A publication of ISCA*:
International Society for Computers
and Their Applications

INTERNATIONAL JOURNAL OF COMPUTERS AND THEIR APPLICATIONS

TABLE OF CONTENTS

Page

Guest Editoral: Special Issue from CAINE 2020 121
Gongzhu Hu, Yan Shi, and Takaaki Goto

Capacity Constrained Broadcast and Multicast Protocols for Clusters in a Pyramid Tree-based Structured P2P Network 122
Indranil Roy, Swathi Kaluvakuri, Koushik Maddali, Abdullah Aydeger, Bidyut Gupta, and Narayan Debnath

Validation of the Improved Increment Asssignment Algorithm 132
Pinzhi Wang, Youssou Gningue, and Haibin Zhu

Efficient Secured Data Lookup and Multicast Protocols with Anonymity in RC-Based Two-level Hierarchical Structured P2P Network 140
Swathi Kaluvakuri, Indranil Roy, Koushik Maddali, Bidyut Gupta, and Narayan Debnath

Understanding the Anti-Mask Debate on Social Media Using Machine Learning Techniques 150
Luca Cerbin, Jason DeJesus, Julia Warnken, and Swapna S. Gokhale

*** "International Journal of Computers and Their Applications is Peer Reviewed".**

Volume 28, No. 3, September 2021 **ISSN 1076-5204**

International Journal of Computers and Their Applications

A publication of the International Society for Computers and Their Applications

EDITOR-IN-CHIEF

ASSOCIATE EDITORS

ISCA Headquarters···•···278 Mankato Ave, #220, Winona, MN 55987···•···Phone: (507) 458-4517
E-mail: isca@ipass.net • URL: http://www.isca-hq.org

Guest Editor's Editorial

This Special Issue of IJCA is a collection of four refereed papers, three of which are selected from the 33rd International Conference on Computer Applications in Industry and Engineering (CAINE 2020), October 19-20, 2020. The fourth paper in this issue is from a regular submission to IJCA.

Each paper submitted to the conference was reviewed by at least two members of the International Program Committee, as well as by additional reviewers, judging the originality, technical contribution, significance, and quality of presentation. After the conferences, eight papers were recommended by the Program Committee members to be considered for publication in this Special Issue of IJCA. The authors were invited to submit a revised version of their papers. After extensive revisions and a second round of review, three papers were accepted for publication in this issue of the journal.

Indranil Roy, Swathi Kaluvakuri, Koushik Maddali, Abdullah Aydeger, Bidyut Gupta, Southern Illinois University Carbondale, USA, and Narayan Debnath, Eastern International University, Vietnam, proposed in their paper "*Capacity Constrained Broadcast and Multicast Protocols for Clusters in a Pyramid Tree-based Structured P2P Network*" practical approaches for broadcasting and multicasting in clusters of peers in P2P networks using pyramid trees.

Pinzhi Wang and Youssou Gningue, Laurentian University, Canada, and Haibin Zhu, Nipissing University, Canada, in the paper "*Validation of the Improved Incremental Assignment Algorithm*" presented an innovative algorithm for the incremental assignment problem. The improved algorithm is evaluated for saving the operation time in finding the optimal solution (the maximum weighted matching) of the bipartite graph.

Swathi Kaluvakuri, Indranil Roy, Koushik Maddali, Bidyut Gupta of Southern Illinois University Carbondale, USA, and Narayan Debnath of Eastern International University, Vietnam, addressed efficiency problem in two-level non-DHT-based (distributed hash table) hierarchical structured peer-to-peer networks in the paper "*Efficient Secured Data Lookup and Multicast Protocols with Anonymity in RC-Based Two-level Hierarchical Structured P2P Network*". They proposed efficient ways to make already existing communication protocols secured.

Luca Cerbin, Jason DeJesus, Julia Warnken, and Swapna S. Gokhale, University of Connecticut, Storrs, USA, presented an analysis of social media data in the paper "*Understanding the Anti-Mask Debate on Social Media Using Machine Learning Techniques*." The analysis is on the problem of mask-wearing debate on social media (Twitter) during the Covid-19 pandemic using various classification methods. It did informal opinion mining, analyzed social parameters, and built a classification framework.

As guest editors we would like to express our genuine appreciation for the encouragement and support from Dr. Wenying Feng, the editor-in-chief of IJCA. I also thank the authors for their contributions, as well as the experts who reviewed all the papers submitted to this issue.

We hope you enjoy this special issue of IJCA. More information about ISCA society can be found at http://www.isca-hq.org.

Guest Editors
Gongzhu Hu, Central Michigan University, USA
Yan Shi, University of Wisconsin Platteville, USA
Takaaki Goto, Toyo University, Japan

September 2021

Capacity Constrained Broadcast and Multicast Protocols for Clusters in a Pyramid Tree-based Structured P2P Network

Indranil Roy*, Swathi Kaluvakuri*, Koushik Maddali*
Abdullah Aydeger*, Bidyut Gupta*
Southern Illinois University at Carbondale, Carbondale, Illinois, USA

Narayan Debnath†
Eastern International University, VIETNAM

Abstract

In this paper, we have considered an existing non-DHT based structured P2P network. It is an interest-based system. At the heart of the architecture, there exists a tree like structure, known as Pyramid Tree, even though it is not a conventional tree. A node on the tree represents a cluster of peers with common interest. There is no limit on the size of such clusters. Residue Class based on modular arithmetic has been used to realize the structure of a cluster. It has been shown that overlay diameter of each such cluster is just one (one overlay hop). Thus, each cluster is a completely connected network. Therefore, theoretically any peer in such a cluster is logically connected to every other peer in the cluster. However, since peers are heterogeneous in nature, therefore, in practice we have to consider their different capacities while designing any communication protocol inside the cluster. In this paper, we have addressed this issue and offered reasonably efficient solutions for broadcasting and multicasting considering peer heterogeneity.

Key Words: P2P network, structured, non-DHT based, pyramid tree, capacity constrained.

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 [2] 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 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) [10, 15, 18]. Such overlay architectures can offer efficient, flexible, and robust service [7, 10, 15, 16, 18]. 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 [4, 6, 9, 14, 17]; 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 [3, 11-13]. It offers advantages of DHT-based systems, while it attempts to reduce the complexity involved in churn handling. Authors in [11, 13] 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’ [5]. The P2P architecture has two levels. At the heart of it, it is a tree structure (pyramid tree); it is not a conventional tree. This tree forms the first level of the system. Each node on the tree represents uniquely a cluster-head of a cluster of peers with common interest and the cluster head is the first peer to join the system among the peers in this cluster. Such clusters form the second level of the architecture. Residue class based on modular arithmetic has been used to realize the architecture. Some of the main advantages of the system are its low data lookup efficiency and ease of churn handling. In this paper, we have considered such architecture and have dealt with a practical issue related to the architecture as detailed in below.

Problem Statement. In our earlier proposed pyramid tree P2P architecture [11, 13], every cluster has an overlay diameter of 1. Each such cluster may consist of a very large number of peers with common interest. It means that every peer in any such cluster C_i has direct logical connection to all other peers inside the cluster. In reality, peers are capacity constrained and it is most likely that any cluster will have heterogeneous peers; therefore, peers can be differently capacity constrained. Hence, even though the overlay diameter is 1, in practice a peer can

[*] E-mail: [indranil.roy, swathi.Kaluvakuri, Koushik]@siu.edu, [aydeger, Bidyut]@cs.siu.edu,
[†] E-mail: Narayan.debnath@eiu.edu.vn

communicate only to few other peers at a given time depending on its capacity. In this paper, we address this issue and offer reasonably efficient solutions for broadcasting and multicasting considering peer heterogeneity.

Our Contribution. We have earlier designed an inter-cluster broadcast protocol [13] in which a participating node (cluster-head) may have to activate a maximum of only three of its links at a time for the propagation of a broadcast packet along the pyramid tree. So, the protocol appears to have followed implicitly an effective capacity constrained approach, even though that was not the objective at the time of designing the protocol. Consider the following reasonably efficient capacity constrained architecture consisting of the peers in any cluster C_i. We logically restructure the peers inside cluster C_i in the following way: we partition the peers in C_i in a number of pyramid trees of identical sizes (except possibly the last one, explained later) and implement the idea of our already designed broadcast protocol on these trees inside the cluster. Note that in the original version of the inter-cluster broadcast protocol, a node in the tree is a cluster-head, whereas when applied inside a cluster a node in the tree can be any peer in the cluster that includes the cluster-head of the cluster as well. Let us first state the capacity constrained broadcast protocol. Later, we shall consider multicasting.

This paper is organized as follows. In Section 2, we talk about some related preliminaries and in Section 3, we present the capacity constrained broadcast along with the proposed restructuring method of peers inside a cluster. In Section 4, we present the capacity constrained multicast protocol. Section 5 draws the conclusion.

2 Related Preliminaries

In this section, we present some relevant results from our recent work on the pyramid tree based P2P architecture [11, 13] for interest-based peer-to-peer system.

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}, $0 \leq i \leq n-1$, 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.*

2.1 Pyramid Tree

The following overlay architecture has been proposed in [13].

1) The tree consists of n nodes. The ith node is the ith cluster head C_i^h.

2) Root of the tree is at level 1.

3) 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 [5].

4) 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.

5) The tree is a complete one if at each level j, there are j number of nodes (i.e., j number of cluster-heads).

6) 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.

7) Joining of a new cluster always takes place at the leaf level.

8) A node that does not reside either on the left branch or on the right branch of the root node is
an internal node.

9) Degree of an internal non-leaf node is 4.

10) 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 $S_n = \{0, 1, 2, \ldots (n-1)\}$. 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 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:

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

In the P2P architecture, we use the numbers belonging to different classes as the logical (overlay) addresses of the peers with a common interest and the number of residue classes is the number of distinct resource types; for the sake of simplicity we shall use only the positive integer values. Before we present the mechanism of logical address assignments, we state the following relevant property of residue class [13].

Lemma 1. *Any two numbers of any class r of S_n are mutually congruent.*

2.3 Assignments of Overlay Addresses

Assume that in an interest-based P2P system there are n distinct resource types. Note that n can be set to an extremely large value a priori to accommodate large number of distinct resource types. Consider the set of all peers in the system given as S = {C_i}, $0 \leq i \leq n-1$. Also, as mentioned earlier, for each subset C_i (i.e. cluster C_i) peer C_i^h is the first peer with resource type R_i to join the system and hence, it is the cluster-head of cluster C_i.

The assignment of overlay addresses to the peers in the clusters and the resources happens as follows:

1) The first cluster-head to join the system is assigned with the logical (overlay) address 0 and is denoted as C_0^h. It is also the root of the tree formed by newly arriving cluster-heads (see the example in Figure 1).

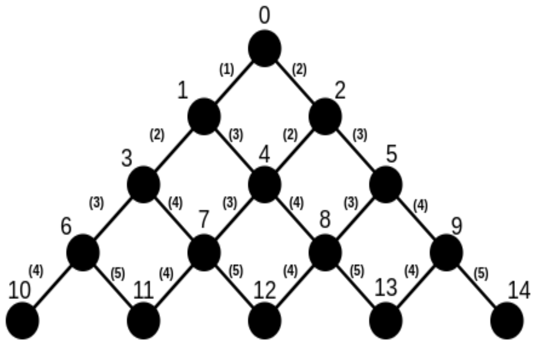


Figure 1: A complete pyramid tree with root 0

2) The (i+1) th newly arriving cluster-head possessing the resource type R_i is denoted as C_i^h and is assigned with the minimum nonnegative number (*i)* of *residue class i (mod n)* of the residue system S_n as its overlay address.

3) In this architecture, cluster-head C_i^h is assumed to join the system before the cluster-head C_{i+1}^h.

4) All peers having the same resource type R_i (i.e., 'common interest' defined by R_i) will form the cluster C_i. Each new peer joining cluster C_i is given the cluster membership address (i + j.n), for i = 0, 1, 2, …

5) Resource type R_i possessed by peers in C_i is assigned the code *i* which is also the logical address of the cluster-head C_i^h of cluster C_i.

Definition 3. *Two peers of a group G_r are logically linked together if their assigned logical addresses are mutually congruent.*

Lemma 2. *Each group C_i forms a complete graph.*

Observation 1. *Any intra-group data look up communication needs only one overlay hop.*

Observation 2. *Search latency for inter-group data lookup algorithm is bounded by the diameter of the tree.*

2.4. Virtual Neighbors [13]

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 (overlay) 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*.

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 + increment) mod 15 = (6 + 4) mod 15 = 10.

Also, note that a *left branch* originating at node 2 on the right branch of the root node is *2 → 4 → 7 → 11*. Similarly, we can identify all other left branches originating at the respective nodes on the right branch of the root node. In a similar way, we can identify as well all right branches originating at the respective nodes on the left branch of the root node as well.

Remark 1. *The sequence of increments on the segments along the left branch of the root appears to form an AP series with 1st term as 1 and common difference as 1.*

Remark 2. *The sequence of increments on the segments along the right branch of the root appears to form an AP series with 1st term as 2 and common difference as 1.*

Remark 3. *Along the 1st left branch originating at node 2, the sequence of increments appears to form an AP series with 1st term as 2 and common difference as 1. Note that the 1st term is the increment on the segment 0 → 2.*

Remark 4. *Along the 2nd left branch originating at node 5, the sequence of increments is an AP series with 1st term as 3 and common difference as 1. Note that the 1st term is the increment on the segment 2 → 5.*

Authors [13] have presented some important structural properties of the pyramid tree P2P system. According to the authors, no existing structured P2P system, either DHT or non-DHT based, possesses these properties. These are stated below.

Let S_Y be the set of logical links which connect a node Y to its neighbors in a complete pyramid tree T_R with root R. Assume that the tree has n nodes (i.e., n group heads / n clusters). Let another tree T'_R be created with the same n nodes but with a different root R'. Let S'_Y be the set of logical links connecting Y to its neighbors in the tree T'_R.

Property 1. $S_Y \neq S'_Y$

Property 2. *Diameter of T_R = Diameter of T'_R*

Property 3. *Number of levels of T_R = Number of levels of T'_R*

Property 4. *Complexity of broadcasting in T_R with root R as the source of broadcast is the same for T'_R with root R'*

Property 5. Both *T_R and T'_R are complete pyramid trees.*

An example: Consider the complete pyramid tree of 5 levels as shown in Figure 2. Note that root of this tree is node 13, whereas root of the tree of Figure 1 is 0.

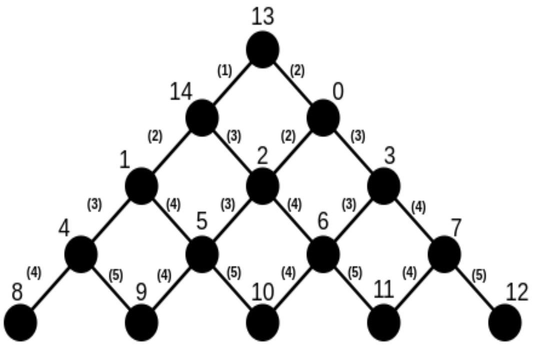


Figure 2: A complete pyramid tree with root 13

It is seen that S'$_4$ = {1,8,9} and S$_4$ = {1,2,7,8}. Therefore, property 1 holds.

Diameters of both trees are the same; it is 8 in terms of number of overlay hops. Besides, both trees use the same 15 nodes and have the same total number of levels. Broadcasting from either root 0 in the tree of Figure 1 or from root 13 in the tree of Figure 2 can be completed in 4 hops. Finally, both trees are complete pyramid trees. Thus, all properties as mentioned above hold.

Remark 5. *Set of the neighbors of a given node Z may vary as the root of the tree varies. Hence, it is termed 'virtual'. However, time complexity of broadcasting remains same, i.e., it is O(d),* where *d* denotes the number of levels of the tree

The following note on broadcasting will help in understanding better the proposed approach on capacity constrained broadcast.

2.5 Broadcast in Complete and Incomplete Pyramid Trees

We now state an informal sketch of the broadcast protocol [13] for complete pyramid tree architecture. It has been shown that the protocol does not generate any duplicate packet. The protocol uses properties 1 to 4 as stated above. It works as follows: whenever a node X on the tree wishes to broadcast, it will assume itself as the root of the overlay tree during broadcasting.

Step 1: *Root X sends packets to its neighbors on left and right branches.*

Step 2: *Each receiving node on the left branch sends packets to its neighbor on this branch till a receiving node is a leaf node.*

Step 3a: *The ith receiving node on the right branch sends packets to its neighbor on the ith left branch originating at the ith node until the ith receiving node is a leaf node.*

Step 3b: *The ith receiving node sends packets to its neighbor, the (i+1)th node on the right branch until it is a leaf node.*

Step 4: *Propagation along the ith left branch continues as in Step 2.*

For incomplete pyramid tree architecture, a broadcast source unicasts its packets to the root of the tree and the root then broadcasts the packets in the tree following the broadcast protocol designed for complete trees. Justification of the source itself not broadcasting its packets has been worked out in detail in one of our on-going projects [8]. It has been proven that only one duplicate packet will be generated per broadcast packet and it is independent of the total number of peers present in the P2P system [8].

3 Capacity Constrained Broadcast

Before we state the protocol formally, we first illustrate in detail the partitioning of the peers in a cluster.

3.1 Partitioning of Peers

We illustrate the partitioning process with an appropriate example. Let us consider a pyramid tree architecture consisting of cluster-heads of different distinct interests. Each cluster-head in turn is connected to peers of common interest belonging to its cluster. Let us call this tree the Global Pyramid Tree (GPT). Assume that a mod value of M has been used for the formation of the GPT. To illustrate the partitioning scheme, let us consider a cluster C_i with cluster-head C_i^h in the GPT. Let C_i consist of 40 peers. Let us assume that inside cluster C_i we use a mod value of 10 to build four complete pyramid trees T_1, T_2, T_3, and T_4 such that each one consists of 10 nodes. It is shown in Figure 3. Observe that the last tree T_4 not necessarily has to be a complete one, it depends on the number of peers inside a cluster; however, it has no effect on our explanation of the idea. We call each such tree inside cluster C_i a local pyramid tree (LPT). Note that since the mod value is 10 for building the LPTs, the peers in any such LPT will have secondary overlay addresses from 0 to 9, respectively (Figure 3). We shall use these secondary overlay addresses for restructuring the LPTs. Why are we considering pyramid trees inside a cluster? The reason is that we have already designed efficient communication protocols for pyramid tree architecture, and it is logical as well as reasonable to consider similar protocols inside a cluster; it offers uniformity in terms of implementation both at the levels of GPT and LPT.

Let the logical address of the cluster-head C_i^h be i based on the mod value of n used in the GPT formation. Then, the other 39 peers in C_i will have the respective overlay addresses as (i+n), (i+2n), …, (i+39n) based on their sequence of joining the cluster. These addresses are termed as primary overlay addresses. The roots of the four trees have the respective addresses as i, (i+10n), (i+20n), and (i+30n). Since C_i is completely connected, therefore, any two peers in the cluster are logically connected. However, to incorporate the idea of 'capacity constrained', we assume that among the roots of the four trees, only the links that connect the neighboring roots are present. That is, only links between T_1 and T_2, T_2 and T_3, and

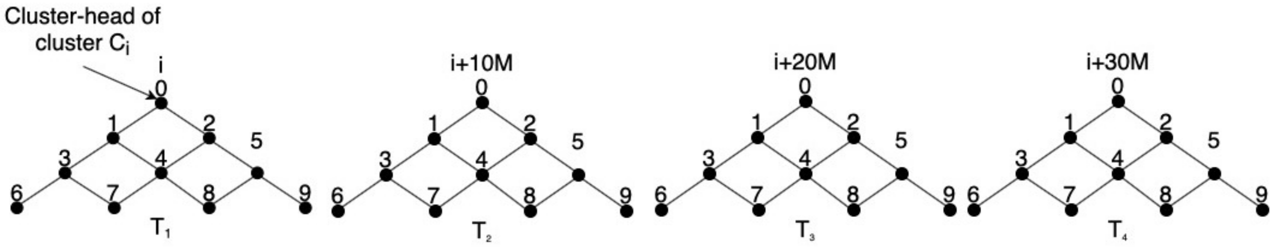


Figure 3: Cluster C_i has four component trees T_1, T_2, T_3, and T_4

In addition, we shall not use in our broadcast protocol any logical link that connects two peers in two different LPTs, none of which is a root in the corresponding tree.

Besides the above-mentioned primary overlay addresses based on the mod value of n, we also assign tertiary overlay addresses only to the roots of the LPTs inside a cluster. If a cluster C_i has n number of LPTs, the root of the first LPT formed will be assigned a tertiary overlay address 1, the root of the second LPT with address 2; in a similar way the root of the jth LPT will have the address j. These tertiary overlay addresses are used in the proposed broadcast and multicast protocols inside a cluster.

Cluster-head C_i^h assigns the addresses (i+n) to (i+9n) to the first nine peers joining the cluster besides the cluster-head itself. It forms the tree T_1 with itself as the root. Note that the virtual links' information among the peers in T_1 define implicitly its pyramid tree structure. Cluster-head C_i^h assigns the next 10 arriving peers with addresses (i+10n) to (i+19n) and imparts the responsibility of becoming the root of the tree T_2 and forming the complete tree T_2 to the peer with address (i+10n). In this way, the other trees are also formed. As pointed out earlier, the last tree T_4 may not be a complete one; it depends on the number of the arriving peers joining the tree. For broadcasting (also for multicasting) inside a tree, we assume that the root as well as each pear in the tree will maintain a local list of the overlay and IP addresses of all peers in the tree. Thus, in this example, each peer in T_1 will maintain the addressing information of 10 peers including that of itself. In general, for an LPT T_j, we denote the table as $\mathcal{T}_j$. In addition, the link information of a peer in the tree (i.e., which other peers it is connected to) is present in the table as well. Thus, effectively, structure of the pyramid tree is actually embedded (implicitly) in the table in the form of the links' information that a peer in the tree is linked with some particular other peers.

Besides, cluster-head C_i^h and the roots of all other LPTS will maintain a separate table $\mathcal{T}_r$ containing the tertiary overlay and the IP addresses of the root of each LPT in the cluster. Observe that table size depends on the mod value used to create the LPTs. At one end it will be a single tree with all peers in the cluster in it; this is not a practical approach since a cluster may have tremendously large number of peers resulting in high broadcast/multicast latency; in addition, table size maintained by each peer will be as large as the number of peers in the cluster. *So, a practical approach will be to use reasonably small mod value to create the trees and this mod value can be the choice of the designers; it can also be dynamically changed because after all these trees are virtual.*

3.2 Restructuring of the LPTs

For the proposed protocols (broadcast and multicast) to work correctly, we need to consider the effect on an existing LPT caused by peers joining and leaving (churn handling). Let us start with the peers joining first.

When we consider new peer joining, only the last LPT may get affected structurally from the following viewpoint. According to our proposed way of forming the LPTs in a cluster, say C_i, process of any new peer joining the cluster will be taken care of by only the root of the last LPT. For example, in Figure 3, the last LPT is the tree T_4 and its root is the peer with overlay address (i+30n). If the tree T_4 is already a complete one, then a new tree T_5 will be formed with its root having the address (i+40n) and all subsequent joins will take place in T_5 unless it is full; and the process of new tree-formation will go on as needed as explained above.

Observation 3. *Any new peer joining a cluster C_i will not affect structurally any LPT other than the last one.*

Observation 4. *If the existing last LPT is an incomplete one, new peers joining may turn it into a complete one, or it may remain an incomplete one.*

We now consider the effect on the structures of the LPTs due to peers leaving. Unlike joins, leaving of peers can take place at any time in any LPT and therefore, it can affect structurally any LPT. Therefore, a complete LPT may become incomplete after some peer(s) leave it. In addition, a complete tree may remain a complete one as well if multiple peers leave the tree; however, the new one will have a smaller number of levels. Besides, based on the positions of the leaving peers in an LPT, the pyramid tree architecture of the concerned LPT may be destroyed completely. See Figures 4 and 5. In the trees shown in these figures a peer with secondary overlay address X appears as X(k); k is the IP address of peer X. A detailed explanation of the trees in these figures appears later in this section. The need for such a representation will be clear shortly. The structure of an LPT may be destroyed after a peer leaves; it depends on the position of the leaving peer on the tree. Therefore, some efficient restructuring process need to be executed after peers leave the LPTs so that the characteristics of pyramid tree architecture can be retained, be it a complete or an incomplete one after peers leave and hence, the existing broadcast / multicast protocols can be applied in the restructured trees with some possible graceful degradation. Below we have stated the restructuring method after peers leave.

3.2.1 Restructuring Method. As mentioned earlier, each peer in the j^th^ LPT, viz. T_j maintains the addressing information of all peers in T_j including that of itself in a table T_j. This information includes a peer's overlay address and the IP address.

The following two situations are considered and we state the methods to handle these so that the characteristics of pyramid tree architecture can be retained.

Case 1: Any peer p with secondary overlay address X in an LPT T_j other than its root j leaves.

We assume graceful degradation; right before leaving the tree T_j, peer p unicasts a 'leave' message to the root j of T_j. Root j will delete the entry corresponding to the peer p with address X from its table T_j and assigns a virtual address X to the peer which had earlier the address (X+1) mod m where *mod m* is used to build the tree T_j. In this way, readdressing of the overlay addresses for all peers following the leaving one will take place. Root j forms a new table T_j' and unicasts this updated table to the rest of the peers of the tree T_j. Thus, the structure of a pyramid tree remains intact after the peer p leaves; of course, the tree T_j now can become an incomplete one. It may also remain as a complete one depending on the number of the peers leaving; however, in that case its level will be smaller than the original version of the tree T_j. Observe that structure of the pyramid tree is actually embedded in the table in the form of the links' information that a peer in the tree is linked with some particular other peers, along with the overlay addresses of the peers.

Case 2: Root j of T_j leaves

Step 1. Root j unicasts a 'leave' message to the peer p' with the next higher overlay address.

Step 2. Peer p' becomes the next root and assigns its new tertiary overlay address as j.

Step 3. Peer p' assigns new overlay addresses to the rest of the available peers, i.e., a previous address Y now becomes (Y-1).

Step 4. New root p' forms a new table T_j' and unicasts it to the other peers.

Step 5. A new tree T_j' is built. This new tree may be an incomplete or complete one depending on the total number of the leaving peers.

Observation 5. *Any combination of peers leaving an LPT can be handled as in either Case 1 or 2. If the combination involves the root, case 2 will be considered; otherwise it is Case 1.*

Example (Case 1)

Let us consider the j^th^ LPT of some cluster as shown in Figure 4, peers' secondary overlay addresses are from 0 to 9 and the tree is a complete one. To explain the situation of Case 1 clearly, we assume that the peers' respective IP addresses are a, b, …, j. Assume that peer 4 leaves. The structure will be no more that of a pyramid tree after the leave (Figure 5). However, based on the proposed restructuring method, peer 5 in Figure 4 now has the logical address 4, similarly peer 6 in Figure 4 now has the logical address 5 and in that way peer 9 has 8. The new pyramid tree after restructuring is shown in Figure 6. Note that in Figure 6 peer 4 with IP address f is different from peer 4 with IP address e in Figure 4. That is, peer 4 in Figure 6 is actually peer 5 in Figure 4. Note that peers with logical addresses 0 to 3 have not gone through any change. Root j constructs a new table T_j' and unicasts to the other peers in this tree.

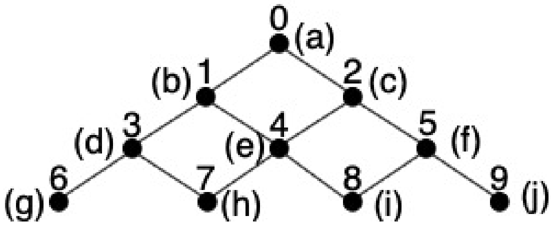
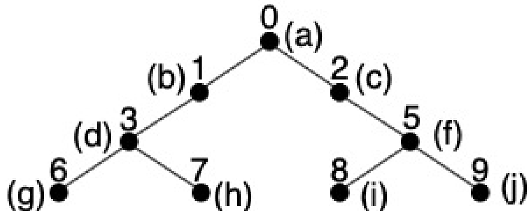
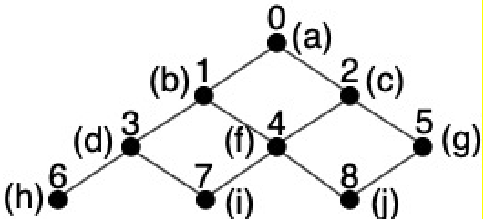


Figure 4: Before peer 4 leaves





Example (Case 2)

Consider the tree of Figure 4. Assume that the root peer with overlay address 0 and IP address a is leaving. Based on the restructuring method, right before leaving peer 0 unicasts a leave message to peer 1. Peer 1 (b) now becomes the new root and its overlay address becomes 0 and it will convert any other overlay address Y to (Y-1). It builds a new table T_j' and unicasts it to the rest of the peers. The new information about the links along with the changed overlay addresses produces the tree as shown in Figure 7. Thus, the structural properties remain intact except that the tree is now an incomplete one.

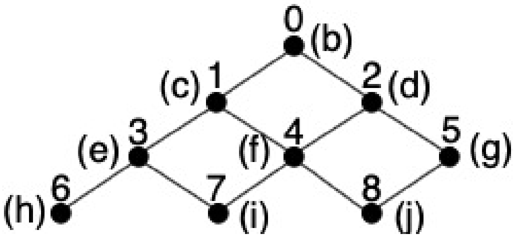


Figure 7: After restructuring, it remains a pyramid tree

3.3 Protocol Capacity-Constrained-Broadcast

Earlier we have discussed the effect of the leaving of peers on the existing structure of an LPT. Since after such leaving an LPT may become an incomplete pyramid tree even after restructuring, therefore, we shall consider only the protocol Broadcast-Incomplete in an LPT. We do it irrespective of the completeness of the tree because protocol Broadcast-Complete is applicable only to complete LPT, whereas Broadcast-Incomplete work both in complete and incomplete LPTs. It may be noted that in Broadcast-Incomplete in an LPT a source peer first unicasts its packets to the root of the tree (after restructuring wherever applicable); the root in turn broadcasts the packets to the rest of the tree. It generates only one extra packet per packet broadcast. This is the only bad effect of the protocol. Note that if the root itself is the source of broadcast, neither any extra packer is generated, nor there is any additional unicast.

In the proposed protocol (see Figure 8), we assume the following: a cluster C_j has k number of LPTs and the tertiary overlay address of the root of the j^{th} LPT is j ($1 \leq j \leq k$). We denote the j^{th} LPT as LPT_j and we call the root simply as root j (its tertiary address is r also). As mentioned earlier, cluster-head C_j^h maintains a separate table T_r containing the tertiary overlay and the IP addresses of the root of each LPT in the cluster; whereas every other root saves the tertiary overlay and the IP addresses of the root(s) of its neighboring LPT(s) only. In the following protocol, unicasting to an LPT means unicast to the root of the LPT.

Protocol

Based on the location of the broadcast source, we need to consider the following three possible situations:

Source peer is present in LPT_k; or is present in LPT_1; or is present in some LPT_r, $r \neq k$ and $r \neq 1$

```
if j = k          / Source peer is present in the last LPT ($LPT_k$)
    unicasts to $LPT_{j-1}$
    root j executes Broadcast-Incomplete protocol in its $LPT_k$
    j = j-1
    while 1< j
        root j unicasts to $LPT_{j-1}$
        root j executes Broadcast-Incomplete protocol in $LPT_j$
        j = j-1
    continue
    root 1 executes Broadcast-Incomplete in $LPT_1$
-----------------------------------------------------------------
if j = 1                / Source is present in the first $LPT_1$
    while j < k
        root j unicasts to $LPT_{j+1}$
        root j executes Broadcast-Incomplete protocol in $LPT_j$
        j = j +1
    continue
    root k executes Broadcast-Incomplete protocol in $LPT_k$
-----------------------------------------------------------------

if 1 < j < k
        / source is present in some other $LPT_r$, $r \neq k$ and $r \neq 1$

step 1.   root j unicasts to $LPT_{j-1}$ and $LPT_{j+1}$
          root j executes Broadcast-Incomplete protocol in
          $LPT_j$
step 2.   t = j-1
          while 1< t
               root t unicasts to $LPT_{t-1}$
               root t executes Broadcast-Incomplete protocol
               in $LPT_t$
               t = t-1
           continue
           root 1 executes Broadcast-Incomplete protocol in
           $LPT_1$
step 3.   t' = j+1
          while t' < k
              root t unicasts to $LPT_{t'+1}$
              root t' executes Broadcast-Incomplete protocol
              in $LPT_{t'}$
              t' = t'+1
           continue

          root k executes Broadcast-Incomplete protocol in
          $LPT_k$
```

Figure 8: The protocol

Theorem 1. Each peer in a cluster C_j receives a copy of each broadcast packet.

Proof. The protocol ensures that each root gets a copy of each broadcast packet. Each root then executes Broadcast-Incomplete' in its tree. Since 'Broadcast-Incomplete' guarantees that each peer in a tree receives a copy of each broadcast packet; hence is the proof. □

Performance

We shall discuss first broadcast latency followed by memory utilization by any root.

Let a cluster C_j be divided into n LPTs. Let us first consider the worst case in which the source of broadcast is either in the 1st LPT or in the last one, the kth LPT. Let it be in the 1st LPT. Let us assume that all LPTs have identical number of levels and it is *l*. It takes (k-1) hopes for a broadcast packet to reach the kth root (i.e., the root of the kth LPT). Observe that the concept of pipelining is implicitly present in the protocol; that is, by the time a broadcast packet arrives at the l^{th} level of the kth tree, broadcast is already complete in all other trees. Therefore, the number of hops to complete the broadcast in the worst case is [(k-1) + (*l*-1)]. In addition, it is clear that if the kth LPT has less than *l* number of levels, the total number of hops as mentioned above is sufficient to complete the broadcast.

If we consider the best case, i.e., when the broadcast source is present in a tree located at the middle. Total number of hops = [(k-1)/2 + (*l*-1)].

From the viewpoint of the memory used by a peer in any LPT_j to save its table T_j or by the cluster-head C_j^h (and any other root) to save its tables T_j and T_r, we have observed the following using clusters of different sizes. Broadcast latency does not vary much with m (m is the mod value used to create the LPTs inside the cluster); however, there is considerable increase in the memory requirement to save the tables as we increase the value of m. In fact, memory requirement varies linearly with cluster size. Therefore, it may be suggested that the designers select a reasonably small value of m for efficient use of the memory of the peers.

4 Capacity Constrained Multicast

We denote the LPT containing the source of multicasting as LPT_s. The source peer, say peer p, in LPT_s registers with the root s. In general, the corresponding root s can itself be the source of multicasting as well. Source peer p first registers with the root s and then during multicasting, it unicasts its packets to the root s and the root s in turn sends the multicast packets to the peers in its tree which have joined the multicast session; actually, these peers form a core-based tree (CBT) [1] with s as its root. In general, we denote a CBT with core j as CBT_j. In addition, root s is also responsible to send multicast packets to the roots of the other trees that are interested in receiving the packets. So effectively the root s acts as the source of multicast during multicasting. The proposed protocol will use the relevant information present in the tables of the roots as in the case of capacity constrained broadcast. A multicast session consists of three phases which are stated below.

Phase 1: Roots learning about the interested peers

Step 1. Source root s of the LPT_s broadcasts a 'query' message in all the component trees. This can be accomplished by executing the capacity constrained broadcast protocol.

Step 2. Source root s forms the CBT_s with s as its root if some peers in LPT_s join the core s for receiving multicast packets from the source peer p of multicasting.

/ CBT_s is formed with s as its root

Phase 2: Formation of core-based trees (CBTs) in other component trees and across the cluster

Step 1. Root j unicasts a join request to the source root s if it receives any join request(s) from peers in LPT_j

Step 2. Root j forms a CBT_j consisting of the interested peers in LPT_j

/ this is the CBT inside LPT_j

Step 3. A 2-level CBT with root s is formed with other joining roots as its leaves. The maximum diameter of this tree is 2.

/ this 2-level CBT with root s is formed across the cluster

Phase 3: Multicasting from source p

Step 1. Source peer p unicasts multicasts packets to source root s

Step 2. Root s sends the packets to the joining cores in the 2-level CBT.

Root s multicasts the packets to the peers in the CBT_s in LPT_s.

Step 3. Each joining core j multicasts the packets to the peers in the CBT_j in LPT_j.

Performance

We assume that for a given mod value of m, the cluster contains k number of LPTs. The trees have the same level *l*, except possibly the last one that may have fewer peers not enough to make the level *l*. However, for simplicity we assume that all LPTs have the same level. Therefore, k.m is the total number of peers in the cluster before partitioning. The protocol builds CBTs with maximum level *l* inside some LPTs containing multicast receivers and a 2-level CBT across the cluster with leaves as some roots of some LPTs interested in receiving multicast packets.

A multicast packet travels from a source peer p to the source root s in a maximum of (*l*-1) hops, then to all other roots in the 2-level CBT in 1 hop, and also to all group members in the receiving LPTs in a maximum of (*l*-1) hops.

Note that multicast in the source LPT goes on along with the multicasts in other LPTs; so, idea of pipelining is implicitly present. Hence, we can safely assume that approximately in (*l*-1) hops multicast in the related LPTs can be completed.

Therefore, multicast latency in hops is (*l*-1) +1+ (*l*-1), i.e., (2*l*-1); and it is independent of the size of the cluster. Hence, time complexity is $O(l)$. Since $l \approx 2^m$, therefore, it may be suggested that the designers select a reasonably small value of m for low latency multicast in the cluster.

5 Conclusions

In this paper, we have considered a recently reported non-DHT based structured P2P system. The main advantages of the architecture are its very low data lookup latency and ease of churn handling compared to most DHT-based P2P systems. In this architecture, a cluster consists of peers with common interest and its overlay diameter is only one hop. In reality, because of peer heterogeneity, peers are differently capacity constrained and therefore, lookup latency of $O(1)$ inside a cluster may not be achievable in reality. It has led us to propose practical approaches for both broadcasting and multicasting inside a cluster of peers considering peer heterogeneity.

We are now investigating how the proposed structure can be used/modified in order to reduce the traffic and operating costs of Internet Service Provider (ISP) and P2P Service Provider.

References

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

[2] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, "Making Gnutella-Like P2P Systems Scalable," *Proc. ACM SIGCOMM*, Karlsruhe, Germany, pp. 407-418, August 25-29 2003.

[3] 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*, 18(9):1188-1201, Sep. 2007.

[4] 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, 2:1250-1260, March 30 - April 1, 2003.

[5] Bidyut Gupta and Mohammad Mohsin, "Fault-Tolerance in Pyramid Tree Network Architecture," *J. Computer Systems Science and Engineering*, 10(3):164-172, July,1995.

[6] M. Kleis, E. K. Lua, and X. Zhou, "Hierarchical Peer-to-Peer Networks using Lightweight SuperPeer Topologies," *Proc. IEEE Symp. Computers and Communications*, pp. 143-148, 2005.

[7] 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.

[8] Koushik Maddali, Indranil Roy, Swathi Kaluvakuri, and Bidyut Gupta, "Efficient Broadcast Protocols for Complete and Incomplete Pyramid Tree P2P Architecture," under preparation.

[9] 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.

[10] 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.

[11] 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, *Proceedings of 32nd Int. Conf. Computer Applications in Industry and Engineering*, Las Vegas, NV, 63:121-129, 2019.

[12] Indranil Roy, Koushik Maddali, Swathi Kaluvakuri, Banafsheh Rekabdar, Ziping Liu, Bidyut Gupta, and 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)*, Helsinki, Finland, pp. 1351-1357, July 2019.

[13] Indranil Roy, Nick Rahimi, Koushik Maddali, Swathi Kaluvakuri, Bidyut Gupta, and Narayan Debnath, "Design of Efficient Broadcast Protocol for Pyramid Tree-**b**ased P2P Network Architecture," EPiC Series in Computing, *Proceedings of 33rd Int. Conf. Computer Applications in Industry and Engineering*, San Diego, CA, 63:182-188, 2020.

[14] K. Shuang, P. Zhang, and S. Su, "Comb: A Resilient and Efficient Two-Hop Lookup Service for Distributed Communication System," *Security and Communication Networks*, 8(10):1890-1903, 2015.

[15] 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*, 11(1):17-32, Feb. 2003.

[16] M. Xu, S. Zhou, and J. Guan, "A New and Effective Hierarchical Overlay Structure for Peer-to-Peer Networks," *Computer Communications*, 34:862-874, 2011.

[17] M. Yang and Y. Yang, "An Efficient Hybrid Peer-to-Peer System for Distributed Data Sharing," *IEEE Trans. Computers,* 59(9):1158-1171, Sep. 2010.

[18] 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*, 22(1):41-53, Jan. 2004.

Indranil Roy (photo not available) is currently a PhD student in Computer Science Department of Southern Illinois University, Carbondale. He has completed his B.E in Electronics & Communication from RCCIIT, Kolkata, in the year 2016. He received his M.S degree in Computer Science from Southern Illinois University, Carbondale in 2018. His main research interests include Blockchain along with interest-based p2p architecture.

Swathi Kaluvakuri (photo not available) is a Ph.D. candidate from Southern Illinois University Carbondale – School of Computing. She graduated from Jawaharlal Nehru

Technological University with a Bachelor of Technology degree in Computer Science major. She holds a keen interest in the areas of Peer to Peer Networking and Blockchain and worked as a Software Engineer, Technical Product Support and IBM AS400 developer for Net Cracker Pvt Ltd from 2012-2014.

Koushik Maddali (photo not available) is a Ph.D. candidate in Department of Computer Science at Southern Illinois University Carbondale. He received his MS from the same university and his BS from Jawaharlal Nehru Technological University, India. His research interests include Peer to Peer Networking, Blockchain and worked on a Virtual Terminal project of Cisco from 2017-2018.

Abdullah Aydeger (photo not available) is an Assistant Professor in the Department of Computer Science at Southern Illinois University, Carbondale. He received his M.S. and a Ph.D. degree from the Department of Computer Engineering at Florida International University in 2016 and 2020 and his B.S. degree in Computer Engineering from Istanbul Technical University in 2013. His research interests include Software Defined Networking, Network Function Virtualization, Moving Target Defense, and their utilization for different network security and resiliency problems. He applies the ideas not only to traditional ISP networks but also to emerging network domains within cyber-physical systems and IoT. He has published papers in reputable journals and conferences. He has also contributed two book chapters. Dr. Aydeger has served as a reviewer for many conferences and journals.

Bidyut Gupta (photo not available) received his M. Tech. degree in Electronics Engineering and Ph.D. degree in Computer Science from Calcutta University, Calcutta, India. At present, he is a professor at the School of Computing (formerly Computer Science Department), Southern Illinois University, Carbondale, Illinois, USA. His current research interest includes design of architecture and communication protocols for structured peer-to-peer overlay networks, security in overlay networks, and Blockchain. He is a senior member of IEEE and ISCA.

Narayan Debnath (photo not available) earned a Doctor of Science (D.Sc.) degree in Computer Science and also a Doctor of Philosophy (Ph.D.) degree in Physics. 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. Dr. Debnath has been the Director of the International Society for Computers and their Applications (ISCA) since 2014. Formerly, Dr. Debnath served as a Full Professor of Computer Science at Winona State University, Minnesota, USA for 28 years (1989-2017). Dr. Debnath has been an active member of the ACM, IEEE Computer Society, Arab Computer Society, and a senior member of the ISCA.

Validation of the Improved Incremental Assignment Algorithm

Pinzhi Wang*, Youssou Gningue*
Laurentian University, Ontario, CANADA

Haibin Zhu†
Nipissing University, Ontario, CANADA

Abstract

The Assignment Problem is a basic combinatorial optimization problem. In a weighted bipartite graph, the Assignment Problem is to find the largest sum of weights matching. The Hungarian method is a well-known algorithm, which is combinatorial optimization. Adding a new row and a new column to a weighted bipartite graph is called the Incremental Assignment Problem (IAP). The algorithm of the Incremental Assignment Problem utilizes the given optimal solution (the maximum weighted matching) and the dual variables to solve the matrix after extending the bipartite graph. This paper proposes an improvement of the Incremental Assignment Algorithm (IAA), named the Improved Incremental Assignment Algorithm (IIAA). The improved algorithm will save the operation time and operation space to find the optimal solution (the maximum weighted matching) of the bipartite graph.

Key Words: Assignment problem, weighted bipartite graph, Hungarian algorithm, incremental assignment problem.

1 Introduction

A matching or independent edge set in a graph is a set of edges without common vertices. There are several algorithms of the matching problem, such as matching in weighted bipartite graphs, matching in unweighted graphs, matching in general graphs. In unweighted graphs, weighted bipartite graphs and matching in unweighted graphs, maximum cardinality matching is sought [7]. In matching in weighted bipartite graphs, each edge has an associated value. A maximum weighted bipartite matching is defined as a matching where the sum of the values of the edges in the matching has a maximal value. If the graph is not complete bipartite, missing edges are inserted with value zero. Finding such a matching is known as the assignment problem. A common variant consists of finding a minimum-weighted perfect matching [2]. In this research, we are interested in the maximum weighted matching problem in bipartite graphs. The well-known algorithm for the assignment problem is the Kuhn-Munkres algorithm or the Hungarian algorithm, originally proposed by Kuhn in 1955 [3] and refined by Munkres in 1957 [4]. This algorithm has $O(n^3)$ complexity when it is implemented with proper data structures.

The incremental assignment problem is described as given a weighted bipartite graph and its maximum weighted matching, determine the maximum weighted matching of the graph extended with a new pair of vertices, one on each partition, and weighted edges connecting these new vertices to all the vertices on their opposite partitions [5]. It can be solved with the Hungarian Algorithm as the ordinary assignment problem. But in [6], Toroslu and Üçoluk propose an algorithm that utilizes the given maximum weighted matching of the maximum-weighted-matched part of the bipartite to determine the maximum weighted matching of the whole (extended) bipartite graph. The complexity of the algorithm is $O(n^2)$.

Consider the situation that there will be thousands or millions of weights. It is costly to calculate the extended feasible labels by using the incremental assignment problem algorithm. The goal of this paper is to present an algorithm to improve the incremental assignment problem algorithm, which reduces the complexity to $O(n)$ in four situations.

2 Background

A graph $G = (V, E)$ is bipartite if there exist two disjoint partitions X and Y $(V = X \cup Y, X \cap Y = \emptyset)$ and no edge connects vertices in the same partition $(E \subseteq X \times Y)$. A matching M is a subset of the edges E $(M \subseteq E)$, such that $\forall v \in V$ at most one edge in M is incident upon v. The size of matching is $|M|$, the number of edges in M. A maximum matching is a matching of maximum size (maximum number of edges). In a maximum matching, if any edge is added to it, it is no longer matching [1].

A weighted bipartite graph $G = (X \cup Y, X \times Y)$ is the graph in which rows correspond to the X partition and columns correspond to the Y partition of vertices. Each entry W_{ij} represents the weight of the edge between the vertices X_i and Y_j. The weight of matching M is the sum of the weights of edges in M.

Given a matching M, an alternating path is a path that begins with an unmatched vertex and whose edges belong alternately to the matching and not to the matching. An augmenting path is an alternating path that starts from and ends on free (unmatched) vertices. All alternating paths originating from a given unmatched node form a Hungarian tree.

*pwang@laurentian.ca, ygningue@laurentian.ca.
†haibinz@nipissingu.ca.

In Figure 2, let M be a matching of G.Vertex v is matched if it is an endpoint of edge in M; otherwise, v is free. $Y_2, Y_3, Y_4, Y_6, X_2, X_4, X_5, X_6$ are matched, other vertices are free. Y_5, X_6, Y_6 is an alternating path. $Y_1, X_2, Y_2, X_4, Y_4, X_5, Y_3, X_3$ is an augmenting path.

3 Improved Incremental Assignment Algorithm

The original matrix adds a new pair of vertices to the maximum-weighted-matched bipartite graph, in which feasible vertices and the maximum weighted matching are given. Any feasible label to the newly added pair of vertices is assigned. By using these labels, we can determine the maximum weighted matching of the whole extended bipartite graph. Before presenting the algorithm, we study the different cases where a solution can be derived in a very fast manner.

3.1 Analysis of the Different Cases

With the adding of vector row and vector column $n + 1$ and their respective costs, we can evaluate the marginal change associated with each row and column. The final values of the dual variables α_i and β_j for $i \in 1 \dots n$ and $j \in i \dots n$. This provides the evaluation of the differences for each row DR_j and column DC_i

$$DC_i = W_{i,n+1} - \alpha_i \quad and \quad DR_j = W_{j,n+1} - \beta_j$$

This allows us to evaluate maximal values of each column DCK and row DRT

$$\begin{aligned} DCK = max_{1 \le i \le n}(W_{i(n+1)} - \alpha_i) = DC_k \ \ and \ \ DRT \\ = max_{1 \le j \le n}(W_{(n+1)j} - \beta_j) = DR_t \end{aligned}$$

Their respective number (LC and LR) are also evaluated by

$$\begin{aligned} LC = Card(Argmax_{1 \le i \le n}(W_{i(n+1)} - \alpha_i)) \ \ and \ \ LR \\ = Card(Argmax_{1 \le j \le n}(W_{(n+1),j} - \beta_j)) \end{aligned}$$

CASE I. When the following relation is satisfied

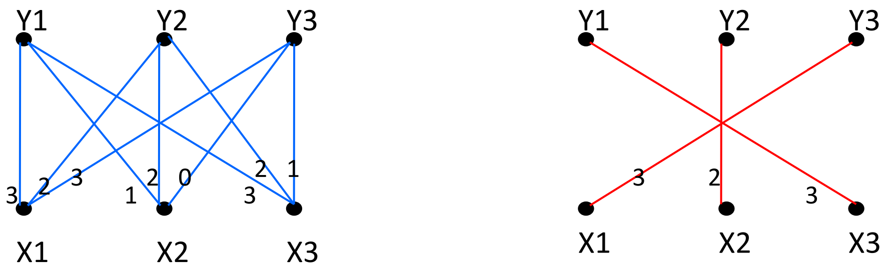
$$W_{n+1,n+1} \ \geq \ DCK + DRT$$

They are presented below.

CASE II. When there is only one-row k and column t reaching the maximal marginal changes with $X_{k,t} = 1$ i.e.

$$LC = LR = 1 \text{ and } X_{k,t} = 1$$

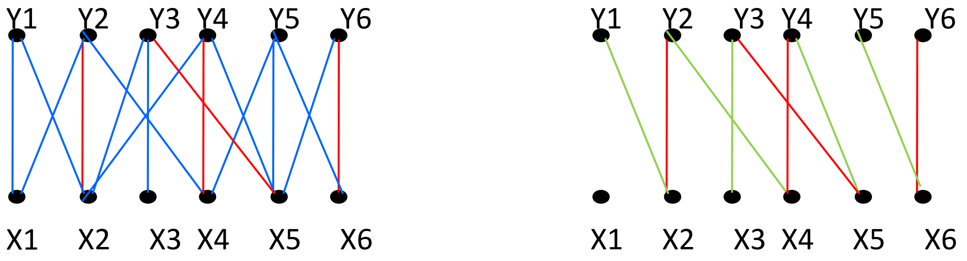
This means that the crossing variable of row k and column t is assigned in the optimal solution of the n assignment problem. Setting $X_{k,t} = 0$ the $n + 1$ assignment problem frees $X_{k,n+1}$ and $X_{n+1,t}$. Then by setting $X_{k,n+1} = 1$, and $X_{n+1,t} = 1$,



bipartite graph

bipartite weighted graph

Figure 1: Bipartite graph and bipartite weighted graph



bipartite graph

alternating/ augmenting path

Figure 2: Bipartite graph and alternating/ augmenting graph

the maximal possible increase of the objective function is reached. This provides an optimal solution to the $n+1$ assignment problem.

CASE III. When there is only one-row k and column t reaching the maximal marginal changes with $X_{k,t} = 0$ i.e.

$$LC = LR = 1 \text{ and } X_{k,t} = 0$$

Moreover, we have a column s and a row r such that

$$X_{k,s} = 1 \text{ and } X_{r,t} = 1 \text{ with } C_{rs} = \alpha_r + \beta_s$$

This means that the crossing variable of row k and column t is not assigned in the optimal solution of the n assignment problem. However, the complementary column s of the basic variable on row k and the complementary row of the basic variable on column t yield a variable X_{rs} which satisfies $C_{rs} = \alpha_r + \beta_s$. Since X_{rs} in the Equality Graph, setting $X_{k,s} = 0\ and\ X_{r,t} = 0$. The $n+1$ assignment problem frees $X_{k,n+1}$ and $X_{n+1,t}$. Then by setting $X_{k,n+1} = 1$, and $X_{n+1,t} = 1$, the maximal possible increase of the objective function is reached. This provides an optimal solution to the $n+1$ assignment problem

CASE IV. When there is more than one row or more than one column reaching the maximal marginal changes

$$\text{if } max(LC, LR) > 1$$

This case presents two subcases

$$max(LC, LR) = LR \geq LC \text{ or } max(LC, LR) = LC > LR$$

SUBCASE IV-a. The subcase is encountered when

$$max(LC, LR) = LR \geq LC \text{ and } LR > 1$$

For each $j = 1, \ldots, LR,$ we find the basis variable $X_{r\sigma(j)} = 1$ on the column $\sigma(j)$.

Then we test if $DC_k = (W_{r(n+1)} - \alpha_r) = DC_r$. This means that r and $\sigma(j)$, satisfies the second condition of Case II. We obtain an optimal solution by setting

$$X_{r\sigma(j)} = 0,\ \ X_{(n+1)\sigma(j)} = 1 \text{ and } X_{(n+1)\sigma(j)} = 1$$

SUBCASE IV-b. The subcase is encountered when

$$max(LC, LR) = LC > LR \text{ and } LC > 1$$

For each $i = 1, \ldots, LR,$ we find the basis variable $X_{\theta(i)s} = 1$ on the column $\theta(i)$.

Then we test if $DR_t = (W_{(n+1)s} - \beta_s) = DR_s$. This means that $\theta(i)$ and s satisfy the second condition of Case II. We obtain an optimal solution by setting

$$X_{\theta(i)s} = 0\ \ X_{\theta(i)(n+1)} = 1 \text{ and } X_{(n+1)s} = 1$$

3.2 Algorithm Presentation

This algorithm is adopted from the Hungarian algorithm, which uses the feasible labels of the vertices together with the maximum weighted matching.

Improved Incremental Assignment Algorithm:
Input:

- n assignment problem comprising a bipartite graph, $\{V, E\}$ (where $V = X \cup Y, X \cap Y = \emptyset,\ |X| = |Y| = n+1$) and $(n+1) \times (n+1)$ matrix of edge weights W_{ij}
- An optimal solution to the $n \times n$ sub-problem of the above assignment problem, comprising a matching M^* of the first n nodes of X to the first n nodes of Y, and the final values of the dual variables α_i and β_j for $i \in 1 \ldots n$ and $j \in i \ldots n$

Output: An optimal matching M, for the $(n+1) \times (n+1)$ problem.

1. Perform initialization:

    (a) Find the difference of each role and each column as follows:

$$DCK = -10^6$$

For $i = 1, \ldots, n$ do

S et $t = \sigma(1)$ $\qquad DC_i = \mathrm{W_{i,n+1}} - \mathrm{\alpha_i}$

If $DCK < DC_i$ then set $DCK = DC_i$ and *LC=1*; $\theta(1) = i$

```
        Else If $DCK = DC_i$ then $LC = LC + 1$; $\theta(LC) = i$

            Else ($DSK$ did not change)
            EndIf

        EndIf

    EndFor

    Set $k = \theta(1)$
    The previous procedure provides

                $DCK = \max_{1\le i\le n}(W_{i(n+1)} - \alpha_i) = DC_k$

                $LC = Card(Argmax_{1\le i\le n}(W_{i(n+1)} - \alpha_i))$

                        $DRT = -10^6$

    For $j = 1, \dots, n$ do

                        $DR_j = W_{j,n+1} - \beta_j$

        If $DRT < DR_j$ then set $DRT = DR_j$ and LR=1; $\sigma(1) = j$
        Else If $DRT = DR_j$ then $LR = LR + 1$; $\sigma(LR) = j$

            Else ($DRT$ did not change)
            EndIf

        EndIf

    EndIf

    Set $t = \sigma(1)$
    The previous procedure provides

                $DRT = max_{1\le j\le n}(W_{(n+1)j} - \beta_j) = DR_t$

                $LR = Card(Argmax_{1\le j\le n}(W_{(n+1),j} - \beta_j))$

                        $I = DCK + DRT$

    (b) If $I \le W_{(n+1)(n+1)}$, then

            $X_{(n+1)(n+1)} = 1$, then go to step 3

        Else If $LC = LR = 1$ then

            If $X_{kt} = 1$,
            Set $X_{k(n+1)} = 1$ and $X_{(n+1)t} = 1$,

                Then go to step 3.

            Else ($X_{kt} \neq 1$)

                Find the complementary column $s$ of the basic variable on row $k$
                Find the complementary row $r$ of the basic variable on column $t$
                If $X_{rs}$ satisfies $C_{rs} = \alpha_r + \beta_s$ (Equality Graph) then
```

```
                    Set $X_{rs} = 1$ then $X_{k(n+1)} = 1$ and $X_{(n+1)t} = 1$
                    Then go to step 3

                Else go to step (c).

        Else if $max(LC, LR) > 1$

            If $max(LC, LR) = LC$   then

                For $j = 1, \dots, LR$ do

                    Find the basis variable $X_{r\sigma(j)} = 1$
                    If $DC_k = (W_{r(n+1)} - \alpha_r)$ then set

                      $X_{r\sigma(j)} = 0$, $X_{(n+1)\sigma(j)} = 1$ and $X_{r(n+1)} = 1$

                      Go to step 3

                End if

            EndFor

        EndIf

    Else if $max(LC, LR) > 1$

        If $max(LC, LR) = LR$   then

            For $i = 1, \dots, LC$ do
            Find the basis variable $X_{\theta(i)s} = 1$
            If $DR_t = (W_{(n+1)s} - \beta_s)$ then set

                  $X_{\theta(i)s} = 0$, $X_{\theta(i)(n+1)} = 1$ and $X_{(n+1)s} = 1$

                Go to step 3

            End if

        EndFor

    EndIf

EndIf
```

(c) Incremental assignment algorithm: Assign feasible values to the dual variables α_{n+1} and β_{n+1} as follows:

$$\beta_{n+1} = max\left(DC_k, W_{(n+1)(n+1)}\right)$$

$$\alpha_{n+1} = DR_t$$

2. Perform the Stage from the basic Hungarian algorithm detailed in the Hungarian Algorithm.

3. Output the resulting matching M.

4 Implementation and Performance Experiments

When properly implemented, the Incremental Assignment Algorithm (IAA) can operate with the computational complexity of $O(n^2)$ [5]. To verify the performance of the Improved Incremental Assignment Algorithm (IIAA), a program is implemented based on that of the IAA.

Five cases are designed for the Improved Incremental Assignment Algorithm (IIAA) in 100 dimensions (a matrix with 100×100 elements) and 200 dimensions (a matrix with 200×200 elements). Case I shows the operation time of the circumstances that $X_{kt} = 1$ and $I \leq W_{(n+1)(n+1)}$. Case II shows the running time of the matrix that $X_{kt} = 1$ and $I > W_{(n+1)(n+1)}$. Case III displays the operation time when $X_{kt} \neq 1$. Find the complementary column s of the basic variable on row k and the complementary row of the basic variable on column t, if X_{rs} satisfies $C_{rs} = \alpha_r + \beta_s$. Case IV reveals the running time when we have more than one largest difference of the row and more than one largest difference of the column. Case V is a general case, which expresses that the random matrix can be solved by the IIAA.

All the experiments are workable in both algorithms. Matrix is formed by randomly creating weights. The test takes 100 random new pairs of vertices in each dimension matrix.

$$\text{Chances} = \frac{\text{Numbers of each case}}{\text{Total number of the cases}} \times 100$$

Figure 3 provides the trend line of chances that a random matrix is solved by the Improved Incremental Assignment Algorithm (IIAA).

Figure 4 presents the average time of different cases, which have been solved by the improved incremental assignment algorithm and incremental assignment algorithm. The time unit is millisecond.

Table 1 presents the average operation time of different pairs of vertices, which have been solved by improved incremental assignment algorithm and incremental assignment algorithm

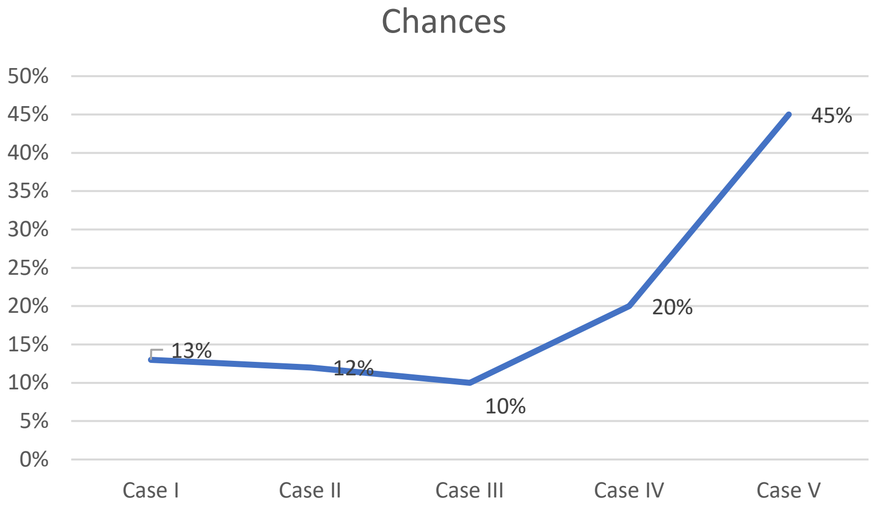


Figure 3: Trend lines for the chances of five cases happen

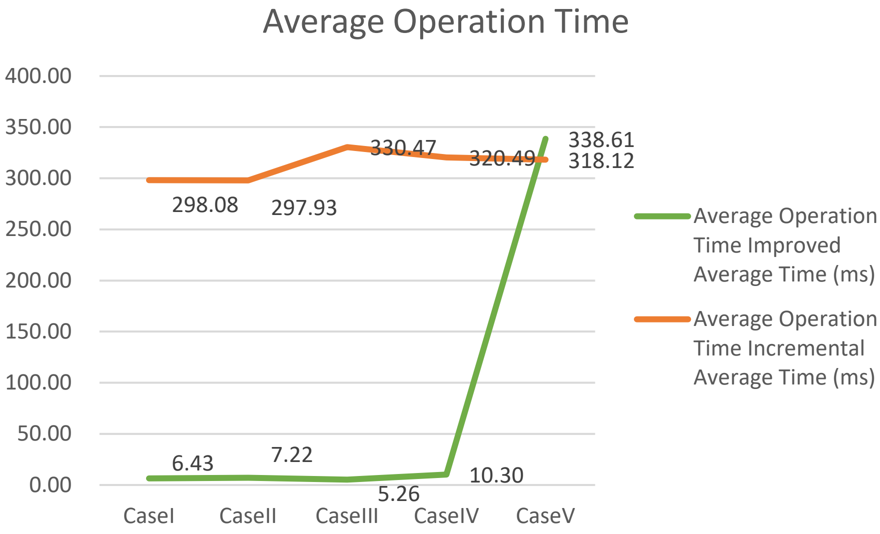


Figure 4: Trend lines for average process time for different cases

for each case in each dimension. The time unit is millisecond.

Table 2 shows the improvement percentage of improved IAA.

$$\text{Percentage} = \frac{\text{Improved IAA average time} - \text{IAA average time}}{\text{IAA average time}}$$

5 Performance Analysis

Table 1 shows the typical data collected from the experiments stated previously. The average operation time of the Incremental Assignment Algorithm (IAA) required by the random matrix is linearly increased. It is possible for a particular smaller dimension matrix to take more time than a larger dimension matrix because the value distributions significantly affect the time needed in the relevant algorithm. The average running time of IIAA is stable regardless of the increase of the dimension. The average running time difference between IIAA and IAA becomes higher because of the different numbers of iterations.

Table 2 indicates that with a higher dimension, the number of iterations of IAA will go higher. But IIAA still has only one iteration no matter how high the dimension is.

In Case I, Case II, and Case III, IIAA saves up to 96% of

Table 1: The process time for each case

Dimension Case & Time	200 × 200 dimension	100 × 100 dimension
Case I		
Improved Average Time	7.74	5.13
Incremental Average Time	462.92	133.24
Case II		
Improved Average Time	9.75	4.71
Incremental Average Time	476.35	119.52
Case III		
Improved Average Time	6.67	4.47
Incremental Average Time	528.81	132.14
Case IV		
Improved Average Time	8.08	12.53
Incremental Average Time	519.46	121.51
Case V		
Improved Average Time	545.78	131.44
Incremental Average Time	515.84	120.41

Table 2: The percentage that the improved IAA improves

Dimension Percentage	200 × 200 dimension	100 × 100 dimension
Case I	-98.32%	-96.15%
Case II	-97.95%	-96.03%
Case III	-98.74%	-96.62%
Case IV	-98.45%	-89.69%
Case V	5.80%	9.16%

operation time in 100 dimensions matrix and nearly 98% in 200 dimensions matrix. In Case IV, IIAA saves about 90% of operation time. The data express that the Improved Incremental Assignment Algorithm will save much more time than the Incremental Assignment Algorithm.

For all the 200 random matrices, 55% matrices meet the conditions of IIAA, the rest of the matrices will go to IAA that is the Case V (general case).

If the matrix is produced randomly, the general expectation of improvement is 46.5% compared with the operation time using the incremental assignment algorithm.

Overall, when the matrix is solved by IIAA, the processing time will be much faster than IAA.

6 Conclusion

In this paper, an improved algorithm has been proposed to solve the incremental assignment problem. From the view of the overall program, the algorithm improves a lot with the running time of operation. From finding the largest difference of the row and the column, with the use of exchange with its maximum weighted matching, the problem has been reduced. Consequently, the step of iteration can be reduced largely.

From the perspective of functions, our solution directly provides an improved way to solving the Incremental Assignment Problem. Not only the operation time will be saved, but also the occupied space will be reduced.

The computational complexity of our algorithm is $O(n^2)$, because the most complicated case for our algorithm will go to the incremental assignment algorithm. As the complexity of the incremental assignment algorithm is $O(n^2)$, our algorithms are also at the same level.

Our solution has many advantages, but it is still necessary to point out the disadvantages. About 55% of the 200 random matrices meet the conditions of the improved incremental assignment algorithm, the rest of the matrix will go to the incremental assignment algorithm. The chance is limited and needs to be improved in the future.

References

[1] R. E. Burkard, M. Dell'Amico, and S. Martello, “Assignment Problems,” Society for Industrial and Applied Mathematics, pp. 35-144, 2009.

[2] https://en.wikipedia.org/wiki/Assignment_problem, 2019.

[3] H. W. Kuhn, “The Hungarian Method for the Assignment Problem,” Naval Research Logistics Quarterly 2(1–2):83-97, March 1955.

[4] J. Munkres, “Algorithms for the Assignment and Transportation Problems,” *Journal of the Society for Industrial and Applied Mathematics* 5(1):32-38, March 1957.

[5] I. H. Toroslu and G. Üçoluk, “Incremental Assignment Problem,” *Information Sciences* 177(6):1523–29, March 15, 2007.

[6] I. H. Toroslu, and G. Üçoluk, Authors’ Response to “An Addendum on the Incremental Assignment Problem,” by Volgenant, Vol. 178, 2008.

[7] Y. Xie, “An o(n2.5) Algorithm: For Maximum Matchings in General Graphs,” *Journal of Applied Mathematics and Physics* 6(9):1773-82, 2018.

Pinzhi Wang (photo not available) received the M.S. degree in computational sciences from Laurentian University, Sudbury, ON, Canada, in 2020. She is currently a Mathematics Teacher in Private School.

Youssou Gningue (photo not available) graduated with a Doctorate, Ph. D., in Applied Mathematics from the University of Sherbrooke (Quebec, Canada) in 1992. He has been a professor in the Mathematics and Computer Science Department of Laurentian University since September 1991 where he is cur-rently an emeritus full Professor since May 2021. Originally, from the country of Sénégal, Professor Youssou Gningue teaches at the third cycle of Applied Mathematics at the University Cheikh Anta Diop (UCAD), Dakar, Sénégal where he supervises doctoral students. Professor Youssou Gningue is also a Pan-Africanist who campaigns for the Unity of Africa

H**aibin Zhu** is a Full Professor and the Coordinator of the Computer Science Program, the Founding Director of Collaborative Systems Laboratory, a member of the University Budget Plan committee, Arts and Science Executive Committee, and the Research Committee, Nipissing University, Canada. He is also affiliate professor of Concordia Univ. and adjunct professor of Laurentian Univ., Canada. He received B.S. degree in computer engineering from the Institute of Engineering and Technology, China (1983), and M.S. (1988) and Ph.D. (1997) degrees in computer science from the National Univ. of Defense Technology (NUDT), China. He was the chair of the department of Computer Science and Mathematics, Nipissing University, Canada (2019-2021), a visiting professor and a special lecturer in the College of Computing Sciences, New Jersey Institute of Technology, USA (1999-2002) and a lecturer, an associate professor and a full professor at NUDT (1988-2000). He has accomplished (published or in press) over 200 research works including 29 IEEE Transactions articles, six books, five book chapters, three journal issues, and three conference proceedings. He is Senior member of ACM, a full member of Sigma Xi and a senior member of IEEE.

Efficient Secured Data Lookup and Multicast Protocols with Anonymity in RC-Based Two-level Hierarchical Structured P2P Network

Swathi Kaluvakuri*, Indranil Roy*, Koushik Maddali*, Bidyut Gupta*
Southern Illinois University Carbondale,Carbondale, IL USA

Narayan Debnath†
Eastern International University, VIETNAM

Abstract

In this paper, we have considered a recently reported 2-level non-DHT-based structured P2P network. It is an interest-based architecture. Residue Class (RC) based on modular arithmetic has been used to realize the overlay topology. Such an architecture has been the choice because it offers low latency in both inter or intra group communications. In the present work, we have proposed efficient ways to make these already existing communication protocols secured. In addition, we have extended these protocols further to include anonymity as well.

Key Words: Overlay multicast, residue class, interest - based, theorem, structured P2P networks, secured protocols, anonymity.

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. P2P networks are classified into two classes: unstructured and structured ones. In unstructured systems [2] peers are organized into arbitrary topology. It takes the 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. In unstructured networks, lookups are not guaranteed. On the other hand, structured overlay networks provide deterministic bounds on data discovery. They provide scalable network overlays based on a distributed data structure which actually supports the deterministic behavior for data lookup. Recent trend in designing structured overlay architectures is the use of distributed hash tables (DHTs) [6, 9, 18]. Such overlay architectures can offer efficient, flexible, and robust service [6, 9, 11, 18-19].

However, maintaining DHTs is a complex task and needs substantial amount of effort to handle the problem of churn. So, the major challenge facing such architectures is how to reduce this amount of effort while still providing an efficient data query service. In this direction, there exist several important works, which have considered designing hybrid systems [5, 14, 16, 20]. These works attempt to include the advantages of both structured and unstructured architectures. However, these works have their own pros and cons [1].

2 Preliminaries

Some of the preliminaries of this RC-based low diameter two level hierarchical structured P2P network [7-8, 12], have been considered here. In this section, we present a structured architecture for an interest-based peer-to-peer system. The following notations along with their interpretations will be used while we define the architecture.

Definition 1. We define a resource as a tuple <*Res$_i$*, *V*>, where *Res$_i$* denotes the type of a resource and *V*is the value of the resource. Note that a resource can have many values.

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

We now describe our proposed architecture suitable for interest-based peer-to-peer system. Generalization of the architecture is considered in [8].

We use the following notations along with their interpretations while we define the architecture.

2.1 Two Level Hierarchy

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

1) At level-1, we have a ring network consisting of the peers H_i ($0 \leq i \leq n-1$). The number of peers on the ring is n which is also the number of distinct resource types. This ring network is used for efficient data lookup and so we name it as transit ring

[*] School of computing. E-mail: [swathi.kaluvakuri, indranil.roy, Skoushik, bidyut]@siu.edu.

[†] School of Computing and Information Technology. E-mail: ndebnath@gmail.com

network.

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

3) Each peer on the transit ring network maintains a global resource table (GRT) that consists of n number of tuples. GRT contains one tuple per group and each tuple is of the form <Resource Type, Resource Code, Group Head Logical Address>, where Group Head Logical Address refers to the architecture. Also, Resource Code is the same as the group-head logical address.

4) Any communication between a peer $G_{x,i} \in$ group G_x and $G_{y,j} \in$ group G_y takes place only through the corresponding group heads H_x and H_y.

The proposed architecture is shown in Figure 1.

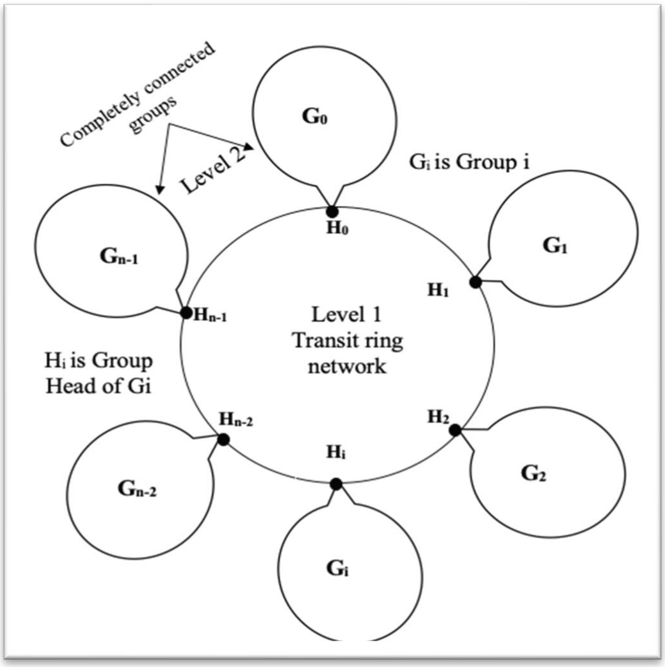


Figure 1: A two-level RC based structured P2P architecture with n distinct resource types

2.2 Relevant Properties of Modular Arithmetic

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

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

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

Thus, any class r (mod n) of S_n can be written as follows:

$[r] = \{\dots, (r - 2n), (r - n), r, (r + n), (r + 2n), \dots, (r + (j-1). n), (r + j.n), (r + (j+1).n), \dots\}$

A few relevant properties of residue class are stated below.

Lemma 1. Any two numbers of any class r of S_n are mutually congruent.

2.3 Assignments of Overlay Addresses

Assume that in an interest-based P2P system there are n distinct resource types. Note that n can be set to an extremely large value *a priori* to accommodate a large number of distinct resource types. Consider the set of all peers in the system given as $S = \{P^{R_i}\}$, $0 \leq i \leq n-1$. Also, as mentioned earlier, for each subset P^{R_i} (i.e. group G_i) peer H_i is the first peer with resource type R_i to join the system.

The assignment of logical addresses to the peers at the two levels and the resources happen as explained in [7-8, 12].

Remark 1. GRT remains sorted with respect to the logical addresses of the group-heads.

Definition 3. Two peers H_i and H_j on the ring network are logically linked together if (i + 1) mod n = j.

Remark 2. The last group-head H_{n-1} and the first group-head P_0 are neighbors based on Definition 3. It justifies that the transit network is a ring.

Definition 4. Two peers of a group G_r 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 G_r forms a complete graph.

2.4 Sailent Features of Overlay Architecture

We summarize the salient features of this architecture.

1) It is a hierarchical overlay network architecture consisting of two levels; at each level the network is a structured one.

2) Use of modular arithmetic allows a group-head address to be identical to the resource type owned by the group. We will show in the following section the benefit of this idea from the viewpoint of achieving reasonably very low search latency.

3) Number of peers on the ring is equal to the number of distinct resource types, unlike in existing distributed hash table-based works some of which use a ring network at the heart of their proposed architecture [11].

4) The transit ring network has the diameter of n/2. Note that in general in any P2P network, the total number of peers N >> n.

5) Each overlay network at level 2 is completely connected. That is, in graph theoretic term it is a complete graph consisting of the peers in the group. So, its diameter is just 1. Because of this smallest possible diameter (in terms of number of overlay hops) the architecture offers minimum search latency inside a group.

2.5 Our Contribution

In this paper, we have considered interest based P2P systems [20-21]. We have considered designing secured protocols for Inter and Intra lookup algorithms. The concepts of symmetric key, asymmetric key cryptography with public and private keys have been used. We have also considered making the capacity-constrained multicast algorithms more efficient with security both inside a group and for the two-level architecture. In addition, we have also considered anonymity. In section III we present the secured data lookup algorithms and in section IV, we have present multicast algorithms with both security and anonymity.

3 Data Lookup Algorithms with Security

Cryptography is the research and implementation of encrypted communication techniques. It is concerned with the creation and analysis of protocols that prevent malicious third parties from accessing information exchanged between two organizations, thereby adhering to various aspects of information security.

A situation in which a message or data exchanged between two parties cannot be accessed by an adversary is referred to as secure communication. In cryptography, an adversary is a malicious party that attempts to retrieve useful information or data by breaching information security principles.

Cryptographic algorithms are used to achieve stability in peer-to-peer networks in terms of authentication and confidentiality. Secret key cryptographic algorithms and public key cryptographic algorithms are the two types of cryptographic algorithms that are most used. Since the same key is used for encryption and decryption and is shared by all parties concerned, secret key cryptographic algorithms are also known as symmetric key algorithms. Asymmetric key algorithms, on the other hand, are also known as public key cryptographic algorithms. A pair of keys, one for encryption and the other for decryption, are used in this form. One of the keys, known as the public key, is made public, while the other, known as the private key, is kept private. Only the pair's secret key will decrypt a message encrypted with a public key. Similarly, a message encrypted with a private key can only be decrypted by the pair's public key [13]. Data lookup algorithms [7, 12] both Inter and Intra are presented in this section with the concept of security. Cryptographic functions and their applications in 2 level RC based architecture is explained in Figure 2.

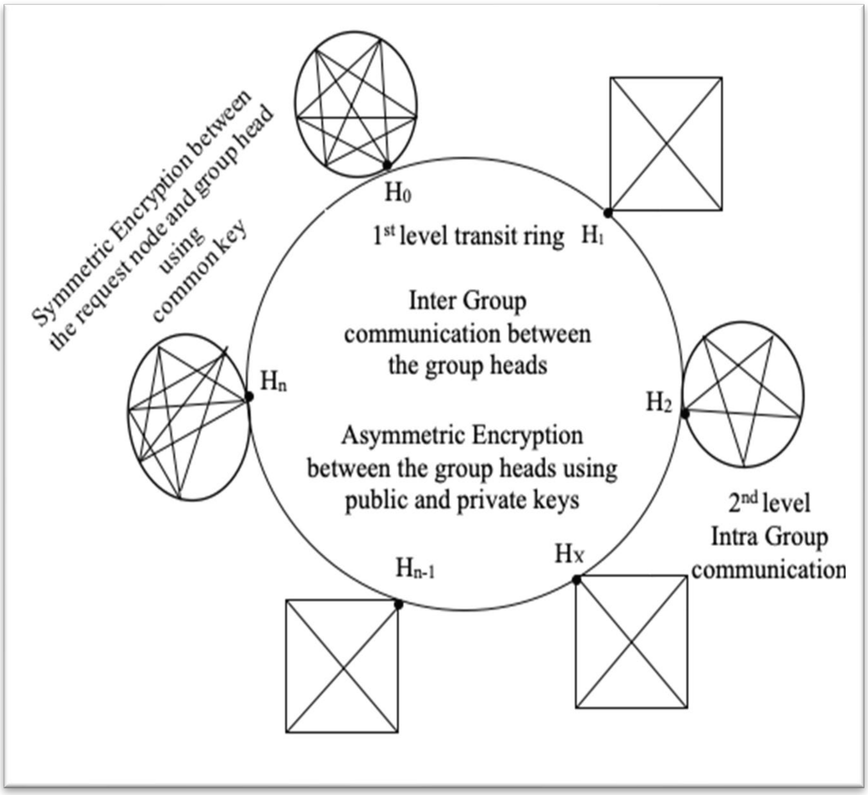

Symmetric Encryption between the request node and group head using common key
H0
1st level transit ring
H1
Inter Group communication between the group heads
Hn
H2
Asymmetric Encryption between the group heads using public and private keys
2nd level Intra Group communication
Hn-1
Hx

Figure 2: Cryptographic functions and their applications in 2 level architecture

3.1 Intra Group Lookup Algorithm with Security in RC based Architecture

In this case the resource lookup happens within the group, i.e., the resource type is the same for both the parties but the value is different. So, we use the concept of symmetric key cryptography where the same cryptographic key is used for encoding the message (request) by requesting peer and decoding the message (request) by group head. The algorithm is explained as follows.

Let us assume that in a group G_x , a peer $G_{x,i}$ with resource $<Res_x, V_i>$ is looking for a resource $<Res_x, V_j>$. *Let* $SyKey_{x,i}$ is the common/symmetric key shared by the requesting node $G_{x,i}$ and the corresponding group head H_x of the same group G_x as explained in Figure 3.

```
Secured Intra-Lookup Algorithm

  1. Request node $G_{x,i}$ will encrypt the message $<Res_x, V_j>$
     with symmetric key $SyKey_{x,i}$ and sends it group head $H_x$
     through a unicast message.

     // To make it clear, this symmetric key information is
        known only to the requestor and the group head so
        other nodes in the same interest group will not be
        able to decrypt the request.

  2. The group head $H_x$ will then decrypt the encrypted
     request with the symmetric key $SyKey_{x,i}$
  3. Later, this request $<Res_x, V_j>$ will be broadcasted in the
     interest group $G_x$ by the group head $H_x$
  4. If a node $G_{x,j}$ in group $G_x$ has the requested resource
     $<Res_x, V_j>$

       a. it encrypts the resource $<Res_x, V_j>$ with symmetric
          key $SyKy_{x,j}$ and unicasts it to the group head $H_x$
       b. The group head $H_x$ will use the symmetric key
          $SyKy_{x,j}$ and decrypts the response from $G_{x,j}$
       c. $H_x$ will now encrypt the response $<Res_x, V_j>$ with
          the symmetric key $SyKey_{x,i}$ and unicasts it to the
          requesting node $G_{x,i}$
       d. Finally, $G_{x,i}$ will decrypt the response using the
          symmetric key $SyKey_{x,i}$

   else search for $<Res_x, V_j>$ fails
```

Figure 3: Intra Group Lookup Algorithm with Security in RC based architecture

3.2 Inter Group Lookup Algorithm with Security in RC based Architecture

When it comes to Inter group, the communication happens between the nodes from two different interest-based groups, so here comes the concept of public and private keys. Hence the asymmetric key security. In our secured RC based architecture, any sort of communication between the peers $G_{x,i} \in$ group G_x and $G_{y,j} \in$ group G_y takes place only through the corresponding group heads H_x and H_y.

The following notations are used to denote the public and private keys of the requesting and responding group heads.

- Pbl_x and Pvt_x to denote respectively the public and private keys of group-head H_x of group G_x.
- Pbl_y and Pvt_y to denote respectively the public and private keys of group-head H_y of group G_y.

Without any loss of generality, let a peer $G_{x,i} \in$ group G_x requests for a resource $<Res_y, V_j>$. Peer $G_{x,i}$ and group head H_x is aware of the fact that that $Res_y \notin$ group G_x. The secured inter-lookup algorithm is explained in Figure 4.

```
Secured Inter-Lookup Algorithm

  1. Request node $G_{x,i}$ encrypts the request $<Res_y, V_j>$ using
     the common key $SyKey_{x,i}$ and unicasts it to the group
     head $H_x$
  2. The group head $H_x$ will then decrypt the encrypted
     request with the symmetric key $SyKey_{x,i}$ . Because the
     group head $H_x$ is aware of the fact that that $Res_y \notin$ group
     $G_x$, it finds the Group head address of $H_y$ along with its
     public key $Pbl_y$ from the GRT table.

     // address code of $H_y$ = resource code of $Res_y = y$ //

  3. $H_x$ encrypts the message with $Pbl_y$ and computes $|x - y| = h$
  4. if $h > n / 2$ (where n is the number of distinct resource
     types)

        $H_x$ forwards the request along with the IP address of the
     request node $G_{x,i}$ to its immediate predecessor $H_{x-1}$

       else $H_x$ forwards the request along with the IP address of
     the request node $G_{x,i}$ to its immediate successor $H_{x+1}$

        // Looking for minimum no. of hops

     end

  5. Each intermediate group-head $H_k$ forwards the
     encrypted request until $H_k = H_y$

  // In the worst case it will take around n/2 hops

  6. Now $H_y$ will decrypt the message using its private key
     $Pvt_y$
  7. if $H_y$ itself has the resource $<Res_y, V_j>$

        $H_y$ encrypts the message with the public key $Pbl_x$ of
        $H_x$ and unicasts it to $H_x$

     else

     $H_y$ broadcasts the request for $<Res_y, V_j>$ in group $G_y$
```

```
    if ∃ $G_{y,j}$ $(\epsilon G_y)$ which has the resource $<Res_y, V_j>$

        • $G_{y,j}$ encrypts the message with symmetric key $SyKey_{y,j}$ and unicasts it to $H_{y.}$
        • $H_y$ decrypts the message with $SyKey_{y,j}$
        • $H_y$ encrypts the decrypted message with the public key $Pbl_x$ of $H_x$ and sends it to $H_x$
        • $H_x$ decrypts the message with its own private key $Pvt_x$
        • Now, $H_x$ encrypts the message $<Res_y, V_j>$ with $SyKey_{x,i}$ and sends it to the requesting peer $G_{x,i}$
        • $G_{x,i}$ will then decrypt the received message using the symmetric key $SyKey_{x,i}$

    else

        $H_y$ unicasts 'search failed message' to $H_x$

    end
end
```

Figure 4: Intra Group Lookup Algorithm with Security in RC based architecture

4 Multicast Algorithms with Anonymity and Security

The basic multicast algorithms of RC based architecture presented in [12] have been enhanced in this section with the concepts of security and anonymity. In [12] we have considered designing a highly efficient capacity-constrained overlay multicast protocol. Our architecture is a 2-level one. Number of nodes (group-heads) *n* on the level-1 ring is just the number of distinct resource types and in any group (cluster) at level 2 there can be any number of nodes. Note that the number of distinct resources *n* is much smaller than the total number of nodes *N* on the ring in [4]. It has inspired us to use some idea from [4], especially transforming the multicast problem to a broadcast one and appropriately augmenting it with ours to design a highly efficient any source capacity-constrained multicast protocol suitable for the RC-based architecture with much less hop and communication complexities compared to the work in [4]. The multicast algorithm with anonymity where $c^s_x \geq n_r$ is explained in Figure 5.

4.1 Multicast Algorithm [12] with Anonymity where capacity of group head ≥ #groupheads

```
Scenario 1: $c^s_x \geq n_r$ (with Anonymity)

1. Source peer $G_{x,i}$ unicasts mcast_msg to its group head $H_{x.}$
2. $H_x$ gathers all ip_addresses of fellow groups heads from the GRT. $H_x$ replaces the ip_address of $G_{x,i}$ to its own in the mcast_msg it received and unicasts to all fellow group heads participating in the multicast.
3. If a receiver group head is also a multicast group member,
    a. It makes a copy of the mcast_msg and keeps it for itself.
    b. Replaces the ip_address of $H_x$ in the mcast_msg to itself and unicasts to each of its members.

   else

       Replaces the ip_address of $H_x$ in the mcast_msg to itself and unicasts to each of its members.

   end
```

Figure 5: Multicast protocol with anonymity where capacity of the group head $(c^s_x) \geq$ number of receiver group-heads (n_r)

4.2 Multicast Algorithm [12] with Security and Anonymity where capacity of group head ≥ #groupheads

The multicast algorithm where $c^s_x \geq n_r$ considering the concepts of anonymity and security is explained in Figure 6.

```
Scenario 2: $c^s_x \geq n_r$ (with Anonymity and Security)

1. Source peer $G_{x,i}$ encrypts the message mcast_msg using the symmetric key $SyKey_{x,i}$ and unicasts it to the group head $H_{x.}$
2. Group head $H_x$ decrypts the received mcast_msg using the symmetric key $SyKey_{x,i}$ and then replaces the ip_address of the $G_{x,i}$ to its own. // Anonymity
        // Note: GRT is modified in this scenario. public key of each.
3. $H_x$ then gathers ip_addresses and the corresponding public keys of fellow group heads from the GRT.
4. Then, $H_x$ will encrypt the modified mcast_msg with the public keys of the respective target/multicast group heads.
5. $H_x$ will now unicast the encrypted mcast_msg to the target group head and repeats the same for all other group heads participating in the multicast.
6. When the message is received, each receiver group head decrypts the received mcast_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 mcast_msg and keeps it for itself.
    b. Replaces the ip_address of $H_x$ to its own, encrypts the message using symmetric key $SyKey_{a,b}$ (where a is the group head number and b is the number of group member) and unicasts it to each receiver.

    Else

    Replaces the ip_address of $H_x$ to its own, encrypts the
```

message using symmetric key $SyKey_{a,b}$ (where a is the group head number and b is the number of group member) and unicasts it to each receiver.

Figure 6: Secured Multicast protocol with Anonymity where capacity of the group head (c^s_x) ≥ number of receiver group-heads (n_r)

Example 1:

Let us consider a scenario where $c^s_x \geq n_r$

# group heads (n_r) = 7 (H_0 to H_6)
Assume that the capacity of each group head (c^s_x) = 9
Source Peer is $G_{5,12}$

In the example, Figure 7, source peer $G_{5,12}$ encrypts the *mcast_msg* using the common shared key $SyKey_{5,12}$ to the head of the group H_5. When group head H_5 receives the message, it decrypts *mcast_msg* using the common key $SyKey_{5,12}$ and then replaces the ip_address of the $G_{5,12}$ with its own address.

Now, H_5 gets the necessary information (ip_addresses and their respective public keys) of the target multicast group say

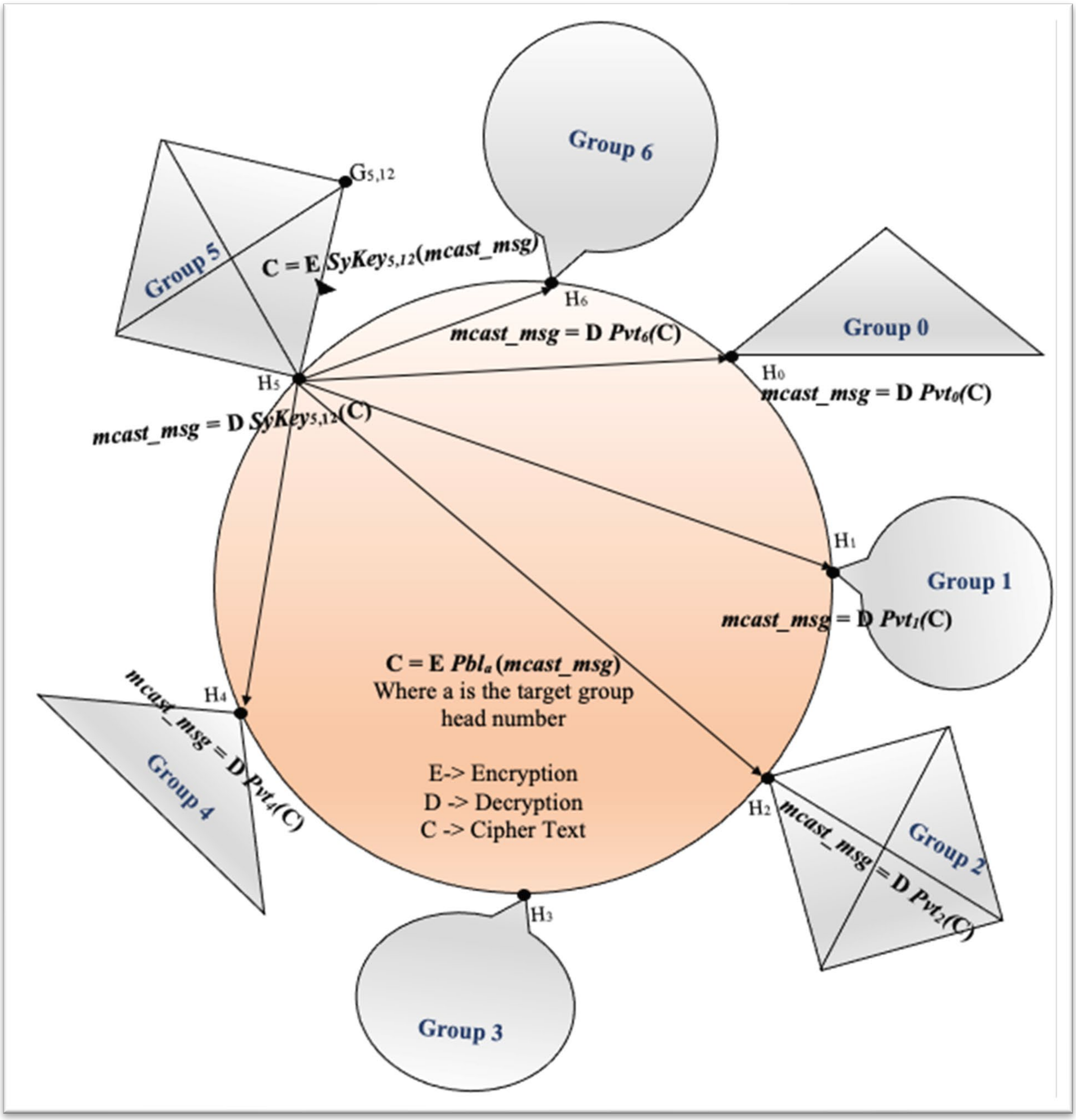


Figure 7: Example of secured multicast protocol when $c^s_x \geq n_r$

H_0, H_1, H_2, H_4 and H_6 from the global resource table (GRT) and then modifies *mcast_msg* by encrypting it with the publickeys *Pbl_0, Pbl_1 , Pbl_2, Pbl_4 and Pbl_6* respectively and unicasts the messages. On the receiving end H_0, H_1, H_2, H_4 and H_6 decrypts the *mcast_msg using the private keys Pvt_0, Pvt_1 , Pvt_2, Pvt_4 and Pvt_6.*

Each head of the target group unicasts the message in two scenarios. For example,

- Group head H_0 on receiving the *mcast_msg* from *H_5.* makes a copy before encrypting it with symmetric keys and unicasting the message to each receiver in its group.
- Group head H_1 on receiving the *mcast_msg* from *H_5.* encrypts the received message using the symmetric key and unicasts it to each receiver in its group without making a copy for itself because it is not a multicast group member.

The multicast algorithm with anonymity where $c^s_x < n_r$ is explained in Figure 8, with anonymity and security is explained in Figure 9.

4.3 Multicast Algorithm [12] with Anonymity where capacity of group head < #groupheads

Scenario 3: $c^s_x < n_r$ (with Anonymity)

1. Source peer *$G_{x,i}$* unicasts *mcast_msg* to its group head *H_x.*
2. *H_x* replaces the ip address of *$G_{x,i}$* with its own.
3. *H_x* then gathers ip_addresses of all group heads participating in the multicast from the GRT.
4. *H_x* then randomly selects the group heads and unicasts the *mcast_msg* based on its capacity.
5. Every receiver group head changes the ip_address of *H_x* present in the received mcast_msg to its own and forwards it to its successor group head on the ring.
6. Additionally, if the receiver group head is a multicast member, it saves a copy for itself.
7. Each receiver group head also sends the *mcast_msg* to all of its members.
8. Message propagation among successor group heads continues around the 1st level circle.
9. A receiver group head drops the received *mcast_msg* if it has received it already from a different source.

Figure 8: Multicast protocol with anonymity where capacity of the group head (*c^s_x*) < number of receiver group-heads (*n_r*)

4.4 Multicast Algorithm [12] with Security and Anonymity where capacity of group head < #groupheads

Scenario 4: $c^s_x < n_r$ (with Security and Anonymity)

1. Source peer *$G_{x,i}$* encrypts the message *mcast_msg* using the symmetric key *$SyKey_{x,i}$* and unicasts it to the group head *H_x.*
2. Group head *H_x* decrypts the received *mcast_msg* using the symmetric key *$SyKey_{x,i}$* and then replaces the ip_address of the *$G_{x,i}$* to its own. *// Anonymity and Step 1 & 2 is similar to scenario 1*
3. *H_x* then gathers ip_addresses and the corresponding public keys of fellow group heads from the GRT.
4. *H_x* randomly selects those many groups heads equal to its capacity/degree.
5. *H_x* will also retrive it's successor's ip address and public key from the GRT table.
6. It encrypts the modified *mcast_msg* with the public key of the selected group head *Pbl_a* where a is the number of the group head on the transit ring and unicasts it.

   // address code of H_a = resource code of Res_a = a //

7. *H_x* will now unicast the encrypted *mcast_msg* to the selected group heads as well as its successor

   // one logical hop to each group head

8. **If** receiver group head receives the *mcast_msg* for the first time (unique),

   a. Each receiver group head decrypts the received *mcast_msg* using private key *Pvt_a*.

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

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

      ii. Replaces the ip_address of *H_x* to its own, encrypts the message using symmetric key *$SyKey_{a,b}$* (where a is the group number and b is the number of group members) and unicasts it to members.

      else

      Replaces the ip_address of *H_x* to its own, encrypts the message using symmetric key *$SyKey_{a,b\ b}$* (where a is the group number and b is the number of group member) and unicasts it to members.

      end

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

   c. Message propagation continues similarly in the 1st level ring until the message reaches all the group heads on level 1 transit ring

   else

Receiver group head drops the duplicate message

Figure 9: Secured multicast protocol with anonymity where capacity of the group head (c^s_x) < number of receiver group-heads (n_r)

Example 2:

Let us consider a scenario where $c^s_x < n_r$
# group heads (n_r) = 7 (H_0 to H_6)
Assume that the Capacity of each group head (c^s_x) = 2
Source Peer is $G_{5,12}$

In this example, Figure 10, source peer $G_{5,12}$ encrypts the *mcast_msg* using the common shared key $SyKey_{5,12}$ to the head of the group H_5. When group head H_5 receives the message, it decrypts *mcast_msg* using the common key $SyKey_{5,12}$ and then replaces the ip_address of the $G_{5,12}$ with its own address (same as Example 1).

H_5 selects any 2 group heads in random (say H_1, H_4) and encrypts the *mcast_msg* with the public keys Pbl_1, Pbl_4, respectively and unicasts the message. The private keys Pvt_1, Pvt_4 respectively are used at the receiving end by the group heads to decode/decrypt the message. H_5 also unicasts the message to its successor H_6 as explained above.

Each group head that receives the encrypted message will unicast the message in the following scenarios.

- Group head H_1 on receiving the *mcast_msg* from H_5. makes a copy before encrypting it with symmetric keys and unicasting the message to each receiver in its group. H_1 also encrypts the message and forwards it to the successor

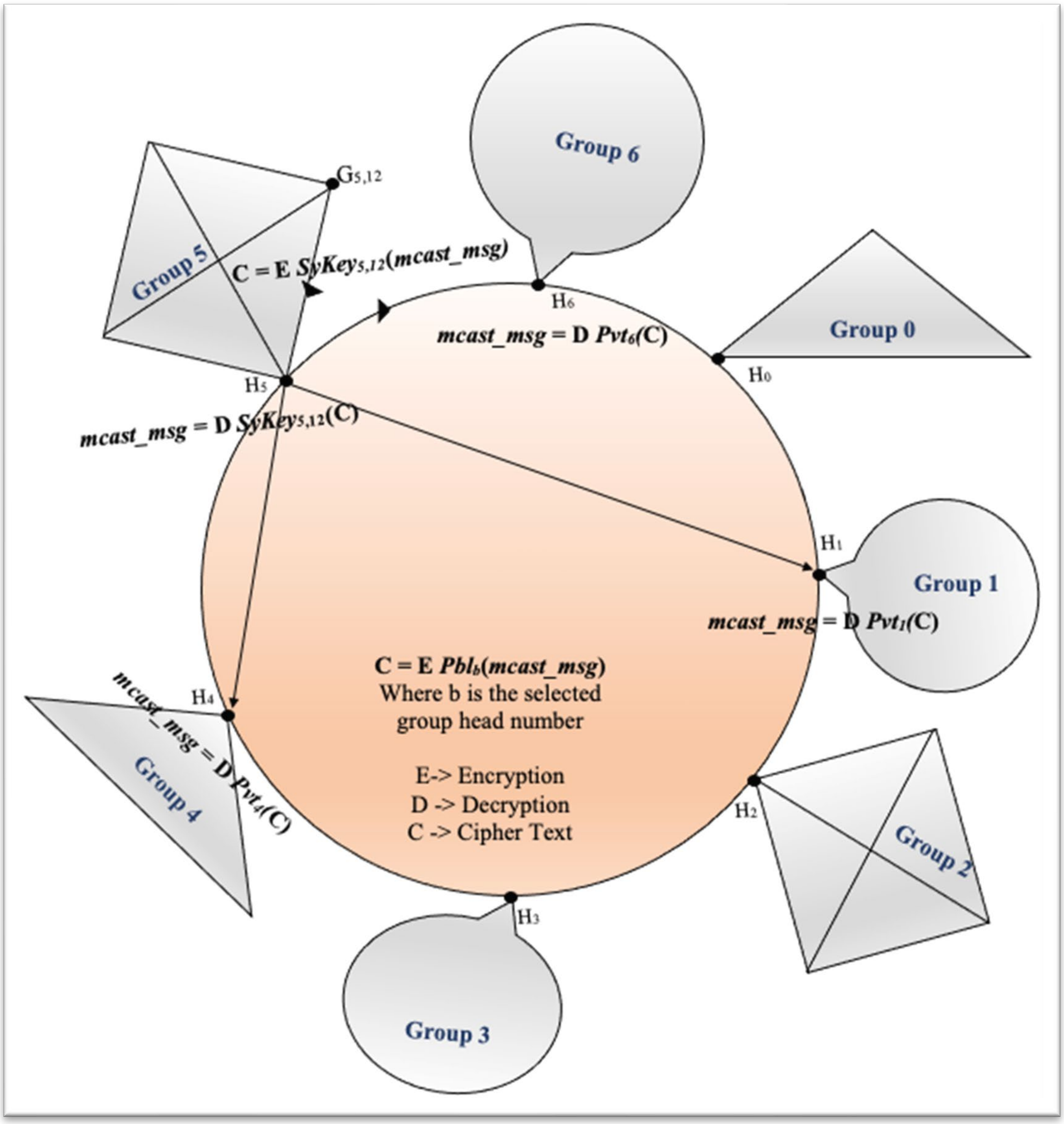


Figure 10: Example of Secured Multicast protocol when $c^s_x < n_r$

on the transit ring.

- Group head H_4 on receiving the *mcast_msg* from H_5. encrypts the received message using the symmetric key and unicasts it to each receiver in its group without making a copy for itself because it is not a multicast group member.

5 Conclusion

In this paper, we have considered a 2-level non DHT-based P2P architecture. This interest-based architecture has been the choice because

1. We have shown earlier [8] its superiority from the viewpoint of search latency of the data lookup protocols compared to those in some very prominent DHT-based contributions [15, 17, 22] and
2. Its superiority over several existing interest-based architectures [1, 3, 6, 9-10, 18]. In this paper, we have incorporated in a very effective way both security and anonymity in both inter and intra-group communication protocols which have appeared in [8].

Future work is directed at designing secured communication protocols for P2P federation built with multiple RC-based P2P components.

References

[1] L. Badis, M. Amad, D. Aîssani, K. Bedjguelal and A. Benkerrou, "ROUTIL: P2P Routing Protocol Based on Interest Links," 2016 International Conference on Advanced Aspects of Software Engineering(ICAASE), Constantine, pp. 1-5, 2016, doi: 10.1109/ICAASE.2016.7843852

[2] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, "Making Gnutella-like P2P Systems Scalable," *Proc. ACM SIGCOMM*, Karlsruhe, Germany, pp. 407-418, August 25-29 2003.

[3] Wen-Tsuen Chen, Chi-Hong Chao and Jeng-Long Chiang, "An Interested-based Architecture for Peer-to-Peer Network Systems," 20th International Conference on Advanced Information Networking and Applications - Volume 1 (AINA'06) , Vienna, 2006, pp. 707-712, doi: 10.1109/AINA.2006.93

[4] Shiping Chen, Baile Shi, Shigang Chen, and Ye Xia, "ACOM: Any-Source Capacity-Constrained Overlay Multicast in Non-DHT P2P Networks," *IEEE Tr. Parallel and Distributed Systems*, 18(9):1188-1201, Sep. 2007.

[5] 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, 2:1250-1260, March 30 - April 1 2003.

[6] M. Hai and Y. Tu, "A P2P E-Commerce Model Based on Interest Community," 2010 International Conference on Management of e-Commerce and e-Government, Chengdu, pp. 362-365, 2010, doi: 10.1109/ICMeCG.2010.80

[7] Swathi Kaluvakuri, Koushik Maddali, Bidyut Gupta and Narayan Debnath, "Design of RC Based Low Diameter Hierarchical Structured P2P Network Architecture," EMENA-ISTL, 2019; *LAIS (Learning and Analytics in Intelligent Systems)*, Springer, 7:312-320, 2020.

[8] Swathi Kaluvakuri, Koushik Maddali, Nick Rahimi, Bidyut Gupta and Narayan Debnath, "Generalization of RC-Based Low Diameter Hierarchical Structured P2P Network Architecture", *International Journal of Computers and Their Applications (IJCA)*, 27(2):77-83, June 2020.

[9] Khambatti, Mujtaba & Ryu, Kyung and Dasgupta, Partha, "Structuring Peer-to-Peer Networks Using Interest-Based Communities," Lecture Notes in Computer Science, 1st International Workshop, DBISP2P 2003, Berlin, September 2003.

[10] 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, pp. 1-5, 2009, doi: 10.1109/CCNC.2009.4784959.

[11] 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.

[12] Koushik Maddali, Banafsheh Rekabdar, Swathi Kaluvakuri and Bidyut Gupta, "Efficient Capacity-Constrained Multicast in RC based P2P Networks," EPiC Series in Computing, *CAINE*, 63:121-129, September 2019

[13] Koushik Maddali, Swathi Kaluvakuri, Bidyut Gupta and Narayan Debnath, "On Designing Secured Communication Protocols along with Anonymity for CRT based Structured P2P Network Architecture," EPiC Series in Computing, *CAINE*, October 2020 (accepted).

[14] Z. Peng, Z. Duan, J. Jun Qi, Y. Cao, and E. Lv, "HP2P: A Hybrid Hierarchical P2P Network," *Proc. Intl. Conf. Digital Society*, pp. 18-28, 2007.

[15] 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.

[16] K. Shuang, P Zhang, and S. Su, "Comb: A Resilient and Efficient Two-Hop Lookup Service for Distributed Communication System," *Security and Communication Networks*, 8(10):1890-1903, 2015.

[17] 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*, 11(1):17-32, Feb. 2003.

[18] Z. Tu, W. Jiang and J. Jia, "Hierarchical Hybrid DVE-P2P Networking Based on Interests Clustering," 2017 International Conference on Virtual Reality and Visualization (ICVRV), Zhengzhou, China, pp. 378-381, 2017, doi: 10.1109/ICVRV.2017.00087.

[19] M. Xu, S. Zhou, and J. Guan, "A New and Effective Hierarchical Overlay Structure for Peer-to-Peer Networks," *Computer Communications*, 34:862-874, 2011.

[20] M. Yang and Y. Yang, "An Efficient Hybrid Peer-to-Peer System for Distributed Data Sharing," *IEEE Trans. Computers*, 59(9)1158-1171, Sep. 2010.

[21] R. Zhang and Y.C. Hu, "Assisted Peer–to-Peer Search with Partial Indexing," *IEEE Trans. Parallel and Distributed Systems*, 18(8):1146-1158, 2007.

[22] 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*, 22(1):41-53, Jan. 2004.

Swathi Kaluvakuri (photo not available) is a Ph.D. candidate from Southern Illinois University Carbondale – School of Computing. She graduated from Jawaharlal Nehru Technological Unversity with a Bachelor of Technology degree in Computer Science major. She holds keen interest in the areas of Peer to Peer Networking and BlockChain and worked as a Software Engineer, Technical Product Support as IBM AS400 developer for NetCracker Pvt Ltd from 2012-2014.

Indranil Roy (photo not available) is currently a PhD student in Computer Science department of Southern Illinois University, Carbondale. He has completed his B.E in Electronics and Communication from RCCIIT, Kolkata, in the year 2016. He received his M.S degree in Computer Science from Southern Illinois University, Carbondale in 2018. His main research interests include Blockchain along with interest-based p2p architecture.

Koushik Maddali (photo not available) is a Ph.D. candidate in Department of Computer Science at Southern Illinois University Carbondale. He received his MS from the same university and his BS from Jawaharlal Nehru Technological University, India. His research interests include Peer to Peer Networking, BlockChain and worked on a Virtual Terminal project of Cisco from 2017-2018.

Bidyut Gupta (photo not available) received his M. Tech. degree in Electronics Engineering and Ph.D. degree in Computer Science from Calcutta University, Calcutta, India. At present, he is a professor at the School of Computing (formerly Computer Science Department), Southern Illinois University, Carbondale, Illinois, USA. His current research interest includes design of architecture and communication protocols for structured peer-to-peer overlay networks, security in overlay networks, and block chain. He is a senior member of IEEE and ISCA.

Narayan Debnath (photo not available) earned a Doctor of Science (D.Sc.) degree in Computer Science and also a Doctor of Philosophy (Ph.D.) degree in Physics. 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. Dr. Debnath has been the Director of the International Society for Computers and their Applications (ISCA) since 2014. Formerly, Dr. Debnath served as a Full Professor of Computer Science at Winona State University, Minnesota, USA for 28 years (1989-2017). Dr. Debnath has been an active member of the ACM, IEEE Computer Society, Arab Computer Society, and a senior member of the ISCA.

Understanding the Anti-Mask Debate on Social Media Using Machine Learning Techniques

Luca Cerbin*, Jason DeJesus*, Julia Warnken*, and Swapna S. Gokhale*
University of Connecticut, Storrs, CT 06269 USA

Abstract

Masks are believed to slow the spread of Covid-19, and can prevent many deaths, yet this inexpensive, common sense public health measure has ignited a fierce debate in the United States. Opponents of masks or anti-maskers have resorted to measures such as organizing protests and marches to make their views public. They have also taken to social media platforms to vigorously argue against the use of masks. Even with the advent of vaccines, masks are still likely to be recommended for a long time. It then becomes important to mine the debate around masks to understand the concerns of the detractors and the arguments used by the proponents to counter these concerns. This paper analyzes the mask dialogue on Twitter, using the data collected in July and August 2020, which coincided with the time when the stay-at-home orders were being relaxed, and the opening of schools and other activities was being contemplated. These tweets are explored in three ways – informal opinion mining is used to reveal the reasons for concerns and support, social parameters of the tweets and tweeters are analyzed to expose the dynamics of the two communities, and classification framework is built to distinguish between pro- and anti-mask tweets so that the latter can be tagged to prevent the spread of discordant information. Our results indicatethat the concerns of anti-maskers are more political and ideological rather than related to the adverse health impacts of masks. Members of the close-knit, small anti-mask community promote each other's views compared to the pro-maskers, although the anti-maskers themselves are not fringe by any means. The classification framework can detect anti-mask tweets with excellent accuracy of over 90%, and hence, it can be used to label tweets that sow misinformation about masks before they spread through the ether and influence people.

Key Words: Masks, anti-mask, pro-mask, twitter, classification, machine learning.

1 Introduction and Motivation

The coronavirus pandemic has upended every single tenet and ritual of our modern society. Discussion and practice of measures such as masks, social and physical distancing, vaccines, hand hygiene, and disinfectants have now become a part of our daily routines. Of these, one of the most contentious issues that has bitterly divided the U.S. society is the wearing of masks. A seemingly simple act of wearing a facial covering that covers both the mouth and the nose serves as a stark reminder of the pandemic, and has also been the topic of a fierce debate. Proponents of masks point to several studies that recommend their use to slow the spread of Covid-19 [19]. Opponents, however, contend that most of the studies have looked at the use of face masks in health care, and not community settings. They further claim that these studies were observational, not the gold standard of science, a randomized controlled trial. It does not help that early in the pandemic public health officials in the U.S. discouraged the use of masks by the general public. At the time "mass masking" was not recommended either by the CDC or the WHO, perhaps to conserve them for healthcare and other front-line workers [9]. Later, however, they backtracked from this initial position and vigorously advocated the use of masks to blunt the spread of the virus and prevent deaths. The u-turn regarding masks and the subsequent political divide over them has come to symbolize the chaos of the U.S. response to the still-raging pandemic [50].

Expressions of pro-mask and anti-mask opinions are plentiful and varied in the physical, offline world. In some counties, where the coronavirus has surged out of control, mask mandates have been imposed and this has further outraged their residents. Those opposed to mask mandates have staged protests, and one local health official had to even quit her job after receiving a death threat for a mask order [33]. In addition to expressing their views through their actions by either wearing or not wearing masks in public spaces and/or organizing protests, people have often turned to social media platforms such as Twitter and Facebook to express their support or opposition to masks. These social media platforms havenot only been woven tightly into the fabric of our society, but sharing on these platforms has skyrocketed especially during the pandemic, because a number of people are either in self-imposed or government-mandated isolation and lockdown. Therefore, in addition to the offline expression of the pro- and anti-mask opinions, this debate over masks has been playing out vociferously over these platforms as well.

Compliance with masks has been spotty at best through the U.S., even though the CDC and other public health experts have repeatedly indicated, on multiple occasions, that wearing

* Computer Science & Engg. Email: {swapna.gokhale} @uconn.edu.

masks could save a significant number of lives [13, 23]. Furthermore, the use of masks is likely to continue despite the approval and roll out of vaccines. In fact, masks and social distancing will probably be recommended at least for a while, because a lot is still unknown about what protections vaccines can afford in terms of preventing infection, its severity and its spread [44]. It is thus believed that masks are and will continue to be an effective tool against fighting the pandemic. Given the usefulness of masks, it is then imperative to understand the public outlook towards their use. Based on such understanding we can launch educational and public awareness initiatives to dispel the myths and misinformation and encourage their adoption broadly. Moreover, understanding the drivers and spread of misinformation can be valuable during future pandemics.

The novelty of the paper lies in understanding the debate over masks through social media dialogue. Using the data collected in July 2020 and August 2020 from Twitter, just when the "stay home, stay safe" orders were beginning to be relaxed, and the opening of schools was being contemplated, this paper seeks to answer the following research questions:

(i) Will masks be embraced by the community at large, or are there a significant numberof detractors and skeptics (anti-maskers) who will continue to defy the simple, inexpensive and most innocuous of the public health guidelines? What misgivings do the detractors and skeptics express? What misinformation about masks is circulating on social media, which if left unchecked will make an eventual broad scale acceptance of masks by the public almost impossible? (ii) How socially cohesive and tight knit is the community of anti-maskers, compared to the group of supporters? (iii) To curb the spread of discordant information, is it feasible to automatically detect the tweets that carry misinformation and express skepticism about masks before they make their way through the ether? This is especially important as social media users are more likely to believe false information about Covid-19 and ignore public health advice [43].

Our results expose the culture wars associated with the use of masks. Concerns of anti-maskers appear to be more motivated by politics and ideology, rather than driven by actual health, convenience or any other pragmatic reasons. Benefits to public health, advocated by pro-maskers to counter this anti-mask rhetoric is weak and unlikely to be convincing and per-suasive. Although the anti-mask views are not fringe, the group of anti-mask users is small, tight knit, and very supportive and encouraging of each other. Despite the small size of their network, anti-maskers have effectively spread their opinions and views widely. Separating the anti-mask tweets from the pro-mask ones is feasible, and can be accomplished with high accuracy by employing a combination of linguistic, auxiliary, and social features to train machine learning models. Most ML classifiers, including Support Vector Machines, RandomForest, Gradient Boosting, achieve an accuracy of over 90% in separating the anti-mask tweets from the pro-mask ones. Importance analysis shows that a bulk of the contribution towards classification comes from the text of the tweets, and from the social parameters that indicate the reach and popularity of the tweets and the tweeters.

The rest of the paper is organized as follows: Section 2 explains the process of collecting and preparing the data. Section 3 describes opinion mining. Section 4 summarizes the findings of social analysis. Section 5 presents the sequence of steps involved in building the classification framework. Section 6 discusses the results. Section 7 compares and contrasts related research. Section 8 offers concluding remarks and directions for future research.

2 Data Preparation

This section discusses three steps in the preparation of data: data collection, data labeling, and data pre-processing.

2.1 Data Collection

Data was collected twice, one month apart, using the crawling seeds *#wearadamnmask*, *#nomaskforme*, *#maskupamerica*, *#masksareforsheep*, *#nomasks*, *#nomaskmandate*, *#antimaskers*, *#maskitorcasket* in July 2020 and August 2020. These two-time frames were chosen as they represented two significant epochs in the mask debate. In July 2020, as the country was emerging from the lockdown, masks were viewed as a way to restore a sense of normalcy. Furthermore, masks came into sharp spotlight in this one-month period because of the tussle surrounding the reopening of schools, and students returning to college campuses. Masks also became a hot button issue during this period when the Democratic presidential candidate Joe Biden suggested that if elected he will issue a national mask mandate [39]. In the same period, leading public health experts, including the CDC promoted the use of masks as "life-saving", highlighting that if everyone committed to wearing masks, we could save a significant number of American lives [13]. Thus, the two data collection epochs one month apart occurred during an eventful period for the fate of the masks and their acceptance. Both data sets were collected using the using the rtweet library in R [28]. The following represent examples of pro-mask and anti-mask tweets from the July 2020 and August 2020 data sets.

July 2020:

Erry time I ride @trimet to work there's always a couple ppl not wearing masks... like really? #WearADamnMask (**P**)

History REPEATING itself #NoMasks #NoMaskOnMe #BLMIsADomesticTerroristGroup https://t.co/oJjoPA9WW9 (**A**)

August 2020:

At least my mask hides my pimple #moreimportantly issaveslives #WearADamnMask https://t.co/hbxquonf8R (**P**)

@healthvermont @CDCgov We are tired of the government telling us to stay safe. Freedom Trumps safety in America. Fellow Vermonters tell the government. We will not comply! #NoMasks #freedom #Vermont (**A**)

2.2 Data Labeling

This set of crawling seeds was harvested because it included both the anti-mask and pro-mask perspectives. For example, we expected that hashtags such as *#maskupamerica* and *#maskitorcasket* would be used in tweets that support masks, whereas hashtags such as *#nomasksforme* and *#masksareforsheep* would be used to show opposition. We anticipated that the tweets would neatly separate according to support and opposition, consistent with the corresponding hashtags. Such clear, neat separation would obviate the need for manual labeling and facilitate weak supervised learning with the hashtags serving as labels. Skimming through the tweets, however, invalidated this assumption and many hashtags were creatively embedded in both supporting and opposing tweets. For example, the hashtag *#nomask* is used in the following two tweets, the first one is clearly pro-mask whereas the second one is anti-mask. In fact, the use of anti-mask hashtags in tweets that express pro-mask opinion has been found to be prevalent. It is believed that such use inadvertently boosts the anti-mask movement, making it difficult to automatically separate such tweets [12]. Such mocking may also fuel the anti-maskers.

Save lives - wear a mask, clean your hands keep a safe distant. #nomask (**P**)

I feel fine cause I dont wear one! #nomask (**A**)

Manual annotation of the tweets seemed inevitable, and was undertaken to classify each tweet into one of two groups –'A' for anti-mask, and 'P' for pro-mask. The entire data set was labeled twice, independently, with a gap of about one week between the two labelings. Duplicates were eliminated before the labeling. Only those tweets where the labels matched on two independent occasions were included in the final corpus, which consisted of 4042 tweets.

About 500 tweets were eliminated because of mismatch of labels. In the corpus, about 57% of the tweets are pro-mask, and 43% are anti-mask. This data also contained a number of public safety announcements (PSAs) from schools, colleges and sports teams. There were tweets that expressed political opinion regarding the conventions, wildfires in California, and the BLM protests without the express mention of masks other than the hashtag. In the manual labeling process, we eliminated these tweets to build a high-quality data set that truly reflects the public opinion about masks instead of other peripheral and allied political issues.

2.3 Data Pre-processing

The labeled data was pre-processed in the following steps shown in Figure 1. It was converted to UTF-8 encoding, and transformed to lower case. Then, numbers, punctuation and stop words were removed. After word stemming and stripping white space, domain specific words that occur in both pro-mask and anti-mask tweets with a similar frequency were removed as they are likely to be uninformative.

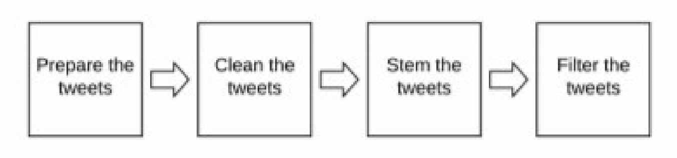


Figure 1: Pre-processing steps

3 Opinion Mining

We represent the remaining words in each of the pro-mask and anti-mask categories into word clouds as shown in Figures 2a and 2b. We read words from these word clouds, and find associations between them to reveal interesting insights into the opinions of supporters and detractors.

Proponents point to the life-saving benefits associated with the use of masks. They promote the use of masks through several phrases such as "masks save lives", "wear a mask save a life". They advocate the covering of both the mouth and nose. Medical terms such as doctor, hospital, patient appear which points to the vital role played by medical and front-line workers during the pandemic. Pro-maskers are also believers of allied public health recommendations such as staying home, washing hands, and practicing social distancing to curb the spread of the virus. The role of masks in bringing

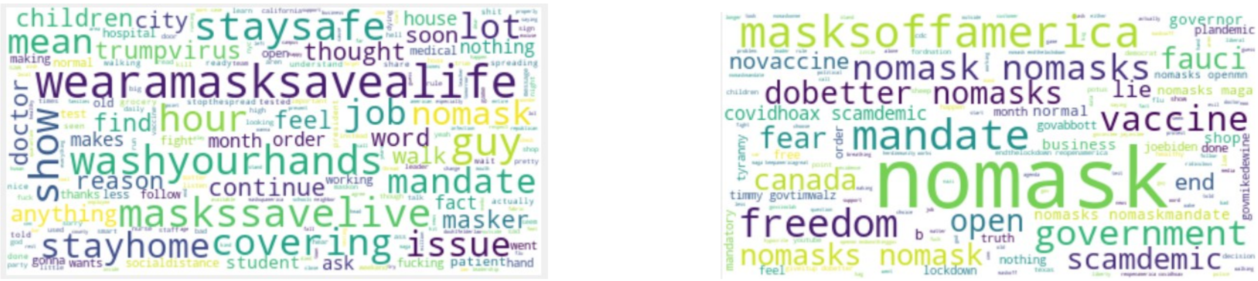


Figure 2: Word clouds of pro-mask and anti-mask tweets

the lives of students and children back to normal is highlighted. Blame for mismanaging the Covid-19 pandemic is laid at President Trump's feet with the phrase "trumpvirus". Some pro-maskers also appear to be supporting a more aggressive stance over the use of masks through a mandate. It can be imagined that some pro-maskers may have had encounters with anti-maskers, through the use of the words no mask, walk, guy, masker.

Anti-maskers persistently and vigorously oppose masks, as evident through the repeated use of phrases such as nomasks, nomasksforme, masksoffamerica, nomaskmandate. The overlap or intersection between anti-maskers and anti-vaxxers is on display through the terms novaccine and vaccine. This group is of the fervent belief that Covid is a hoax, as expressed through the terms covidhoax, scamdemic, and plandemic. Pandemic is Dr. Milkovitz's documentary that claims to have exposed Dr. Fauci's fraud [15]. Views such as masks take away freedom, and amount to tyranny are voiced. Calls to open businesses, end the lockdown, especially directed at Republican politicians such as Gov. Abbott and Gov. Mike

Devine who opposed mask mandates are found in these clouds. The sentiment that the government is fear mongering and withholding the truth are also expressed. Joe Biden and his proposed mask mandate after inauguration also appear in the list of concerns.

This mining and analysis of associations exposes that the concerns against the use of masks are more political and ideological, rather than being rooted in any health or convenience matters. This divide may have largely been fueled by the Trump administration's defiance towards the wearing of masks. Proponents and opponents can be seen to be split along the political divide. Those leaning left view the virus as a serious threat, and those leaning right tend to downplay its seriousness. In viewing a mask as a political symbol, proponents may be viewed as trying to hype the seriousness of the virus, whereas, opponents may be viewed as purposely trivializing the virus and prioritizing the economy over health and safety. Interestingly, both proponents and opponents argue that masks block their respective happiness, to the proponents of masks, blocking of happiness is tantamount to threatening their safety, whereas to the opponents the blocking of happiness arises from hijacking their liberty. Many medical reasons are cited for not wearing masks, these include claustrophobia, panic attacks, autism spectrum disorder, and sensory processing issues.

4 Social Analysis

Social media platforms are conducive to a viral spread of rumors and misinformation. Tweet data collected using the rtweet library also records a number of parameters that may indicate the reach of the tweets and tweeters, which may further offer insights into how a tweet may circulate over the platform. In this section, we explore the social parameters of the tweets [26] to examine the conjecture that the anti-mask community is close knit and much better organized and connected compared to the pro-mask com-munity. A brief description of these parameters follows. Values of these parameters, along with their classification into user- or tweet-level are reported in Table 1.

Table 1: Social features

Parameter	Pro-Mask	Anti-Mask	User/Tweet
Percentage	57.74	42.26	Tweet
Avg. Tweet Length	152.20	154.83	Tweet
Avg. Retweet Count	1.11	4.95	Tweet
Avg. Favorite Count	4.43	11.42	Tweet
Avg. Follower Count	5596.72	3099.60	User
Avg. Friend Count	2624.66	2475.14	User
Avg. Status Count	30976.34	25587.25	User
Avg. Favorites Count	67	3	User
Avg. List Count	102.72	28.54	User
Avg. Quote Retweet Count	5586.10	3342.84	Tweet
Avg. Quoted Favorite Count	21599.03	12764.23	Tweet
Avg. Quoted Follower Count	2163732.13	12677596.95	Tweet
Avg. Quoted Friend Count	7906.06	9343.33	Tweet
Avg. Quoted Status Count	76204.73	47290.24	Tweet
Percent Verified	2.18	0.18	Tweet
Percent Mentions	30.21	48.36	Tweet
Percent Replies	28.28	43.03	Tweet
Percent Quoted	22.41	24.47	Tweet

- **Avg. Tweet Length**: Number of characters in each tweet.
- **Avg. Favorite Count**: Number of times a tweet has been liked by Twitter users.
- **Avg. Quoted Favorite Count**: Number of likes the quoted tweet received.
- **Avg. Quoted Retweet Count**: Number of retweets the quoted tweet received.
- **Avg. Quoted Followers Count**: Number of followers of the user who tweeted the quoted tweet.
- **Avg. Quoted Statuses Count**: Number of status updates of the user who created the quoted tweet.
- **Avg. Quoted Friends Count**: Number of friends of the user who tweeted the quoted tweet currently has.
- **Avg. Retweet Count**: Number of times a tweet is retweeted.
- **Avg. List Count**: Number of public lists in which the tweeter claims membership.
- **Avg. Statuses Count**: Number of tweets posted by the tweeter.
- **Avg. Followers Count**: Number of followers of the tweet owner.
- **Avg. Friends Count**: Number of friends of the tweet owner.
- **Percent verified**: Percentage of tweets that were shared from verified accounts.
- **Percent quoted**: Percentage of tweets that quote other tweets.
- **Percent replies**: Percentage of tweets that were replies to existing tweets.
- **Percent mentions**: Percentage of tweets that mentioned other users.

The average length for pro-mask and anti-mask tweets is similar, however, the average retweet and favorite counts of these two classes of tweets are significantly different. Roughly,these counts are about 3-4 times higher for anti-mask compared to pro-mask tweets. If retweeting and favoriting (liking) is viewed as akin to endorsing the content of the tweets, then a potential explanation for this discrepancy could be the strength of the passion regarding anti-mask opinions, and a lack of similar passion when expressing pro-mask opinions. When a pro-mask tweeter shares a tweet supporting opinion. On the anti-mask side, however, because the nature of the position is not popular or mainstream, someone who agrees with the opposing view is more likely to make it known masks, that opinion may be assumed to be more mainstream, and it may not be considered valuable to favor a normal

Comparing the metrics at the user-level, the average follower count is higher for pro-mask users compared to anti-mask users. This skew could be explained by the much higher percentage of pro-mask tweets being posted from verified accounts compared to anti-mask tweets. Verified accounts usually belong to the more famous, elite and educated people, and they tend to have many more followers than the ordinary users. These accounts could also belong to public health authorities and organizations, who tweet to encourage people to wear masks. In terms of absolute numbers, a total of 67 verified users have shared pro-mask tweets, whereas only 3 verified users have shared anti-mask tweets. This suggests that more prominent users are in favor of masks compared to the few known ones that oppose them.

An interesting difference is in the average status count, which is slightly higher for pro-mask compared to anti-mask users. This may be possible because pro-mask users may be more extroverted and comfortable sharing their mainstream opinion. A user who shares many statuses is more likely to desire to keep their friends/followers abreast of what is happening, which is a social characteristic. Pro-mask users also have a higher favorites count, perhaps showing they are generally more active on Twitter. As further evidence of this, the average quote retweet count is significantly higher for pro-mask users than anti-mask users. Average quote favorite count and average quote status count are also higher for pro-mask users. The average list count, which indicates the number of lists or groups of which a user is a member is substantially higher for pro-mask users compared to anti-mask users. These social parameters indicate that pro-mask users appear to be more active on the platform and they are more likely outgoing, because many metrics that require active participation on the platform are significantly higher for pro-mask users. On the other hand, the percent of tweets that are replies to existing tweets, and the percent of tweets that mention other users is higher for anti-mask users. Replies always have user mentions, but not all tweets that contain user mentions are replies. The percent of tweets that contain quoted tweets is very similar for both pro- and anti-mask users. This suggests that the anti-mask community although small is deeply engaged in supporting and promoting the anti-mask view.

In summary, the pro-mask community is open and active in encouraging the use of masks (perhaps through the sharing of useful, public health benefits), but pro-maskers are less engaged with each other. Anti-mask users, on the other hand, form a tight-knit group and appear strongly interested in endorsing and propagating the anti-mask view. However, it is important to remember that about 40% of the tweets support the anti-mask view, so although the number of anti-maskers may be few and close knit, the view itself is not fringe.

5 Tweet Classification

While masks may not be completely effective, they certainly do not amount to a “dangerous waste of time”. They can be at least partially effective, and prevent a significant number of deaths, as indicated by the CDC [13, 23]. Anti-maskers seem just like many other conspiracy cultists including anti-vaxxers and flat earthers, and they share misinformation about masks. Such discordant information spreads virally over these platforms, and provides impetus in the use of these platforms to organize marches, rallies and demonstrations. Such anti-mask rhetoric can persuade people on the fence to further denounce the use of masks. Pro-maskers seem to be appealing to the collective goodwill to adhere to public health measures such as washing hands, staying home, and maintaining a safe distance, which is likely to be ineffective. It then becomes imperative to identify and label the anti-mask dialogue, in the hope of limiting its persuasive power. Given the excessive volume of content that gets sharedon these platforms, manual separation of anti-mask tweets is impossible, highlighting the need for automated detection. This section presents a classification approach to distinguish between pro-mask and anti-mask tweets, labeled as ‘P’ and ‘A’ respectively.

5.1 Feature Extraction

The first step is to extract features that abstract away the important properties of the tweets while ignoring the unnecessary details. We considered linguistic, auxiliary, social, psycho-linguistic, and sentiment features as discussed below in the classification framework.

5.1.1 Linguistic Features. Tweets were processed using natural language techniques so that the key features including the semantic relationship between the words and the contextual information of the words and sentences were numerically encoded in high-dimensional vectors. We considered a number of vector representations such as bag-of-words [52], *n*-grams, Term Frequency-Inverse Document Frequency (TF-IDF) [35] and word2vec and doc2vec [36] that are commonly used for classification. Of these, we used the *n*-grams/TF-IDF and word embeddings.

In the *n*-grams method, a sample of text is represented by the most frequent instances of every unique *n* continuous words as a dimension. The most frequent word grams are selected from the entire corpus. The tweets were represented through unigram (1-gram) vectors, and the weight for each unigram is its TF-IDF score which is given by:

$$TF\text{ - }IDF = tf * log(_{df}) \tag{1}$$

In Equation (1), *tf* is the number of times a particular term occurs in a tweet, *T* is the total number of tweets, and *df* is the number of tweets containing that particular term. The main advantage of a $TF - IDF$ score over the simple frequency counts of the *n*-gram method is that it assigns a higher weight to the terms that occur more frequently through the entire data set. Thus, the $TF - IDF$ score should assign a higher weight to those phrases that are the most important in determining whether a tweet is anti-mask or pro-mask. After pre-processing, the size of our corpus (number of unique words) is over 9000. Of these, we calculated the TF-IDF vector representations of the top 2000 most relevant unigrams. We used the TF-IDF implementations from the NLTK library to extract these features [31].

Although the TF-IDF score provides a differentiated representation of the words based on their frequency of occurrence, it does not preserve any relationship between the words. Word embeddings are a powerful technique that represent semantically related words as closely related vectors. Words with similar meanings are mapped to low-dimensional, non-sparse vectors that exist near each other in a predefined vector space. A good word embedding can preserve the contextual information behind words in a tweet that a *n*-gram/TF-IDF scheme cannot. We use Word2Vec, which is a popular technique to create distributed numerical representations of word features using a two-layer neural network with back propagation [36]. Word2vec trains words against other words that neighbor them in the input corpus. Word2Vec allows us to encode the context of a given word by including information about preceding and succeeding words in the vector that represents a given instance of a word. Therefore, the results obtained from using Word2vec may result in a much better classification.

We implemented Word2Vec using the gensim library [45]. From the preprocessed tweets, we generated a list of tokens, and built a model to represent each word by a 10-dimensional vector, where the parameter *min count* is 1. The number of workers, which is the number of partitions during testing is 8. The model considers all the words in the corpus. We created the vector representations for all the tokens, and the total number of epochs used is 25. We used the continuous bag of words (CBOW) model to generate the representations. The other option was to use the skip gram model. Skip gram works well with a small amount of data and is found to represent rare words well. On the other hand, CBOW is faster and has better representations for more frequent words [27, 40]. We chose CBOW based on earlier success with this model to classify the anti-vaxx dialogue [42].

We also included POS (part-of-speech) tagging using the NLTK library [31]. The NLTK library provides the ability to classify each word as one of 35 parts of speech. POS tagging occurred before removing stop words to capture any differences in the raw text. The occurrences of each part of speech is counted for each tweet and fed as input to our models.

5.1.2 Psycho-Linguistic Features. Some studies show that refusal to wear masks may be linked to sociopathic, narcissistic and psychopathic tendencies [51]. These leanings are reflected in an excessive use of first-person pronouns “I” and “me”, in written and spoken language. Therefore, we considered the use of these first-person pronouns in the anti-mask and pro-mask tweets. The use of these pronouns, however, did not appear significantly different in these two groups. In total, the pro-mask tweets used “I” 8 times compared to the use of “I” 14 times in the anti-mask tweets. Counting the instances of both “I” and “me”, the pro-mask tweets had 103 occurrences, while the anti-mask tweets had 102. Because the differences appeared insignificant, these first-person pronouns were not considered further in the classification.

5.1.3 Auxiliary Features. Written texts including social media feeds do not carry with them clues that can be gathered from facial expressions and body language that accompany face-to-face or spoken communication. Therefore, in social media texts, users may use a variety of punctuation marks and other means such as hashtags and emoticons to emphasize their point. These auxiliary features are believed to somewhat substitute the clues that can be learned from communicating in the physical space, and are known to improve classification accuracy [14]. Therefore, we included numbers of hashtags, mentions, punctuations, links, words in upper case letters, question marks, exclamation marks, periods, quotations, and all punctuations as features.

5.1.4 Social Features: We used the social features listed in Table 1 in the classific**c**ation framework. Because their values differed widely, we transformed each feature using the MinMaxScaler in sklearn library [1]. This function scales and translates each feature individually such that it lies in the range of 0 and 1. This transformation is often used as an alternative to zero mean, unit variance scaling [1].

5.1.5 Sentiment Features: Textblob [32] and Vader [25] sentiment scores, computed for each original tweet (before preprocessing) were used in the classification. TextBlob calculates the sentiment polarity for each tweet, which ranges from −1 to +1, where −1, 0 and +1 indicate negative, positive and neutral sentiment respectively. Vader computes a compound score as a normalized and weighted composite score obtained by analyzing each word in a tweet for its direction of sentiment - a negative (positive) valency for negative (positive) sentiment. It therefore ranges from -1 to +1 depending on the net sentiment of the tweet. The compound score provides a single unitary measure for the sentiment analysis of the tweet.

5.2 ML Models

We considered the following popular supervised machine learning models for classification. Implementations of these models in the Scikit package, were used [7], and the parameters chosen for implementation are listed below.

- **Random Forests**: Random Forests is an ensemble learning classification technique based on Decision Trees [30]. By using bagging to reduce variance, the method generates a number of decision trees with different training sets and parameters. Random Forests is easy to apply and a flexible approach. To a certain degree it eliminates the overfitting problem that often occurs when using decision trees. The number of trees was 100, the number of features in each tree was equal to the square-root of the number of total features by default, and each decision tree was allowed to grow fully up to its leaves.

- **Support Vector Machines (SVMs)**: Support Vector Machines (SVMs) is a powerful classification technique that estimates the boundary (called hyper-plane) with the maximum margin [49]. We used SVMs with RBF kernel, the regularization parameter C is set to 1000, and kernel coefficient gamma is set to 0.01. The remaining parameters are set to their default values.

- **Multi-Layer Perceptron (MLP)**: Multi-Layer Perceptron (MLP) is one of the feed-forward Artificial Neural Networks (ANN) that consists of input, hidden, and output layers [11]. The numbers of neurons in these four layers were 10, 8, 5, and 2. We used rectifier linear unit (ReLu) activation function to minimize the vanishing problem that the gradients of the loss function goes toward zero that usually occurs in deep neural networks.

- **Gradient Boosting (GB)**: Gradient Boosting is another ensemble learning technique which builds classifier trees in a forward stagewise fashion [16]. Each stage takes a small step towards the minimization of classification error from the previous step. The algorithm continues until a maximum number of trees are built or there is no significant improvement in minimizing the error. Finally, predictions for the test data are obtained by combining predictions of the trees built in each stage using a weighted sum to obtain the final prediction. The parameters of the gradient boosting algorithm can be classified into tree specific, boosting and miscellaneous parameters. The number of trees is set to 1600, the fraction of observations to be selected for each tree (subsample) is set to 0.55, the maximum depth of each tree is set to 5, the minimum samples in each leaf is set to 1, the learning rate which determines the impact of each tree on the final outcome is set to 0.05.

- **Long Short-Term Memory (LSTM)**: LSTM is an artificial recurrent neural network architecture used in deep learning [24]. We used Keras library to implement the model [10]. Keras computations require vectors of the same length, we truncated and pad the input sequences to 360. The model knows that zero values carry no information. In the LSTM model, the first layer is the embedded layer that uses vectors of length 100 to represent each word. The next layer is the LSTM layer with 100 memory units (smart neurons). Finally, because this is a classification problem, we use a dense output layer with a single neuron and a sigmoid activation function to predict either 0 or 1 for the anti-mask and pro-mask classes. Binary cross entropy is used as the loss function. The efficient ADAM optimization algorithm was used, and the model is batch sizes of 64 and 100 epochs.

5.3 Performance Metrics

Our objective is to identify anti-mask tweets, and hence, to define the performance metrics, we designate the anti-mask and pro-mask classes as positive and negative respectively.

Tweets can thus be classified into four groups – true positive (TP) (anti-mask labeled anti- mask), true negative (TN) (pro-mask labeled pro-mask), false positive (FP) (pro-mask labeled anti-mask), and a false negative (FN) (anti-mask labeled pro-mask). These four groups lead to the following metrics to compare classifier performance:

- **Accuracy**: Accuracy is defined as the percentage of tweets that are labeled correctly:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \qquad (2)$$

- **Precision**: Precision measures the percentage of the tweets that are actually anti-mask out of all the tweets that are predicted as anti-mask:

$$\text{Percision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad (3)$$

- **Recall**: Recall measures how many of the anti-mask tweets are actually labeled as anti-mask:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad (4)$$

- **F-score**: F-score seeks a balance between Precision and Recall:

$$\text{F1} = 2\text{x}\,\frac{\text{Prescision}*\text{Recall}}{\text{Precision+Recall}} \qquad (5)$$

Precision is the percentage of relevant from the set detected and recall is the percentof relevant from within the global population [34]. Precision is an important measure to determine when the costs of a false positive is high. Applying symmetrical logic, recall would be the metric of significance when the cost of a false negative is high. In the context of detecting anti-mask tweets, false positive labeling implies that a pro-mask tweet is labeled as anti-mask, whereas a false negative labeling implies that an anti-mask tweet is labeled as pro-mask. In false positive labeling, because a pro-mask tweet may be labeled as anti-mask it may be subject to actions such as being censored or tagged for misinformation. However, any additional stringent punitive actions such as removing the tweet altogether may lead to freedom of speech violations. In false negative labeling, an anti-mask tweet will slip through the cracks and will not be tagged for carrying misinformation. While such mislabeling may cause damage by spreading discordant information, it will not lead to any violations of people's individual rights. Therefore, in this problem, precision may be a more important metric than recall. A balance may also be sought between precision and recall to trade off in fringing freedom of speech against the spread of discordant information.

6 Results and Discussion

We split the entire corpus using stratified sampling into two partitions; the training partition consisted of 75% and the testing partition contained 25% of the tweets. All the models listed in Section 5.2, except for LSTM, were trained and tested on a combination of TF- IDF, word embedding, POS tags, auxiliary and social features. LSTM was fed pre-processed text directly along with auxiliary and social features. We combined all the features for model training, guided by its success in their use in detecting tweets that spread vaccine misinformation [42]. The results of the performance metrics for all the models are noted in Table 2.

The table shows that all the classifiers except for SVM can distinguish between anti- mask and pro-mask tweets of accuracy and F1-score over 90%. Moreover, the accuracy of the SVM is only slightly lower than 90%. For some models, the accuracy reaches as high as 96%. These results show that anti-mask tweets that can sow discordant information about masks, and promote non-compliance can be accurately separated from social media dialogue. They also show that this accuracy can be achieved even after data from different time periods is combined. Each time period presents a different context or a backdrop against which this dialogue played out, in July it was lifting the lockdown, and in August it was reopening schools and restarting the sports and other activities. However, without regard to the underlying background information, pro- and anti-mask sentiment can be detected.

We use the Random Forest model to determine the importance of scores of the various types of features. The relative scores are summarized in Table 3. The table indicates that the bulk of the contribution, around 82%, which includes TF-IDF plus word embeddings plus POS tags, comes from the text of the tweets. Social features which determine the reach of the tweet and the popularity and level of activity of the tweeters contribute about 10%. Sentiment scores have very little contribution, around 3%. This could be because we found that the sentiment scores were not sufficiently different

Table 2: Performance of ML models

Model	Accuracy	Precision	Recall	F1-Score
RF	95.64	0.9359	0.9913	0.9628
LSTM	93.66	0.9305	0.9640	0.9458
SVM	89.81	0.9056	0.9224	0.9139
GB	95.71	0.9647	0.9441	0.9780
MLP	94.46	0.9382	0.9672	0.9525

between the pro- and anti-mask tweets, with anger and aggressiveness being the most dominant emotion in both, as illustrated in the two examples below:

Table 3: Importance scores for feature types

Feature Type	Importance Score
TF-IDF	0.4666
Embeddings	0.2598
Social Features	0.1398
POS Tags	0.1036
Sentiment	0.0310
Auxiliary	0.000

If masks are so effective then why did the mandatory rule not apply to shop staff? So COVID will kill me, the customer but not the shop worker? The insanity is breathtaking in its stupidity, incomprehension and indefensibility. #NoMasks **(P)**

#WearADamnMask with over 140k #COVID deaths Passengers cheer as 'Karen' is kicked off flight for refusing to wear mask https://t.co/wRd0iaJ1WF via @nypost. **(P)**

The first tweet is anti-mask and expresses anger towards the hypocrisy surrounding the use of masks, and the second tweet is pro-mask expressing anger and cynicism towards those who choose not to wear masks. Because anger was the dominant sentiment in both pro-mask and anti-mask tweets, sentiment scores may not have been effective in the classification. There is a good degree of sarcasm and irony expressed on both sides as well, the first tweet above notes the hypocrisy of imposing a mask mandate on the customers but not the staff. Detecting of emotions [38] could shed light on which emotions are expressed in the data set.

7 Related Research

Social media conversations are spontaneous and unfiltered, and hence, can offer genuine insights into people's opinions on a variety of offline events, topics, and policies. Because the donning of masks is relatively recent and controversial, efforts that have analyzed social media conversations around masks are gaining prominence. Ahmed *et. al.* [4] build a network of users from mask-related conversations on Twitter, and analyze this network using centrality measures to find the most influential users. Even when face masks were recommended, there remained widespread confusion about who should be wearing a mask – whether healthy people should be wearing it, and for whose protection [47]. A geographical analysis of anti- mask activity based on Twitter content has been conducted [48]. Lang *et al.* [29] examine the uses of pro- and anti-mask hashtags, and find that an increase in the volume of these hashtags is correlated with an increase in the cases. The further classify pro-mask hashtags into those urging the use of masks and issuance of mask mandates, and assertions of the efficacy, altruistic value, and positive masculinity associated with mask wearing. Anti-mask hashtags are further sorted into rejection of mask wearing, insults to mask wearers, and disinformation that asserts the negative effects of mask wearing. Al-Ramahi *et. al.* [5] also find the volume of anti-mask hashtags correlated with the volume of Covid cases. They identify three themes in anti-mask tweets, namely, constitutional rights, conspiracy theories, and fake news, pandemic, and data. Pascual-Ferra [41] analyze the toxic speech in the mask debate, and find that the tweets that included anti-mask hashtags were more likely than tweets with pro-mask hashtags to contain toxic language. He *et. al.* [22] understand the common attitudes and reasons for resistance towards the wearing of masks, and corroborate some of the reasons for the opposition as found by Al-Ramahi [5]. They use supervised machine learning to separate tweets that are relevant to the wearing of masks, and further filter those that do not express any personal opinions.

Our work can be distinguished from these contemporary efforts in that it tries to automatically separate anti-mask tweets from pro-mask tweets using machine learning. As is shown in these efforts, anti-mask tweets pedal misinformation and incite anger and hatred against government restrictions designed to curb the spread of the virus. Misinformation may discourage the adoption of this common-sense public health measure, whereas, provoking people may ultimately lead to violence and bloodshed in the physical world. Identifying anti-mask tweets may prevent the damage that they may cause, but due to the sheer volume of content shared on social media platforms, it is impossible to do so manually. Our approach is therefore valuable to automatically separate and tag such anti-mask tweets.

Overall, social media feeds have been mined to understand the public outlook on hot button medical and other health-related issues, the most notable topic that is related to masks is vaccines. The issue of masks and vaccines are inextricably linked together in the Covid world, especially, because it is believed that there is a significant overlap between anti-vaxxers and anti-maskers. Therefore, we also review the work on identifying anti-vaxx dialogue on social media as closely related to this work. Research at the intersection of vaccines and social media use both unsupervised and supervised learning for harnessing informal opinions, and also classify these perceptions into support or opposition. Some works consider specific vaccines such as Dengavaxia [2], MMR [3], Flu [8], and Zika [18], while some mine general attitudes about vaccines (anti-vaxx opinions, adversity and safety signals, fake news and rumors and interference from trolls) without reference to any particular vaccine [21, 37, 6, 35, 17, 53], and recently the Covid-19 vaccine [46, 42].

8 Conclusions and Future Research

This paper analyzes the debate around masks on Twitter using the tweets collected during the months of July and August 2020, just as many states were beginning to lift their stay home, stay safe orders, and plans were being conceived to reopen schools. Our initial analysis mines the opinions of anti-mask and pro-mask groups and compares their social features. A classification framework is then built which can differentiate between the two groups of tweets with an accuracy over 90%. Our research reveals that concerns of anti-maskers are mostly centered around politics and ideology, rather than on pragmatic issues of convenience and health. The anti-mask group is small, close-knit and supportive of each other's opinions, and hence, may be surprisingly effective at spreading the anti-mask hysteria. The benefits for public health, advocated by pro-maskers is unlikely to convince the politically motivated anti-maskers to change their views and habits. The classification framework, by the virtue of separating anti-mask tweets from pro-mask ones accurately can label tweets that sow such incorrect information about masks. Such labeling can warn other users that the views promoted by these tweets are not mainstream, and detrimental to public health.

Longitudinal analysis of the mask dialogue, with data collected at several other points during the pandemic, especially after President Trump was hospitalized due to Covid-19 is a topic of the future. A detailed topic modeling [20] framework to discover both the pro- and anti-mask themes, similar to pro-vaxx and anti-vaxx themes is also underway. Finally, collecting data from other social media platforms such as Facebook, and incorporating it in the analysis is also ongoing.

References

[1] "sklearn.preprocessing.MinMaxScaler,".https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html, 2007.

[2] A. B. C. Abrigo and M. R. J. E. Estuar, "A Comparative Analysis of N-Gram Deep Neural Network Approach to Classifying Human Perception on Dengavaxia," *Proc. of 2nd Intl. Conf. on Information and Computer Technologies*, pp. 46-51, 2019.

[3] A. Addawood, "Usage of Scientific References in MMR Vaccination Debates on Twitter," *Proc. of IEEE/ACM Intl. Conf. on Advances in Social Network Analysis and Mining (ASONAM)*, pp. 971-979, 2018.

[4] W. Ahmed, J. Vidal-Alaball, F. L. Segui, and P. A. Moreno-Sanchez, "A Social Network Analysis of Tweets Related to Masks during the Covid-19 Pandemic," *Intl. Journal on Environ Res Public Health*, 17(21):8235, doi: 10.3390/ijerph17218235, November 2020.

[5] M. Al-Ramahi, A. Elnoshokaty, O. El-Gayae, T. Nasralah, and A. Wahbeh, "Public Discourse Against Masks in the COVID-19 Era: Infodemiology Study of Twitter Data," *MIR Public Health Surveillance*, 7(4):e26780, doi: 10.2196/26780, April 2021.

M. Asghari, D. Sierra-Sosa, and A. Elmagharby, "Trends on Health in Social Media: Analysis using Twitter Topic Modeling," *Proc. of Intl. Symposium on Signal Processing and Information Technology*, pp. 558-563, December 2018.

[6] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and Gäel Varoquaux, "API Design for Machine Learning Software: Experiences from the Scikit-Learn Project," *Proc. of ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108-122, 2013.

[7] A. Celesti, A. Galletta, F. Celesti, M. Fazio, and M. Villari, "Using Machine Learning to Study Flu Vaccines Opinions of Twitter Users," *Proc. of Fourth Intl. Workshop on ICT Solutions for Health*, pp. 1103-1106, 2019.

[8] K. K. Cheng, T. H. Lam, and C. C. Leung, "Wearing Face Masks in the Community during the Covid-19 Pandemic: Altruism and Solidarity," *Lancet*, April 2020.

[9] Francois Chollet, Keras blog, (Keras is a deep learning library), 2015.

[10] W. H. Delashmit and M. T. Manry, "Recent Developments in Multilayer Perceptron Neural Networks," *Proc. of the 7th Annual Memphis Area Engineering and Science Conference*, 2005.

[11] C. Dotto and L. Morrish. "Coronavirus: How Pro-Mask Posts Boost the Anti-Mask Movement," https://firstdraftnews.org/latest/coronavirus-how-pro-mask-posts-boost-the-anti-mask-movement/, August 2020. Accessed: 2021-01-21.

[12] Centers for Disease Control and Prevention, "CDC Calls on Americans to Wear Masks to Prevent Covid-19 Spread," https://www.cdc.gov/media/releases/2020/p0714-americans-to-wear-masks.html, July 2020. Accessed: 2021-01-21.

[13] E. Forslid and N. Wiken, *Automatic Irony- and Sarcasm Detection in Social Media*, Master's Thesis, Uppsala Univeritet, August 2015.

[14] S. Frenkel, B. Decker, and D. Alba, "How the 'Plandemic' Movie and its False-hoods Spread Widely Online," https://www.nytimes.com/2020/05/20/technology/plandemic-movie-youtube-facebook-coronavirus.html, May 2020. Accessed: 2021-01-21.

[15] J. H. Friedman, "Stochastic Gradient Boosting," *Computational Statistics & Data Analysis*, 38(4):367-378, 2002.

[16] B. Gambini, "Study: Fake Russian Twitter Accounts Politicized Discourse About Vaccines," http://www.buffalo.edu/news/releases/2020/03/024.html, March 2020. Accessed: 2020-06-28.

[17] A. Ghenai and Y. Mejova, "Catching Zika Fever: Application of Crowdsourcing and Machine Learning for Tracking Health Misinformation on Twitter," *Proc. of Intl. Conf. on Healthcare Informatics*, 2017.

[18] M. Godoy, "Yes, Wearing Masks Helps, Here's Why," https://www.npr.org/sections/health-shots/2020/06/21/880832213/, June 2020. Accessed: 2021-01-21.

[19] S. Gokhale, "Monitoring the Perception of Covid-19 Vaccine using Topic Models," *Proc. of the 13th IEEE Intl. Conf. on Social Computing and Networking*, Virtual,

Due to Covid-19, December 2020.

[20] S. K. Habibabadi and P. D. Haghighi, "Topic Modelling for Identification of Vaccine Reactions in Twitter," *Proc. of ACSW*, Sydney, Australia, January 2019.

[21] L. He, C. He, T. L. Reynolds, Q. Bai, Y. Huang, C. Li, K. Zheng, and Y. Chen, "Why Do People Oppose Mask Wearing? A Comprehensive Analysis of U.S. Tweets During COVID-19 Pandemic," *Journal of the American Medical Informatics Association*, 28(7):1564-1573, July 2021.

[22] Healthline.com, "As Many As 130,000 Lives Could Be Saved the Next 3 Months If Everyone Wore a Mask," https://www.healthline.com/health-news/as-many-as-130000-lives-could-be-saved-the-next-3-months-if-every one-wore-a-mask, November 2020. Accessed: 2021-01-21.

[23] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, 9(8):1735-1780, November 1997.

[24] C. J. Hutto and C. Gilbert, "Vader: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text," *Proc. of Intl. AAAI Conference on Weblogs and Social Media*, 2014.

[25] E. Israfilova, A. Arslan, N. Yildrim, and T. Kaya, "Influencer Identification System Design Using Machine Learning Techniques," C. Kahraman, S. Cebi, S. Cevik Onar, B. Oztaysi, A. C. Tolga, and I. U. Sari, Editors, *Intelligent and Fuzzy Techniques in Big Data Analytics and Decision Making*. Springer, January 2021.

[26] D. Karani, "Introduction to Word Embedding and Word2Vec," https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec, September 2018.

[27] M. Kearney, "Collecting Twitter Data," https://cran.r-project.org/web/packages/rtweet/rtweet.pdf, January 2020.

[28] J. Lang, W. W. Erickson, and Z. Jing-Schmidt, "#maskon! #maskoff! Digital Polarization of Mask-Wearing in the United States During COVID-19," *Public Library of Science ONE*, 16(4):e0250817, doi: 10.1371/journal.pone.0250817. eCollection, April 2021.

[29] A. Liaw and M. Wiener, "Classification and Regression by Random Forest," *R news*, 2(3):18-22, 2002.

[30] E. Loper and S. Bird, "NLTK: The Natural Language Toolkit," *CoRR*, cs.CL/0205028, 2002.

[31] S. Loria, P. Keen, M. Honnibal, R. Yankovsky, D. Karesh, E. Dempsey, W. Childs, J. Schnurr, A. Qalieh, L. Ragnarsson, J. Coe, A. L. Calvo, N. Kulshrestha, J. Eslava, *et al*., "Textblob: Simplified Text Processing," *Secondary TextBlob: Simplified Text Processing*, 3, 2014.

[32] V. Luckerson, "How Mask Mandates were Beaten Down in Rural Oklahoma," https://www.newyorker.com/news/dispatch/how-mask-mandates-were-beaten-down-in-rural-Oklahoma, May 2020. Accessed: 2021-01-21.

[33] S. MacAvaney, H. R. Yao, E. Yang, K. Russell, N. Goharian, and O. Frieder, "Hate Speech Detection: Challenges and Solutions," *PLOS ONE*, August 2019.

[34] R. Mahajan, W. Romine, M. Miller, and T. Banerjee, "Analyzing Public Outlook towards Vaccination using Twitter," *Proc. of Intl. Conf. on Big Data*, pages 2763-2772, December 2019.

[35] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and Their Compositionality," *Neural Information Processing Systems*, 2013.

[36] T. Mitra, S. Counts, and J. W. Pennebaker, "Understanding Anti-Vaccination Attitudes in Social Media," *Proc. of the Tenth Intl. AAAI Conf. on Web and Social Media*, pp. 269–278, 2016.

[37] A. Mondal and S. Gokhale, "Mining Emotions on Plutchik's Wheel," *Proc. of Intl. Workshop on Sentiment Analysis and Mining of Social Networks*, Virtual, Due to Covid-19, December 2020.

[38] S. Morrison, "Biden Wants a National Mask Mandate, Can He Do That?" https://www.vox.com/2020/8/21/21395570/biden-mask-mandate-for-all-national-states, August 2020. Accessed: 2021-01-21.

[39] C. Nicholson, "A Beginner's Guide to Word2Vec and Neural Word Embeddings," https://pathmind.com/wiki/word2vec, 2019.

[40] P. Pascual-Ferra, N. Alperstein, D. J. Barnett, and R. N. Rimal, "Toxicity and Verbal Agression on Social Media: Polarized Discourse on Wearing Face Masks During the COVID-19 Pandemic," *Big Data & Society*, pp. 1-17, January-July 2021.

[41] N. Paul and S. Gokhale, "Analysis and Classification of Vaccine Dialogue in the Coronavirus Era," *Proc. of IEEE Big Data Workshop on Smart and Connected Communities*, Virtual, Due to Covid-19, December 2020.

[42] S. Perry, "Social Media Users are More Likely to Believe False Information About Covid-19 and to Ignore Public Health Advice, Study Suggests," https://www.minnpost.com/second-opinion/2020/07/social-media-users-are-more-likely-to-believe-false-infor-mation-about-covid-19-and-to-ignore-public-health-advice-study-suggests/, July 2020. Accessed: 2021-01-21.

[43] The Associated Press, "Can I Stop Wearing a Mask After Getting a Covid-19 Vaccine," https://apnews.com/article/mask-wearing-after-coronavirus-vaccine-f69b720444bd08565bf1f3a42e4a24ef, December 2020. Accessed: 2021-01-21.

[44] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," *Proc. of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pp. 45-50, May 2010.

[45] R. F. Sear, N. Velasquez, R. Leahy, N. Johnson Restrepo, S. E. Oud, N. Gabriel, Y. Lupu, and N. F. Johnson, "Quantifying Covid-19 Content in the Online Health Opinion War using Machine Learning," *IEEE Access*, 8:91886-91893, May 2020.

[46] Y. Shen, "Covid-19 Outbreak: Tweet Analysis on Face Masks," https://towardsdatascience.com/covid-19-out break-tweet-analysis-on-face-masks-27ef5db199dd, March 2020. Accessed: 2021-01-21.

[47] Knau Staff, "Twitter Analysis Shows Arizona is #1 in Anti-Face Mask Activity," https://www.knau.org/post/twitter-analysis-shows-arizona-1-anti-face-mask-activity, July 2020. Accessed: 2021-01-21.

[48] J. Suykens and J. Vandewalle, "Least Squares Support Vector Machine Classifiers," *Neural Processing Letters*, 9(3):293-300, 1999.

[49] A. Taylor, "How the Split Over Masks Sums Up America's Chaotic Coronavirus Response," https://www.washingtonpost.com/world/2020/06/25/face-masks-america-divided/, June 2020. Accessed: 2021-01-21.

[50] A. Whalen, "Narcissists and Psychopaths are More Likely to Refuse to Wear Masks, Says New Research," https://www.newsweek.com/narcissists-psychopaths-face-mask-requirement-mandate-social-distancing-covid-19-coronavirus-1519732, July 2020. Accessed: 2021-01-21.

[51] R. Zafarani, M. A. Abbasi, and H. Liu, *Social Media Mining: An Introduction*, Cambridge University Press, New York, NY, USA, 2014.

[52] W. Zhao, "Misinformation Correction Across Social Media Platforms," *Proc. of Intl. Conf. on Computational Science and Computational Intelligence*, pp. 1371-1376, 2019.

Luca Cerbin (photo not available) is a senior at Swarthmore college. Luca is majoring in Computer Science with a double minor in Applied Math and English. He works in a biology lab on image processing and machine learning on planarians. His research interests are in machine learning and understanding the limitations associated with it. This work was done when he was a visiting student at the Univ. of Connecticut.

Jason DeJesus (photo not available) is a graduate student in the Electrical and Computer Engineering Department at the University of Connecticut. He works as a Graduate Assistant in a cybersecurity lab, with a focus on hardware security.

Julia Warnken (photo not available) is pursuing her Masters of Engineering in Computer Science and Engineering at the University of Connecticut. Her research interests include machine learning, data mining, and genetic or biomedical applications.

Swapna S. Gokhale (photo not available) is an Associate Professor of Computer Science and Engineering at the University of Connecticut. She received her PhD in Electrical and Computer Engineering from Duke University in 1998. Her research interests lie in the areas of performance and dependability analysis of computer systems, software engineering education, and mining and analysis of social media, epidemiological and sustainability data. She has published over 150 conference and journal papers on these topics. She is a recipient of a NSF CAREER award and a Senior Member of the IEEE.

Journal Submission

The International Journal of Computers and Their Applications is published four times a year with the purpose of providing a forum for state-of-the-art developments and research in the theory and design of computers, as well as current innovative activities in the applications of computers. In contrast to other journals, this journal focuses on emerging computer technologies with emphasis on the applicability to real world problems. Current areas of particular interest include, but are not limited to: architecture, networks, intelligent systems, parallel and distributed computing, software and information engineering, and computer applications (e.g., engineering, medicine, business, education, etc.). All papers are subject to peer review before selection.

A. Procedure for Submission of a Technical Paper for Consideration

1. Email your manuscript to the Editor-in-Chief, Dr. Wenying Feng. Email: wfeng@trentu.ca.
2. Illustrations should be high quality (originals unnecessary).
3. Enclose a separate page (or include in the email message) the preferred author and address for correspondence. Also, please include email, telephone, and fax information should further contact be needed.
4. **Note**: Papers shorter than 10 pages long will be returned.

B. Manuscript Style:

1. **WORD DOCUMENT**: The text should be **double-spaced** (12 point or larger), **single column** and **single-sided** on 8.5 X 11 inch pages. Or it can be single spaced double column.

   LaTex DOCUMENT: The text is to be a double column (10 point font) in pdf format.
2. An informative abstract of 100-250 words should be provided.
3. At least 5 keywords following the abstract describing the paper topics.
4. References (alphabetized by first author) should appear at the end of the paper, as follows: author(s), first initials followed by last name, title in quotation marks, periodical, volume, inclusive page numbers, month and year.
5. The figures are to be integrated in the text after referenced in the text.

C. Submission of Accepted Manuscripts

1. The final complete paper (with abstract, figures, tables, and keywords) satisfying Section B above in **MS Word format** should be submitted to the Editor-in-Chief. If one wished to use LaTex, please see the corresponding LaTex template.
2. The submission may be on a CD/DVD or as an email attachment(s). **The following electronic files should be included:**
   - Paper text (required).
   - Bios (required for each author).
   - Author Photos are to be integrated into the text.
   - Figures, Tables, and Illustrations. These should be integrated into the paper text file.
3. Reminder: The authors photos and short bios should be integrated into the text at the end of the paper. All figures, tables, and illustrations should be integrated into the text after being mentioned in the text.
4. The final paper should be submitted in (a) pdf AND (b) either Word or LaTex. For those authors using LaTex, please follow the guidelines and template.
5. Authors are asked to sign an ISCA copyright form (http://www.isca-hq.org/j-copyright.htm), indicating that they are transferring the copyright to ISCA or declaring the work to be government-sponsored work in the public domain. Also, letters of permission for inclusion of non-original materials are required.

Publication Charges

After a manuscript has been accepted for publication, the contact author will be invoiced a publication charge of **$500.00 USD** to cover part of the cost of publication. For ISCA members, publication charges are **$400.00 USD** publication charges are required.

Revised 2020

ISCA INTERNATIONAL JOURNAL OF COMPUTERS AND THEIR APPLICATIONS Vol. 28 No. 3, Sept. 2021