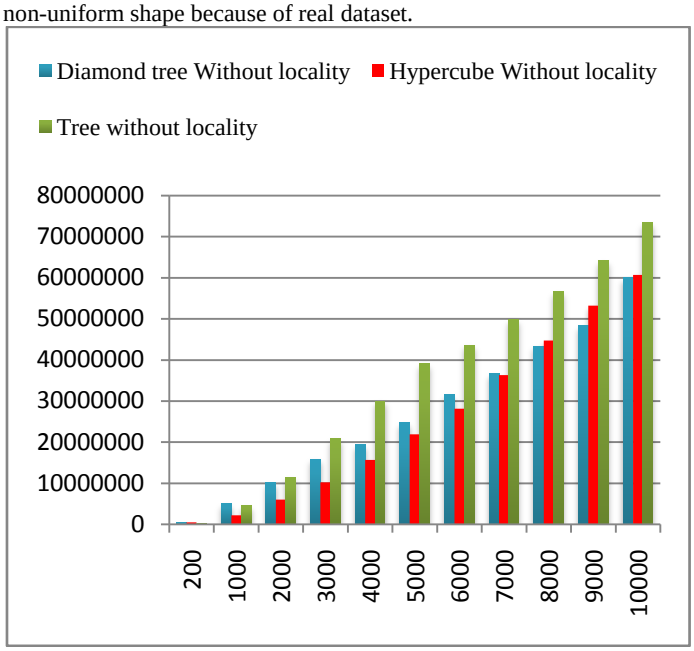

Figure 4, Traffic results before providing locality algorithm

For comparison we have presented the traffic after providing the locality algorithm in Figure5.

The results show that there is a little overhead for traffic after providing the locality algorithm and both figures have an ascending form.

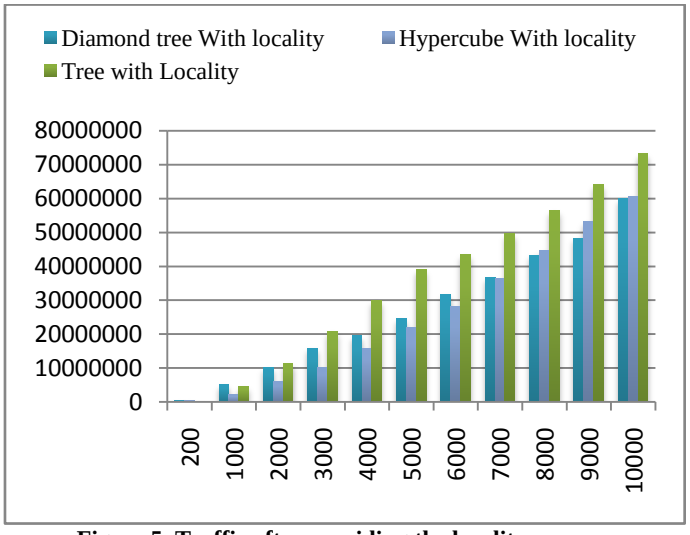

Figure 5, Traffic after providing the locality

The results for computing the time is presented in Table 2. Again, we have the time before and after providing the locality algorithm. The time before providing locality contains the broadcast time and the time to find a place for new node to sit in the overlay network, but the time after providing the locality contains previous time and also the time of address updating after sitting the new node in the overlay network. The network size is 10000 nodes.

Table 2. Time results before and after providing the locality

	Tree	Diamond Tree	Hypercube	Chord
Before locality	381.312 (ms)	365.515 (ms)	361.546 (ms)	493.953 (ms)
After locality	480.981(ms)	469.515(ms)	412,184(ms)	493.953(ms)

All analysis is the same as traffic results showed but one difference is obvious. In hypercube topology, the number of neighbor nodes will increase when the network size increases and this results in to take less time for finding a place for the new node. In comparison to time before and after locality we find little overhead, and this is a good property for our algorithm.

In Figure 6 we expand the simulation with 200 to 10000 nodes before providing locality for showing the time results.

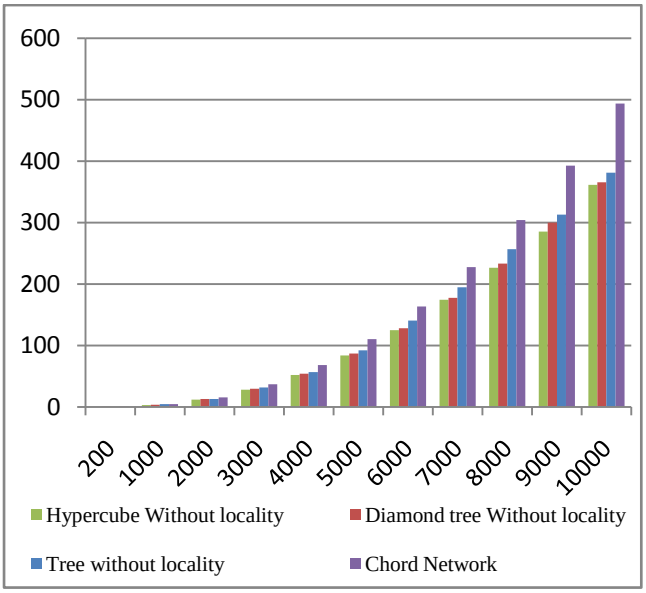

Figure 6, Connection Time before locality

The results show that the time will increase by increasing the number of nodes in the network. Chord has the highest time and the time for hypercube is less than all. In Figure 7 the connection time after providing locality is appeared.

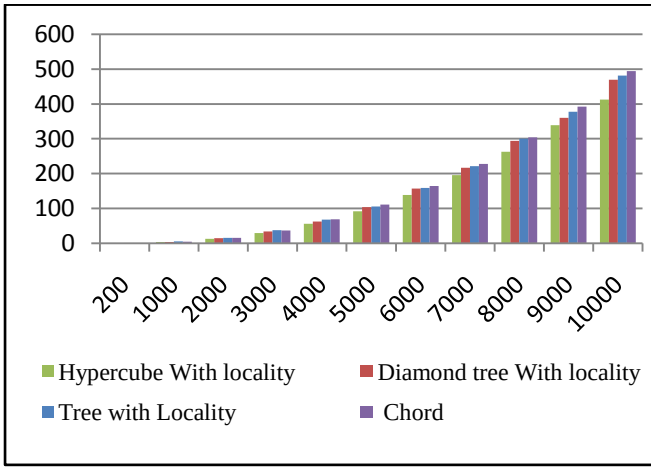

Figure 7, Connection time after providing locality

Again the time for Chord is higher than the rest, and hypercube has less time for joining the overlay network.

Some of the advantages of proposed algorithm are mentioned here: This algorithm is fully distributed because search and join will happen without any need to a central server. In addition, joining or leaving a node in this algorithm does not need huge changes in the network structure, because all nodes situation will be updated after joining or leaving the node. The metrics for locality in this algorithm are round trip time and hop count, but in other algorithms [7, 8] there is just one parameter like time as a locality metric. Also, more important advantage of this algorithm is that we use real datasets [25] for broadcast procedure and for certain computations. Another advantage is that, this algorithm is applicable on both tree and hypercube topologies.

5. CONCLUSIONS

The proposed algorithm improves locality with distributed method. In addition, the good properties of the algorithm are low overhead and good network performance due to supply locality. Moreover, by applying this algorithm on hypercube; we have proved that this algorithm can be useful in all applicable networks that utilize tree as an overlay. The simulation results showed that diamond tree has some good characteristics against the binary tree. Diamond tree is symmetric and regular, there is more routing path between two nodes and the degree of node is constant, so it causes less traffic and time in comparison to Binary tree. Also, hypercube is similar to diamond tree and the results are very close. The results showed that traffic increases and the time reduce in comparison to diamond tree. In hypercube, the neighbors will increase when the network size increases, so more messages will be sent but it is guaranteed to find a location for new node in a less time in comparison to the diamond, because it has more neighbors.

We are intrested in working on load balancing of each node in the overlay network. Also it is necessary to find the effect of other locality metrics like physical location and bandwith on network traffic and costs.

6. ACKNOWLEDGMENT

The authors would like to thank Mr. M. Tahery for his insightful comments in implementation step and Mr.A.Hadighi for the language editing of the paper.

7. REFRENCES

[1] L. Napster, "Napster," *URL: http://www. napster. com*, 2001.

[2] Y. Wang, *et al.*, "Analyzing the characteristics of gnutella overlays," in *Fourth International Conference on Information Technology, ITNG '07*, 2007, pp. 1095-1100.

[3] R. A. Ferreira, *et al.*, "Locality in structured peer-to-peer networks," *Journal of Parallel and Distributed Computing*, vol. 66, pp. 257-273, 2006.

[4] V. Aggarwal, *et al.*, "Improving user and ISP experience through ISP-aided P2P locality," in *INFOCOM Workshops*, 2008, pp. 1-6.

[5] L. G. A. Sung, *et al.*, "A survey of data management in peer-to-peer systems," *School of Computer Science, University of Waterloo*, 2005.

[6] O. Abboud, *et al.*, "Underlay awareness in P2P systems: Techniques and challenges," in *IEEE International*

Symposium on Parallel & Distributed Processing, IPDPS 2009., 2009, pp. 1-8.

[7] C. Miers, *et al.*, "A taxonomy for locality algorithms on peer-to-peer networks," *Latin America Transactions, IEEE* vol. 8, pp. 323 - 331, 2010.

[8] M. Gharib, *et al.*, "A Novel Method for Supporting Locality in Peer-to-Peer Overlays Using Hypercube Topology," in *International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, 2010, pp. 391-395.

[9] M. Gharib, *et al.*, "The Effect of Using Cube Connected Cycle for Improving Locality Awareness in Peer-to-Peer Networks," in *12th International Conference on Computer Modelling and Simulation (UKSim),*, 2010, pp. 491-496.

[10] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Middleware*, Heidelberg, Germany, 2001, pp. 329-350.

[11] B. Y. Zhao, *et al.*, "Tapestry: A resilient global-scale overlay for service deployment," *IEEE Journal on elected Areas in Communications,*, vol. 22, pp. 41-53, 2004.

[12] P. F. SylviaRatnasamy, *et al.*, "A Scalable, Content-Addressable Network," *SIGCOMM*, vol. 1, pp. 161–172, 2001.

[13] M. J. Freedman, *et al.*, "Geographic locality of IP prefixes," in *the 5th ACM SIGCOMM conference on Internet Measurement*, 2005, pp. 3-17.

[14] F. Dabek, *et al.*, "Vivaldi: A decentralized network coordinate system," in *Proceedings of the 2004 conference on Applications, technologies, architectures,SIGCOMM '04*, 2004, pp. 15-26.

[15] D. Doval and D. O'Mahony, "Overlay networks: A scalable alternative for P2P," *Internet Computing, IEEE*, vol. 7, pp. 79-82, 2003.

[16] K. Dhara, *et al.*, "Overview of Structured Peer-to-Peer Overlay Algorithms," *Handbook of Peer-to-Peer Networking*, pp. 223-256, 2010.

[17] D. Cvetkovic, *et al.*, "Graphs for small multiprocessor interconnection networks," *Applied Mathematics and Computation*, vol. 217, pp. 2468-2480, 2010.

[18] J. Duato, *et al.*, *Interconnection networks: An engineering approach*: Morgan Kaufmann, 2003.

[19] T. Feng, "A survey of interconnection networks," *Computer*, vol. 14, pp. 12-27, 1981.

[20] I. Stoica, *et al.*, "Chord: a scalable peer-to-peer lookup protocol for internet applications," *Networking, IEEE/ACM Transactions on*, vol. 11, pp. 17-32, 2003.

[21] Y. Liu, *et al.*, "Location-aware topology matching in P2P systems," in *in Proc INFOCOM*, 2004, pp. 2220-2230

[22] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," *Peer-to-Peer Systems*, vol. 2429, pp. 53-65, 2002.

[23] D. Malkhi, *et al.*, "Viceroy: A scalable and dynamic emulation of the butterfly," in *Proceedings of the twenty-first annual symposium on Principles of distributed computing,PODC '02*, 2002, pp. 183-192.

[24] K. Fujii and S. Goto, "Correlation between hop count and packet transfer time," *APAN/IWS2000, February*, 2000.

[25] N. S. Woo and A. Agrawala, "A symmetric tree structure interconnection network and its message traffic," *Computers, IEEE Transactions on*, vol. 100, pp. 765-769, 1985.

[26] (22 Feb). Available: http://www-01.ibm.com/software/analytics/spss/

[27] (Nov 23). Available: http://iplane.cs.washington.edu/data/data.html

[28] J. P. Ahulló and P. G. López, "PlanetSim: An extensible framework for overlay network and services simulations," in *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops,Simutools '08*, 2008, pp. 660-670.

[29] P. García, *et al.*, "Planetsim: A new overlay network simulation framework," *Software Engineering and Middleware*, vol. 3437, pp. 123-136, 2005.