A Tree based Routing DHT to Evaluate Maxdisjoint for Multihop Networks

M.Lavanyaprabha
M.E., Department of Computer Science and Engineering,
Mahendra Institute of Technology,
Mahendhirapuri, Mallasamudram West, Tiruchengode,Tamil Nadu 637503.

Dr.P.Balakumar
Associate Professor
Department of Computer Science and Engineering,
Mahendra Institute of Technology,
Mahendhirapuri, Mallasamudram West, Tiruchengode, Tamil Nadu 637503.

Abstract- Distributed hash tables (DHTs) share storage and routing responsibility among all nodes in a distributed network, which possess bounded path length and 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. The evaluation of impact of MAXDISJOINT, using simulations of pastry, with comparison to other various positions when nodes are randomly compromised in a contiguous run is performed. The route diversity mechanism which cans effectively path messages to a open 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 approach that keeps comeback 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 from different region and it must be routed properly to the user.

Keywords – DHT,Routing,Replica Placement.

I. INTRODUCTION

PEER-TO-PEER (p2p) networks are a popular substrate for building distributed applications because of their effectiveness, flexibility to failure, and ability to self organize. The p2p network architecture even smaller network of nodes properly maintain routing information or route messages, when they encounter failure in securing data objects, the integrity of a large scale system may be compromised. The overall efficiency of lookups has become a major focus of p2p design because many popular applications, like Skype, publish-subscribe information system and IP communication, rely on a lookup service as core functionality.
Sit and Morris classify attacks on DHTs into three categories:

1. Storage attacks, which target the manner in which peers manage data items.
2. Routing attacks, which target the manner in which peers route messages.
3. Miscellaneous attacks, that affect factors like admission control or the underlying network routing service.

II. RELATED WORKS

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

2.1 Replica Placement

Replica placement has long been studied in the area of distributed environment. Studies have compare performance of placement schemes in terms of quality of availability, and time of recovery in different types of server less systems. The first DHT-based replication schemes were only concerned with ease of use and thus local replication, i.e., replicas position at close to the master copy in the ID space, was used.

2.2 Peer-to-Peer routing protection

A number of works with the aim of look to improve routing security are center on the notion of route diversity mechanism. 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 resemblance classes. Since the paths are self-sufficient and do not differ in the destination, Cyclone does not effects of storage and retrieval attacks. Furthermore, each p2p network is required to maintain additional information about routing table, which incurs overhead. In contrast, our replica place creates disjoint routes withoutneed of any additional routing state or modifying the underlying routing scheme. Communication takes place Messages are split in two, encrypted, and sent to the destination across to the diverse paths. Route diversity is created by routing messages through the routing table entries that minimize route overlie. The suitable entries are chosen using experimental results that depend on the network size. Routing table entries does not exceed the little of compromised nodes in the network.

III. DISTRIBUTED HASH TABLES WITH TREE-BASED ROUTING

Distributed hash tables are often the geometry impacts neighbor set route selection, which can have an impact on better performance, tolerance to failure, and proximity performance as studied in although we use the term tree-based routing, to refer the geometry, Tree-based routing algorithms have specific properties that we define herein.

3.1 Tree-Based Routing DHTs

Consider a DHT with a structured 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 has the following properties:
1. The node u partitions the entire id space into contiguous allocation segments and selects one node from each id subdivision 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 applied again to repartition I0 until parts of size one be created. The element 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.

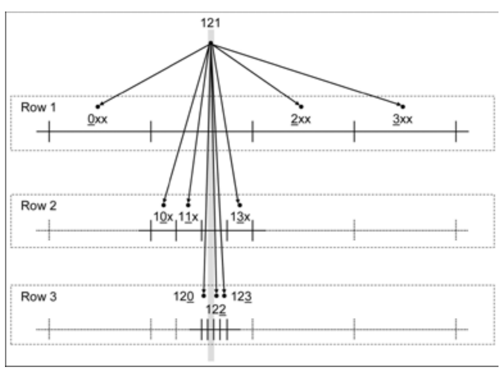

Fig 1. Node Transmission in Routing Table

IV. THE MAXDISJOINT REPLICA PLACEMENT

Using Pastry as an example in the following sections, we will show 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 need to placed replicas such that a given replica set will use a diverse group of routing table entries for possible source nodes. Using an example to provide the necessary perception. 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 10 xs and the other is forwarded via 11 xs. However, consider another source node 221. This node routes to the purpose 101 and 111 through the same entry marked 1xxand, 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. 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 context of a full DHT routing table, but we will show, through experimentation, that these properties hold even in sparsely populated DHTs.

V. EXPERIMENTS

To confirm that our analytical results hold for lightly populated DHTs or DHTs with clustered id spaces conducted a series of experiments through replication. First, to measure the impact of replica placement on routing robustness, we take into consideration, the number of disjoint routes created for several replica placements. Moreover, 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.

5.1 Simulation Environment:

All of our experiments it performed using a Java-based simulator it developed. They 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 probability of failure or in a run of several nodes. The simulator is extensible to model other DHT routing table implementations. 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

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
σ	Cluster Width (standard deviation)

Table.1Pastry Simulation Parameters

Pastry Simulation Parameters by randomly selecting an uncompromised query node and a fixed point of a key. In real, if the query node is compromised, it can affect the outcome of the lookup. However, if it assumes 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 fixed point of the 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 a suitable 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.

5.2.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 equally 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 try to reduce load by querying the entire replica set with a single lookup message. This naturally creates route overlap; for a fair assessment, it dispatches a separate lookup for each replica in the replica set.

5.3 Measurement of Creation of Disjoint Routes

First, to check 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 nearest neighbor set placement does not create a significant number of disjoint routes as expected. Routes toward fixed keys that are closely related to each other in the identifier space are likely to converge. Since the neighbor set placement clusters replicas, an adversary can exclude 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.

5.4 Replica Placement and Neighbor Set Routing

It believes replica placement is an efficient way of creating more number of disjoint routes because it does not require significant changes 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 efficiency. To evaluate this claim, it considers the route diversity technique introduced by Castro et all which it call neighbor set routing. Castro et al. use neighbor set routing to find diverse routes toward the nearest neighborhood of a fixed point 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 equally 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:

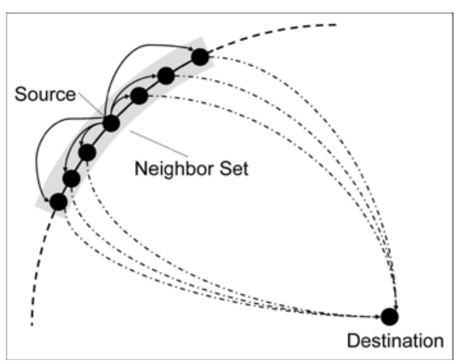

Fig. 2.Graphical depiction of neighbor set routing.

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.

VI. 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 maximum disjoint routes in a fully 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 rarely populated. In addition, MAXDISJOINT creates disjoint routes without modification of the underlying routing scheme, considered some of the practical restrictions 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 tolerant to changes in the fraction of nodes compromised.

Byzantine-fault tolerant replication and requires that clients reclaim multiple replicas and perform a voting operation to ensure correctness. It believes that both one-hop and two-hop DHTs and multilevel hierarchies will be used in the future for a diversified range of applications. These applications with an extremely large user base that are not very latency responsive in the lookup phase, e.g., BitTorrent and Skype, will continue to prefer the better scalability of multilevel hierarchies. Also, the availability of open-source software such as FreePastry 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 huge numbers of clients will prefer One-hop DHT technology.

REFERENCES

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

[2] Y. Chen, R.H. Katz, and J. Kubiatowicz, Dynamic Replica Placement for Scalable Content Delivery, Proc. Intl Workshop Peer-to-Peer Systems (IPTPS 02), pp. 306-318, 2002.

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

[4] C. Harvesf and D.M. Blough, The Effect of Replica Placement on Routing Robustness in Distributed Hash Tables, Proc. IEEE IntlConf. Peer-to-Peer Computing (P2P 06), pp. 57-66, 2006.

[5] A. Singh, M. Castro, P. Druschel, and A. Rowstron, Defendingagainst Eclipse Attacks on Overlay Networks, Proc. ACMSIGOPS 04, pp. 115-120, 2004.

[6] E. Sit and R. Morris, Security Considerations for Peer-to-Peer Distributed Hash Tables, Proc. Intl Workshop Peer-to-Peer Systems (IPTPS 02), pp. 261-269, 2002.

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