# Secured Data Lookup in LDE Based Low Diameter Structured P2P Network

Nick Rahimi, Bidyut Gupta, and Shahram Rahimi
Department of Computer Science
Southern Illinois University
Carbondale, IL, USA
{nick, Bidyut, Rahimi }@cs.siu.edu

## Abstract

In this paper, we have considered a recently reported non-DHT based structured P2P system. The architecture is based on Linear Diophantine Equation (LDE) and it is an interest-based system; it offers very efficient data lookup. In this paper, we have a two-fold objective: first we will incorporate security in the data look-up algorithms designed for the LDE based P2P systems, which is absent in many existing works; second we will present some important simulation results which will show the superiority of our work compared to some important existing works.

***Keywords:*** Structured P2P Network, Linear Diophantine Equation, Security

## 1 Introduction

Peer-to-Peer (P2P) overlay networks are widely used in distributed systems. There are two classes of such networks: unstructured and structured ones. In unstructured systems [2] peers are organized into arbitrary topology. Flooding is usually used for data look up. Problem arising due to frequent peer joining and leaving the system, also known as churn, is handled effectively in unstructured systems. However, it compromises with the efficiency of data query and the much-needed flexibility. Unstructured networks have excessive lookup costs and lookups are not guaranteed. On the other hand, structured overlay networks provide deterministic bounds on data discovery. They provide scalable network overlays based on a distributed data structure which actually supports the deterministic behavior for data lookup. Recent trend in designing structured overlay architectures is the use of distributed hash tables (DHTs) [4], [5], [9]. Such overlay architectures can offer efficient, flexible, and robust service [3] - [5], [7], [8].

However, maintaining DHTs is a complex task and needs substantial amount of effort to handle the problem of churn. So, the major challenge facing such architectures is how to reduce this amount of effort while still providing an efficient data query service. In this direction, there exist several important works, which have considered designing hybrid systems [1], [6], [10] - [12]; their objective being incorporation of the advantages of both structured and unstructured architectures. However, these works have their own pros and cons.

We have earlier presented a new hierarchical P2P network architecture [13] in which at each level of the hierarchy existing networks are all structured. We have used Linear Diophantine Equation (LDE) as the mathematical base to realize the architecture. Note that most structured approaches use DHTs to realize their architectures. Use of Linear Diophantine Equation in designing P2P architecture is a completely new idea. We have explored the many different possible advantages that can be fetched using LDEs [14]; some of these advantages include efficient handling of data look-up, node (peer) join/leave, anonymity, load balancing among peers, to name a few; besides achieving fault-tolerance is reasonably simple. We have shown that the complexity involved in maintaining different data structures is much less than that involved in the maintenance of DHTs. On several points, LDE-based overlay architecture can outperform DHT-based ones. The presented architecture [13] has considered interest-based P2P systems [6], [15]. The rationale behind this choice is that users sharing common interests are likely to share similar contents, and therefore searches for a particular type of content is more efficient if peers likely to store that content type are neighbors.

*Problem Formulation*: In the above architecture, we have designed two efficient data look-up algorithms [14]: one for intra-group look-up query and the other for inter-group look-up query. The first one works inside a cluster while the second one involves more than one cluster. In this paper, we have a two-fold objective: first we will incorporate security in the data look-up algorithms which is essential for any P2P system even though it is absent in many existing works; second we will state some important simulation results which will show the superiority of our work compared to some important existing works.

The paper is organized as follows. In Section 2, we have briefly stated some relevant materials from [13] and [14]. In Section 3 we have presented the secured intra-group and inter-group look-up algorithms and in Section 4, we have presented the results of simulation. Section 5 draws the conclusion.

## 2 Preliminaries

Some of the preliminary ideas of the hierarchical P2P architecture proposed in [13] have been considered in this paper. For the sake of completeness, we reproduce here from [13] some of the notations and the basic idea of using Linear Diophantine equations for generating the logical addresses of the nodes (peers) of the overlay network.

We define a resource as a tuple $<R_i, V>$, where $R_i$ denotes the type of a resource and V is the value of the resource. A resource can have many values. For example, let $R_i$ denote the resource type 'songs' and V' denote a particular singer. Thus $<R_i, V'>$ represents songs (some or all) sung by a particular

singer V'. In the model for interest-based P2P systems [13], we assume that no two peers with the same resource type $R_i$ can have the same tuple; that is, two peers with the same resource type $R_i$ must have tuples $<R_i, V'>$ and $<R_i, V''>$ such that $V' \neq V''$. In [13], the assumption is that no peer can have more than one resource type.

We define the following. Let S be the set of all peers in a peer-to-peer system. Then $S = \{P^{Ri}\}$, $0 \leq i \leq r-1$. Here $P^{Ri}$ denotes the subset consisting of all peers with the same resource type $R_i$ and no two peers in $P^{Ri}$ have the same value for $R_i$ and the number of distinct resource types present in the system is r. Also for each subset $P^{Ri}$, $P_i$ is the first peer among the peers in $P^{Ri}$ to join the system. We now describe the P2P architecture suitable for interest-based peer-to-peer system

## 2.1 Two Level Hierarchy

In [13] we have proposed a two level overlay architecture and at each level, networks of peers are all structured. It is explained in detail below.

1) At level-1, we have a ring network consisting of only the peers $P_i$ ($0 \leq i \leq r-1$). Therefore, number of peers on the ring is r, the number of distinct resource types. This ring network is used for efficient data lookup and so it is called as transit network.
2) At level-2, there are r numbers of completely connected networks of peers. Each such network, say $N_i$ is formed by the peers of the subset $P^{Ri}$, ($0 \leq i \leq r-1$), such that all peers ($\epsilon P^{Ri}$) are directly connected (logically) to each other, resulting in the network diameter of 1. Each such $N_i$ is connected to the transit ring network via the peer $P_i$. Peer $P_i$ acts as the group-head of network $N_i$. From now on network $N_i$ will be referred to as $group_i$ (in short as $G_i$) with $P_i$ as its group-head. The architecture is shown in Figure. 1.
3) Each node in the transit ring network maintains a *global resource table* (GRT) that consists of tuples of the form *<Resource Type, Resource Code, Group Head Logical Address, Group Head IP address>*, where *Group Head Logical Address* refers to the logical address assigned to a node by our proposed architecture.

## 2.2 Linear Diophantine Equation (LDE) and Its Solutions

Let us consider the LDE as stated below.

$$an \equiv b \pmod{c}, \quad a, b, \text{ and } c \text{ are integers.} \qquad (1)$$

Let $d \mid b$, where $d = \gcd(a,c)$. It means that (1) has d mutually incongruent solutions.

The above equation can also be stated as

$$an + (-c)k = b, \quad k \text{ is an integer.} \qquad (2)$$

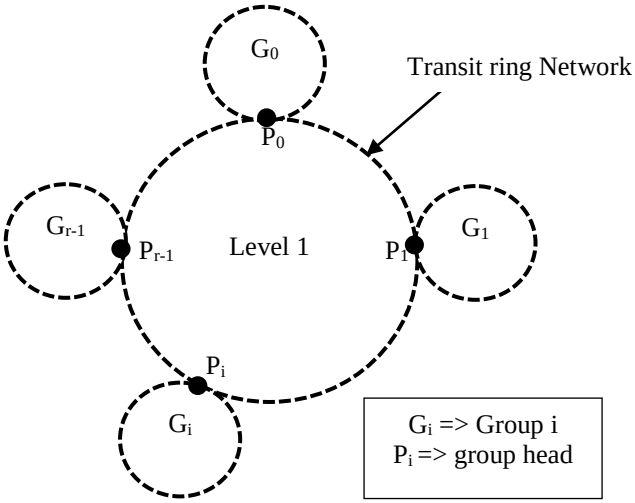


Figure 1: A two-level structured architecture with distinct resource types

Each solution of Equ. (1) (& hence of (2) as well) has the form: $n = n_0 + ct/d$, $k = k_0 + at/d$
where $n_0$ and $k_0$ constitute one specific solution and t is any integer.

Among the different values of n described by $n = n_0 + ct/d$, we note that the d values
$n_o$, $n = n_0 + c/d$, $n = n_0 + 2c/d$, --- , $n = n_0 + (d-1)c/d$
are all mutually incongruent modulo c, because the absolute difference between any two of them is less than c.

Also the values of a, b, and c can be so chosen as to make d very large whenever needed. Observe that there are infinite other solutions which are congruent to each of the d solutions. For example, all solutions of the form
$(n_o + mc)$, m is an integer, are mutually congruent. Similarly all solutions of the form $[(n_0 + c/d) + mc]$ are mutually congruent.

## 2.3 Implementation of the Architecture

Assume that in an interest-based P2P system there are r distinct resource types ($r \leq d$). That is, a maximum of d resource types can be present. Note that this is not a restriction, because d can be set to an extremely large value a priori by choosing an appropriate LDE. Consider the set of all peers in the system given as
$S = \{P^{Ri}\}$, $0 \leq i \leq r-1$.

As mentioned earlier, for each subset $P^{Ri}$ (i.e. group Gi) peer Pi is the first peer with resource type Ri to join the system. Now we use the mutually incongruent solutions of a given LDE to define the architecture as follows.

The ring network (Figure. 1) at level-1 will consist of all such $P_i$'s, for $0 \leq i \leq r-1$, and $r \leq d$, such that

a) Each $P_i$ will be assigned the logical address ($n_0$ + i.c/d). Note that ($n_0$ + i.c/d) is the $i^{th}$ mutually incongruent solution where $0 \leq i \leq d-1$.

b) The transit network is a ring by default, because of modulo operation. Two peers in the ring network are neighbors if their assigned addresses differ by c/d, with the exception that the first peer $P_0$ and the last peer $P_{r-1}$ will be considered as neighbors even though their addresses differ by (r-1).c/d. This structure has made the joining of new peers with new resource types very simple.

c) Resource type $R_i$ possessed by peers in $G_i$ is assigned the code ($n_0$ + i.c/d) which is also the logical address of the group-head $P_i$ of group $G_i$.

d) Diameter of the ring network can be at most d/2.

At level-2 all peers having the same resource type $R_i$ will form the group $G_i$ (i.e. the subset $P^{Ri}$). Only the group-head $P_i$ is connected to the transit ring network. Observe that any communication between any two groups $G_i$ and $G_j$ takes place via the respective group-heads $P_i$ and $P_j$. Peers in $G_i$ will be assigned with the addresses

$$[(n_0 + i.c/d)+ m.c], \text{ for } m = 0, 1, 2, \ldots \quad (3)$$

Note that m = 0 corresponds to the address of group-head $P_i$ of $G_i$.

*Observation 1.* All addresses in $G_i$ are mutually congruent solutions for a given i.

*Observation 2. Congruence Relation* is reflexive, symmetric, and transitive. Therefore, it can be concluded that all peers in a group $G_i$ are *directly connected (logically*) to each other forming a network of *diameter 1* only.

## 2.4 Intra-Group Data Lookup

Without any loss of generality let us consider data lookup in group $G_i$ by a peer $p_a$ possessing $< R_i , V_a >$ and requesting for resource $< R_i , V_b >$. The algorithm for intragroup data lookup is presented in algorithm *Intra-Group-Lookup* (Algorithm 1).

```
1 node pa (ϵ Gi) broadcasts in Gi for < Ri, Vb >
  // one-hop communication since Gi is a complete graph

2 if pb with < Ri, Vb > then
3       node pb unicasts < Ri, Vb > to node pa
4 else
5       search for <Ri, Vb > fails
6 end
```

Figure 2: Algorithm 1: Intra-Group-Lookup

## 2.5 Inter-Group Data Lookup

In our proposed architecture, any communication between a node $p_i \in G_i$ and $p_j \in G_j$ takes place only via the respective group-heads $P_i$ and $P_j$. Without any loss of generality let a peer $pi \in G_i$ request for a resource $< R_j , V^* >$. The following steps are executed to answer the query:

Peer $p_i$ knows that $R_j \notin G_i$. Assume that there are $r$ distinct resource types and $r \leq d$. Then, in order to locate resource $R_j$,a search along the transit ring network is required. We call this method as algorithm *Inter-Group-Lookup* (Algorithm 2).

# 3 Secured Data Look-Up

To achieve security from the viewpoints of authentication and confidentiality, we apply symmetric cryptography [9] for intra-group data communication and asymmetric cryptography for inter-group communication. Symmetric key technique uses the same key for ciphering and deciphering. In symmetric cryptography, generating strong keys for the ciphers are relatively easier compared to its asymmetric counterpart. The encryption and decryption computations are faster since we use one key for both operations. In addition, in general it is more difficult to break symmetric keys compared to asymmetric keys. However, it requires a secure way to distribute the shared keys among the peers. In our P2P architecture use of symmetric keys for intra-group communication appears to be suitable since all peers in a group form a complete graph and hence they all are one hop away from the group-head and from each other. In our system, we assume that group-heads are trustworthy peers and they act as trusted key distributed centers. Also, when a group-head crashes or leaves, the new group-head acts as a trusted center as well. However, for inter-group communication, we take advantage of asymmetric cryptography. In asymmetric cryptography [18], the keys are not identical. For each secure communication, there is a pair of keys for encoding and decoding interchangeably. The key in the pair that can be shared openly is called the public key. The matching key, which is kept secret, is called the private key. Both keys can be used to encrypt a message; the other key can act in reverse. Furthermore, to be able to support the use of asymmetric cryptography, we do a minor modification of Global Resource Table (GRT). A new entry is used in the GRT to represent the public key of each group-head. Therefore, the new GRT consists of tuples of the form; *<Resource Type, Resource Code, Group Head Logical Address, and Group Head Public Key>*. Group-head $G_0$ is responsible for updating the GRTs to reflect the effect of churn caused by group-heads leaving / joining the P2P system. In addition, we assume that in each group, its members share a unique master key each with the group-head for secure intra-group communication.

## 3.1 Secured Intra-Group data lookup

For Intra-Group data look up, without any loss of generality, let us consider that in group $G_i$, peer $p_a$ possesses $<R_i,V_a>$ and requests for resource $<R_i,V_b>$. Notation $K_{mn}$ denotes the master key shared only by a peer $p_n$ ($\epsilon$ $G_m$) and the corresponding group-head $P_m$ of group $G_m$. Thus, $p_a$ has the master key, $K_{ia}$, known only to itself and the group-head $P_i$. For secure intra-group data look-up the following steps are followed in (Figure. 4):

1 Node $p_i$ ($\epsilon$ $G_i$) unicasts request for < $R_j$, V*> to group-head $P_i$
2 $P_i$ determines resource <$R_j$, $V^*$> group-head $P_j$'s address code from GRT // address code of $P_j$ = resource code of $R_j$ = $n_0 + (c/d)j$
3 $P_i$ computes h ← $|(n_0 + i(c/d)) - (n_0 + j(c/d))|$ // looking for minimum no. of hops along the transit ring

4 **if** h > r/2 **then**
5 $P_i$ forwards the request along with the IP address of $p_i$ to its predecessor $P_{i-1}$
6 **else**
7 $P_i$ forwards the request along with the IP address of $p_i$ to its successor $P_{i+1}$
8 **end**
9 Each intermediate group-head $P_k$ forwards the request until the request arrives at $P_j$
10 **if** $P_j$ possesses < $R_j$, V*> **then**
11 $P_j$ unicasts < $R_j$, V*> to $p_i$
12 **else**
13 $P_j$ broadcasts the request for <$R_j$, V*> in group $G_j$
14 **if** $P_j$ possesses <$R_j$, V*> **then**
15 $P_j$ unicasts <$R_j$, V*> to $p_i$
16 **else**
17 $P_j$ unicasts search failed to $p_i$
18 **end**
19 **end**

Figure 3: Algorithm 2: Inter-Group-Lookup

1. $p_a$ issues an encrypted request for resource $<R_i,V_b>$ to the group-head $P_i$.
   // This requested message is encrypted by the shared key $K_{ia}$ of $P_i$ and $p_a$. Thus, $P_i$ is the only one who //can successfully read the message and $P_i$ knows that it has originated at peer $p_a$
2. Group-head $P_i$ decrypts the message with $K_{ia}$
3. Group-head $P_i$ broadcasts in $G_i$ for $<R_i,V_b>$
4. If peer $p_b$ possesses $<R_i,V_b>$, it encrypts $<R_i,V_b>$ with $K_{ib}$ and sends it to $P_i$
5. $P_i$ decrypts the message with $K_{ib}$
6. $P_i$ encrypts the message $<R_i,V_b>$ with $K_{ia}$ and sends it to the requesting peer $p_a$
7. $p_a$ decrypts the received message with $K_{ia}$ and now has the resource $<R_i,V_b>$

Figure 4: Algorithm 3: Secured Intra-Group-Lookup

### 3.2 Secured Inter-Group data look up

In our architecture, as we have discussed before, any communication between two peers $p_i$ ($\in G_i$) and $p_j$ ($\in G_j$) takes place only via the respective group-heads $P_i$ and $P_j$. We use . We use the notations $Pu_m$ and $Pr_m$ to denote respectively the public and private keys of group-head $P_m$. Without any loss of generality, let a peer $p_i \in G_i$ request for a resource <$R_j$,$V^*$>.
Peer $p_i$ knows that $R_j \notin G_i$ . Assume that there are $r$ distinct resource types and $r \leq d$. The steps in Algorithm 3 are executed to answer the query (Figure. 4)

## 4 Experimental Results

In Table 1, an analytical comparison of the LDE based system with two of the most well established P2P systems, viz., Chord and Pastry is presented. It shows the superiority of our LDE based architecture compared to the other two.

Table 1: Data Lookup Complexity Comparison

| | **Chord** | **Pastry** | **LDE-based** |
|---|---|---|---|
| **Architecture** | Structured P2P Overlay | Structured P2P Overlay | Interest based, Two-level Structured Hierarchical |
| **Lookup Protocol** | Matching key and NodeID. | Matching key and prefix in NodeID. | Inter-Group: Routing through Group-heads Intra-group: Complete Graph |
| **Parameters** | *N*-number of peers in network. | *N*-number of peers in network b-number of bits ($B = 2^b$) used for the base of the chosen identifier. | *r* - Number of distinct resource types. *N*-number of peers in network. r << N |
| **Lookup Performance** | $O(\log N)$ | $O(\log_B N)$ | Inter-Group: $O(r)$ Intra-group: $O(1)$ |

1. Peer $p_i$ ($\epsilon$ $G_i$) encrypts the request for $<R_j,V^*>$ with $K_{ii}$
2. $P_i$ dycrypts the message with $K_{ii}$ and finds group-head $P_j$'s address code from *GRT*
   // *address code* of $P_j = n_0 + j\ (c/d)$
3. $P_i$ computes $h \leftarrow |\ (n_0 + i\ (c/d)) - (n_0 + j\ (c/d))\ |$
   // *looks for minimum no. of hops along the transit ring to reach* $P_j$
4. **if** $h > r/2$ then
   $P_i$ encrypts the message with $Pu_j$ and forwards the request to its predecessor $P_{i-1}$
5. **else**
   $P_i$ encrypts the message with $Pu_j$ and forwards the request to its successor $P_{i+1}$
6. **end**
7. Each intermediate group-head $P_k$ forwards the request until the request arrives at $P_j$
8. $P_j$ decrypts the message with its own private key $Pr_j$
9. **if** $P_j$ possesses $<R_j,V^*>$
10. $P_j$ encrypts the message with the public key $Pu_i$ of $P_i$ and unicasts it to $P_i$
11. **else**
12. $P_j$ broadcasts the request for $<R_j,V^*>$ in group $G_j$
13. **if** $\exists\ p_k\ (\epsilon\ G_i)$ which possesses $<R_j,V^*>$
14. $p_k$ encrypts the request message with $K_{jk}$
15. $P_j$ decrypts the message with $K_{jk}$
16. $P_j$ encrypts the decrypted message with the public key $Pu_i$ of $P_i$ and sends it to $P_i$
17. $P_i$ decrypts the message with its own private key $Pr_i$
18. $P_i$ encrypts the message $<R_i,V_b>$ with $K_{ii}$ and sends it to the requesting peer $p_i$
19. $p_i$ decrypts the received message with $K_{ii}$
20. **else**
21. $P_j$ unicasts *'search failed'* to $p_i$
22. **end**
23. **end**

Figure 5: Algorithm 4: Secured Inter-Group-Lookup

In addition, we have also done simulation to demonstrate the efficiencies of the data lookup mechanisms used in LDE based system, Chord, and Pastry. In this simulation, we have used PeerfactSim.Kom software [15]. In the simulation, we have focused on measuring the average of hop counts in presence of churn; we have experimented with networks consisting of 100, 200, and 300 peers for each of the 3 P2P systems. Figures 6 a, b, and c present the results of the simulation. Table 2 contains the summarized results as well. We infer that LDE based P2P system offers more efficient data lookup mechanisms than the other two.

# 4 Conclusion

We have extended our earlier work to incorporate security in data communication. We have done extensive simulations and the results of the simulation show the superiority of the LDE based architecture compared to some important existing works from the viewpoint of data lookup efficiency in presence of frequent churn (i.e. peers leave and join the system randomly). This work is a part of an ongoing research project with the goal of designing P2P federation consisting of small P2P systems so that bandwidth cannot be an issue

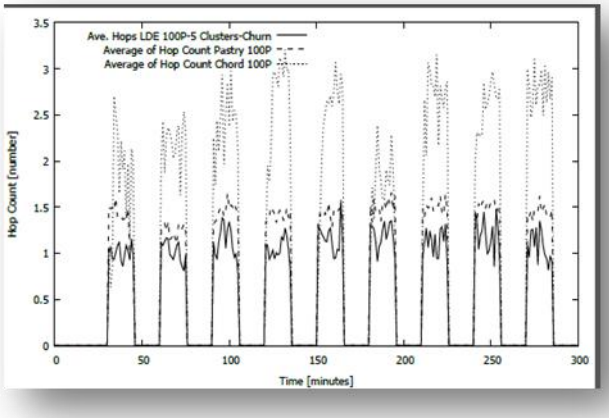


Figure 6a: 100 Peers

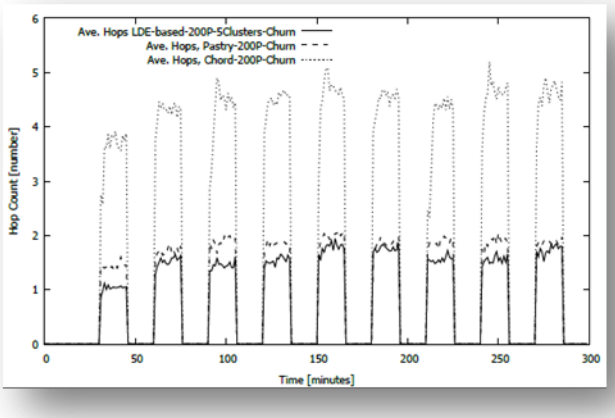


Figure 6b: 200 Peers

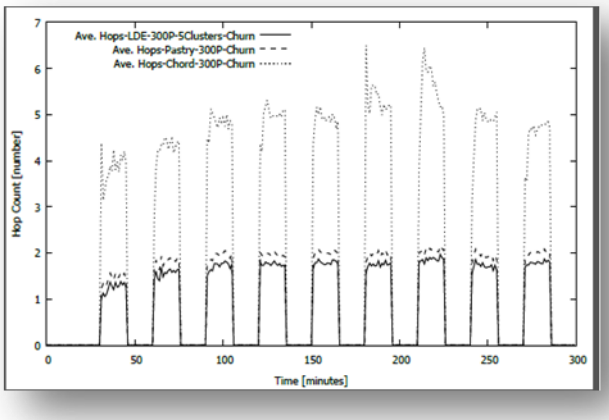


Figure 6c: 300 peers

Table 2: Ave. Hop Counts in Simulation results

| | **100 Peers** | **200 Peers** | **300 Peers** |
|---|---|---|---|
| **LDE-Based with 5 Clusters** | 1.3063 hop/min | 1.5417 hop/min | 1.685 hop/min |
| **Pastry** | 1.6332 hop/min | 1.8102 hop/min | 1.907 hop/min |
| **Chord** | 3.7325 hop/min | 4.321 hop/min | 4.775 hop/min |

Figure 6 (a, b, & c): Average of Hop Counts in LDE-Based vs Chord & Pastry in networks of 100, 200, and 300 Peers

# References


[1] P. Ganesan, Q.Sun, and H. Garcia-Molina, "Yappers: A peer-to-peer lookup service over arbitrary topology," Proc. IEEE Infocom, pp. 1250-1260, 2003, San Francisco, USA, March 30 - April 1 2003.

[2] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, "Making gnutella-like p2p systems scalable," Proc. ACM SIGCOMM, Karlsruhe, Germany, August 25-29, 2003.

[3] B. Y. Zhao,L. Huang, S. C. Rhea, J. Stribling, A. Zoseph, and J. D. Kubiatowicz, "Tapestry: a global-scale overlay for rapid service deployment," IEEE J-SAC, vol. 22, no. 1, pp. 41-53, Jan. 2004.

[4] A. Rowstron and P. Druschel, "Pastry: scalable, distributed object location and routing for large scale peer-to-peer systems," Proc. IFIP/ACM Intl. Conf. Distributed Systems Platforms (Middleware), pp. 329-350, 2001.

[5] I. Stocia, R. Morris, D. Liben-Nowell, D. R. Karger, M. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup protocol for internet applications," IEEE/ACM Tran. Networking, vol. 11, No. 1, pp. 17-32, Feb. 2003.

[6] M. Yang and Y. Yang, "An efficient hybrid peer-to-peer system for distributed data sharing," IEEE Trans. Computers, vol. 59, no. 9, pp. 1158-1171, Sep. 2010.

[7] M. Xu, S. Zhou, and J. Guan, "A new and effective hierarchical overlay structure for peer-to-peer networks," Computer Communications, vol. 34, pp. 862-874, 2011.

[8] D. Korzun and A. Gurtov, "Hierarchical architectures in structured peer-to-peer overlay networks," Peer-to-Peer Networking and Applications, Springer, pp. 1-37, March 2013.

[9] W. Stallings, Cryptography and Network Security: Principles and Practice, 7th ed., Prentice Hall.

[10] Z. Peng, Z. Duan, J.Jun Qi, Y. Cao, and E. Lv, "HP2P: a hybrid hierarchical p2p network," Proc. Intl. Conf. Digital Society, 2007.

[11] K. Shuang, P Zhang, and S. Su, "Comb: a resilient and efficient two-hop lookup service for distributed communication system," Security and Communication Networks, vol. 8(10), pp. 1890-1903, 2015.

[13] Bidyut Gupta, Shahram Rahimi, Ziping Liu, and Sindoora Koneru, "Design of structured peer-to-peer networks using linear diophantine equation," Proc. CAINE, pp. 147-151, New Orleans, Oct., 2014.

[14] N. Rahimi, K. Sinha, B. Gupta, and S. Rahimi, "LDEPTH: A low diameter hierarchical p2p network architecture," Proc. 2016 IEEE Int. Conf. on Industrial Informatics (INDIN 2016), Poitiers, France, July, 2016.

[15] Feldotto, M., & Graffi, K. (2013, July). Comparative evaluation of peer-to-peer systems using PeerfactSim. KOM. In High Performance Computing and Simulation (HPCS), 2013 International Conference on (pp. 99-106). IEEE.