Design of a Hybrid Interest-Based Peer-to-Peer Network Using Residue Class-based Topology and Star Topology

Indranil Roy*
Southeast Missouri State University, Cape Girardeau, USA
Swathi Kaluvakuri†
Enterprise Fleet Management, St. Louis, Missouri, USA.
Nick Rahimi ‡
University of Southern Mississippi, USA
Bidyut Gupta §
Southern Illinois University; Carbondale, IL, USA.
Narayan C. Debnath ¶
Eastern International University, Vietnam.

Abstract

In this work, a hybrid interest-based peer-to-peer network is presented. It consists of the existing Residue-class (RC) based peer-to peer (P2P) network and the Star interconnection network. The former one has been considered because of its two main advantages viz.(1) all peers with the same interest (or possessing same resource type) structurally form a group of diameter one, and (2) the group heads are connected in the form of a ring and the ring always remains connected even in presence of any churn. The Star topology has been considered because of its low diameter. We have incorporated these above advantages of the two architectures in the presented hybrid topology in a way that the diameter of the hybrid topology becomes comparable to the diameter of the star topology. In fact, it is equal to the diameter of the star topology plus 2. The proposed hybrid topology takes the shape of an n-star, and a remarkable improvement of data lookup latency with message complexity of $O(n)$is achieved, where n is the number of group heads, and n is usually very small compared to the total number of peers in the network.

Key Words: Residue Class, Interest-based, P2P Network, Star Interconnection Network, Diameter, Unicast Query, Lookup Latency.

*Department of Computer Science, Southeast Missouri State University, Cape Girardeau, MO. Email: iroy@semo.edu.

†Enterprise Fleet Management, St. Louis, Missouri.Email: Swathi.kaluvakuri@efleets.com.

‡School of Computing Sciences Computer Engineering, University of Southern Mississippi, Hattiesburg, MS. Email: nick.rahimi@usm.edu.

§School of Computing, Southern Illinois University; Carbondale, IL. Email: bidyut@cs.siu.edu.

¶School of Computing and Information Technology, Eastern International University, Vietnam. Email: NdebnathC@gmail.com .

1 Introduction

Peer-to-peer (P2P) overlay networks are widely used in distributed systems due to their ability to provide computational and data resource sharing capability in a scalable, self-organizing, distributed manner. There are two classes of P2P networks: unstructured and structured ones. In unstructured systems [1] peers are organized into arbitrary topology. It takes help of flooding 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. Besides, in unstructured networks, lookups are not guaranteed. On the other hand, structured overlay networks provide deterministic bounds on data sential 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) [2]-[4]. Such overlay architectures can offer efficient, flexible, and robust service [2]-[6]. 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 DHT-based hybrid systems [17]-[21]; 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 [27]. It offers advantages of DHT-based systems, while it attempts to reduce the complexity involved in churn handling.

There exists yet another important approach; it is interest-based P2P networks. It is a structured approach without using DHT. Therefore, it does not have the problems that DHT based architecture faces, while at the same time it supports the deterministic behavior for data lookup. There exist several works in the literature in this direction [7]-[10], [14]-[16], [22]-[26].

It is an interest-based system. Two of the most prominent advantages relevant to the present work are: (1) all peers with the same interest (or possessing same resource type) structurally form a group of diameter one, and (2) the group heads are connected in the form of a ring and the ring always remains connected even in presence of any churn. These two advantages help in designing a very efficient data look-up algorithm [10]. However, we observe that there is scope to improve further the latency of the data look-up algorithm; all that is required for this purpose is to reduce the diameter of the ring, because this latency is dependent on the diameter (k/2) of the ring (with k group heads). As k increases, latency of the existing look up algorithm increases as well. We observe that a star topology with same number of nodes (group heads) has much smaller diameter compared to the RC based architecture. Therefore, to reduce the diameter further the work reported in [13] has suggested the possibility of designing a hybrid architecture that combines the structural advantages of both RC-based and star-based topologies [11]. Objective is to reduce data lookup latency which reduces as diameter reduces. Therefore, in this paper, we have considered designing a hybrid architecture consisting of RC-based ones and Star topology; we name this architecture as hybrid star architecture. We have identified several important architectural properties of Star topology and have used them to design an efficient low latency unicast query propagation algorithm suitable for the hybrid architecture. We have made sure that the diameter of the redesigned RC based hybrid architecture which takes the shape of a star network (say n-star) is comparable with that of a non-hybrid n-star network with the same number of nodes (group heads), thereby ensuring remarkable improvement of the data look-up latency. In the worst-case, the total number of message transmissions is only (2n-3) for an n-star hybrid network. In this work, we shall use the words, architecture and topology, interchangeably.

2 RC Based Topology and Star Topology

We first state briefly the architecture of RC based network followed STAR topology.

2.1 RC Based architecture [10]

Definition 1: We define a resource as a tuple $\langle R_i, V \rangle$, 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 'movies' and V' denote a particular actor. Thus $\langle R_i, V' \rangle$ represents movies (some or all) acted by a particular actor V'.

Definition 2: Let S be the set of all peers in a peer-to-peer system. Then $S = \{PR_i\}$, where $0 \leq i \leq n-1$, and PR_i denotes the subset consisting of all peers with the same resource type R_i. The number of distinct resource types present in the system is n. Also, for each subset PR_i, we assume that P_i is the first peer among the peers in PR_i to join the system. We call P_i the group-head of group G_i, which is formed by the peers in the subset PR_i.

A. Two level P2P architecture

It is a two-level overlay architecture and at each level structured networks of peers exist. It is explained below.

1. At level-1, we have a ring network consisting of the peers P_i where $0 \leq i \leq n-1$. The number of peers (i.e., group heads) on the ring is n, which is also the number of distinct resource types. This ring network is used for efficient data lookup and is named the transit ring network.

2. At level-2, there are n completely connected networks (groups) of peers. Each such group, say G_i, is formed by the peers of the subset PR_i, where $0 \leq i \leq n-1$, such that all peers $\in PR_i$ are directly connected (logically) to each other, resulting in a network diameter of 1. Each G_i is connected to the transit ring network via its group-head P_i.

3. Any communication between a peer $p'_i \ \varepsilon \ G_i$ and $p'_j \ \varepsilon \ G_j$ takes place only via the respective group-heads P_i and P_j.

The architecture is shown in Figure 1.

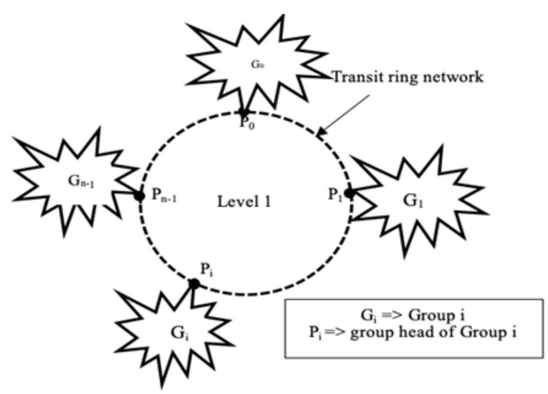

Figure 1: An RC-based P2P network

B. Assignments of overlay addresses

Consider the set S_n of nonnegative integers less than n, given as $S_n = \{0, 1, 2, \ldots, n-1\}$. 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 labeled as $[0], [1], [2], \ldots, [n-1]$, where

$$[r] = \{a : a \text{ is an integer}, a \equiv r \, (\text{mod } n)\}.$$

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

[0] = ..., 6, 3, 0, 3, 6, . . .

[1] = ..., 5, 2, 1, 4, 7, . . .

[2] = ..., 4, 1, 2, 5, 8, . . .

A relevant property of residue class is stated below.

Lemma 1. Any two numbers of any class r of Sn are mutually congruent. Assume that in an interest-based P2P system there are n distinct resource types. Consider the set of all peers in the system given as $S = \{PR_i\}$, where $0 \leq i \leq n-1$. Also, as mentioned earlier, for each subset PR_i (i.e., group G_i), peer P_i is the first peer with resource type R_i to join the system.

In this overlay architecture, the positive numbers belonging to different classes are used to define the following parameters:

1. Logical addresses of peers in a subnet PR_i (i.e., group G_i). The use of these addresses has been shown to justify that all peers in G_i are directly connected to each other (logically), forming an overlay network of diameter 1. In graph theoretic terms, each G_i is a complete graph.
2. Identifying peers that are neighbors to each other on the transit ring network.
3. Identifying each distinct resource type with unique code.

The assignment of logical addresses to the peers at the two levels and the resources happen as follows:

At level-1, each group-head P_r of group G_r is assigned the minimum nonnegative number r of residue class $r \pmod{n}$ of the residue system S_n.

At level-2, all peers having the same resource type R_r will form the group G_r (i.e., the subset P_{R_r}) with the group-head P_r connected to the transit ring network. Each new peer joining group G_r is given the group membership address $r + j \cdot n$, for $j = 1, 2, \ldots$.

3. Resource type Rr possessed by peers in Gr is assigned the code r which is also the logical address of the group-head Pr of group Gr.

4. Each time a new group-head joins, a corresponding tuple <Resource Type, Resource Code, Group Head Logical Address >is entered in the global resource table (GRT).

Definition 3. Two peers Pi and Pj on the ring network are logically linked together if (i + 1) mod n = j. Remark 2. The last group-head $P_n - 1$ and the first group-head P0 are neighbors based on Definition 3. It justifies that the transit network is a ring.

Definition 4. Two peers of a group Gr are logically linked together if their assigned logical addresses are mutually congruent.

Lemma 2. Diameter of the transit ring network is n/2.

Lemma 3. Each group Gr forms a complete graph.

2.2 STAR Architecture [11]

The address of a node in an n-star Sn is identified by a unique permutation of the digits {1, 2, 3, ..., n}. Let fi, $(2 \leq i \leq n-1)$ be a function that maps permutation Pk of the digits 1, 2, 3, ..., n to another permutation P'_k, i.e. $f_i(P_k) = P'_k$ where the first and the i^{th} digits in Pk are interchanged to generate P'_k.

The mapping function f_1 on P_k is such that the first and the nth digits in Pk are interchanged to generate P_m. That is, $f_1(Pk) = P_m$. In other words, the last digit of Pm is identical to the first digit of P_k and vice-versa; meaning thereby that Pk and Pm belong to two different Sn-1 stars which are the components of Sn and the nodes represented by these addresses, i.e., P_k and P_m are directly connected to each other. For example, in a 4-star, any two of the component 3-stars have two links connecting them. Thus, we see that number of nodes in a complete n-star is n!

An example of a complete 4-star, S4 is shown in Figure 2. Note that it has 24 (4!) nodes and each component 3-star, S3 is also a complete one and has 6 (3!) nodes. That is, the S4 has four component complete 3-stars. However, in general, a star may be an incomplete one as well. Later when we use the topological properties of a star to design the modified RC based network, a node in a star means a group head as in the RC based one.

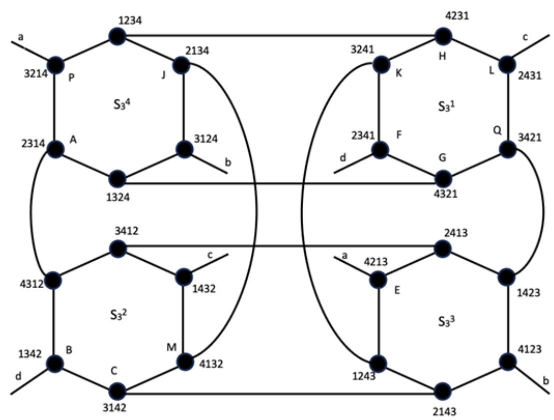

Figure 2: A complete 4-star architecture

2.3 Comparison of the diameters of the two architectures We shall consider number of overlay hops between nodes as the parameter to determine diameters. For this purpose, in RC based one, we consider only the group heads located on the ring. The peers inside a group form a network of diameter 1 and so, we may not need to count it while determining the diameter of the (transit) ring because a group head may not have any other member present at a given time. For Star architecture [11] we consider only the nodes that form the architecture. Later in this paper, we shall propose a hybrid star topology using both RC-based ones and star networks in which a node represents a group head of peers with identical interest. The comparison of diameters of RC-based networks and star interconnection networks is stated below.

From the information stated above, we observe that as the number of group heads/nodes increases, diameter of RC based architecture increases much faster compared to the diameter of a star architecture. Therefore, latency of any data look-up protocol will be much smaller in a Star network compared to that in an RC based network.

RC-based ring network		Star interconnection network		
Number of Group heads	**Diameter**	**Complete Star**	**Number of nodes**	**Diameter**
6	3	3-Star	6	3
24	12	4-Star	24	5
120	60	5-Star	120	7
720	360	6-Star	720	9

Table 1: Comparison of RC-based ring network and Star interconnection network

3 A Six-node Basic RC Component and A Complete 3-Star

We highlight below the similarities and dissimilarities between a six-node basic RC-based component and a complete 3-Star.

Let us consider the two architectures as shown in Figure 3 and Figure 4. We name the first one as a basic RC-based component. The second one shows a complete 3-star. One main reason to consider such an RC based component is that it is very unlikely that there will exist less than six distinct types of resources at any time. Even if it is, as will be clear later that (see Section 4, Figure 5 and Figure 6) the architecture will simply be an RC-based ring with less than six nodes. The structural similarities and dissimilarities of the two architectures are stated below. As is seen in the figures, the overlay addresses in the two architectures are based on RC based idea and star-based idea respectively. The two architectures have the same diameter (3 in terms of hop). So, data look-up latency in both can be at most 3. This is the similarity. Now, consider that node f has left. In Figure 3, the topology remains connected, i.e., it remains a ring [10]. So, performance of any data look-up protocol does not degrade. However, in Figure 4, the same is not true because the topology is a disconnected one resulting in an incomplete 3-star. This is the dissimilarity. This simple and yet important observation will be used in constructing the proposed architecture.

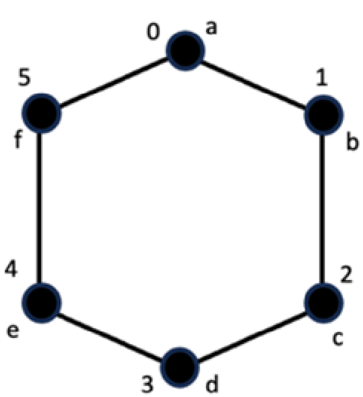

Figure 3: RC-based component

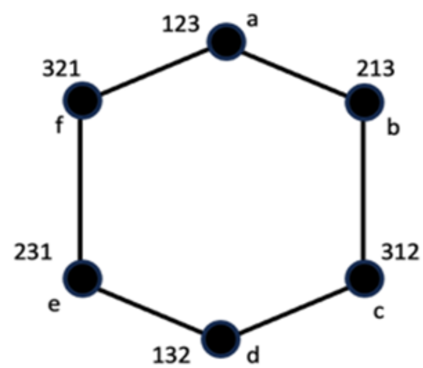

Figure 4: A complete 3-star

4 The Proposed Hybrid Architecture and Some Relevant Properties

In the proposed architecture we assume that it will consist of a number of basic RC-based components and the components will be connected to form a complete n-star. Data look-up inside an RC-based component will follow the existing data look-up protocol reported in [10] using the RC-based overlay addresses of the nodes and inter-component routing (i.e., to send any query from component to component) will use the star-based overlay addresses. Note that in the architecture, every node in the star will be a group-head of peers with the same resource type. We shall use group-head and node interchangeably. The reason for consideration of such intra-component and inter-component query protocols is two-fold: 1) under no circumstances an RC-based component can be disconnected, so there will be no degradation of performance of the look-up process, and the latency of intra-component query look-up will be limited by 3 hops, 2) overlay distance between any two nodes in the hybrid architecture will at most be equal to the diameter of the n-star plus 2; this 2 is due to the fact that diameter of each source group and destination group is 1. Therefore, it results in a much less latency required for the query propagation compared to that in an RC based ring topology with the same number of nodes n!.

In the above architectural design, a node needs to have a tuple of overlay addresses (a1, a2), in which a1 represents the node's RC-based overlay address and a2 represents its star-based overlay address. For example, in Figure 3, the address tuple of node 'a' will now become (0, 123). The first one is used for intra-component query propagation and the second one is used for the inter-component query propagation.

We now illustrate briefly how the proposed architecture is constructed. Let us start with a single node ‘a’. The address tuple of the node ‘a’ is (0, 123). Now consider that a second node ‘b’ joins (Figure 5). The address of this second node is (1, 213). Note that it is now a ring of two nodes (see the dotted link between the two nodes). Now consider that a third node ‘c’ joins. Its address tuple is (2, 312). The basic component structure is still a ring (see the dotted link in Figure 6). However, if we consider a 3-star with these three nodes, it is just incomplete and definitely the performance of any query in it will degrade. This will go on unless the component contains all six nodes. This is the reason which we mentioned earlier, why each component will be constructed using RC based idea. However, when a new node, say ‘g’, (seventh one) joins, it will form the second basic component (of a 4-star) with one node only and the overlay address tuple of ‘g’ will be 4231 while at the same time the second component address of each of the nodes in the first basic component will be updated as xxx4 (see Section 2.2). In addition, based on star topology, node ‘a’ and node ‘g’ will have a direct overlay link for inter-component query propagation. The structure is shown in Figure 7. In this way, using RC based components, star topologies of higher dimensions can be constructed. In this context, it may be mentioned that while the number of the distinct resource types is limited [12], the number of members in a group can be enormously large (theoretically infinite). This is one of the main characteristics (advantages) of the residue class-based design and it has been incorporated in the Star architecture (of the proposed design) consisting of basic RC Based components.

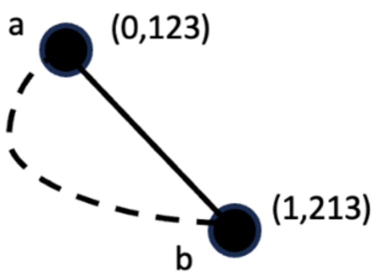

Figure 5: A two-node component

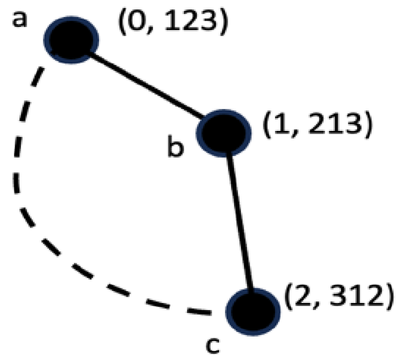

Figure 6: A three-node component

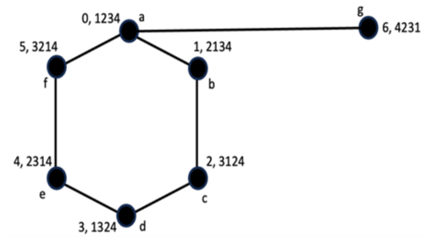

Figure 7: An incomplete 4-star

Note that in general, for an n-star, the star-based overlay address of any node will be a specific permutation of the n literals, viz., 1,2,3,. . . ,n.

We now define ‘neighbors’ of a node x in a basic RC-based component using its star-based overlay address. Let this address be ABC. Then, nodes with addresses BAC and CBA are the neighbors of node x. It means that node x has direct logical connections with these two nodes in the basic component which is also a complete 3-star. These connections can be used by node x for intra-basic component or simply intra-component routing. Now assume that the overlay topology appears as a complete 4-star. So, we now add another literal, say D in the overlay address of node x and the address becomes ABCD. Then, according to star topology node x will be logically connected to node DBCA that belongs to a different 3-star. Therefore, degree on node x increases from 2 in a basic component to 3 in 4-star topology. Thus, node x has three neighbors in 4-star topology. This connection to the 3rd neighbor implies that node x can use this for inter-component routing in the 4-star. Now consider that this 4-star is a part of a complete 5-star. Then, the address of x will consist of 5 literals and so it becomes ABCDE. Then, according to 5-star topology node x will be logically connected to node EBCDA. Therefore, node x now has 4 neighbors. To conclude, in an n-star overlay hybrid architecture, degree of any node is (n-1). Thus, any node can use these addresses (of neighbors) to multicast its information to different basic components of the topology. This idea may be used to incorporate parallelism in broadcasting by any node in the network. In this context, it is understood that the overlay address (be it basic RC-based component or not) of a node will be paired with its IP address.

Example:
In Fig 4, node with address 123 has its neighbors as 213 and 321.

In Fig. 2, the 4-star address of the above node will have 4 literals and it is 1234. Neighbors of the node with 4-star based overlay address 1234 are 2134, 3214, and 4231. The first two are in the same RC-based component as is the node 1234, and the 3rd one is in another RC-based component in the parent 4-star. We name this 3rd one as an inter-component neighbor of the node 1234. For inter component propagation of query/data from node 1234 to node 4231, and vice-versa, this 3rd neighbor is necessary. Note that the 3rd one is obtained by interchanging the first and the 4th literals of 1234.

If it is a 5-star, address of the above node will have 5 literals and it is 12345; its 4 neighbors are 21345, 32145, 42315, and 52341. The first two are in the same RC-based component as is the node 12345. In this case, the 3rd and 4th nodes are the inter-component neighbors of node 12345. The 3rd one is used for propagation of information from node 12345 to a different RC-based component belonging to the same parent 4-star. The 4th neighbor belongs to a different 4-star component of the 5-star topology and its address is obtained by interchanging the first and the 5th literals of 12345; it is used for information propagation between two 4-stars.

In general, a node in an n-star will have (n-1) different neighbors; two of them belong to its parent RC-based component and the rest are one for each higher order star respectively. It may be noted that the n-star is a nested architecture in a sense that it consists of (n-1) stars; each (n-1) star consists of (n-2) stars and so on till it comes down to component 3-stars.

For low latency propagation (limited by the diameter of a given star topology) we consider that the following information to be present with each node in an n-star. Each node maintains the following tables:

1) Resource table for basic component (RT-Basic)

2) Resource tables for next higher order stars (RT-4-Star, RT-5-Star, etc.)

3) Each node maintains a list of its (n-1) neighbors.

Example:
Consider the RC-based component and the 3-star of Figures 3 and 4. The RT-Basic for the nodes in a basic component appears as

Table 2: RT-basic

nodes	resource-code (ar)	3-star-based address (as)
a	0	123
b	1	213
c	2	312
d	3	132
e	4	231
f	5	321

In the above table, RT-based overlay address of a node and the resource code it possesses are the same ones.

Table for the 4-star of Fig. 2 is shown below:

Table 3: RT-4-Star

resource-code-range	4-star-based-address-range
0, 5	xxx4
6, 11	xxx2
12, 17	xxx1
18, 23	xxx3

In this table, each row corresponds uniquely to one 3-star component of a 4-star. In a row, the first entry denotes the lower and the upper code values of the six resources present uniquely with the six nodes belonging to a component 3-star; the second entry denotes that the star-based overlay addresses of the six nodes in it end with the same literal. Note that overlay addresses of the six nodes can be obtained by permutations of the first three literals (denoted as xxx) while the 4th literal is fixed in all these six addresses.

Table 4: RT-5-Star

resource-code-range	5-star-based-address-range
0, 23	xxxx5
24, 47	xxxx2
48, 71	xxxx3
72, 95	xxxx4
96, 119	xxxx1

In this table for a 5-star, each row corresponds uniquely to one 4-star component of a complete 5-star. The corresponding 5-star topology is not shown here; however, the above table actually gives the information of the topology. In a row, the first entry denotes the ranges of resources (resource codes) present in the component, and the second entry denotes that the overlay addresses of the 24 nodes in it end with the same literal. Therefore, we observe that for an i-star topology, it will have i numbers of component (i-1) stars. We denote a component as the m^th component in which the i^th literals in the overlay addresses of all nodes in the component are m. For example, in the above RT-5-star table, there are five 4-star components, and the respective 5^th literals in the overlay addresses of the five component 4-stars are 1, 2, 3, 4, and 5. That is, all nodes in component one will have 1 as the last literal in their overlay addresses; all nodes in component two will have 2 as the last literals in their overlay addresses; and so on.

In a similar way as above, tables can be built for higher order stars. However, it has been shown that in reality the number of distinct resource types is limited [12].

Therefore, we suggest that the design consideration should be restricted to a maximum of 6-star that contains 720 distinct resource types (Section 2.3). If necessary, a federation of 6-stars or even of 5-stars can be built (not covered in this paper). However, the proposed unicast algorithm in Section 5 is a general one that considers any hybrid n-star topology. In this context, we repeat that each node is the group-head of a group of peers possessing similar kind of resources, and the construction of each group is such that it can contain very large number of such peers (theoretically infinite). Note that for an n-star hybrid topology, the total number of rows of the tables together is (6+4+5+6+…+n), i.e., n(n+1)/2. Thus, for a 6-star topology, it will be 21x2 only, which is reasonably quite small for any group-head (node) to store.

5 Data Look-up Algorithm.

Let us present an algorithm for unicast communication. We assume that in an n-star hybrid topology a query for an instance of a resource with code ar' is originated at a node with a tuple of overlay addresses (ar, as). Each node maintains tables such as its parent RT-basic, RT- 4-star, RT-5-star, and so on. Each node maintains a list of its neighbors.

5.1 Algorithm Unicast

We start with a description of searching inside a RC-based component as this will be used in the main body of the unicast algorithm as a subroutine.

Algorithm 1 Intra-RC-Based-Component

if $ar' \in RT$-basic $[ar - ar'] > 3$
Diameter of a RT-based component is 3 *ar* forwards the query to its immediate predecessor
else
ar forwards the query to its immediate successor
for each intermediate receiving node N
N forwards the query until its code is equal to ar'
if$N$ has the answer to the query N unicasts the answer to node *ar* using the IP address of *ar*
else
One-hop communication since the diameter of each group-head is one N broadcasts the query in its group-heads
if $\exists p'$ with ar'
Peer p' unicasts ar' to *ar*
else
Search latency in the group-heads is minimal, i.e., only two hops Search latency in the group-heads is minimal, i.e., only two hops

else
Execute Algorithm Unicast

Algorithm Unicast:

1. The originator node, identified by the tuple of overlay addresses (ar, as), starts searching through the resource tables of higher-order stars.The search continues until a resource with the code ar' is found in some m^{th} component of an $(i-1)$-star of an i-star, where $4 \leq i \leq n$. Since the originator node belongs to a 3-star component, the search begins with its parent 4-star. The search then continues through progressively higher-order stars until the desired resource is located.
2. Node *ar* finds the *i*th literal, i.e., *m*, in the *n*-star-based overlay address of the *m*th component of the $(i-1)$-star.
3. If node ar''s overlay address starts with the literal *m*, node *ar* interchanges the first literal with the *i*th literal of its address. Node *ar* then forwards the query to the node with this modified address in the *m*th component of the $(i-1)$-star, and control flows to step 7.
4. Node *ar* finds its neighbor whose overlay address starts with the literal *m*. Each node in a 3-star component has exactly one neighbor in each higher-order star.
5. Node a_r forwards the query to the neighbor.

6. neighbor node interchanges the first literal with the ith literal of its address and forwards the query to the node with this address in the mth component (i-1)-star.

/ receiving node is an inter-component neighbor of the sending node

7. receiving node in the mth (i-1)-star now becomes the new originator node and the following steps 7.1, 7.2 are executed. /vspace10pt

7.1. new originator node searches its resource table to identify the (i-2)-star that has the resource; steps 2 to 6 are executed with this new originator and the (i-2)-star; this process continues until a 3-star is identified to have the resource

/ Resource tables of lower order stars from (i-2)-star to 4-star are searched

7.2 control flows to Algorithm Intra-RC-Based-Component

Theorem 1. Message complexity of Algorithm Unicast is O(n).

Proof. Proof is constructive. Worst-case appears when in Step 1 of Algorithm Unicast implies that if the resource is initially found in some component (n-1)-star of the n-star hybrid topology, steps 2 to 7.1 will be executed repeatedly until a 3-star is identified to have the resource. During each repetition, the query is sent to an appropriate node in a star of immediate lower dimension. Thus, so far (n-3) message-transmissions take place. Once in a 3-star it takes at most 3 overlay hops (diameter of a 3-star can be at most 3) to reach the final probable destination. So, all together there are [(n-3) + 3) = n] message-transmissions. However, if Steps 4 and 5 are to be executed in each repetition, (n-3) more message transmissions take place. Therefore, total number of message transmissions in the worst-case is n+(n-3), i.e.(2n-3). Hence, the message complexity is O(n).

Example of unicast query propagation

Let us consider the 4-star network of Fig. 2. Let a peer in group j with group-head address tuple $(1, 2134)$ has requested its group-head for a particular instance of a resource with resource code 11. Note that the group-head may itself be the requesting peer. Let us denote this group-head as gj. Group-head gj maintains its tables RT-basic and RT-4-star. It first searches the table RT-basic and does not find the resource 11 in it. It then searches the table RT-4-star, and the 3rd entry (12, 17 xxx1) indicates that resources with codes from 12 to 17 exist with the peers of the RC-based component 3-star in which the 4-star based addresses of all six nodes end with the literal 1. Now g_j finds its neighbor that has its overlay address starting with the literal 1 and it is 1234. Node g_j sends the query to the neighbor 1234.

Node 1234 interchanges the first literal with the 4th literal of its address to obtain 4231 and sends the query to this new node 4231. If node 4231 possesses resource with code 11 and if has the answer to the query, it directly sends the reply to the requesting group-head gj; else it broadcasts the query in its group to learn if a peer in its group has the answer. If so, the peer sends the answer to gj directly; otherwise search fails. On the other hand, if node 4231 has different resource code, it then initiates the searching in its parent RT-based component 3-star. This searching to identify the group-head that owns the resource code 11 takes at most 3 overlay hops as the diameter of any RC-based component can be at most 3. Searching for the correct group-head can also be done without any propagation of the query in the basic component. It can be done by node 4231 via a simple search of the appropriate RT-basic table for the correct owner of the resource with code 11. Node 4231 then has to contact the owner directly informing about the query. However, because of very low diameter of a basic component, query propagation may be a reasonably good idea and therefore, in this work, we have considered the query propagation approach. Finally, as stated above if a peer (be it a peer in a group or a group-head itself) is found to have the answer, it will directly send the answer to the requesting group-head.

5.2 Fault Tolerance

Observe that links in a star topology are physical [11] while in our proposed hybrid architecture using both RC-based and star topology, links are virtual. Hence, there is no need to consider any link failure.

In addition, each node here is actually the group-head of a group of peers with similar kind of resources (interests) and the construction of each group is such that it can contain very large number of such peers (theoretically infinite); therefore, it is very unlikely that a complete group of peers will disappear because of churn [10]. It may be noted that inclusion or exclusion (leaving) of any peer, related to an existing group does not change the diameter of the group (which is one); since peers in a group form a complete graph.

6 Conclusion

In this work we have used some topological properties of Star inter-connection networks to redesign an already existing Residue class-based peer-to-peer architecture. The existing RC-based architecture has been the choice in this work because of its manifold advantages. Specifically, the following two structural properties are behind the choice: these are (1) all peers with the same interest (or possessing same resource type) structurally form a group of diameter one, and (2) the group heads are connected in the form of a ring and the ring always remains connected even in presence of any churn. However, data look-up latency is k/2 for an RC-based network with k group heads. It may appear substantial if k is large. In order to reduce this latency, some pertinent topological properties of Star network have been used to modify the existing RC based design. The proposed hybrid design has much less diameter than in its original design (RC based design). It reduces the data look-up latency remarkably. In an n-star hybrid topology, the overlay distance between any two nodes in the hybrid architecture will at most be equal to the diameter of the n-star network plus 2; this 2 is due to the fact that diameter of each source group and destination group is 1. Therefore, it results in a much less latency required for the query propagation compared to that in an RC based ring topology consisting of the same number of nodes n! as in the n-star hybrid network.

Future works aim at designing secured multicast and broadcast data look-up algorithms in the proposed design.

References

1. 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).
2. 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).
3. 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).
4. R. 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).
5. 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).
6. 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).

7. 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.
8. S.K. Khan and L.N. Tokarchuk, "Interest-based Self Organization in Cluster-Structured P2P Networks," Proc. 6th IEEE Consumer Communications and Networking Conf, Las Vegas, pp. 1-5, 2009.
9. L. Badis, M. Amad, D. Aissani, K. Bedjguelal, and A. Benkerrou, "ROUTIL: P2P Routing Protocol Based on Interest Links," Proc. 2016 Int. Conf. Advanced Aspects of Software Eng., Constantine, pp. 1-5, 2016.
10. Swathi Kaluvakuri, Koushik Maddali, Nick Rahimi, Bidyut Gupta, and Narayan Debnath, "Generalization of RC-based Low Diameter Hierarchical Structured P2P Network Architecture," IJCA, vol. 27, no. 2, June 2020, pp. 74-83.
11. S. B. Akers and B. Krisnamurthy, "A Group Theoretic Model for Symmetric Interconnection Networks," IEEE Trans. Computers, vol. 38, no. 4, pp. 555-566, 1989.
12. Cheng, J., & Donahue, R. (2013), "The pirate bay torrent analysis and visualization," International Journal of Science, Engineering and Computer Technology, 3(2), 38.
13. Indranil Roy, Reshmi Mitra, Swathi Kaluvakuri, Nick Rahimi, and Bidyut Gupta, "On Further Improvement of Data Look-Up Latency in Residue Class (RC) Based Peer-to-Peer Networks," EPiC Series in Computing, volume 98, pp. 69-76, (Proc. 39th CATA 2024, March).
14. H. Shen, G. Liu and L. Ward, "A Proximity-Aware Interest-Clustered P2P File Sharing System," IEEE Transactions on Parallel and Distributed Systems, 26(6):1509-1523, doi: 10.1109/TPDS.2014.2327033, 1 June 2015.
15. I. Roy, S. Kaluvakuri, K. Maddali, Z. Liu, and B. Gupta, "Efficient Communication Protocols for Non-DHT-Based Pyramid Tree P2P Architecture," WSEAS Transactions on Computers, (Invited paper), 20:108-125, July 2021.
16. I. Roy, B. Gupta, B. Rekabdar, and H. Hexmoor, "A Novel Approach Toward Designing a Non-DHT Based Structured P2P Network Architecture," EPiC Series in Computing, Proceedings of 32nd Int. Conf. Computer Applications in Industry and Engineering, 63:121-129, 2019.
17. 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, pp. 1250-1260, Vol. 2, March 30 - April 1 2003.
18. 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.
19. Z. Peng, Z. Duan, J. Jun Qi, Y. Cao, and E. Lv, "HP2P: A Hybrid Hierarchical P2P Network," Proc. Intl. Conf. Digital Society, pp. 86-90, 2007.
20. K. Shuang, P Zhang, and S. Su, "Comb: Resilient and Efficient Two-Hop Lookup Service for Distributed Communication System," Security and Communication Networks, vol. 8(10), pp. 1890-1903, 2015.
21. M. Kleis, E. K. Lua, and X. Zhou, "Hierarchical Peer-to-Peer Networks Using Lightweight SuperPeer Topologies," Proc. IEEE Symp. Computers and Communications, pp. 1-6, 2005.
22. 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), pp. 944-950, July 2017, Emden, Germany.
23. 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, pp. 832-837, July 2016.
24. 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, pp. 1351-1357, Finland.
25. 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, Volume 63, 2019, pages 121-129, (Proceedings of 32nd Int. Conf. Computer Applications in Industry and Engineering).
26. Indranil Roy, Nick Rahimi, Koushik Maddali, Swathi Kaluvakuri, Bidyut Gupta, and Narayan Debnath, "Design of Efficient Broadcast Protocol for Pyramid Tree-Based P2P Network Architecture," EPiC Series in Computing, Volume 63, 2020, pages 182-188, (Proceedings of 33rd Int. Conf. Computer Applications in Industry and Engineering, San Diego).
27. 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.

Authors

Indranil Roy is an Assistant Professor in the Department of Computer Science at Southeast Missouri State University. He received his MS and Ph.D. degrees in Computer Science from Southern Illinois University, Carbondale in 2018 and 2022, respectively. His current research interests include the design of architecture and communication protocols for structured peer-to-peer overlay networks, security in overlay networks, and Blockchain.

Swathi Kaluvakuri holds a Doctor of Philosophy (Ph.D.) degree in Computer Science from Southern Illinois

University Carbondale, USA, 2021. She graduated from Jawaharlal Nehru Technological University with a Bachelor of Technology degree in Computer Science major in 2018. Her extensive knowledge and expertise in areas such as Blockchain networks, data analysis, Machine Learning, and Cybersecurity have positioned her as a Sr. Software Engineer at Enterprise Fleet Management, R & D wing. She leads the Research & Development wing in the area of St. Louis, Missouri, USA.

Nick Rahimi is the Director of Cyber Innovations Lab and an Assistant Professor at the School of Computing Sciences & Computer Engineering of the University of Southern Mississippi (USM). Dr. Rahimi obtained two Bachelor of Science degrees in Computer Software Engineering and Information Systems Technologies with a concentration in Cybersecurity and received his Master and Ph.D. degrees in Computer Science from Southern Illinois University (SIU). His research interests lie in the area of cybersecurity, blockchain, cryptography, internet anti-censorship, machine learning in cybersecurity, distributed systems, and decentralized networks. Prior to joining USM, Dr. Rahimi was a tenure track Assistant Professor at SIU for 2 years and Southeast Missouri State University (SEMO) for one year respectively. Moreover, he has over eight years of experience in industry as a software engineer and team leader.

Bidyut Gupta is currently a Professor of Computer Science at the School of Computing, Southern Illinois University at Carbondale. His research interests include fault-tolerant distributed computing, design of P2P network architectures with low latency communication protocols, fog computing and its applications, and high latency networks. He is a Senior member of IEEE and ISCA.

Narayan C. Debnath is currently the Founding Dean of the School of Computing and Information Technology at Eastern International University, Vietnam. He is also serving as the Head of the Department of Software Engineering at Eastern International University, Vietnam. Formerly, Dr. Debnath served as a Full Professor of Computer Science at Winona State University, Minnesota, USA for 28 years, and the elected Chairperson of the Computer Science Department at Winona State University for 7 years. Dr. Debnath has been the Director of the International Society for Computers and their Applications (ISCA), USA since 2014. Professor Debnath made significant contributions in teaching, research, and services across the academic and professional communities. He has made original research contributions in software engineering, artificial intelligence and applications, and information science, technology, and engineering. He is an author or co-author of over 500 research paper publications in numerous refereed journals and conference proceedings in Computer Science, Information Science, Information Technology, System Sciences, Mathematics, and Electrical Engineering.