EPiC Series in Computing
Volume 91, 2023, Pages 116–123
Proceedings of 38th International Conference on Computers and Their Applications

Infrastructure Development of Non DHT-Based Pyramid Tree Network Architecture

Indranil Roy[1], Nick Rahimi[2], Bidyut Gupta[3], and Narayan Debnath[4]
[1] Department of Computer Science, Southeast Missouri State University, Cape Girardeau, MO
iroy@semo.edu
[2] School of Computing Sciences & Computer Engineering, University of Southern Mississippi, Hattiesburg, MS
nick.rahimi@usm.edu
[3] School of Computing, Southern Illinois University; Carbondale, IL
bidyut@cs.siu.edu
[4] School of Computing and Information Technology, Eastern International University; Vietnam
ndebnathc@gmail.com

Abstract

In this paper, we have considered a recently reported 2-layer non-DHT-based structured P2P network. It is an interest-based system and consists of different clusters such that peers in a given cluster possess instances of a particular resource type. It offers efficient data look-up protocols with low latency. However, the architecture lacks in one very important aspect: it is assumed that no peer in any cluster can have more than one resource type, and this could be a very hard restriction practically. This is true for all interest-based works existing in the literature. Therefore, in the present work, we have addressed this issue of generalizing the architecture to overcome this restriction and so far, have come up with some significant initial results. Work is being on to complete the generalization process. We have identified some of our previously reported data look-up protocols that will need to be modified in order to accommodate the new findings toward the generalization and while doing so, we aim at keeping the data look-up latencies of these probable modified protocols unchanged. In addition, our objective is to consider security in communication in the generalized architecture as well. To achieve it, we aim at using mainly public key-based approach for the different look-up protocols reported earlier, because results obtained so far in this direction indicate that the required number of public-private key pairs will be much smaller than the number of symmetric keys if symmetric key-based approach is used.

1 Introduction

Recent trend in designing structured P2P architectures is the use of distributed hash tables (DHTs) [2]-[4]. 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 this effort while still providing an efficient data query service. In this direction, there exist several important works, which have considered designing DHT-based hybrid systems [1], [6], [9]-[11]; these works attempt to include the advantages of both structured and unstructured architectures. However, these works have their own pros and cons. Another design approach has attracted much attention; it is non-DHT based structured approach [13], [14], [16], [18]. It offers advantages of DHT-based systems, while it attempts to reduce the complexity involved in churn handling. Authors in [16] have considered one such approach and have used an already existing architecture, known as Pyramid tree architecture originally applied to the research area of 'VLSI design for testability' [12]. It is an interest-based peer-to-peer system [14] – [17], [20] with peers of common interest are clustered together. Its main focus is to improve the efficiency of data lookup protocols in that a query for an instance of a particular resource type is always directed to the cluster of peers which possess different instances of this resource type. So, success or failure to get an answer for the query involves a search in that cluster only instead of searching the whole overlay network as in the case of unstructured networks. However, that a peer can have only one resource type is a hard restriction practically.

Our Contribution

In the present work, as a continuation of our research in Pyramid tree P2P network area, we present some of our recently obtained significant results towards designing the generalized form of the architecture that will allow any peer to possess multiple different resource types Work is going on to complete the generalization process. We have identified some of our previously reported data look-up protocols that will need to be modified in order to accommodate the new findings toward the generalization and while doing so, we aim at keeping the data look-up latencies of these probable modified protocols unchanged. In addition, our objective is to consider security in communication in the generalized architecture as well. To achieve it, we aim at using mainly public key-based approach for the different look-up protocols reported earlier, because results obtained so far as reported in this paper in our current research indicate that the required number of public-private key pairs will be much smaller than the number of symmetric keys if symmetric key-based approach is used.

The organization of the paper is as follows. In Section 2, we talk briefly about some related preliminaries. Our contributions in the present paper appear in Sections 3 and 4. In Section 3, results obtained so far on the generalization of the architecture have been presented. In Section 4, effect of generalization on the existing communication protocols has been considered and some initial findings related to the security issue have been presented. Section 5 draws the conclusion.

2 Preliminaries

In this section, we present some relevant results from our recent works on the Pyramid tree based P2P architecture [16], [18], [19] for interest-based peer-to-peer system. Residue Class based on modular arithmetic has been used to realize the overlay topology.

Definition 1. *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.*

Note that 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'.

Definition 2. *Let S be the set of all peers in a peer-to-peer system with n distinct resource types (i.e. n distinct common interests). Then* $S = \{C_i\}$, $0 \le i \le n-1$, *where* C_i *denotes the subset consisting of all peers with the same resource type* R_i. *In this work, we call this subset* C_i *as cluster i. Also, for each cluster* C_i, *we assume that* C_i^h *is the first peer among the peers in* C_i *to join the system. We call* C_i^h *as the cluster-head of cluster* C_i.

The overlay network considered is a 2-layer non DHT based architecture. At layer-1, there exists a tree like structure, known as pyramid tree. It is not a conventional tree. A node *i* in this tree represents the cluster-head of a cluster of peers which possess instances of a particular resource type R_i (i.e., peers with a common interest). The cluster-head is the first among these peers to join the system. Layer 2 consists of the different clusters corresponding to the cluster-heads.

2.1 Characteristics of pyramid ree

The following overlay architecture has been proposed in [16], [18], [19].

- The tree consists of n nodes. The i^{th} node is the i^{th} cluster head C_i^h. The tree forms the layer-1 and the clusters corresponding to the cluster-heads form the layer-2 of the architecture.
- Root of the tree is at level 1.
- Edges of the tree denote the logical link connections among the n cluster-heads. Note that edges are formed according to the pyramid tree structure [12].
- A cluster-head C_i^h represents the cluster C_i. Each cluster C_i is a completely connected network of peers possessing a common resource type R_i, resulting in the cluster diameter of 1.
- The tree is a complete one if at each level j, there are j number of nodes (i.e. j number of cluster-heads). It is an incomplete one if only at its leaf level, say k, there are less than k number of nodes.
- Any communication between a peer $p_i \in C_i$ and a peer $p_j \in C_j$ takes place only via the respective cluster-heads C_i^h and C_j^h and with the help of tree traversal wherever applicable.
- Joining of a new cluster always takes place at the leaf level.
- A node that does not reside either on the left branch or on the right branch of the root node is an internal node.
- Degree of an internal non-leaf node is 4.
- Degree of an internal leaf node is 2.

2.2 Residue Class

Modular arithmetic has been used to define the pyramid tree architecture of the P2P system.

Consider the set S_n of nonnegative integers less than n, given as $S_n = \{0, 1, 2, \dots (n-1)\}$. This is referred to as the set of residues, or residue classes (mod n). That is, each integer in S_n represents a residue class (RC). These residue classes can be labelled as [0], [1], [2], …, [n – 1], where [r] = {a: a is an integer, a ≡ r (mod n)}.

For example, for n = 3, the classes are:

$[0] = \{\dots, -6, -3, 0, 3, 6, \dots\}$
$[1] = \{\dots, -5, -2, 1, 4, 7, \dots\}$
$[2] = \{\dots, -4, -1, 2, 5, 8, \dots\}$

In the P2P architecture, each integer representing a residue class is the logical (overlay) address of the cluster-head of a cluster. For example, logical address of the first cluster-head is 0, for the second one it is 1, and so on. We use the integers belonging to different classes as the logical (overlay) addresses of the peers with a common interest (i.e. peers in the same cluster) and the number of residue classes is the number of distinct resource types; for the sake of simplicity only the positive integer values are used for addressing. It becomes clear that mathematically any class consists of infinite number of integers; it means that we do not put any limit on the size of a cluster. In general, number of peers can be too large compared to the number of distinct resource types.

An example of a complete pyramid tree of 5 levels is shown in Figure 1. It means that it has 15 nodes/clusters (clusters 0 to 14, corresponding to 15 distinct resource types owned by the 15 distinct clusters). It also means that residue class with *mod 15* has been used to build the tree. The nodes' respective logical addresses are from 0 to 14 based on their sequence of joining the P2P system.

Each link that connects directly two nodes on a branch of the tree is termed as a *segment.* In Figure 1, a bracketed integer on a segment denotes the difference of the logical addresses of the two nodes on the segment. It is termed as *incremen*t and is denoted as *Inc* This increment can be used to get the logical address of a node from its immediate predecessor node along a branch. For example, let X and Y be two such nodes connected via a segment with increment *Inc*, such that node X is the immediate predecessor of node Y along a branch of a tree which is created using *residue class with mod n.* Then, *logical address of Y = (logical address of X + Inc) mod n*.

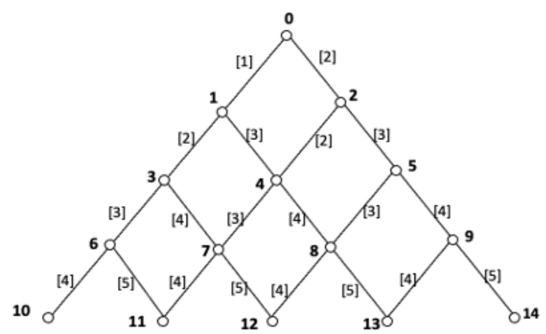

Figure 1: A complete pyramid tree with root 0

Thus, in the example of Figure 1, Logical address of the leftmost leaf node = (logical address of its immediate predecessor along the left branch of the root + Inc) mod 15 = (6 + 4) mod 15 = 10.

3 Generalization of the Architecture

As mentioned earlier, in the architecture, it is assumed that no peer can have more than one resource type, and this could be a very hard restriction practically. To overcome this restriction, we have come up with the concept of *Generalization* i.e., the architecture is generalized in such a way that a peer can have multiple resource types. Generalization of the Architecture needs to deal with two possible scenarios. Below we consider the two possible scenarios and state how to incorporate some necessary changes in the architecture in order to handle the two scenarios. Throughout our presentations, we shall interchangeably use the words 'node' and 'cluster-head'. So, a node on the tree is actually a cluster-head. These are all peers though. However, we strictly use the word 'peer' to represent members of a cluster only to avoid any possible confusion. In addition, we assume that 'resource with type k' and 'resource with code k' mean the same resource.

3.1 Peer with Multiple Existing Resource Types

Scenario 1: Let us consider a situation that in some cluster C_i, its cluster-head C_i^h or a peer p in C_i wants data insertion of another existing resource type, say R_k in the network. Here data-insertion by a peer means the peer in question declares the possession of instances of another resource type that already exists in the system. As mentioned earlier peers in cluster C_k possess instances of the resource type R_k. Also, peer p in C_i already possesses some instances of the resource type R_i.

Solution: The solution for this scenario is as follows. The cluster-head C_i^h or peer p will now become a member of cluster C_k as well. So, it is understood that the IP address of C_i^h /p will be known to members of both the clusters C_i and C_k. It means that, in the overlay network, C_i^h /p will appear logically in both the clusters C_i and C_k, and will have direct logical connections to all member peers of clusters C_i and C_k. Therefore, it should be clear that our already reported intra and inter-cluster data-lookup protocols [22], [19] do not need any modification in this scenario. Same is true for broadcasting involving the cluster-heads in the tree. In addition, we have observed that the capacity-constrained broadcast and multicast protocols inside a cluster [18] in the tree need not be modified as well.

3.2 Existing peers declaring new resource types

Scenario 2*:* Consider a P2P interest-based pyramid tree system which has currently r distinct resource types, viz., R_0, R_1, R_2, … R_{r-1}. Assume that the respective resource codes are 0, 1, 2, …, (r-1). Without any loss of generality, let us assume a scenario where cluster-head C_i^h / a peer p in a cluster C_i wants a data insertion of a new resource type R_r currently not present in the network.

Solution: Solution lies in an appropriate modification of the table of information (*TOI*) maintained by each cluster-head [18]. We know that in *TOI*, corresponding to each cluster-head there is an entry (tuple). For example, the tuple for some cluster-head C_i^h appears as <resource code (logical address) owned by peers in C_i^h, IP address of the cluster-head C_i^h >; note that in the architecture resource code and the logical address of a cluster-head are same. That is, one denotes the other. It facilitates packet propagation in the tree. In short, we write the tuple as < Res. Code, IP (C_i^h) >. As new clusters are formed owing to peers joining with new resource types, the *TOI* grows dynamically, and the newest joining cluster-head is assigned with the next largest logical address not yet used and hence its resource code also becomes the largest among all such existing codes. Therefore, this table remains sorted with respect to logical addresses of cluster-heads (i.e., with respect to the resource codes of the resources they possess).

Coming back to the second scenario, a new entry is made in the *TOI* corresponding to the new resource type R_r with resource code r. So currently this code r is the largest one present in the table. Based on if it is the cluster-head C_i^h / or a peer in cluster C_i that wants a data insertion of a new resource type R_r, in the newly entered tuple, the corresponding cluster-head will be either C_i^h or the peer p. That is, if it is C_i^h, it will now represent two different clusters corresponding to two different resource types i and r; So, it will have two different logical addresses i and r as well. Therefore, later any peer wishing to join with resource type r will join the cluster with logical address r. Effectively, C_i^h now will maintain two different clusters C_i and C_r, i.e., one with peers for resource code i and the other with peers with resource code r. It is clear that cluster-head in the second case with resource type r is now C_r^h which is actually C_i^h. In case it is the peer p in cluster C_i, peer p will maintain a cluster of peers with resource type r; thus, p will appear as a peer in Cluster C_i and will also appear as a cluster-head C_r^h with logical address r. Therefore, we have modified the *TOI* to include the relevant information of the new entry. Below we give an example to clear the idea further.

Observation 1. *Generalization of the architecture may require some nodes of the tree represent multiple cluster-heads with the same IP address, but with different distinct resource types.*

We have observed that the existing inter-cluster data look-up protocol as well as the broadcast protocol involving all cluster-heads in the tree [18] will need some appropriate modifications to handle the second scenario. Work is underway to modify these existing protocols.

4 Public and Symmetric Key-Based Secured Communication

We have observed that use of the *TOI* is a must in designing secured protocols in the generalized architecture. Besides, we think that we need to use the broadcast protocol (yet to be designed for the generalized architecture) in the secured protocols wherever needed.

In addition, we will consider using public key-based approach for most of the existing protocols because we think that the number of public-private key pairs will be reasonably smaller than the number of symmetric keys required for the symmetric key-based approach. We also aim to observe if there is any significant advantage for using a combination of both public and symmetric keys, especially in the case of designing a secured multicast protocol.

We have found that prior to designing the secured protocols, first of all we need to consider the following two possible situations for the distribution of public keys among cluster-heads. Same is true for symmetric keys if symmetric key-based approach is followed.

Situation 1: Resource type of a peer p does not exist in *TOI*

Situation 2: Resource type of a peer p exists in *TOI*

Analyzing these two situations we have come up with the following observations.

Observation 2: Considering both Situations 1 and 2, total number of required Public-Private key pairs (N') is the number of cluster-heads present in TOI. Some peer may appear as cluster-head multiple times in a generalized situation; thus, depending on the situation this peer may possess more than one public-private key pair.

Observation 3: Number of required symmetric keys (N'') for secured communication inside a cluster is the number of peers present in the cluster not counting the cluster-head. This number N'' is much larger than N' because number of peers in a cluster is supposed to be very large compared to the number of distinct resource types [22].

As soon as we complete our work on the generalization of the architecture along with all its protocols, we shall use these observations to complete our works on making the protocols in the generalized architecture secured with either public key approach only, or a hybrid combination of public and symmetric key approach.

5 Conclusion

Authors [14], [16], [19], [20]. have exploited the architectural properties of the Pyramid tree P2P network to design different communication protocols with reasonably low search latency. However, these recent contributions still lack in one very important aspect: in the architecture, it is assumed that no peer can have more than one resource type, and this could be a very hard restriction practically. In the present work, we have addressed this issue of generalizing the architecture and have come up with some significant initial results toward defining a generalized architecture; it is reflected in the modified 'Table of Information'; we think that the various protocols for data/query propagation will need to consult with this table of information. Work is going on in this direction. We have observed that generalization has no effect on either of the existing intra-cluster protocol; only inter-cluster data look up and the broadcast protocol need to be modified. In addition, we have also considered security in communication in the generalized architecture. To achieve it, we aim at using mainly public key-based approach for the different look-up protocols reported earlier. because results obtained so far in this direction indicate that the required number of public-private key pairs will be much smaller than the number of symmetric keys if symmetric key-based approach is considered. We also aim to observe if there is any significant advantage for using a combination of both public and symmetric keys, especially in the case of designing a secured multicast protocol.

References

[1] P. Ganesan, Q.Sun, and H. Garcia-Molina, “Yappers: A peer-to-peer lookup service over arbitrary topology,” Proc. IEEE Infocom 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] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, “A scalable content- addressable network, CAN,” Vol. 31, No. 4, pp. 161-172, ACM, 2001.

[4] A. Rowstron and P. Druschel, “Pastry: Scalable, Distributed Object Location and Routing for Large Scale Peer-to-Peer Systems”, Proc. FIP/ACM Intl. Conf. Distributed Systems Platforms (Middleware), pp. 329-350, 2001.

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

[10] 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.

[11] M. Kleis, E. K. Lua,, and X. Zhou, “ Hierarchical Peer-to-Peer Networks using Lightweight SuperPeer Topologies”, Proc. IEEE Symp. Computers and Communications, 2005.

[12] Bidyut Gupta and Mohammad Mohsin, “Fault-Tolerance in Pyramid Tree Network Architecture,” J. Computer Systems Science and Engineering, Vol. 10, No. 3, pp. 164-172, July,1995.

[13] Bidyut Gupta, Nick Rahimi, Shahram Rahimi, and Ashraf Alyanbaawi, “Efficient Data Lookup in Non-DHT Based Low Diameter Structured P2P Network,” Proc. IEEE 15th Int. Conf. Industrial Informatics (IEEE INDIN), July 2017, Emden, Germany.

[14] N. Rahimi, K. Sinha, B. Gupta, and S. Rahimi, “LDEPTH: A Low Diameter Hierarchical P2P Network Architecture”, Proc. IEEE 14th Int. Conf. on Industrial Informatics (IEEE INDIN), Poitiers, France, July 2016.

[15] Indranil Roy, Koushik Maddali, Swathi Kaluvakuri, Banafsheh Rekabdar, Ziping Liu, Bidyut Gupta, Narayan Debnath, “Efficient Any Source Overlay Multicast In CRT-Based P2P Networks ─ A Capacity - Constrained Approach”, Proc. IEEE 17th Int. Conf. Industrial Informatics (IEEE INDIN), July 2019, Helsinki, Finland.

[16] Indranil Roy, Bidyut Gupta, Banafsheh Rekabdar, and Henry Hexmoor, “A Novel Approach Toward Designing A Non-DHT Based Structured P2P Network Architecture”, EPiC Series in Computing, Vol. 63, 2019, pages 121-129, Proceedings of 32nd Int. Conf. Computer Applications in Industry and Engineering.

[17] 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.

[18] Indranil Roy, Swathi Kaluvakuri, Koushik Maddali, Abdullah Aydeger, Bidyut Gupta, and Narayan Debnath, Capacity Constrained Broadcast and Multicast Protocols for Clusters in Pyramid Tree-based Structured P2P Network, IJCA, Vol. 28, No. 3, Sep. 2021.

[19] I. Roy, N. Rahimi, K. Maddali, S. Kaluvakuri, B. Gupta, and N. Debnath, “Design of Efficient Broadcast Protocol for Pyramid Tree-based P2P Network Architecture”, EPiC Series in Computing, *Proceedings of 33rd Int. Conf. Computer Applications in Industry and Engineering, San Diego*, 63:182-188, 2020.

[20] S. K. A. Khan and L. N. Tokarchuk, "Interest-Based Self Organization in Group-Structured P2P Networks," 2009 6th IEEE Consumer Communications and Networking Conference, Las Vegas, NV, 2009, pp. 1-5, doi: 10.1109/CCNC.2009.4784959.

[21] Indranil Roy, Swathi Kaluvakuri, Koushik Maddali, Ziping Liu, and Bidyut Gupta, "Efficient Communication Protocols for Non DHT-based Pyramid Tree P2P Architecture," WSEAS Transactions on Computers, Vol. 20, pp. 108-125, July 2021 (Invited paper).

[22] Koushik Maddali, Indranil Roy, Swathi Kaluvakuri, Bidyut Gupta, Narayan Debnath, "Design of Broadcast protocols for Non DHT-based Pyramid Tree P2P Architecture", IJCA, Vol. 28, No. 4, pp. 193- 203, December 2021.