AENSI Journals

**Australian Journal of Basic and Applied Sciences**

ISSN:1991-8178

Journal home page: www.ajbasweb.com

# A Tree Based Routing DHT to Evaluate MAXDISJOINT for MULTIHOP Networks

[1]Dr. J Vellingiri, [2]Kumarasamy Saravanan and [3]Dr. Ramasamy Asokan

*[1]Professor, Department of Computer Science and Engineering, Kongunadu College Of Engineering And Technology, Anna University, Tamilnadu*
*[2]Assistant Professor, Department of Computer Science and Engineering, Erode Sengunthar Engineering College, Erode, Anna University, Tamilnadu*
*[3]Professor and Principal, Department of Electronics and Communication Engineering, Kongunadu College Of Engineering and Technology, Anna University, Tamilnadu*

**ARTICLE INFO**

***Article history:***
*Received 19 August 2014*
*Received in revised form*
*19 September 2014*
*Accepted 29 September 2014*
*Available online 3 December 2014*

***Keywords:***
*DHT,Tree based routing, Peer to peer network*

**ABSTRACT**

Distributed hash tables (DHTs) share storage and routing responsibility among all nodes in a peer-to-peer network. These networks have bounded path length unlike unstructured networks. Unfortunately, nodes can deny access to keys or misroute lookups. Tree-based routing DHTs, a replica placement that creates route diversity for these DHTs are characterized. The placement creates disjoint routes and finds the replication degree necessary to produce a desired number of disjoint routes is proved. Using simulations of Pastry a tree-based routing DHT, the impact of MAXDISJOINT on routing robustness compared to other placements when nodes are compromised at random or in a contiguous run is evaluated. The route diversity mechanism can successfully route messages to a correct replica even with a quarter of the nodes in the system compromised at random. A family of replica query strategies that can trade off response time and system load and a hybrid query strategy that keeps response time low without producing too high a load. The proposed work can be used in many applications like bit torrent and skype, since this system requires data's form different region and it must be routed properly to the user.



**To Cite This Article:** Dr. J Vellingiri, Kumarasamy Saravanan and Dr. Ramasamy Asokan., A Tree Based Routing DHT to Evaluate MAXDISJOINT for MULTIHOP Networks. ***Aust. J. Basic & Appl. Sci.,*** *8(18): 356-360, 2014*

## INTRODUCTION

PEER-TO-PEER (p2p) networks are a popular substrate for building distributed applications because of their efficiency, Scalability, resilience to failure, and ability to self organize. The p2p architecture relies on the distribution of responsibility among hundreds of thousands, if not millions, of nodes in the network. Therefore, if a small set of nodes failto serve data objects, properly maintain routing information, or route messages, the integrity of a very large-scale system may be compromised. The efficiency of lookups has become a central focus ofp2p design because many popular applications, like name resolution, publish-subscribe, and IP communication, rely on a lookup service as core functionality.

***Sit and Morris classify attacks on DHTs into three categories:***

1. Storage and retrieval attacks, which target the manner in which peers manage data items;
2. Routing attacks, which target the manner in which peers route messages; and
3. Miscellaneous attacks, which target other aspects of the system, such as admission control or the underlying network routing service.

### *2 Related Works:*

To place our work in context, it discuss related work on replica placement, peer-to-peer routing security, and general peer-to-peer security issues.

### *2.1 Replica Placement:*

Replica placement has long been studied in the realm of distributed computing. Many studies have compared the performance of different placement schemes in terms of quality of service, availability, and time to recovery in different types of server less systems. The first DHT-based replication schemes were only

**Corresponding Author:** Dr. J Vellingiri, Professor, Department of Computer Science and Engineering, Kongunadu College Of Engineering And Technology, Anna University, Tamilnadu

concerned with availability and thus local replication, i.e., replicas placed close to the master copy in the ID space, was used.

### *2.2 Peer-to-Peer Routing Security:*

A number of works that look to improve routing security are centered around the notion of route diversity. For example, Artigas *et al*. propose Cyclone, equivalence based routing scheme deployed over an existing structured peer-to-peer overlay Independent lookup paths are created by routing across different equivalence classes. Since the paths are independent and do not differ in the destination, Cyclone does not naturally mitigate the effects of storage and retrieval attacks. Furthermore, each peer is required to maintain additional routing information, which incurs overhead. In contrast, our replica placement creates disjoint routes without requiring any additional routing state or modifying the underlying routing scheme. Messages are split in two, encrypted, and sent to the destination across diverse paths. Route diversity is created by routing messages through the routing table entries that minimize route overlap. The appropriate entries are chosen using empirical results that depend on the network size. routing table entries does not exceed the fraction of compromised nodes in the network.

### *3 Distributed Hash Tables With Tree-Based Routing:*

Distributed hash tables are often The geometry impacts neighbor and route selection, which can have an impact on flexibility, resilience, and proximity performance as studied in Although we use the term "tree-based routing," we are not referring to the geometry, but the routing algorithm. Tree-based routing algorithms have specific properties that we define herein.

### *3.1 Tree-Based Routing DHTs:*

Consider a DHT with an ordered id space I with size N ¼ jIj and a branching factor B such that logB N is integral. The branching factor is used by each node to construct its routing table. The routing table of a node u has the following properties:

1. The node u partitions the entire id space into contiguous segments and selects one node from each id segment to include in its routing table. The partitioning is performed as follows:

a. The node u partitions the id space into B equal size contiguous parts.

b. Of the B id segments, u selects the id segment I0of which it is a member.

c. Steps 1a and 1b are repeated to repartition I0 until parts of size one are created. The part that consists of the node u is discarded..

d. At the end of the partitioning process, u will have created ðB _ 1Þ logB N contiguous parts of the id space, with sizes N

B ; N

B2 ; . . . ; N

BðlogB NÞ_1 ; and 1,

and with B _ 1 parts of each size.

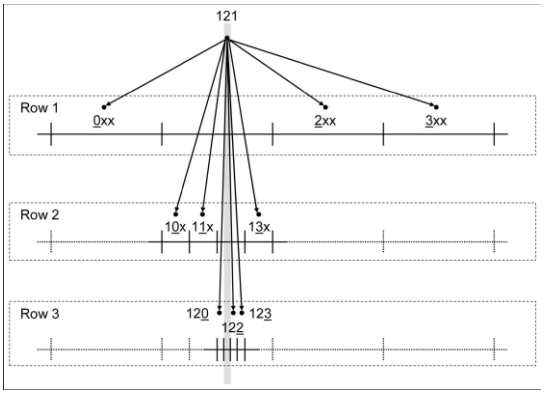


**Fig. 1:** Pastry routing table structure.

### *4 The Maxdisjoint Replica Placement:*

Using Pastry as an example in the following sections, we will demonstrate how the properties of tree-based routing can be used to construct a replica placement that creates disjoint routes.

### *4.1 Intuition behind MaxDisjoint Placement:*

To create route diversity in Pastry via replica placement, it is necessary to place replicas such that a given replica set will use a diverse set of routing table entries for every possible source node. We use an example to provide the necessary intuition. Consider a Pastry ring with id space of size 64 and prefix matching in base-4 digits. We show the Pastry routing table structure graphically for a hypothetical ode 121 in Fig. 1.Suppose we wish to replicate an object with id 101 in this Pastry ring. Node 121 routes to this object through the routing table entry marked "10x" in Fig. 1. Suppose we replicate the object with the id 111 to target the routing table entry "11x" in the example. This approach creates an additional disjoint route for any lookups for object 101 originating at node 121. One route is forwarded via the entry "10x" and the other is forwarded via "11x." However, consider another source node 221. This node routes to the object 101 and 111 through the same entry marked "1xx"and, therefore, does not gain an additional disjoint route.

### *4.2 Evaluation of Disjoint Routes:*

The desired number of disjoint routes d is one of the tunable inputs to the MAXDISJOINT algorithm. Controlling the level of fault tolerance is an important design parameter and, therefore, d is a very useful input. In this section, we will formally prove that the algorithm indeed creates d disjoint routes in support of the intuition provided in earlier sections assume that routing is performed in an identifier space of size N with branching factor B. All of our analytical results are proved within the context of a full DHT, but we will show, through experimentation, that these properties hold even in sparsely populated DHTs.

## *5 Experiments:*

To confirm that our analytical results hold for sparsely populated DHTs or DHTs with clustered id spaces conducted a series of experiments through simulation. First, to measure the impact of replica placement on routing robustness, we consider the number of disjoint routes created for several replica placements. Furthermore, we find the probability of lookup success when nodes are compromised at random or in a run of several nodes. Second, it considers a heuristic used for creating route diversity in [3] that we call neighbor set routing. It measure its ability to create route diversity and the impact on the Probability of lookup success. Finally, having shown that replica placement can improve routing robustness, we consider the impact of parallel queries on response time.

### *1. Simulation Environment:*

All of our experiments it performed using a Java-based simulator it developed. It are able to model Chord and Pastry routing, uniform and clustered node distributions, and two adversarial models. Nodes may be compromised at random with some failure probability or in a run of several nodes. The simulator is extensible to model other DHT implementations, node distributions, and adversarial models. Each data point in our results is representative of over 100,000 lookups performed in 10 different random node distributions. It simulate a lookup

**Table 1:** Pastry Simulation Parameters.

| Symbol | Parameter |
|---|---|
| $N$ | Identifier Space Size |
| $n$ | Number of Nodes |
| $B$ | Branching Factor ($2^b$ for Pastry) |
| $r$ | Replication Degree |
| $f$ | Fraction of Compromised Nodes |
| $C$ | Number of Clusters |
| $\sigma$ | Cluster Width (standard deviation) |

by randomly selecting an uncompromised query node and a target key. In reality, if the query node is compromised, it can affect the outcome of the lookup. However, if it assume that data items are self verifying, a compromised query node can only cause the client to time out and select another query node. It deem a lookup successful if there exists a route consisting of only uncompromised nodes from the query node to any replica of the target key. If all routes from the query node to the replica set contain a

Compromised node, then the lookup is deemed to fail. For most experiments, it is sufficient to compute routes in the network using the appropriate routing protocol. However, to measure response time, it was necessary to modify our simulator to be event driven. The remaining simulation parameters are summarized in Table 1.

***2. Replica Placements Considered:***

Consider four replica placements: MAXDISJOINT, neighbor set, where replicas are placed at distinct nodes in the neighborhood of the root random, where replica locations are uniformly distributed; and spaced, where replica identifiers are separated by a uniform spacing s. It is worth noting that two replicas may be placed at the same node with spaced replication. In the case of neighbor set placement, some implementations may attempt to reduce load by querying the entire replica set with a single lookup message. This naturally creates route overlap; for a fair assessment, it dispatch a separate lookup for each replica in the replica set.

***5.1 Measurement of Disjoint Routes:***

First, to verify the correctness of our analysis, it measures the average number of disjoint routes created using the considered replica placements. These results are depicted. For the parameters tested, MAXDISJOINT placement outperforms the other placements in creating disjoint routes. The neighbor set placement does not create a significant number of disjoint routes as expected. Routes toward keys that are close to each other in the identifier space are likely to converge. Since the neighbor set placement clusters replicas, an adversary can eliminate the entire replica set if he can compromise a node common to all routes destined for that neighborhood. By increasing the route diversity, it eliminates these single points of vulnerability. The performance of the spaced placement scheme is dependent on the spacing chosen. If the spacing is small, then the spaced placement is very similar to the neighbor set placement.

***5.2 Replica Placement and Neighbor Set Routing:***

It believes replica placement is an efficient way of creating disjoint routes because it does not require significant Modification to the underlying DHT routing scheme. Other works have considered reusing the existing routing information to create route diversity. Although these approaches do not create provably disjoint routes, it believes there is value in introducing some additional form of route diversity. Furthermore, it believes that these techniques could be combined with our replica placement to provide additional benefit. To evaluate this claim, it considers the route diversity technique introduced by Castro *et al*l which it call neighbor set routing. Castro *et al*. use neighbor set routing to find diverse routes toward the neighborhood of a key. To create diverse routes, messages are routed via the neighbors of the source node. This is depicted graphically in Fig. 2. Castro *et al*.claim that this technique is sufficient in the case when Replicas are distributed uniformly over the identifier space, as in CAN and Tapestry. It was consider the ability of neighbor set routing to create diverse routes to a replica to enhance the routing robustness of MAXDISJOINT. To evaluate the relative impact of replica placement and neighbor set routing, it was consider four scenarios:

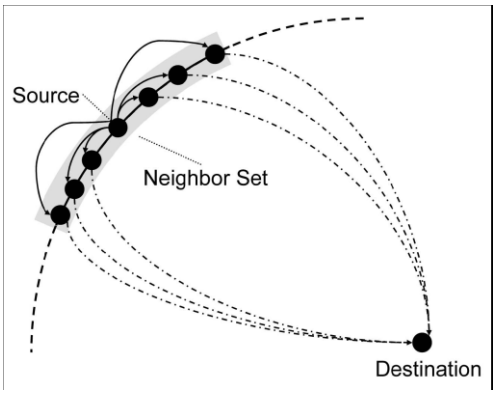


**Fig. 2:** Graphical depiction of neighbor set routing.

***Scenarios:***

1. Neither replication nor neighbor set routing,
2. Only neighbor set routing through eight neighbors,
3. Only MAXDISJOINT placement with eight replicas,
4.Both neighbor set routing and MAXDISJOINT placement.

These results are depicted in Fig. 2.

Both MAXDISJOINT placement and neighbor set routing can have a positive impact on the probability of lookup success. At best, neighbor set routing can create independent routes, since all paths will converge at the destination. If the destination node is compromised, no amount of route diversity can increase the probability of lookup success. Nonetheless, the added route diversity that neighbor set routing provides can benefit MAXDISJOINT placement, especially with a large fraction of compromised nodes.

***Conclusion And Future Work:***

In this paper characterized a class of DHTs, which employ a tree-based routing scheme. It proved that for every DHT of this class there exists a replica placement that can create a provable number of disjoint routes, replica placement that creates disjoint routes in a full distributed hash table that employs a tree based routing scheme. Through simulation, it should that this placement creates disjoint routes effectively in DHTs that are sparsely populated. In addition, MAXDISJOINT creates disjoint routes without modification of the underlying routing scheme, considered some of the practical limitations of using MAXDISJOINT in a real implementation; that is, it evaluated the choice of the replica query strategy on response time. Of particular concern was the impact of a parallel strategy on the system load and, as a result, the response time. It observed that the parallel strategy is adversely affected by an increase in the lookup rate; however, it is resilient to changes in the fraction of nodes compromised.

Byzantine-faulttolerant replication and requires that clients retrieve multiple replicas and perform a voting operation to ensure correctness. It believes that both one-hop DHTs and DHTs with multilevel hierarchies will be used in the future for different types of applications. Applications with an extremely large user base that are not very latency sensitive in the lookup phase, e.g., Bit Torrent and Skype, will continue to prefer the better scalability of multilevel hierarchies. Also, the availability of open-source software such as Free Pastry makes it a popular choice for research use and for rapid development and deployment of new p2p applications. Applications that are latency sensitive for lookups and do not need to scale to massive numbers of clients will prefer One-hop DHT technology.

## REFERENCES

Paxton, C.G., R. Rajaraman and A. Richa, 1997. Accessing Nearby Copies of Replicated Objects in a Distributed Environment, Proc.ACM Symp. Parallel Algorithms and Architectures (SPAA ’97), 311-320.

Chen, Y., R.H. Katz and J. Kubiatowicz, 2002. Dynamic Replica Placement for Scalable Content Delivery,” Proc. Int’l Workshop Peer-to-Peer Systems (IPTPS ’02), 306-318.

Gupta, A., B. Liskov and R. Rodrigues, 2004. Efficient Routing for Peerto-Peer Overlays,” Proc. Conf. Symp. Networked Systems Design and Implementation (NSDI ’04).

Harvesf, C. and D.M. Blough, 2006. The Effect of Replica Placement on Routing Robustness in Distributed Hash Tables,” Proc. IEEE Int’lConf. Peer-to-Peer Computing (P2P ’06), 57-66.

Singh, A., M. Castro, P. Druschel and A. Rowstron, 2004. Defendingagainst Eclipse Attacks on Overlay Networks,” Proc. ACMSIGOPS, 115-120.

Sit, E. and R. Morris, 2002. Security Considerations for Peer-to-Peer Distributed Hash Tables,” Proc. Int’l Workshop Peer-to-Peer Systems (IPTPS ’02), 261-269.

Srivatsa, M. and L. Liu, 2004. Vulnerabilities and Security Threats in Structured Peer-to-Peer Systems: A Quantitative Analysis,” Proc.IEEE Ann. Computer Security Applications Conf. (ACSAC ’04), 252-261.

Ratnasamy, S., P. Francis, M. Handley, R. Karp, and S.Schenker, 2001. A Scalable Content-Addressable Network,” Proc.ACM SIGCOMM ’01, 161-172.