EPiC Series in Computing

Volume 75, 2021, Pages 59–68

CAINE 2020. The 33rd International Conference on Computer Applications in Industry and Engineering

On Designing Secured Communication Protocols along with Anonymity for CRT based Structured P2P Network Architecture

[1]Koushik Maddali, [1]Swathi Kaluvakuri, [2]Nick Rahimi,[1]Bidyut Gupta and [3]Narayan Debnath

[1]School of Computing
Southern Illinois University
Carbondale, IL, USA

[2]Information Systems and Applied Technology
Southern Illinois University
Carbondale, IL, USA

[3]School of Computing and Information Technology
Eastern International University, Vietnam

{koushik, swathi.kaluvakuri, shrahimi }@siu.edu, bidyut@cs.siu.edu, ndebnath@gmail.com

Abstract

In this work, we have considered designing secured communication protocols for Chinese remainder theorem based structured p2p architecture. Such an architecture has been the choice because of the complexity in Inter or Intra group communications are just $O(1)$ [16]. In this work, we have considered efficient way to make the already existing communication protocols [16] secured. We have extended these protocols further to include anonymity. We have considered security separately for multicasting inside a group and multicasting outside the group.

Index Terms **— Overlay multicast, Structured P2P network, Chinese Remainder Theorem, security, anonymity, cryptographic keys**

I. INTRODUCTION

Problems associated with the global deployment of multicast-capable routers, lack of support for network management, and also the scalability problem caused by the simultaneous presence of large number of multicast sessions, are some of the main reasons why the deployment of router-based IP

multicast has been slow. Consequently, researches have started considering application level multicast as an alternative to IP multicast, because the former one can be deployed fast as it does not depend on router infrastructure [1], [2], [3]. Multicast protocols proposed in [4], [5], [6] focus on designing an optimized overlay multicast tree per multicast source. They can work well for certain applications, such as software distribution. There exist some interesting multicast protocols designed to work in DHT-based architectures [7], [8] However, none of these above-mentioned works considers node heterogeneity. In addition, their performance degrades sharply as frequency of node (peer) movements in and out of the network increases. Note that frequent joining of new peers and leaving of existing peers is known as churn.

Authors in [9],[10], [11] have proposed a unique way of designing non-DHT based P2P network architecture. The architecture is an unrestricted ring of nodes (peers) in a sense that a node can be anywhere on the ring unlike in DHT-based ring, such as Chord [12]. Each node n_i has its successor on the ring and also each node randomly selects c_i number of other nodes on the ring as its immediate logical neighbors; c_i is the degree/capacity of node n_i. They have proposed a capacity-constrained any source overlay multicast protocol which uses the unique idea of transforming a multicast problem to a broadcast one and the broadcast of a message is completed using a combination of tree propagation and sequential propagation on the ring. During multicasting, a tree structure is created implicitly even though there is no explicit multicast tree creation unlike in the classical multicast protocols that use either the source-based tree approach [13] or the shared tree approach [14].

The work [16] has used some of the ideas from [10] in designing efficient any source overlay p2p networks with capacity constrained approach.

II. PRELIMINARIES

In this section, we start with a brief description of the CRT-based hierarchical structured P2P architecture [9]; note that we design our multicast protocols for this architecture. This section ends with a summary of our contribution in the present work. This architecture is a two-level structured architecture for interest-based peer-to-peer system [9], [11], [15]. We use the following notations along with their interpretations while we define the architecture.

Two Level Architecture

A resource is defined 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 proposed model for interest-based P2P systems, it is assumed 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 $V' \neq V''$. However, this constraint can easily be relaxed [9].

We define the following. Let S be the set of all peers in a peer-to-peer system. Then $S = \{P^{Ri}\}$, $0 \leq i \leq n-1$. 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 n. Also, for each subset P^{Ri}, P_i is the first peer among the peers in P^{Ri} to join the system.

At level 1, there is a network of peers such that peers are directly connected (logically) to each other. In graph theoretic term, the network at level 1 is a complete graph. Hence, the network diameter is *1 overlay hop*. The periphery of this network appears as a ring network and hence it is named as *transit ring network*. This network consists of the peers P_i ($0 \leq i \leq n-1$). Therefore, number of peers on the ring is n and this number represents the number of distinct resource types present in the P2P system. Each of these n peers is termed *group-head.* The periphery of this network as well as the direct links connecting

any two peers in this network can be used for efficient data lookup. In this architecture, each group-head has a global resource table (GRT) that has every group-head's logical as well as IP addresses.

At level-2, there are n numbers of completely connected networks of peers. Each such network, say N_i is formed by the peers of the subset P^{Ri}, $(0 \leq i \leq n-1)$, such that all peers (ϵ P^{Ri}) are directly connected (logically) to each other, resulting in the network diameter of *1 overlay hop*. Each such N_i also called group (in short as G_i) is connected to the transit ring network via the peer P_i, the group-head of G_i. The architecture is shown in Fig. 1.

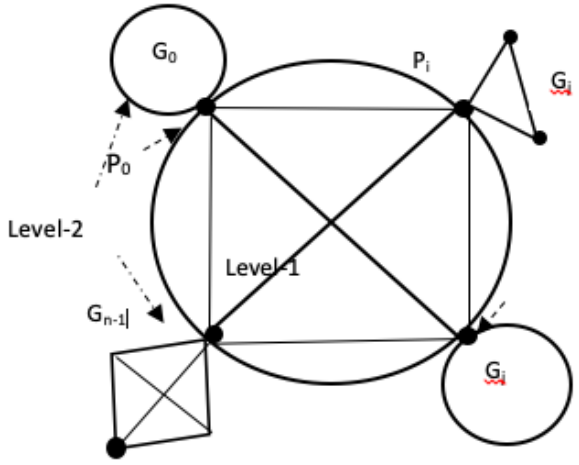

Fig. 1 A two-level structured P2P architecture with r distinct resource types

In any structured P2P system, the mathematical model used to build the architecture defines neighborhood relations among peers. The mathematical model is intimately related to the efficiency of different data lookup schemes used in a given structured P2P system.

We now state a brief sketch of the mathematical model used in this approach to realize the architecture [9]. The authors first determine a simultaneous solution (a positive integer) of a given system of linear congruencies and then determine some more solutions as needed to form the architecture, which are congruent to the simultaneous solution. For this, the authors have used the Chinese Remainder Theorem (CRT). Each such solution will become the logical address of a group head uniquely [9] [16]. At the same time, it requires to determine separately the solutions of each linear congruence as needed and these solutions will represent the logical addresses of the peers present in a group [11]. The following interesting structural facts are revealed.

Observation 1. *Any insertion of a group head P_l always takes place between the current last group head P_{l-1} and the first group head P_0 along the transit ring network.*

Lemma 1. *Diameter of the Level-1 network is 1 overlay hop.*

Lemma 2. *Diameter of a Level-2 group is 1 overlay hop.*

Theorem 1. *Diameter of the hierarchical two-level structured architecture is 3 overlay hops.*

Remark 1. *There are infinite number of solutions which are congruent to the one mutually congruent solution of any Linear Diophantine Equation (LDE) considered in CRT, hence, size of a cluster at Level-2 can be made very large (theoretically unlimited), yet the diameter remains 1.*

Observation 2. *Each group head has two different logical addresses; one from Level-1 assignment and one from Level 2 assignment.*

Observation 3. *Different group heads may get identical Level 2 assigned addresses. It will not affect any intra-cluster lookup query in a cluster, as this address is local to this group only.*

III. OUR CONTRIBUTION

We have considered designing secured multicast protocols for both inside a group and for the two level architecture. In addition, we have also considered anonymity. We have also designed multicast algorithms with both security and anonymity. In section IV we present the secured data lookup algorithms. In section V, we have present multicast algorithms with both security and anonymity.

IV. SECURED DATA LOOKUP ALGORITHMS

Computer security encompasses confidentiality, integrity, availability, authentication, non-repudiation etc. To achieve security in P2P networks from the view point of authentication and confidentiality, cryptographic algorithms are used. Generally, cryptographic algorithms are classified into two types, secret key cryptographic algorithms and public key cryptographic algorithms. Secret key cryptographic algorithms are also termed as symmetric key algorithms as the same key is used for both encryption and decryption which is shared between all parties involved. On the other hand, public key cryptographic algorithms are also known as asymmetric key algorithms. In this type, a pair of keys are used, one for encryption and another for decryption. One of the pair is made available to everyone known as a public key and the other is kept secret; known as the private key. Message encrypted using a public key can only be decrypted by the secret key of the pair. Same way, a message that is encrypted using a private key can only be decrypted by the public key of the pair. In this section, secure data lookup algorithms [9] both Inter and Intra are presented.

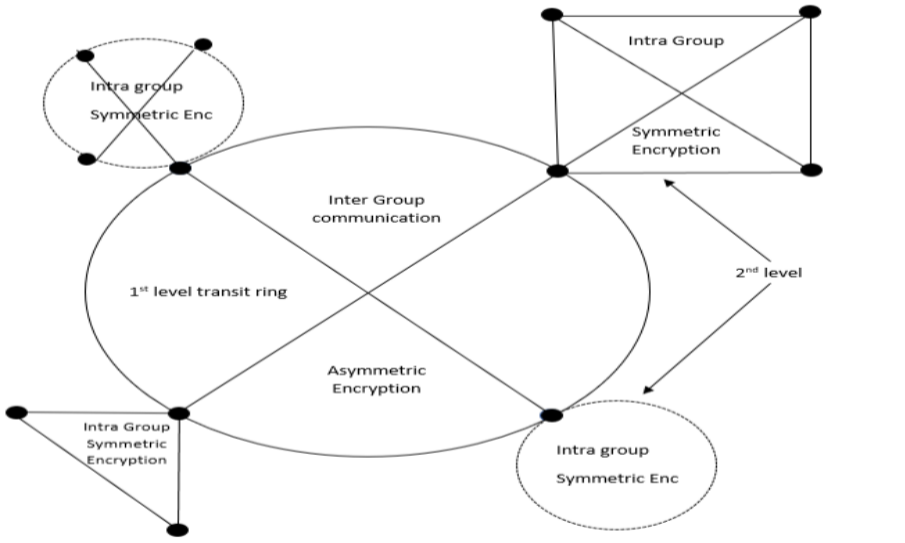

Fig. 2 Types of Cryptographic functions and their application domains

Secure Intra Group Lookup

We use symmetric cryptography for secure intra group lookup. let us consider that in group $Group_i$, peer $G_{i,x}$ has a resource $<Res_i, V_x>$ and it requests for resource $<Res_i, V_y>$. Let $Key_{i\,x}$ denotes the common key shared only by a peer $G_{i,x}$ *that belongs to* $Group_i$ and the corresponding group-head H_i of the same group. To summarize, this common key information is known only to the peer $G_{i,x}$ and group head H_i. So, symmetric encryption takes place with in a group.

1. Request node $G_{i,x}$ encrypts the request $<Res_i, V_y>$ using the common key $Key_{i\,x}$ and unicasts it to the Group head H_i

   // This common key is known only to the requesting node and group head only so other nodes will not be able to decrypt it. Hence the symmetric key security
2. The Encrypted request is decrypted by Group head H_i using the common key $Key_{i\,x}$
3. H_i will broadcast the request $<Res_i, V_y>$ *in* $Group_i$,
4. **If** a node $G_{i,y}$ has the requested resource $<Res_i, V_y>$
   a. it encrypts the resource with common key $Key_{i\,y}$ and unicasts it to the group head H_i
   b. The group head H_i uses common key $Key_{i\,y}$ and decrypts the response
   c. H_i now encrypts the message $<Res_i, V_y>$ with $Key_{i\,x}$ and sends it to the requesting node $G_{i,x}$
   d. $G_{i,x}$ decrypts the message using $Key_{i\,x}$

   else

   search for $<Res_i, V_y>$ fails

Fig. 3 Intra Group Lookup Algorithm with Security

Secure Inter Group Lookup

In this secured architectiure, any sort of communication between two peers $G_{i,x} \in Group_i$, and $G_{j,y} \in Group_j$ takes place only through the corresponsing group heads H_i and H_j. *Because the communication is between two different groups, we use asymmetric encryption, so here comes the concept of Public and Private keys.*

We use the notations Pub_i and Prv_i to denote respectively the public and private keys of group-head H_i *and let* Pub_j and Prv_j be the public and private keys of group-head H_j. Without any loss of generality, let a peer $G_{i,x} \in Group_i$ requests for a resource $<Res_j, V_y>$. Peer $G_{i,x}$ is aware of the fact that that $Res_j \notin Group_i$.

1. Request node $G_{i,x}$ encrypts the request $<Res_j, V_y>$ using the common key Key_{ix} and unicasts it to the Group head H_i

   // This common key is known only to the requesting node and group head only so other nodes will not be able to decrypt it.
2. The Encrypted request is decrypted by Group head H_i using the common key Key_{ix} and finds the Group head address of H_j *along with its public key* Pub_j from the GRT table
3. H_j encrypts the message with Pub_j and forwards the request across the ring
4. Each intermediate group-head H_k forwards the request until the request arrives at H_j
5. Now H_j will decrypt the message using its private key Prv_j
6. **if** H_j itself possesses $<Res_j, V_y>$
7. H_j encrypts the message with the public key Pub_i of H_i and unicasts it to H_i
8. **else**
9. H_j broadcasts the request for $<Res_j, V_y>$ in group $Group_j$
10. **if** $\exists\ G_{j,y}\ (\epsilon\ Group_j)$ which possesses $<Res_j, V_y>$
11. $G_{j,y}$ encrypts the request message with Key_{jy}
12. H_j decrypts the message with Key_{jy}
13. H_j encrypts the decrypted message with the public key Pub_i of H_i and sends it to H_i
14. H_i decrypts the message with its own private key Prv_i
15. H_i encrypts the message $<Res_j, V_y>$ with Key_{ix} and sends it to the requesting peer $G_{i,x}$
16. $G_{i,x}$ decrypts the received message using the common key Key_{ix}
17. **else**
18. H_j unicasts 'search failed message' to H_i
19. **end**
20. **End**

Fig. 4 Inter Group Lookup Algorithm with Security

V. SECURED MULTICAST ALGORITHMS WITH ANONYMITY

In this section we'll present secured multicast algorithms both inter group and for the whole p2p systems. The basic Multicast algorithms of CRT architecture presented in [16] have been enhanced here with security properties.

Secured Multicast Algorithm 1 with Anonymity where capacity ≥ #groupheads

1. Source peer $G_{i,m}$ encrypts *multicast_msg* using common key $Key_{i\,m}$ and unicasts it to the group head H_i
2. Group head H_i decrypts the received multicast_msg using the common key $Key_{i\,m}$ and then replaces the ip_address of the $G_{i,m}$ to its own.
   Note: GRT is modified in this scenario. Public key of each group head is added along with ip_address.
3. H_i then gathers ip_addresses of fellow group heads along with their public keys from the GRT.
4. H_i then encrypts the modified multicast_msg with the public key of the respective target group head.
5. H_i then unicasts the encrypted multicast_msg the target group head and repeats the same for all other group heads.
6. Each receiver group head decrypts the received multicast_msg using their respective private keys.
7. **If** the receiver group head is also a multicast group member,
   a. It makes a copy of the multicast_msg and keeps it for itself.
   b. Replaces the ip_address of H_i to its own, encrypts the message using common key $Key_{a,b}$ (where a is the group number and b is the number of group member) and unicasts it to members.

   else

   Replaces the ip_address of H_i to its own, encrypts the message using common key $Key_{a,b}$ (where a is the group number and b is the number of group member) and unicasts it to members.

Fig. 5 Multicast Algorithm 1 with Security and Anonymity

Example 1:
Scenario : $c_{si} \geq n_r$
group heads (n_r) = 6
Capacity of each group head (c^s_i) = 9
In this example assume that the source peer is $G^s_{4,18}$

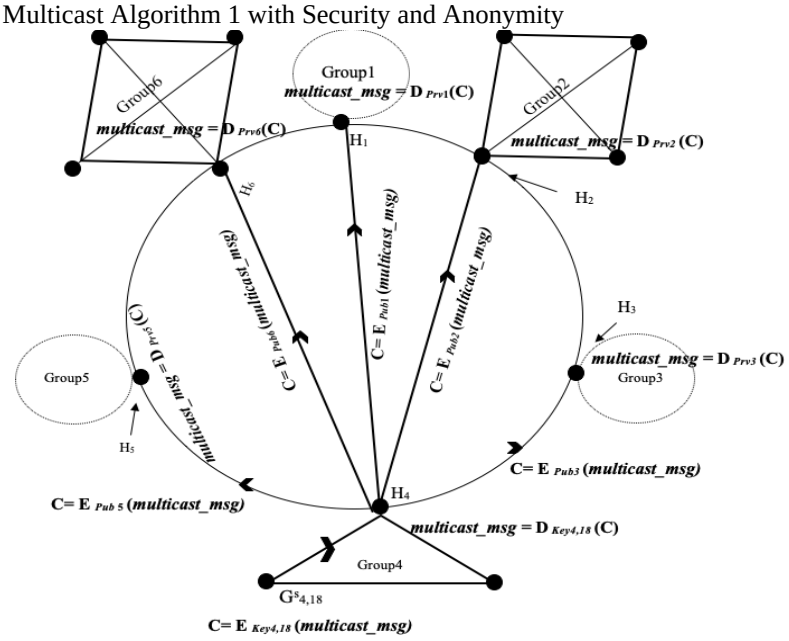

Fig. 6 Multicast Algorithm 1 example with Cryptographic messages

Secured Multicast Algorithm 2 with Anonymity where Capacity < # groupheads

1. Source peer $G_{i,m}$ encrypts multicast_msg using common key $Key_{i\,m}$ and unicasts it to the group head H_i
2. Group head H_i decrypts the received multicast_msg using the common key $Key_{i\,m}$ and then replaces the ip_address of the $G_{i,m}$ to its own.
   for j = 1 to p
3. H_i then gathers ip_addresses of fellow group heads along with their public keys from the GRT.
4. H_i then encrypts the modified multicast_msg with the public key of the respective target group head Pub_a where a is the number of the group head for each subset.
5. H_i then unicasts the encrypted multicast_msg to the target group head and repeats the same for all other group heads p in j.
6. Each receiver group head decrypts the received multicast_msg using private key Prv_a where a is the group head number.
7. **If** the receiver group head is also a multicast group member,
   a. It makes a copy of the multicast_msg and keeps it for itself.
   b. Replaces the ip_address of H_i to its own, encrypts the message using common key $Key_{a,b}$ (where a is the group number and b is the number of group member) and unicasts it to members.

   else

   Replaces the ip_address of H_i to its own, encrypts the message using common key

   $Key_{a,b}$ (where a is the group number and b is the number of group member) and unicasts it to members.

Fig. 7 Multicast Algorithm 2 with Security and Anonymity

Example 2:
$c_{si} < n_r$ (Case1)
Suppose #group heads (n_r) = 13
Capacity of each group head (c^s_i) = 5
In this example assume that the source peer is $G^s_{7,\,18}$
So, Set T of 13 number of receiver group-heads is partitioned into 3 disjoint subsets.
Subset T_1 consists of receiver group-heads H_8 to $H_{12,}$ T_2 consists of receiver group-heads $H_{13,}$ H_1 to H_4 and subset T_3 consists of receiver group-heads H_5 and H_7

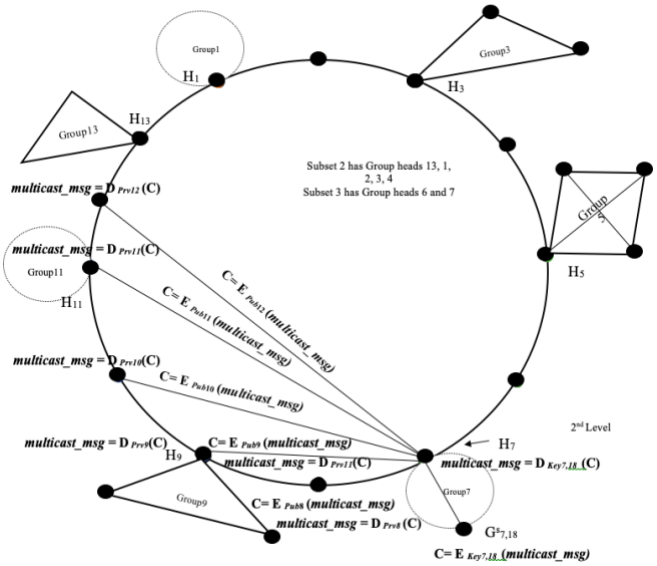

Fig. 8 Multicast Algorithm 2 example with Cryptographic messages

Secured Multicast Algorithm 3 with Anonymity where Capacity < # groupheads

1. Source peer $G_{i,m}$ encrypts *multicast_msg* using common key $Key_{i\ m}$ and unicasts it to the group head H_i
2. Group head H_i decrypts the received *multicast_msg* using the common key $Key_{i\ m}$ and then replaces the ip_address of the $H_{i,m}$ with its own.
3. H_i then gathers ip_addresses of fellow group heads along with their public keys from the GRT.
4. H_i randomly selects groups heads equal to its capacity/degree.
5. It encrypts the modified *multicast_msg* with the public key of the selected group head Pub_a where a is the number of the group head and unicasts it.
6. H_i repeats the same for all other selected group heads based on degree.

If receiver group head receives the *multicast_msg* for the first time (unique),

a. Each receiver group head decrypts the received *multicast_msg* using private key Prv_a.

If the receiver group head is also a multicast group member,

i. It makes a copy of the multicast_msg and keeps it for itself.

ii. Replaces the ip_address of H_i to its own, encrypts the message using common key $Key_{a,b}$ (where a is the group number and b is the number of group member) and unicasts it to members.

else

i. Replaces the ip_address of H_i to its own, encrypts the message using common key $Key_{a,b}$ (where a is the group number and b is the number of group member) and unicasts it to members.

b. It then replaces the ip_address of H_i to its own; acquires ip_address and public key of successor group head; encrypts the modified message using the acquired public key Pub_s where s is the successor group head and forwards it.

c. Message propagation continues similarly in the 1st level ring.

else

Receiver group head drops the message.

Fig. 9 Multicast Algorithm 3 with Security and Anonymity

Example 3:

$c_{si} < n_r$ (Case2)

Suppose #group heads (n_r) = 8

Capacity of each group head (c^s_i) = 3

In this example assume that the source peer is $G^s_{2,\ 23}$

In Fig 10, H_2 selects any 3 group heads in random (say H_4, H_6, H_7)

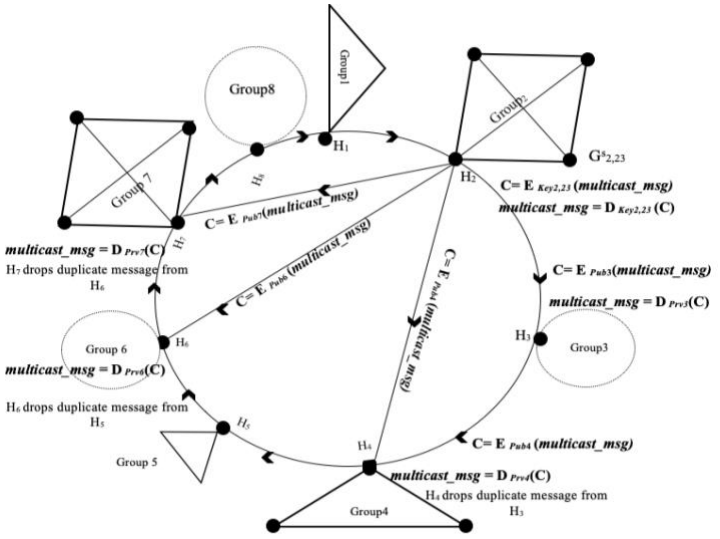

Fig. 10 Multicast Algorithm 3 example with Cryptographic messages

VI. CONCLUSION

In this paper, CRT based structured P2P architecture [9], [16] has been considered. We have designed secured data lookup protocols and secured multicast algorithms both inter and intra. In addition, we also considered the anonymity in case of multicasting.

REFERENCES

[1] Y.H. Chu, S. Rao, S. Seshan, and H. Zhang, "A Case for End System Multicast," IEEE J. Selected Areas in Comm, vol. 20, no. 8, Oct. 2002.

[2] J. Jannotti, D. Gifford, K. Johnson, M. Kaashoek, and J. O'Toole, "Overcast: Reliable Multicasting with an Overlay Network," Proc. Symp. Operating Systems Design and Implementation (OSDI'00), Oct. 2000.

[3] C.K.S. Banerjee and B. Bhattacharjee, "Scalable Application Layer Multicast," Proc. ACM SIGCOMM'02, Aug. 2002.

[4] J. A. Dejan Kosti, A. Rodriguez, and A. Vahdat, "Bullet: High Bandwidth Data Dissemination Using an Overlay Mesh," Proc. Symp. Operating Systems Principles (SOSP,03), Oct. 2003.

[5] S. Banerjee, C. Kommareddy, B.B.K. Kar, and S. Khuller, "Construction of an Efficient Overlay Multicast Infrastructure for Real-Time Applications," Proc. INFOCOM'03, Mar. 2003.

[6] A. Riabov and L. Z. Zhen Liu, "Overlay Multicast Trees of Minimal Delay," Proc. Int'l. Conf. Distributed Computing Systems (ICDCS)'04), Mar. 2004.

[7] R. Zhang and Y. C. Hu, "Borg: A Hybrid Protocol for Scalable Application-Level Multicast in Peer-to-Peer Networks," Proc. Int'l. Workshop Network and Operating System Support for Digital Audio and Video (NOSSDAV'03), 2003.

[8] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Application Level Multicast Using Content-Addrerssable Networks," Proc. Int'l. Workshop Networked Group Comm (NGC'01), 2001.

[9] Bidyut Gupta, Nick Rahimi, Henry Hexmoor, Shahram Rahimi, Koushik Maddali, and Gongzhu Hu, Design of Very Efficient Lookup Algorithms for a Low Diameter Hierarchical Structured Peer-to-Peer Network, Proc. IEEE 16th Int. Conf. Industrial Informatics (IEEE INDIN), July 2018, Porto, Portugal.

[10] Shiping Chen, Baile Shi, Shigang Chen, and Ye Xia, "ACOM: Any-Source Capacity-Constrained Overlay Multicast in Non-DHT P2P networks," IEEE Tran. Parallel and Distributed Systems, vol. 18, no. 9, pp. 1188-1201, Sep. 2007.

[11] 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 (IEEE INDIN), Poitiers, France, July, 2016.

[12] 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.

[13] Stephen E. Deering and David R. Cheriton, "Multicast Routing in Datagram Internetworks and Extended LANs", ACM Trans. on Computer Systems (TOCS), Vol. 8, No. 2, pages. 85-110, May 1990.

[14] Tony A. Ballardie, "Core Based Tree Multicast Routing Architecture", Internet Engineering Task Force (IETF), RFC 2201, September 1997.

[15] 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.

[16] Indranil Roy, Koushik Maddali, Swathi Kaluvakuri, Benafsheh Rekabdar, Ziping Liu, Bidyut Gupta and Narayan C. Debnath, Efficient Any Sourc Overlay Multicast in CRT- Based P2P Networks- A Capacity- Constrained Approach, Proc. IEEE 17th Int. Conf. Industrial Informatics (IEEE INDIN), July 2019, Helsinki-Espoo, Finland.