Computer Networks 64 (2014) 89–111

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

3D P2P overlay over MANETs

S.A. Abid [a], Mazliza Othman [a], Nadir Shah [b,*]

[a] Faculty of Computer Science and Information Technology, University of Malaya, Malaysia
[b] Department of Computer Science, COMSATS Institute of Information Technology, Pakistan

ARTICLE INFO

Article history:
Received 1 October 2013
Received in revised form 3 February 2014
Accepted 4 February 2014
Available online 12 February 2014

Keywords:
P2P overlays
Distributed Hash Tables
MANETs
Distributed systems
3D
Content sharing

ABSTRACT

We study the challenging problems of *the mismatch between the overlay and the physical network* and *the resilience of the overlay structure* in peer-to-peer (P2P) protocols over a mobile ad hoc network (MANET). Existing P2P protocols have used inflexible overlay structures to arrange peers and do not consider intra-neighbor relationships of peers when assigning logical identifiers. The intra-neighbor relationships of peers are crucial to exactly interpret the physical proximity of peers in an overlay and to avoid the mismatch between the overlay and the physical network that causes extensive routing overhead, larger average file discovery delay, increased false-negative ratio, and high average path-stretch. In this paper, we present a novel P2P overlay over MANETs that exploits a 3-dimensional overlay and 3D space that takes into account the physical intra-neighbor relationship of a peer and exploits a 3D-overlay to interpret that relationship. In the proposed protocol, each peer runs a distributed algorithm that exploits a 3D-overlay to calculate a consecutive logical identifier to a peer. Moreover, the protocol utilizes the 3D-overlay to maintain multi-paths to a destination peer that provides resilience against a node/link failure. Simulation results show that the proposed 3D-overlay outperforms the existing P2P overlay protocol in terms of routing overhead, average file-discovery delay, false-negative ratio, and average path-stretch.

1. Introduction

Peer-to-peer (P2P) computing refers to technology that enables two or more peers to collaborate spontaneously in an overlay network by using appropriate information and communication systems without the necessity for central coordination [28]. A P2P overlay network is a robust, distributed and fault-tolerant network architecture for sharing resources like CPU, memory and files. Mobile and wireless technology have experienced great progress in recent years. Today's cell phones, PDAs and other handheld devices have larger memory, higher processing capability and richer functionalities. The user can store more audio, video, text and image data with handheld devices. These devices are also equipped with low radio technology, like Bluetooth and Wi-Fi. By means of these low radio technologies, the devices can communicate with each other without using communication infrastructure (e.g., cellular infrastructure) and form a self-organized mobile ad hoc network (MANET) that can be used to share resources.

Each node in a MANET works as both a host (for sending/receiving the data) and a router (for maintaining the routing information about other nodes). A MANET is deployed in the places, where infrastructure is either not available or too expensive. The P2P overlay network approaches were initially proposed to work at the application layer for P2P overlay over the Internet [1–5]. Later on, due to the advances in wireless and mobile technology, P2P overlays were deployed over MANETs [6–14,18–20,22,38–42].

[*] Corresponding author. Tel.: +92 3329854715.
E-mail addresses: shahbaz.akhtar.abid@gmail.com (S.A. Abid), mazliza@um.edu.my (M. Othman), nadirshah82@gmail.com (N. Shah).

http://dx.doi.org/10.1016/j.comnet.2014.02.006
1389-1286/© 2014 Elsevier B.V. All rights reserved.

The P2P overlay is dynamic, where peers join/leave for content sharing. Such a peer-to-peer communication paradigm is very important in a mobile ad hoc network as the centralized server might not be available or located in the MANET. Therefore, P2P is an interesting alternative for decentralizing services or making its own local resources available in the MANET to serve local user communities [28]. The P2P overlay network provides a lookup service (i.e., searching for resources) handling flat identifiers with an ordinary query response semantic [28].

There are various P2P application scenarios, where a P2P overlay over MANETs can be used to share resources. For example, the users equipped with cell phones, PDAs or other handheld devices can form a P2P network for sharing audio/video clips, pictures, files and other information. Possible resource sharing application scenarios can be found at airport lounges, music concerts, bus stops, railway stations, university campus, cafeterias, etc. where people can utilize their handheld devices to share resources (files, notes, images, etc.) by utilizing ad hoc communication that is free of charge, infrastructure less, and decentralized. Such networks can also be made useful and deployed rapidly in case of political disputes like, London and Egypt strikes in 2011 [29–32].There are few applications [33,34] that can be used for these purposes.

In this paper, we consider a scenario of a structured P2P overlay over a MANET, where not all nodes share and access files, i.e., some nodes are peers and others are non-peers. A peer node is a member of the P2P overlay network that shares and/or accesses a resource (e.g., a video file) while a non-peer node does not. The P2P overlay network works at the application layer and assumes that routing at the network layer is provided by an underlying routing protocol.

Recently, several schemes have been proposed for structured P2P networks over MANET [6–14,18–20,22,38–42]. Constructing a structured P2P overlay over MANET gives rise to some challenges that are imperative to address in order to make the P2P protocol robust and scalable. We have identified two correlated issues that must be considered when designing a structured P2P overlay over a MANET, namely *the mismatch between the overlay and the physical network*, and *the resilience of the overlay structure*. These issues need immediate attention and affect the performance of the structured P2P overlay over MANET in terms of path-stretch ratio (the ratio of the path length available in the P2P network to the length of the shortest path in the physical network between two peers), long routes, and file discovery delay (the average time elapsed from the moment when the file-look-up query is issued to the moment when the first reply is received).

In this paper, we propose a novel protocol for constructing a structured P2P overlay over a MANET that exploits a 3-dimensional overlay structure (3D-Overlay), named the 3-dimensional overlay protocol (3DO), to arrange peers in an overlay that interprets the physical relationship of peers in a 3-dimensional logical space (3D-LID space). The 3D-LID space gives a peer the liberty to exactly interpret the physical relationship of peers in the 3D-Overlay.

3DO differs from the prior work in [14] in the following ways: (i) The proposed protocol in [14] suffers from the mismatch problem when a peer *P* (except the root-peer) has at least one directly connected neighboring peer, say P1, such that P1 is closer to the root-peer than *P*, and *P* stores the portion of the identifier space higher than P1's identifier space. The peer *P* might not be a neighbor in the logical structure to all of its physical adjacent peers; (ii) the LID space distribution among peers in [14] is inconsistent and the peers are not placed in a proper structure (like ring, chord, and two-dimensional/multidimensional spaces etc.) for the overlay network. The physical intra-neighbor relationship of a peer with respect to its neighboring peers are not interpreted logically in [14]; and (iii) a parent peer retrieves the LID space portion of its child peer in case the child peer moves to some other place, generating extensive traffic overhead.

Though the 3D structure has been used for routing at the network layer in [26], to the best of our knowledge, this is the first attempt to use a 3D structure to arrange peers in the overlay network at the application layer and to assign three dimensional LIDs to peers in the P2P overlay over MANET. None of the existing P2P protocols over MANET [1–14,18–20,22,24,38–42] has used a 3D structure to map the physical proximity of a peer in an overlay. 3DO is designed to work at the application layer for content sharing (e.g., files) in P2P over MANETs and it assumes an underlying routing protocol, i.e., OLSR to perform routing of packets at the network layer. In [26], the proposed protocol is a DHT-based routing protocol for MANETs and is designed to work at the network layer for routing purposes. There is no content sharing at the application layer in [26].

In 3DO, each peer considers the intra-neighbor peer relationships, like distinct neighboring peers, adjacent/nonadjacent neighboring peers, and common neighboring peers while computing its LID. This helps to exactly map the physical proximity of a peer in the 3D-Overlay. These relationships are crucial when calculating the relative position of a peer in order to optimally address the mismatch between the overlay and physical network (see Section 3). None of the existing P2P protocols over MANETs has considered all of these relationships. The connecting order of 3D structure is flexible to reflect the intra-neighboring peer relationships in an overlay, which avoids the mismatch between the overlay and physical networks. For instance, *P* represents the set of peers in the network and for all *p1* belonging to *P* there exists T_{p1} as a set of 1-hop logical neighboring peers of peer *p1*. If peers *p2*, *p3* belonging to T_{p1} and both are not neighbors in the overlay network, then it means that peers *p2* and *p3* lie in different dimensions from peer *p1*. Therefore, both peers *p2* and *p3* obtain their LIDs corresponding to two different dimensions of the local 3D-Overlay of peer *p1*. In this way, 3DO uses different sign dimensions of a 3D structure to compute LIDs of distinct neighboring peers of a peer. The 3D structure naturally helps to utilize a peer's local neighbors when forwarding a query and provides resilience against peer failure/mobility by providing an alternative route to a peer. In 3DO, we propose the use of Shepard's interpolation method [23] in 3D to compute the LID of a peer relative to its 1-hop logical neighboring peers.

In summary, 3DO is designed to achieve the following objectives: (i) To avoid long routes, the neighboring peers in an overlay should also be adjacent in the physical network. This would reduce control traffic and lookup latency in the network, as discussed in [13,14]; (ii) to avoid redundant traffic, a peer in an overlay should be logically close to all its physically adjacent peers. This allows a lookup query from the peer to be always forwarded closer to the destination peer in both the physical network and the overlay network. The lookup query, therefore, observes a short route in the physical network that speeds up the lookup process and reduces the routing traffic; (iii) the file should be retrieved from the source peer (the peer with the actual data item) by the requesting peer via the shortest physical route in the network; (iv) the protocol should adapt to node mobility. When the connectivity among the peers changes in the physical network due to node mobility, the overlay network should be updated accordingly; (v) the system should be adaptive to network churn. When a peer joins or leaves the P2P network, the P2P overlay should update itself accordingly; and (vi) The system should be distributed in nature in the following sense. *First*, a system operation should be locally carried out so that the operation has the minimum global effect on the network. *Second*, the system operation should require local information rather than global information about the network.

The rest of this paper is organized as follows: Section 2 explains why structured P2P overlays are appropriate in MANETs. In Section 3, we outline the problem statement by describing the mismatch between the overlay and the physical network and how the shape of the logical structure amplifies the mismatch problem. Section 4 discusses the related work. The motivation behind the proposed 3D-structure and the detail of the proposed 3D Overlay (3DO) over MANETs are explained in Section 5. Modeling and verification of 3DO are performed using formal methods in Section 6. Simulation results are presented in Section 7. Finally, Section 8 concludes the paper.

2. Why structured p2p overlays in MANETs

In this section, we first discuss the classification of approaches that are designed for content sharing in MANETs and then we elaborate: (i) the reason structured P2P overlays are appropriate for MANETs and (ii) the advantages and disadvantages of using DHTs for structured overlays in MANETs. Approaches for content sharing in MANETs can be categorized into: (i) non overlay-based and (ii) overlay-based.

2.1. Non-overlay-based approaches

These approaches use the underlying routing protocol for content searching instead of node-ID searching, i.e., a requesting node broadcasts a lookup query for content (say File1). Upon receiving the lookup query, a receiving node responds by sending a reply message to the requesting node if the receiving node possesses the content (i.e., File1 in this case). Otherwise, the receiving node broadcasts the lookup query to its neighbors in the network. These approaches do not maintain the overlay network at the application layer over the physical network and the functionality of P2P systems is completely implemented/integrated at the routing layer in MANETs. Examples of these approaches can be found [10,12,37,47–50]. However, these approaches produce heavy routing traffic if the network has a low peer ratio. The reason is that every node (whether it is a peer or a non-peer) forwards the lookup query. The peer ratio is the ratio of number of peers to the total number of nodes (i.e., the sum of the number of peers and non-peers) in the network as explained in [22]. Moreover, to support multiple P2P applications at the routing layer would require identification for each of the P2P applications. Therefore, managing multiple P2P applications simultaneously in a MANET would be more difficult.

2.2. Overlay-based approaches

In this category, an overlay network is built over the physical network by connecting only to other peers. These approaches perform better in the scenarios of P2P overlay over MANETs, where not all the nodes are to share and access files, i.e., some nodes are peers and others are non-peers. In this way P2P traffic is only limited to peers [22]. These approaches can be further classified into structured [6–14,18–20,22,24,38–42] and unstructured [35–37] overlays.

2.2.1. Unstructured P2P overlays

2.2.1.1. Unstructured. P2P overlays do not impose a rigid relation between the overlay topology and the places where resources or their indices are stored. In an unstructured P2P network, a peer establishes a connection with a certain number of other peers that are chosen randomly, thus a random overlay network is established among the peers. Each peer in the unstructured P2P network holds its own shared content. Therefore, unstructured P2P networks use flooding at the overlay layer for resource discovery in the P2P network. Examples of these approaches are [51,52]. These are simple and easy to implement, but the major drawback of unstructured P2P overlays is its scalability. A requesting peer in these protocols broadcasts a lookup query for content in the overlay network, which takes a long time and consumes network resources extensively by producing redundant traffic because most of the time there is no relation between the name of resources and their locations [22,28,36]. This redundant traffic increases the routing overhead in the file discovery phase, which increases the probability of query loss due to packet collision in MANETs. Another limitation of unstructured P2P overlays over MANETs is: if a requesting peer does not receive a reply for a lookup query, the requesting peer would be unsure whether the lookup query is lost due to packet collision or the requested file does not exist in the network. The rate of redundant traffic increases exponentially with an increase in the peer ratio [51,52], which leads to higher probability of query loss due packet collision in unstructured P2P in MANETs.

2.2.2. Structured P2P overlays

The structured P2P overlays impose a rigid relationship between the overlay topology and the places where resources or their indices are stored. In structured P2P overlays, each peer is assigned a logical ID (LID) from a predefined address space and peers are arranged in the overlay network following a ring, a chord or a multidimensional structure based on the peers' LIDs. Each peer maintains information of its neighboring peers with LIDs that are close to its own LID in the overlay structure. Each peer applies a hash function over its shared files. This generates a key k for a file F from the same address space that is used for the peers' LIDs. The file F with key k is stored on a peer P with LID closer to that of k. Thus, if another peer P2 wants to retrieve F then P2 sends the lookup query toward P by forwarding to one of its 1-hop logical neighbor that has a LID closer to k. This process is repeated on each intermediate peer till the lookup query arrives at P. After receiving the lookup query, P sends the required file to the requesting peer if the file F with key k exists at P. Otherwise, P sends a NULL value in the reply to P2. This ensures that if the requesting peer does not receive a reply, then either the lookup query or the reply to the lookup query has been lost in the network due to packet collision. The structured P2P overlays removes flooding at the overlay network and thus reduces redundant traffic in the network [46], which in turn reduces both energy consumption and packet collisions.

2.2.2.1. Distributed Hash Tables (DHTs) in structured P2P overlays. The lookup services in P2P overlays are often implemented using Distributed Hash Tables (DHTs) [28]. A DHT defines how an overlay is fabricated (i.e., it defines the logical addressing of peers) and how keys are maintained (i.e., lookup procedure) in an overlay. It maps application data/values to keys that are m-bit identifiers drawn from the logical identifier space. A DHT imposes a structure on the overlay network that enables to choose routing table entries satisfying a certain criteria depending on the respective DHTs [28]. This structure allows DHTs to introduce an upper bound of $O(log N)$ on the number of hops, where N is the number of nodes, which means a node needs to coordinate with only a few other nodes in the overlay structure to reach the destination node that removes flooding and reduces overhead [46].

Although DHTs can route lookup queries very efficiently in comparison with unstructured P2P networks, they usually introduce higher maintenance overhead for their routing tables. Support towards high mobility in a DHT-based structured P2P overlay over a MANET is itself a major challenge, which needs immediate attention. However, DHT-based structured overlay approaches outperform unstructured approaches when the number of nodes, the number of objects, or the query rate increases, since they do not introduce flooding in the network [46].

In our study, we have targeted low mobility scenarios like university campuses, concerts, and airport lounges, where people equipped with handheld devices communicate through low radio range (e.g., Wi-Fi) forming a P2P network for sharing resources (files, notes, images, etc.).

3. Problem statement

In the following paragraphs, we discuss the problem that we address in the context of the design criteria listed in Section 1. Table 1 provides definitions of important terms related to P2P overlay network.

The importance of requirement (i) is discussed in [13,14]. In Chord [1] and Pastry [2], a new node computes its LID by applying a hash function to its IP address or the public key. Then, the new node sets up a connection with neighboring peers in the overlay network that are close to its LID according to the overlay routing algorithm. In MANETs, this can prevent a peer from setting up a connection with neighboring peers that are adjacent in the physical network, resulting in a mismatch between the overlay and the physical network; this is called mismatch problem.

The example network in Fig. 1 has a Chord ring overlay

The neighbors in the overlay are not adjacent in the physical network. Suppose, peer P5 initiates a file-lookup query for a key k. If the key k is at P7, then the query is forwarded along the path that consists of a series of links P1–P11, P11–N3, N3–P6, P6–N3, N3–P11, P11–P12, P12–N2, and N2–P7 in the physical network, shown as black arrows in Fig. 1. We see that the query traverses certain links, e.g. P11–N3, N3–P6, more than once. This results in redundant traffic and a larger lookup latency. This scenario shows that redundant traffic would be produced in maintaining the routing information among the peers by periodically exchanging probe messages. Similarly, maintaining the index information of shared files in the network also produces redundant traffic.

To explain the importance of objective (ii) in Section 1, we take an example scenario in Fig. 1 for the approach in [8], where a peer maintains its physically adjacent peers

Table 1
Definitions of important terms related to a P2P overlay network over a MANET.

Anchor Peer (AP)	A peer that holds the keys of files that are stored at some other peer with respect to its logical identifier space portion (LSP). Any peer in an overlay can act as an anchor peer
Logical Identifier (LID)	A unique ID that identifies a peer in an overlay and it describes the relative position of that peer in the overlay network
Logical Identifier Space (LID Space)	A user defined address space from which each peer and resource (e.g., files, etc.) obtains its LID
Overlay network	A logical network that arranges peers according to their LID following some structure, e.g., a ring [1]
LID Space Portion	A subset of the entire LID space that is disjoint from that of other peers
Universal Identifier (UID)	An identifier of a peer that is public, unique, and remains the same throughout the network lifetime. It could be the IP/MAC address of a peer

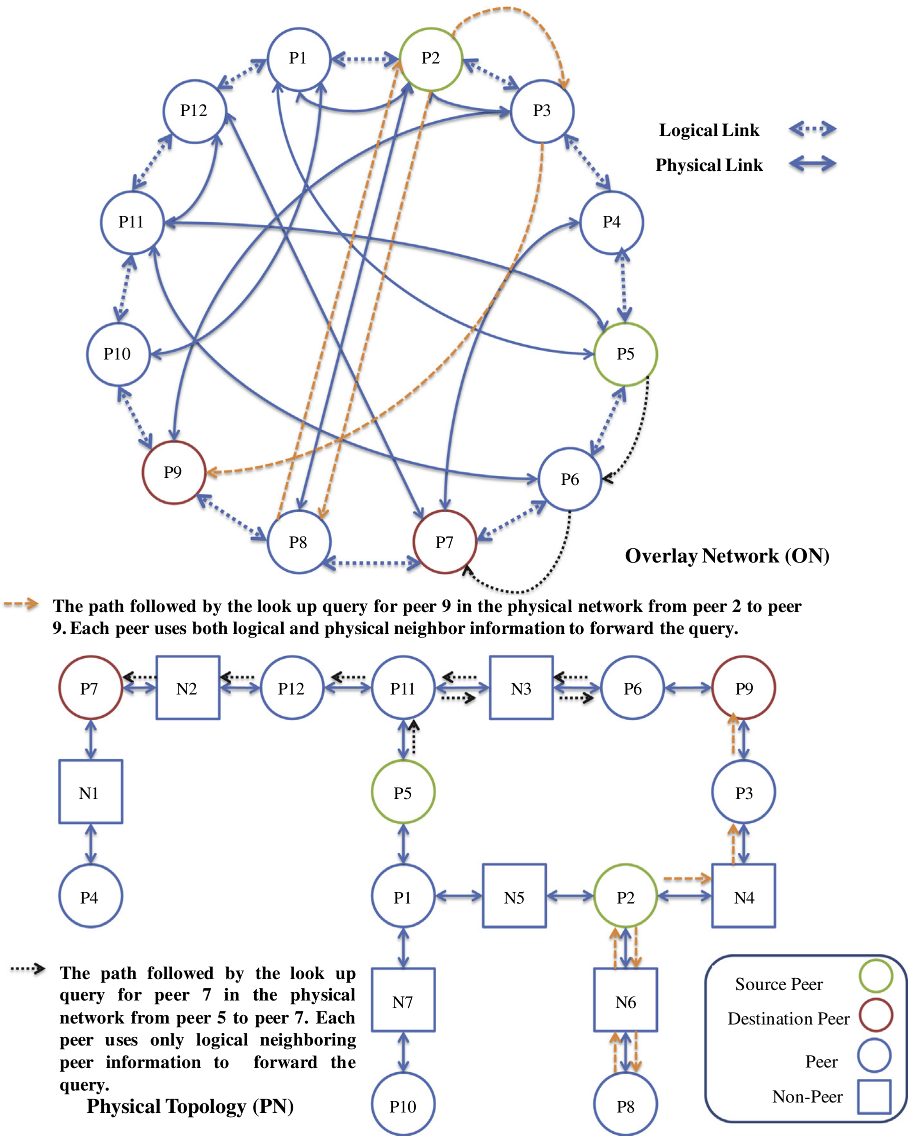

P1
P2
P3
P4
P5
P6
P7
P8
P9
P10
P11
P12
Logical Link
Physical Link
Overlay Network (ON)
The path followed by the look up query for peer 9 in the physical network from peer 2 to peer 9. Each peer uses both logical and physical neighbor information to forward the query.
N1
N2
N3
N4
N5
N6
N7
The path followed by the look up query for peer 7 in the physical network from peer 5 to peer 7. Each peer uses only logical neighboring peer information to forward the query.
Physical Topology (PN)
Source Peer
Destination Peer
Peer
Non-Peer

Fig. 1. Limitation of the DHT model for a MANET.

along with the logical neighboring peers. A lookup query from peer *P* is forwarded to the closest logical neighboring peer among these neighboring peers. But this approach does not avoid a mismatch between the overlay and the physical network that causes redundant traffic and longer lookup latency. Let us say P2 initiates a lookup query for a data item of key $k = 9$ residing at P9. According to the routing scheme in [8], P2 forwards the query to P8 that is logically closer to *k* by utilizing the logical and physical neighboring peer information at P2.

Though the lookup query moves logically closer to P9 in the overlay network, the lookup query moves away from the destination P9 in the physical network, generating two transmissions, i.e., P2–N6, N6–P8 in the physical network as shown by the dashed arrows in Fig. 1. P8 forwards the lookup query to P9 because P9 is the responsible peer for *k* according to the knowledge of P8. This produces five transmissions in the physical network by traversing links P8–N6, N6–P2, P2–N4, N4–P3, and P3–P9 as shown by the dashed arrows in Fig. 1. So, the total number of routing traffic is six for this lookup procedure. However, there exists a shorter route from P2 to P9 via links P2–N4, N4–P3, and P3–P9 that requires only three transmissions. This example shows that even though a peer *P* maintains information of its physically adjacent peers, *P* is not necessarily the neighbor of all its physically adjacent peers in the overlay topology. This leads to increased routing traffic and longer data lookup latency. So, our point is that the peer not only must maintain connections with physically adjacent peers, but also must have a closer logical neighbor relationship with all physically adjacent peers. This ensures that a lookup query is forwarded towards a destination peer via a short route in both the physical network and the overlay network.

The resilience of the overlay network depends on the connecting order of the structure and it plays a key role to avoid the mismatch between the overlay and the physical network. When a peer invokes a joining algorithm, the flexible connection order of the overlay structure leverages a joining algorithm to assign a consecutive LID with respect to the intra-neighbor relationship of a peer. For example, in the tree-based overlay [11], a peer *P* can have two logical neighboring peers, i.e., its child nodes, and *P* can maintain consecutive LIDs with only these two neighboring peers. In case another peer P1 joins through peer *P*, node P1 would obtain a nonconsecutive LID in a different sibling tree to that of *P*. This would result in an overlay network that would not reflect the physical proximity of a peer with respect to its neighboring peers, resulting in longer routes, high path-stretch values when sending queries.

Similarly, ring-based [25] and chord-based [1] overlays are inflexible structures and constrained by their connecting order. In both ring and chord overlay structures, a peer is constrained to maintain consecutive LIDs with only two logical neighboring peers, i.e., its predecessor and successor, which also results in an overlay that does not reflect the physical proximity of a peer with respect to its physically adjacent peers, resulting in a mismatch between the overlay and the physical network.

So far, we have discussed the two correlated problems that play an important role in the performance degradation of P2P overlays over MANETs. In the following section, we present an overview of the existing P2P protocols for MANETs by describing how these approaches fail to avoid the above mentioned problems.

4. Related work

MANET routing protocols can be broadly classified into two categories: reactive and proactive. A reactive routing protocol finds the route in an on-demand fashion, i.e., route discovery is initiated only when there is data to be sent. AODV [16] is one example of a reactive routing protocol. A reactive protocol experiences a larger delivery delay and is inefficient for heavy traffic. Proactive routing protocols periodically update the routing information of all nodes regardless of whether or not the data are to be sent to those nodes. OLSR [15] is one example of a proactive protocol. A proactive routing protocol enables the source node to forward the data packet immediately towards the destination, eliminating the delay for route discovery.

Da Hora et al. [7] improve the performance of Chord [1] over MANET by proposing redundant transmissions of the file-lookup query to avoid frequent loss of query packets due to packet collisions in MANET. The initiating peer sends the lookup query to multiple neighboring peers according to its finger table. This approach has the following limitations. *First*, AODV [16] is used as the underlying routing protocol. Due to the reactive nature of AODV, the requesting peer might not know the shortest route to the source peer when receiving the reply to the query. In this case, the requesting peer can retrieve the file following the route in the P2P overlay that might involve a longer path than the shortest path in the physical network. Thus, the approach does not satisfy requirement (iii) mentioned in Section 1. This approach also results in a large file-retrieval delay. *Second*, this approach does not attempt to construct an overlay that matches the physical network and might perform poorly in a MANET as discussed in [14]. Third, a redundant file-lookup query makes copies in the overlay network that might pass a few common links in the physical network, leading to congestion on the common links that in turn increases the chances of packet collisions in the physical network. Taking the scenario in Fig. 1 as an example, the overlay links from P7 to P12 and from P7 to P11 share the links P7–N2 and N2–P12 in the physical network. Thus, when P1 initiates redundant queries to its neighboring peers, i.e., P2 and P3, the queries pass through the same links P7–N2 and N2–P12 in the physical network, making these common links vulnerable to packet collisions.

Kummer et al. [8] improve Chord over a MANET by maintaining a peer's physically adjacent peers along with its logical neighboring peers. A lookup query from peer *P* is forwarded to the closest logical neighboring peer among *P*'s physically adjacent neighboring peers. This approach also has some limitations. *First*, maintaining the physical adjacent neighboring peers along with the logical neighboring peers generates redundant routing traffic. *Second*, the physically adjacent peers are not necessarily the logical neighboring peers, which leads to a random distribution of the DHT structure rather than a systematic one, resulting in a larger file lookup delay.

Li et al. [9] present an approach called M-Chord that applies Chord to MANET. They classify the nodes into super-nodes and normal nodes. To publish a shared file in M-Chord, the node registers with a super-node according to the LID of the shared file. Similarly, to access a file in M-Chord, the node computes the LID of the file and forwards the query to the corresponding super-node. The super-node in MChord uses a procedure similar to the Chord ring to forward the request to the next super-node until a hit or fail message is returned. This approach cannot avoid mismatch between the overlay and the physical networks.

Sözer et al. [11] use the Distributed Hash Table (DHT) and the topology-based tree structure to store the file index and the routing information, and unify the lookup and routing functionalities. Their approach has four limitations. *First*, two peers cannot communicate if they are separated by some intermediate non-peer(s), resulting in a P2P network partition. This means that, by default, all nodes in the physical network must be peers in the P2P network regardless of whether or not the node wants to be part of the P2P network. *Second*, two peers cannot directly communicate if they do not have a parent-child relationship even if they are within communication range in the physical network. The communication between such peers is only possible by following the route in the tree structure, resulting in a higher lookup delay and more routing traffic. *Third*, routing between the requesting peer and the source peer by following the tree structure often results in a non-shortest path in the physical network [11] that leads to higher path-stretch and does not meet requirement (iii) mentioned in Section 1. *Fourth*, when peer *P* disconnects with one of its child peers, say Q, it regains the responsibilities of Q's LID space. After regaining Q's LID space, *P* broadcasts this event to all peers forcing all peers to restore their shared files that lie in the re-gained LID space. This recovery has a global effect and generates heavy control traffic in the network, thus, violating requirement (vi) in Section 1.

Shin and Arbaugh [13] propose an approach called RIGS (Ring Interval Graph Search) that is suitable for static scenarios in a MANET. Instead of assigning the peer-ID randomly like Chord, RIGS assigns the ID to the peer such that the assignment generates a Ring Interval Graph (RIG). RIGS can build an overlay that closely matches the physical network. However, when the physical topology changes due to node mobility, RIGS cannot maintain the matching of the overlay network and the physical network. RIGS is not a distributed approach because it needs the topology information of the entire network to construct the spanning tree containing all peers in the physical network for RIGS. This is very difficult due to the dynamic nature of both the P2P network and the MANET. RIGS would have to re-construct the spanning tree from scratch when a peer joins or leaves the P2P network or when connections between the peers change due to node mobility. Reconstructing the spanning tree generates extensive redundant traffic in the network. Thus, the RIGS approach cannot meet requirements (iv) and (v) in Section 1. Since building the spanning tree containing all peers in the network requires global topology information, RIGS also cannot satisfy requirement (vi) mentioned in Section 1.

Lee et al. [20], propose an approach called LACMA (location-aided content management architecture) for content-centric MANETs that is designed to address the problem of content placement and replication in highly mobile MANETs. LACMA is based on the concept of named data instead of named hosts and it binds data to a geographic location rather than to nodes. LACMA proactively copies or replicates content to a certain geographical location that decouples the content placement problem in case of node mobility. LACMA maintains multiple layers of the virtual grid with different sizes across the grid, where each layer describes a different content popularity level, i.e., increasing top-down. Popular content is more densely replicated, and it is placed at the bottom of the grid. LACMA assumes that location information is obtained via GPS.

A more recent approach to P2P overlay proposed by Shah et al. [22] introduces a root-peer in the P2P network. In their approach, each peer stores a disjoint portion of the LID space such that the peer closer to the root-peer has a lower portion of the LID space. This approach has two limitations. *First*, due to node mobility, peers frequently exchange information about their LID space and the index of the stored files when their distances to the root-peer change, generating heavy network traffic. *Second*, a peer *P* (except the root-peer) has at least one directly connected neighboring peer, say P1, such that P1 is closer to the root-peer than *P*, and *P* stores the portion of the identifier space higher than P1's identifier space. The peer *P* might not be a neighbor in the DHT structure (logical space) to all of its physical adjacent peers. Thus, this approach does not meet requirement (ii) in Section 1. Moreover, in [22], the LID space distribution among peers is inconsistent, and the peers are not placed in a proper structure (like ring, chord and two-dimensional spaces) for the overlay network. This approach does not logically interpret the physical intra-neighbor relationship of a peer with respect to its neighboring peers. For instance, *P* represents the set of peers in the network and for each *p1* that belongs to *P* there exists T_{p1}, as a set of 1-hop logical neighboring peers of peer *p1*. If peers *p2*, *p3*, *p4* belongs to T_{p1}, and peers *p2*, *p3*, and *p4* are not direct logical neighbors, then it means peers *p2, p3* and *p4* lie in different dimensions of *p1* as shown in the Fig. 2. Therefore, peers *p2*, *p3* and *p4* should obtain their LIDs corresponding to three different dimensions of peer *p1* using a proper structure.

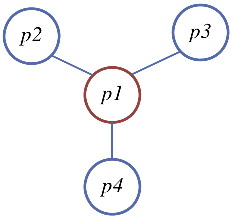


Fig. 2. The intra - neighbor relationship of neighboring peer *p2*, *p3*, *p4*, *p1*.

An optimal solution to the mismatch between the overlay and the physical network would be possible only if the physical relationship of peers is interpreted exactly in an overlay by assigning LIDs to a peers such that the peer's LID is logically close to the LIDs of all its physically close peers and by using a proper structure for the overlay, i.e., peers' LIDs reflect their physical proximity in the overlay.

5. 3D Overlay (3DO)

3DO resorts to an application layer architecture, where each peer has a permanent UID (e.g., IP address/MAC address) that identifies the peer in the physical network, and a transient logical identifier that reflects the peer's relative location with respect to its neighboring peers in the 3D-Overlay. An optimal solution to the mismatch between the overlay and the physical network would be possible only if the physical relationship of peers is interpreted exactly in an overlay by assigning an LID to a peer such that the peer's LID is logically close to the LIDs of its physically close peers, i.e., the LIDs of peers reflect their physical proximity in the overlay. To achieve this goal, each peer in 3DO computes a LID in the form of an ordered three tuple $\{x|y|z\}$, where each tuple is an *M*-bit identifier calculated from a pre-determined logical identifier space (LID space). The 3D-LID space ranges from 1 to $\pm 2^M$ for each axis, i.e., *x*, *y*, and *z*. The protocol uses probe messages between direct logical neighboring peers to maintain an overlay, i.e., it relies on local information. Each node periodically transmits a probe message to its direct logical neighboring peers that contains the peer *P*'s LID, its UID, *P*'s LID space portion, list of *P*'s neighboring peers, and distance of *P* from its neighboring peers. In addition to the LID at each peer, a dimension parameter (*dim*) is maintained to group peers with respect to different dimensions that is helpful while routing packets. The basic idea is that each peer envisions its neighboring peers in a 3D rectangular coordinate system, i.e., local 3D-Overlay consisting of three planes that divide the space into six dimensions and eight octants. Each peer acts as the origin of its local 3D-Overlay, where each neighboring peer obtains its LID that reflects its relationship with other neighboring peers. The basic motivation behind using 3D-Overlay is to satisfy the requirements defined in Section 3. These requirements are essential to avoid the mismatch between the overlay and the physical network. The decision choices explained in Cases 1–4 (see Section 5.1) is to logically interpret the physical intra-neighbor relationship of a peer with respect to its neighboring peers.

Weights are assigned to each link using the inverse distance function, providing connectivity to its neighboring peers on the basis of their hop distances. The hop distance information between two peers is obtained from the underlying routing agent (in our case OLSR [15]). When a peer joins the network, it is placed in the overlay topology in a systematic way as discussed in Section 5.1, meeting requirement (v) in Section 1. In our system, a peer builds up the minimum spanning tree (MST) and assigns LIDs to peers by using local neighboring peers information that makes our protocol distributive in nature, satisfying requirement (vi).

The detail of each component of the 3DO is presented in the following subsections.

5.1. Peer join

To join the P2P network, a peer P_j broadcasts a join request message (JRQST) in the network using the expanded ring search (ERS) algorithm [17,21] in order to find the peers that are physically adjacent to itself in the network. To join the network, a peer is required to listen for a certain waiting time, T_w, to receive a join-reply message (JRPLY) from an existing peer corresponding to JQRST. A peer node sends the JRPLY to P_j upon receiving JRQST. The JRPLY from peer P contains P's LID, its directly connected neighboring peers along with their distances to P. Algorithm 1 illustrates the handling of JRQST and JRPLY.

Upon receiving JRPLY, P_j stores the information in the JRPLY message in its peer-routing table. Then, P_j builds a weighted undirected connected graph consisting of itself, its directly connected neighboring peers and its 2-hop neighboring peers, i.e., neighboring peers of P_j's directly connected peers. The weight of the link in the graph between two logically linked peers is obtained by taking the inverse of the distance (in terms of number of hops) between them in the physical network. The distance information is obtained from the underlying routing agent, i.e., OLSR, by using the cross-layer mechanism and its peer-routing table. Using this graph, the join peer P_j executes its MST with itself as the source vertex. The join peer P_j stores a peer as its directly connected neighboring peer if that peer is directly adjacent in its MST and has not been selected previously as the directly connected neighboring peer. P_j removes a peer from the directly connected neighboring peer list if that peer is no longer adjacent in its MST. This is done in order to obtain the physically closest peers of the joining peer in the network, achieving requirement (i) in Section 1.

A non-peer node simply forwards the JRQST to another node when receiving it, provided the time-to-live (TTL) value of the JRQST message has not expired. P_j stops sending JRQST when T_w reaches the maximum threshold.

After T_w expires, the joining peer computes its LID based on one of the following cases:

- If the joining peer does not receive any JRPLY, it assumes that there is no online peer in the P2P network so it automatically assigns itself the LID {1|1|1}-0.
- If the joining peer receives at least one JRPLY, it computes its LID with respect to its physical adjacent neighboring peer(s) using the heuristics explained in Case 1 to Case 4 below, achieving requirement (ii) in Section 1. Assume node P_i is the existing peer in the P2P network with LID {1|1|1}-0.

Case 1: If peer P_j joins and finds P_i as its only neighboring peer, P_j obtains its distance in hops to P_i from its

Algorithm 1: Handling of JQRST and JRPLY()

```
Joining peer $P$ Broadcast JQRST using ERS
if receive JRPLY then
    store neighbor peer info in $NT_p$
end if
if $T_w$ expires then
    if $NT_p \neq null$ then
        set $X_1$ = { Select $P_{x1} \in NT_p$ such that $P_{x1}$ is a logical neighbor peer of $P$, which
                                      is either 1-hop or 2-hop away }
        set $X_2$ = {Select $P_{x2} \in NT_p$ is a 1-hop logical neighbor peer of $P$}
        Build a connected weighted undirected graph of $P$ and $P_{x1}$ □ $X_1$
              {/*Weight are based on hop distance between $P$ and $P_{x1}$ in the physical network
                and calculated using Eq. 1*/}
        Execute the MST algorithm with $P$ as a source vertex
        set Y = { $P_y : P_y$ is a directly connected neighbor of $P$ in MST}
        set Z = { select $P_w \mid P_w \in$ Y and $P_w \notin X_2$}
        send Cprob towards $P_w \in$ Z
        remove all $P_d \in NT_p \mid P_d \in X_2$ and - $P_d \notin$ Y
        call LIDComputation()
    else
        set LIDp = {1|1|1}-0
    end if
end if
```

routing agent and checks P_i's neighboring peers information received in JRPLY. If P_i does not have any neighbor except P_j, peer P_j selects the first available dimension of peer P_i (say, + x-dimension) out of the six dimensions along the positive and negative axes in the local 3D-Overlay of P_i and calculates its LID$_j$ using the following formula: $\{T_{ix} + \left(\frac{\text{LSP}_{ix}+}{4}\right)|T_{iy}|T_{iz}\}$, where T_{ix}, T_{iy}, and T_{iz} are the three tuple of the LID$_i$ of peer P_i, and LSP$_{ix}$+ is the maximum range of peer P_i's LID space portion in the positive x-dimension. By using this formula, peer P_j obtains ¾ of peer P_i's LSP$_{ix}$+. The purpose here is to give a greater LID space portion to the corner peers so that they can accommodate new peers in the future. After obtaining its LID$_j$, node P_j sets its dimension parameter to 1 as LID$_j$ belongs to the positive x-dimension. Algorithm 2 illustrates the computation of LID in Case 1.

Algorithm 2: LIDComputation() (Case1)

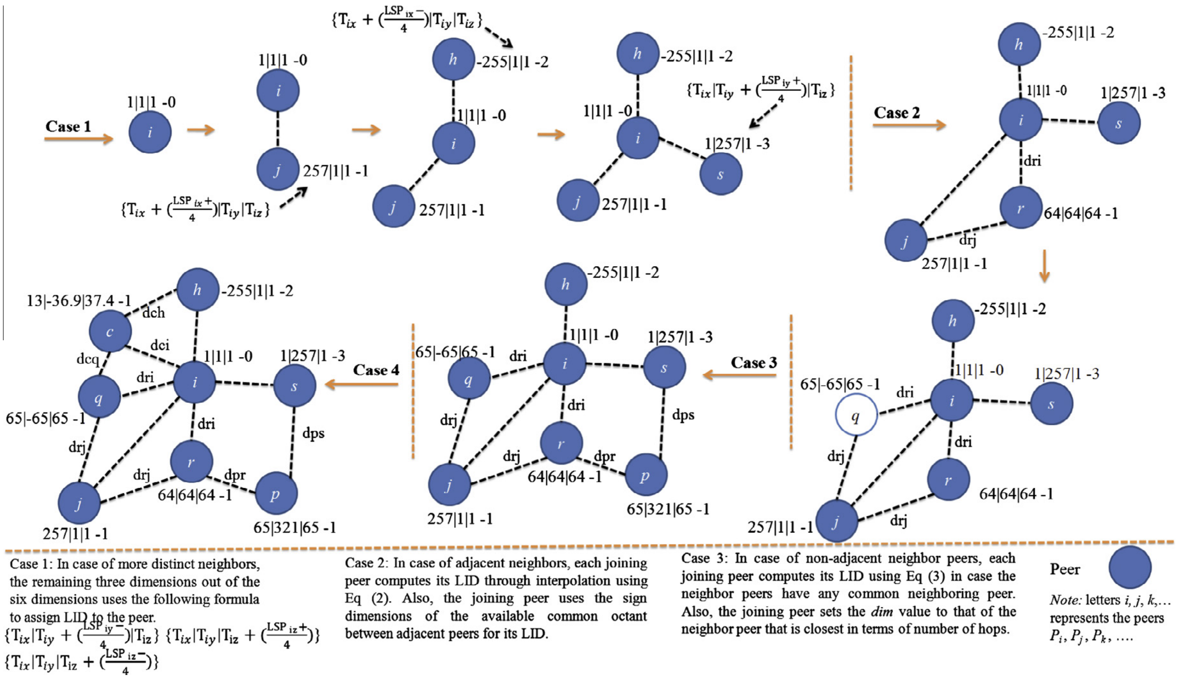
Required: Information related to neighbor peer P_i is stored in P's peer-neighbor table (NT_p) and hop distance to P_i is obtained using routing agent, i.e., OLSR at joining peer P.

```
if $\exists P_i \in NT_p$ : $(\exists P_m \in \text{Nbr}(P_i))$ != $P_j$ then
    $dim_p \leftarrow$ NextAvailableDim($P_m$, $P_i$)
else
    $dim_p \leftarrow$ first($dim(P_i)$)
end if
LID$_p \leftarrow$ ComputeLID($P_i$, $dim_p$)
```

Using a similar procedure, the joining process of peers P_h and P_s along with their LIDs are shown in Fig. 3. Peers P_h and P_s compute their LID$_h$ and LID$_s$ corresponding to the negative x-dimension and the positive y-dimension by using the following formula: $\{T_{ix} + \left(\frac{\text{LSP}_{ix}-}{4}\right)|T_{iy}|T_{iz}\}$ and $T_{ix}|T_{iy} + \left(\frac{\text{LSP}_{iy}+}{4}\right)|T_{iz}\}$, respectively, where LSP$_{ix}$– is the maximum range of the LSP for peer P_i in the negative x-dimension and LSP$_{iy}$+ is the maximum range of LSP for peer P_i in positive y-dimension. Moreover, peers P_h and P_s set their dim value to 2 and 3, respectively. Fig. 3 illustrates the joining of peers P_j, P_h, and P_s.

The basic motivation for the decision choices made in Case 1 is to map the physical intra-neighbor relationship of a peer in the 3D-Overlay. If two neighboring peers are not directly linked logically to each other, it means that these two neighboring peers physically exist in two different dimensions of the peer. In such scenarios, 3DO is capable of assigning LIDs to peers in the overlay that reflects the physical intra-neighbor relationship of peers.

Case 2: If peer P_r joins the overlay, as shown in the Fig. 3, and receives JRPLY from two peers P_i and P_j, and both P_i and P_j are logically adjacent in overlay network, then peer P_r first obtains the distances d_{qi} and d_{qj} in number of hops using its routing agent. After this peer P_r assigns weights to each of these links using a simple inverse distance weighting function,



Case 1: In case of more distinct neighbors, the remaining three dimensions out of the six dimensions uses the following formula to assign LID to the peer.
$\{T_{ix}|T_{iy} + (\frac{\text{LSP}_{iy}-}{4})|T_{iz}\}$ $\{T_{ix}|T_{iy}|T_{iz} + (\frac{\text{LSP}_{iz}+}{4})\}$
$\{T_{ix}|T_{iy}|T_{iz} + (\frac{\text{LSP}_{iz}-}{4})\}$

Case 2: In case of adjacent neighbors, each joining peer computes its LID through interpolation using Eq (2). Also, the joining peer uses the sign dimensions of the available common octant between adjacent peers for its LID.

Case 3: In case of non-adjacent neighbor peers, each joining peer computes its LID using Eq (3) in case the neighbor peers have any common neighboring peer. Also, the joining peer sets the *dim* value to that of the neighbor peer that is closest in terms of number of hops.

Note: letters *i, j, k,*… represents the peers $P_i, P_j, P_k, \ldots$.

Fig. 3. The peer-joining process. Black dashed lines represent the physical links between neighboring peers in the physical network.

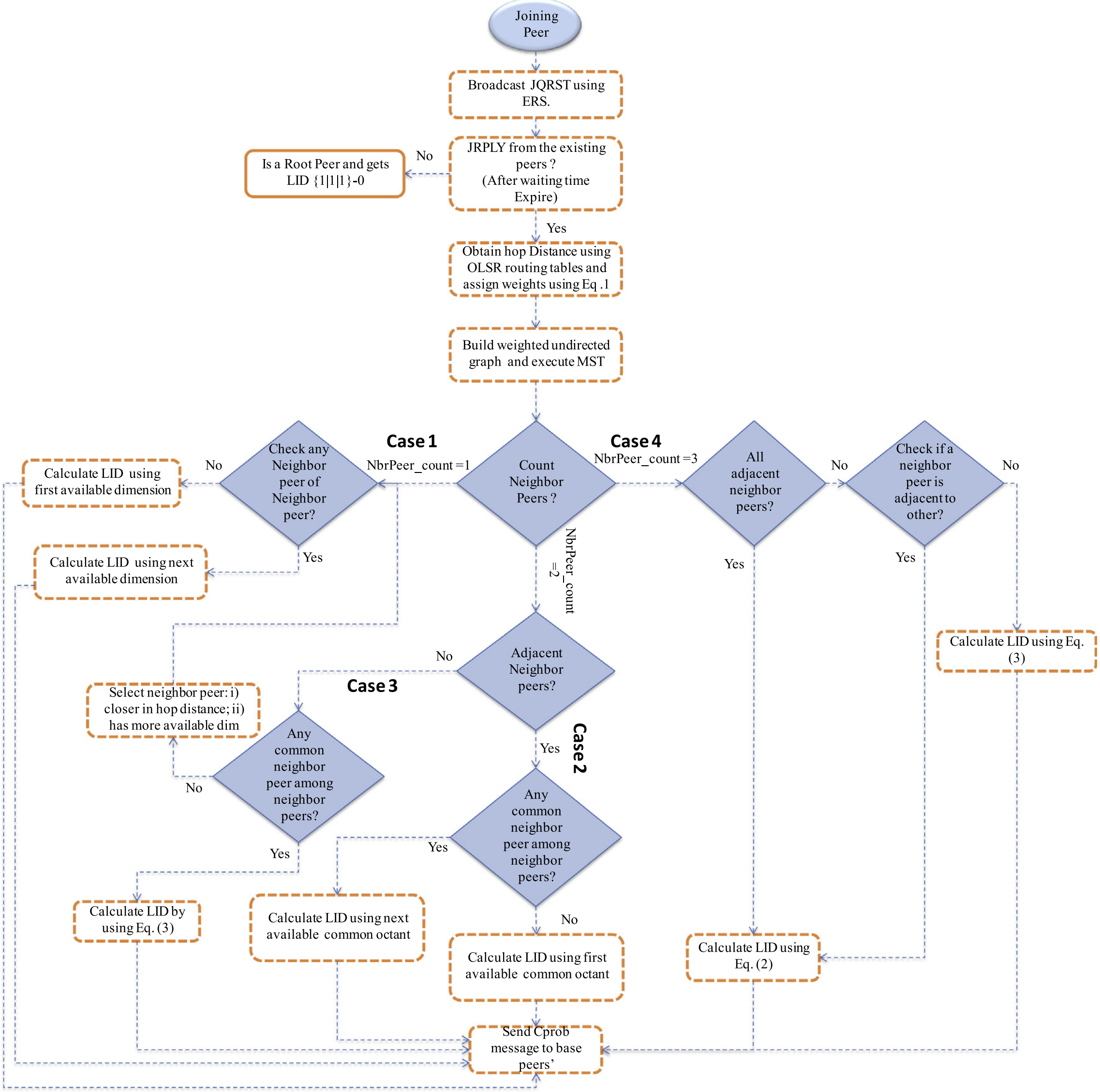

Joining Peer
Broadcast JQRST using ERS.
JRPLY from the existing peers ? (After waiting time Expire)
No
Is a Root Peer and gets LID {1|1|1}-0
Yes
Obtain hop Distance using OLSR routing tables and assign weights using Eq .1
Build weighted undirected graph and execute MST
Case 1
NbrPeer_count =1
Count Neighbor Peers ?
Case 4
NbrPeer_count =3
Check any Neighbor peer of Neighbor peer?
No
Calculate LID using first available dimension
Yes
Calculate LID using next available dimension
All adjacent neighbor peers?
No
Check if a neighbor peer is adjacent to other?
No
Yes
Yes
Calculate LID using Eq. (3)
NbrPeer_count =2
Adjacent Neighbor peers?
No
Case 3
Select neighbor peer: i) closer in hop distance; ii) has more available dim
Any common neighbor peer among neighbor peers?
No
Yes
Case 2
Yes
Any common neighbor peer among neighbor peers?
Yes
No
Calculate LID by using Eq. (3)
Calculate LID using next available common octant
Calculate LID using first available common octant
Calculate LID using Eq. (2)
Send Cprob message to base peers'

Fig. 4. Flow chart of the peer joining algorithm.

$$W_{mk} = \frac{1}{d(P_m, P_k)^p} \tag{1}$$

where W_{mk} is the weight assigned to the link between a newly joining peer P_m and its neighboring peer P_k, d is the distance in hops between P_m and P_k, and p is a positive real number, called the power parameter whose value is assumed to be 2. Greater values of p assign greater influence to peers closest to the joining peer. Here the weight decreases as the distance in hops increases from the joining peer. After assigning weights to its neighboring peers, P_r checks for a common neighboring peer between P_i and P_j.

$$\mathrm{LID}_m = \left\{ \sum_{k=1}^{P_{nbr}} \frac{W_{mk}}{\sum_{j=1}^{P_{nbr}} W_{mj}} * T_{kx} \middle| \sum_{k=1}^{P_{nbr}} \frac{W_{mk}}{\sum_{j=1}^{P_{nbr}} W_{mj}} * T_{ky} \middle| \sum_{k=1}^{P_{nbr}} \frac{W_{mk}}{\sum_{j=1}^{P_{nbr}} W_{mj}} * T_{kz} \right\} \tag{2}$$

where m is the newly joining peer and $P_{nbr} \geqslant 2$ are 1-hop neighboring peers of m, W_{mk} and W_{mj} are the weights assigned by m to its neighboring peers P_k and P_j, respectively, using the inverse distance function, and T_{kx}, T_{ky}, and T_{kz} are the corresponding tuples P_k's LID in the x, y and z dimensions.

(i) If there exists no common peer between P_i and P_j in the overlay, P_r finds an available common octant between P_i and P_j, and computes LID_r corresponding to a common octant using Eq. (2). Furthermore, P_r sets the signs of each tuple according to the sign dimensions of an available common octant between peers P_i and P_j. (ii) If a common neighboring peer exists (say, peer P_q) between P_i and P_j in the overlay, P_r finds the next available common octant between P_i and P_j, and calculates LID_r corresponding to the dimensions of that common octant using Eq. (2). Algorithm 3 illustrates the computation of LID in Case 2 and Case 3.

The decision choices made in Case 2 is to address a peer's physical adjacency to its neighboring peers and to assign a relative LID to the peer with respect to its adjacent neighboring peers. To obtain the exact relative position of the peer, 3DO exploits Shepard's interpolation method to assign LID to a newly joining peer with respect to its discrete set of neighboring peers. This is an attempt to exactly map the relative position of a peer in the overlay network with respect to its neighboring peers in the physical network.

Algorithm 3: LID Computation() (Case 2 and Case 3)

Required: Information related to neighbors P_i, P_q is stored in P's peer-neighbor table (NT_p) and hop distance to P_i and P_q is obtained using routing agent, i.e., OLSR at joining peer P.

```
if $\exists P_i, P_q \in NT_p : P_p \in \mathrm{Nbr}(P_i)$ and $P_i \in \mathrm{Nbr}(P_q)$ then
    $\mathrm{oct_p} \leftarrow \mathrm{first}(\mathrm{oct_{iq}})$
    if $\exists P_c \in \mathrm{Nbr}(P_i)$ and $\mathrm{Nbr}(P_q)$ then    /* $P_c$ refers to any common neighbor peer other than $P$
        $\mathrm{oct_p} \leftarrow \mathrm{NextCommonOctant}(P_c, P_i, P_q)$
    else
        $\mathrm{oct_p} \leftarrow \mathrm{FirstCommonOctant}(P_i, P_q)$
    end if
else if $\exists P_i, P_q \in NT_p : P_q \notin \mathrm{Nbr}(P_i)$ and $P_i \notin \mathrm{Nbr}(P_q)$ then
    Peer_adj $\leftarrow$ false
    if $\exists P_c \in \mathrm{Nbr}(P_i)$ and $P_c \in \mathrm{Nbr}(P_q)$ where $P_c \neq P$ then
        $\mathrm{LID_j} \leftarrow \mathrm{Compute}(Pi, P_q, \mathrm{Peer\_adj})$  /* using Eq. 3
    else
            $P_{flag} = null$
        if $\mathrm{hopDist}(P_i, P) < \mathrm{hopDist}(P_q, P)$ then
            $dim_p \leftarrow \mathrm{NextAvailableDim}(P_i)$
            $P_{flag} = P_i$
        else if $\mathrm{hopDist}(P_i, P) > \mathrm{hopDist}(P_q, P)$ then
            $dim_p \leftarrow \mathrm{NextAvailableDim}(P_q)$
            $P_{flag} = P_q$
        else
            if ($\mathrm{CountAvaialbleDim}(P_i) > \mathrm{CountAvailableDim}(P_j)$) then
                $dim_p \leftarrow \mathrm{NextAvailableDim}(P_i)$
                $P_{flag} = P_i$
            else
                $dim_p \leftarrow \mathrm{NextAvailableDim}(P_q)$
                $P_{flag} = P_j$
            end if
          $\mathrm{LID_p} \leftarrow \mathrm{ComputeLID}(P_{falg}, dim_p)$
        end if
    end if
end if
```

Case 3: Suppose peer P_p joins and has access only to peer P_r and P_s and both P_r and P_s are not adjacent to each other in the overlay network, as shown in Fig. 3. P_p first calculates its distance, i.e., d_{pr} and d_{qs} in number of hops using its routing agent and assigns weights to each of these links using Eq. (1). Then, peer P_p checks for common neighboring peers between P_r and P_s in the overlay. (i) If a common neighboring peer exists (say, node P_i), between P_r and P_s, then peer P_p computes LID_p simply by adding each tuple of peer P_r and P_s using Eq. (3):

$$\mathrm{LID}_m = \left\{ \sum_{k=1}^{P_{nbr}} T_{kx} \middle| \sum_{k=1}^{P_{nbr}} T_{ky} \middle| \sum_{k=1}^{P_{nbr}} T_{kz} \right\} \tag{3}$$

where m is a newly joining peer and $P_{nbr} \geq 2$ are 1-hop neighboring peers of m. T_{kx}, T_{ky}, T_{kz} are tuples of non-adjacent neighboring peers corresponding to each dimension.

(ii) If there is no common neighboring peer, then P_p calculates LID_p by using the available dimension of either P_r or P_s depending on two parameters: (a) a neighboring peer that is closer in terms of distance, i.e., number of hops and (b) a neighboring peer that has more available dimensions. Algorithm 3 is used to obtain a relative position with respect to non-adjacent neighboring peers in Case 3.

The decision made in Case 3 addresses the physical non-adjacency of neighboring peers and assigns a relative LID to the joining peer in the 3D-Overlay with respect to its non-adjacent neighboring peers. 3DO also exploits the information about a common neighboring peer between two non-adjacent neighboring peers before assigning a LID to a joining peer. If a common neighboring peer exists, it shows some kind of relationship between these two non-adjacent neighboring peers. 3DO uses this relationship to assign a relative position in the overlay to a joining peer in order to minimize the path-stretch caused by the mismatch between the overlay network and the physical network as explained in Section 3.

Case 4: If P_c joins the P2P network and receives JRPLY messages from peers P_i, P_q, and P_h as shown in Case 4 of Fig. 3, then one of the following cases holds. (i) If P_i, P_q, and P_h are adjacent, then P_c calculates LID_c by using Eq (2). (ii) If there exists two non-adjacent peers P_q and P_h, then P_c checks for a common neighboring peer between P_q and P_h. If node P_i is a common neighboring peer to P_q and P_h, P_c computes LID_c using Eq (2). If there exists either a common neighboring peer other than P_i, P_q, or P_h or there is no common neighboring peer, P_c calculates LID_c using Eq. (3). Algorithm 4 illustrates the computation of LID in Case 4.

Algorithm 4: LID Computation() Case 4

Required: Information related to neighbors P_i, P_q and P_h is stored in P's peer-neighbor table (NT_p) and the hop distances to P_i, P_q, and P_h are obtained using the routing agent, i.e., OLSR at the joining peer P.

```
PeerCommon← false
if $\exists P_i,P_q,P_h \in NT_p$ : $P_q \in$ Nbr($P_i$),Nbr($P_h$) and $P_i \in$ Nbr($P_q$),Nbr($P_h$) and $P_h \in$ Nbr($P_i$),Nbr($P_q$) then
    $\text{LID}_\text{p}$ ← ComputeLID($P_i$,$P_q$,$P_h$)
    return
else if $P_i \in$ Nbr($P_q$),Nbr($P_h$) and $P_q \notin$ Nbr($P_h$)) and $P_h \notin$ Nbr($P_q$) or $P_q \in$ Nbr($P_i$),Nbr($P_h$) and $P_i \notin$ Nbr($P_h$) and $P_h \notin$ Nbr($P_i$) or $P_h \in$ Nbr($P_q$),Nbr($P_i$) and $P_q \notin$ Nbr($P_i$)) and $P_i \notin$ Nbr($P_p$) then
    PeerCommon← true
    $\text{LID}_\text{p}$ ← ComputeLID($P_i$,$P_q$,$P_h$) /* using Eq. 2
    return
else
    $\text{LID}_\text{p}$ ← ComputeLID($P_i$,$P_q$,$P_h$, PeerCommon) /* using Eq. 3
end if
```

The sequence diagram in Fig. 4 summarizes the peer joining algorithm.

In addition to calculating its LID, each joining peer in Cases 1–4 sets its *dim* value by checking the *dim* value of its base peer(s). The term 'base peer' refers to peers that are involved in the computation of the joining peer's LID. If the base peers are in the same dimension, the joining peer sets its *dim* value to that of its base peers. If the base peers have different *dim* values, the joining peer sets its *dim* value to the *dim* value of a base peer that is closer in terms of distance.

After computing its LID, a joining peer P sends a connection-probe message (CProb) to each of its base peers, containing P's computed LID, the list of P's directly connected neighboring peers and their distances from P. When a base peer receives a CProb, it stores the information of CProb in its peer routing table and sends a connection-reply prob message (CRProb) to P. Any change to the peer-routing table at a peer triggers an information update to its directly connected neighboring peers.

After joining the P2P network, P computes the ID of its shared files by applying the hash function. Then P stores the index information of the shared files similar to the way in which a lookup query is forwarded, as discussed in Section 5.4.

Fig. 5 illustrates the local view of the 3D-Overlay of peer P_i and its neighboring peers in the P2P network that is built according to the joining process of 3DO.

Fig. 5 is helpful in visualizing the arrangement of peers according to their LIDs in the 3D-Overlay. The black dotted lines are the physical links between the peers. The dashed lines are the three planes of the local 3D-Overlay of P_i. The alphabets represent the IP addresses of the peers while roman numerals represent the eight octants of P_i's 3D-Overlay. Fig. 5 shows the logical mapping of the physical relationships of P_i with its 1-hop neighboring peers shown in Fig. 3. This relationship is expressed in terms of LIDs and logical dimensions of nodes in peer P_i's 3D-Overlay that allows the peers to calculate their LIDs such that the physically close peers have close LIDs. It can be seen from Figs. 3 and 5 that 3DO exactly maps the physical intra-neighbors relationship of peer P_i with its 1-hop neighboring peers in terms of their LIDs. The neighboring peers of P_i in the overlay network are adjacent in the physical network, and peer P_i in its local 3D-Overlay is logically close to all its physically adjacent neighboring peers. This avoids long routes and redundant traffic overhead, and decreases the end-to-end delay caused by the mismatch between the overlay and the physical network discussed in Section 3. Other peers in the P2P network build their local 3D-Overlay in the same way by arranging their 1-hop neighboring peers according to their LIDs computed by 3DO.

In addition, 3DO is resilient against peer/link failures and it facilitates multipath routing, because each peer maintains all its physically adjacent neighboring peers to leverage an alternative route if the next peer towards the destination peer fails/moves. When a peer P wants to leave the P2P network, it informs its neighboring peers and transfers the indexes of its stored files to the corresponding neighboring peers. In this paper, we assume that a peer P does not inform any of its neighboring peers before leaving the network. In this case, the index information of files at P can be retrieved from the secondary anchor peer (see Section 5.5 for detail), but the shared files that are stored at P are lost. This problem can be handled by replicating the shared files at one of a peer's logical neighbor of its local 3D-overlay.

5.2. Update

In 3DO, each peer periodically exchanges probe messages with its directly connected neighboring peers to update and maintain connectivity. For two logically linked peers, the peer with the lower peer ID initiates the sending of the probe message to maintain the link. This rule avoids redundant probe message transmissions. The probe message of a peer *P* contains its LID in the form of a three-tuple and *P*'s directly connected neighboring peers along with their distances from *P*. When *P* does not receive any probe message from a base peer P1 within a timeout, *P* assumes that P1 has either failed or moved out of its transmission range. After detecting this, *P* sets a timer T_{fail} to wait for a specific period of time to allow P1 to reconnect with *P*. When this timer expires, *P* computes a new LID if it no longer connects to any of its base peers. For this purpose, it builds up the MST consisting of itself as the source vertex, its directly connected neighboring peers and its 2-hop neighboring peers, i.e., neighboring peers of *P*'s directly connected peers. Then, it computes a new LID based on the updated list of the directly connected peers. 3DO attempts to maintain the overlay closer to the physical network when the physical network changes due to node mobility. Therefore, requirement (iv) in Section 1 is satisfied.

Each peer in the P2P network also periodically refreshes its shared file index information by sending probe messages to the corresponding peers, i.e., anchor nodes. Similarly, if *P* does not receive a probe message of the index information of one of its stored files and the lifetime of the index information expires, *P* removes that index information from its LID space portion.

5.3. Primary anchor peer computation and file's index information storage

After computing its LID, each peer performs two major operations: (i) it retrieves from its neighboring peers the index information about files with keys closest to its LID and acts as the primary anchor peer (PAP) for these files and (ii) it computes the keys for its stored files and sends the index information of these files to their respective PAPs.

To send the file *f*'s index information (i.e., *P*'s LID, file's key) to some other peer in the P2P network that would later act as a PAP for the files stored at peer *P*, peer *P* applies the hash function to file *f* and generates a key *k* whose value is within the range of the LID space. The LIDs of peers in the P2P network and the key *k* are drawn from the same LID space. Therefore, the key *k* is in the form of an ordered three tuple, where each tuple is within the specified range of the LID space on each axis.

After computing the file key *k*, the peer *P* builds a file-index information store (FIS) message for key *k*. The FIS contains *P*'s LID and UID, and the key *k*. A peer whose LID is closest to *k* stores the index information of that file. To route the FIS message with destination address *k*, each peer uses information about its 1-hop logical neighboring peers (P_{nbr}) and neighboring peers of P_{nbr} ($P_{nbr\to nbr}$) received in the probe messages. The FIS is forwarded to one of its P_{nbr} or $P_{nbr\to nbr}$ that has the same dimension parameter as that of *k* and offers the closest position in every tuple of its LID with respect to *k*, i.e., with the least sum of difference (LSD) to *k*. This is achieved simply by computing the sum of the difference (SD) of each tuple of the P_{nbr} or $P_{nbr\to nbr}$ with the corresponding tuple of *k* using Eq. (4) and then selecting one of them as a next hop with the LSD to the *k* using Eq. (5). Algorithm 5 illustrates the handling of FIS message.

Algorithm 5. File information Storage ()

```
Require: Received FIS message (M) for file's key $k$ at peer P (LID$p$ and UID$p$) and information related to all local neighbor peers is stored in the peer-neighbor table ($NT_p$)
if $dim_p == dim_k$ then
    $flag \leftarrow true, SD_p \leftarrow$ SumDiff($LID_p, k$)
end if
Select $P_i \in NT_p$ such that $\forall P_j \in NT_p, i \neq j : SD_i \leftarrow$ SumDiff($LID_i, k$) $< SD_j \leftarrow$ SumDiff($LID_j, k$)
if flag == $true$ and $SD_p < SD_i$ then
    store File's location information in FIS
else
    send M to $P_i$
end if
```

If such a neighbor does not exist, the peer simply forwards the FIS to its base peer. If *P* has the LSD, then *P* examines its peer-routing table for index information that matches *k*. Otherwise, *P* forwards the FIS to one of its neighboring peers that has the least SD. This process is repeated at each peer until the FIS arrives at the peer closest to the key *k* (PAP for the key *k*).

For example, P_u in Fig. 6 applies a hash function over file *f* and generates a key *k*, i.e., hash (*f*) = *k* = {420|100|14}-3. P_u then builds the FIS message for the file *f*'s key *k*.

In order to forward the FIS, peer P_u first checks its 1-hop logical neighboring peers ($P_{u\text{-}nbr}$), i.e., peer P_s {1|257|1}-3 and its neighbors of neighboring peer ($P_{u\text{-}nbr\to nbr}$), i.e., P_t with LID$_t$ {513|257|1}-3 in its peer-routing table as shown in Fig. 7.

Both the neighboring peers, P_s and P_t, have the same *dim* value as that of the key *k*. Peer P_u then calculates the sum of difference of its neighboring peers P_s and P_t using Eq. (3), i.e., $SD_s = \{|1 - 420| = 419\} + \{|257 - 100| = 157\} + \{1 - 14 = 13\} \to \{419 + 157 + 13\} \to \{589\}$ and $SD_t = \{|513 - - 420| = 93\} + \{|257 - 100| = 157\} + \{1 - 14 = 13\} \to \{93 + 157 + 13\} \to \{263\}$, respectively, with respect to the key *k*, i.e., {420|100|14}-3. P_u then sends the FIS as shown in Fig. 6 to the peer P_t having the least SD, i.e., 263 to the key *k*. Upon receiving the FIS, P_t finds itself closest to *k* and stores the file *f*'s index information received in the FIS.

5.4. File discovery

To retrieve a file *f*, *P* applies the hash function to *f*, producing a key *k* within the range of the LID space. After obtaining the key *k*, peer *P* builds the file-lookup query

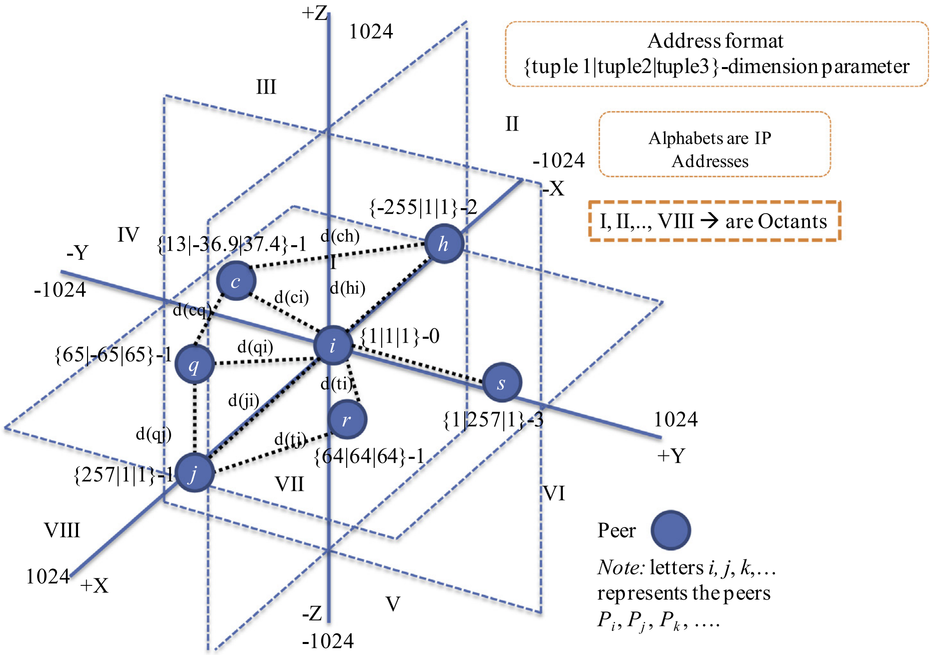
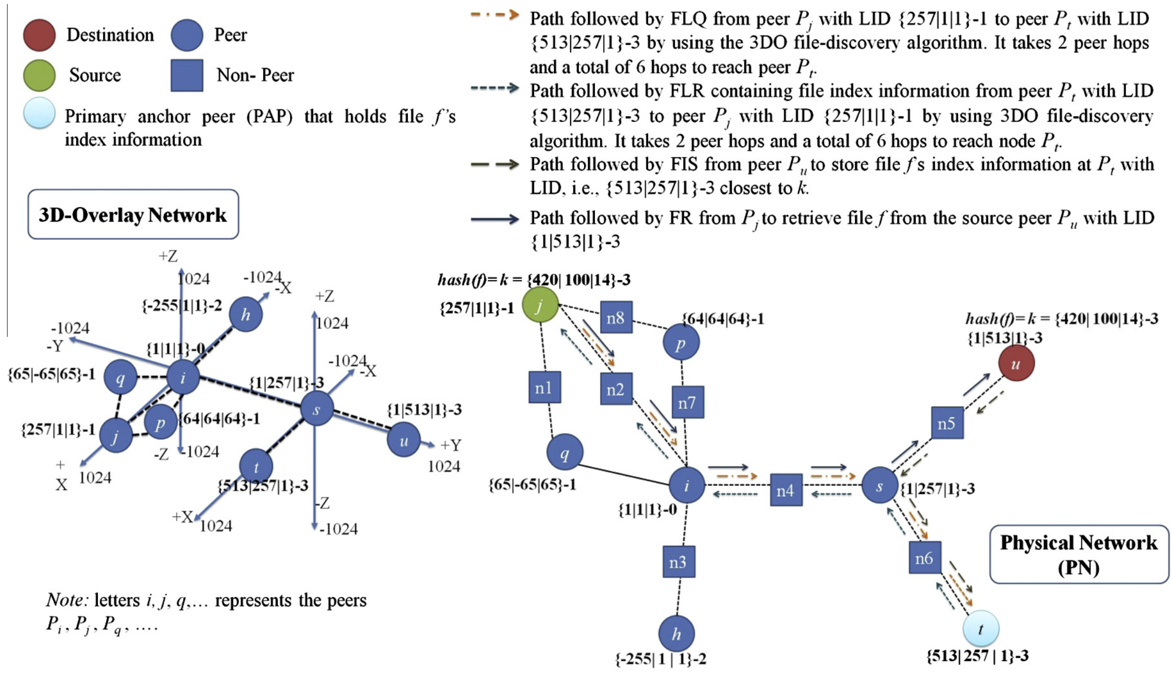


Fig. 5. A logical view of the physical arrangement of neighboring peers in the local 3D-Overlay of peer P_i maintained by the 3DO.


Path followed by FLQ from peer P_j with LID {257|1|1}-1 to peer P_t with LID {513|257|1}-3 by using the 3DO file-discovery algorithm. It takes 2 peer hops and a total of 6 hops to reach peer P_t.

Path followed by FLR containing file index information from peer P_t with LID {513|257|1}-3 to peer P_j with LID {257|1|1}-1 by using 3DO file-discovery algorithm. It takes 2 peer hops and a total of 6 hops to reach node P_t.

Path followed by FIS from peer P_u to store file f's index information at P_t with LID, i.e., {513|257|1}-3 closest to k.

Path followed by FR from P_j to retrieve file f from the source peer P_u with LID {1|513|1}-3

Fig. 6. File's index information storage, lookup and retrieval process in 3DO. The overlay network on the left side shows the arrangement of peers in the 3D-Overlay. The physical network on the right describes the physical arrangement of nodes in the P2P network.

(FLQ) for the key k. FLQ contains the requesting peer's LID and UID, and the key k. A peer whose LID is closest to k is responsible for k. To route FLQ with destination address k, each peer (similar to the routing of FIS) uses information about its P_{nbr} and $P_{nbr\rightarrow nbr}$ that is received in the probe messages. The peer forwards the FLQ to one of its peers P_{nbr} or $P_{nbr\rightarrow nbr}$ that has the same *dim* value to that of k and offers the closest position in every tuple of its LID with respect to k, i.e., with the least sum of difference (LSD) to key k.

$$\mathrm{SD}_{nbr} = (|T_{nbrx} - T_{kx}|) + (|T_{nbry} - T_{ky}|) + (|T_{nbrz} - T_{kz}|) \qquad (4)$$

$$\mathrm{LSD}_{nbr} = \min_{nbr \in L_{nbr}} \mathrm{SD}_{nbr} \tag{5}$$

where SD_{nbr} is the sum of differences of each element of L_{nbr}'s LID to the corresponding element of k or hashed value; T_{nbrx}, T_{nbry}, T_{nbrz} are three tuples of L_{nbr}'s LID, and T_{kx}, T_{ky}, T_{kz} are three tuples of k or hashed value; LSD_{nbr} is the least sum of difference of L_{nbr}, and L_{nbr} are P_{nbr} and $P_{nbr \to nbr}$.

When the anchor peer responsible for the key k receives the FLQ, it sends the file-lookup reply message (FLR) to the requesting peer. FLR contains either the index information of file f or a NULL value in case there is no index information of file f at that peer. To limit the lifetime, the FLQ is associated with a TTL value. Receiving the index information for file f, the requesting peer retrieves the file directly from the source peer by sending a file retrieval (FR) message via a short route in the physical network using the underlying routing due to the proactive nature of OLSR. This satisfies requirement (iii) mentioned in Section 1. Algorithm 6 illustrates the handling of the FLQ message during the file discovery process.

Algorithm 6. File Discovery ()

Require: File name (f) and information related to all local neighbor peers is stored in the peer-neighbor table (NT_p) of P.

```
hash (f) = k    /* k is the form of an ordered three tuple with a random dim value. Each element of k is
                   drawn from the same LID space, i.e., used to assign LIDs to peers.
bulid FLQ for k
if $\dim_p == \dim_k$ then
    flag ← true, $SD_p$ ← SumDiff($\mathrm{LID}_p$, k)
end if
Select $P_i \in NT_p$ such that $\forall P_j \in NT_p, i \neq j : SD_i \leftarrow \mathrm{SumDiff}(\mathrm{LID_i}, k) < SD_j \leftarrow \mathrm{SumDiff}(\mathrm{LID_j}, k)$
if flag == true and $SD_p < SD_i$ then
    get File with key k
else
    send FLQ to $P_i$
end if
```

For the sake of simplicity, a scheduling algorithm similar to [14] can be used to retrieve the file in blocks. Since a peer in our system has the information up to its 2 hop neighboring peers, the lookup query moves two hops towards the destination peer. Therefore, in our system, a lookup query for the data item of the key k can be resolved in $O([p/2])$ time, where p is the number of peers in the P2P network. However, this does not mean that 3DO has a longer lookup delay than Chord, as Chord with a finger table can resolve a lookup query in $(\log p)/2$ on average, where p is the number of peers in the P2P network. But, this is the hop-cost of resolving a lookup query in the overlay network. If the overlay network does not match the underlying physical network as is the case with Chord, the real cost in the physical network might be much higher.

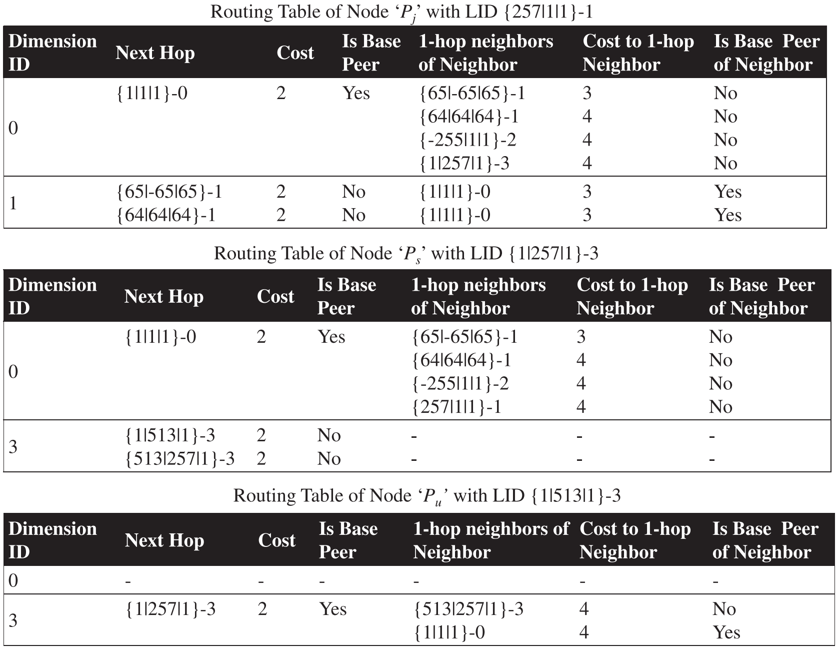
For example, peer P_j with LID {257|1|1}-1} initiates a file-lookup query (FLQ) for the key k = {420|100|14}-3. P_j first checks its 1-hop logical neighboring peers ($P_{j\text{-}nbr}$), i.e., peer P_i {1|1|1}-0 and peer P_q {65|−65|65}-1 and their neighboring peers ($P_{j\text{-}nbr \to nbr}$), i.e., P_s {1|257|1}-3 and peer P_h {−255|1|1}-2, in its peer-routing table as shown in Fig. 7. P_j checks the *dim* value and calculates the sum of difference of $P_{j\text{-}nbr}$ and $P_{j\text{-}nbr \to nbr}$. Peer P_s has the same *dim* value as that of k, i.e., *dim* = 3. So, P_j forwards the FLQ for k towards peer P_s as shown in Fig. 6. P_s then checks its $P_{s\text{-}nbr}$ and $P_{s\text{-}nbr \to nbr}$ in its peer-routing table as shown in Fig. 6. P_s has two $P_{s\text{-nbr}}$, i.e., P_t {513|257|1}-3 and P_u {1|513|1}-3 with the same *dim* value as that of key k. Peer P_s then calculates the sum of the differences using Eq. (3), i.e., SD_t = {|513 − 420| = 93} + {|257 − 100| = 157} + {14 − 1 = 13} → {93 + 157 + 13} → {263} and SD_u = {|1−420| = 419} + {|513 − 100| = 413} + {1 − 14 = 13} → {419 + 413 + 13} → {845} of its neighboring peers P_t and P_u, respectively, with respect to key k, i.e., {420|100|14}-3. P_s finds the LSD using Eq. (4). P_s sends the FLQ towards the P_t that has the least SD, i.e., 263.

P_t then checks its file index table to search for the index information of the file against the corresponding key k. P_t then sends the FLR containing the required file index information towards P_j. Upon receiving the file index information, P_j can then communicate directly with the source peer P_u that holds the corresponding file. For this purpose, P_j sends the FR towards P_u to retrieve file f as shown in Fig. 6.

In the above example, we observe that FLQ observes the shortest route in the overlay network without generating any redundant traffic in the physical network. Moreover, we observe that the intra-neighbor relationships with peers in the overlay network is the same as that of the physical network to avoid the mismatch between overlay network and the physical network.

5.5. Replication

3DO adopts a simple replication strategy to avoid the loss of a file's index information in case the primary anchor peer (PAP) moves or fails. As we mentioned above, the PAP is the peer whose LID is closest to the file f's key k. After receiving the file f's index information, the PAP selects a peer from its 1-hop neighboring peers with the LID second closest to k as a secondary anchor peer (SAP) and replicates the index information of files at the SAP. The SAP becomes active in case the PAP fails or moves.



Routing Table of Node 'P_j' with LID {257\|1\|1}-1

Dimension ID	Next Hop	Cost	Is Base Peer	1-hop neighbors of Neighbor	Cost to 1-hop Neighbor	Is Base Peer of Neighbor
0	{1|1|1}-0	2	Yes	{65|-65|65}-1	3	No
				{64|64|64}-1	4	No
				{-255|1|1}-2	4	No
				{1|257|1}-3	4	No
1	{65|-65|65}-1	2	No	{1|1|1}-0	3	Yes
	{64|64|64}-1	2	No	{1|1|1}-0	3	Yes

Routing Table of Node 'P_s' with LID {1\|257\|1}-3

Dimension ID	Next Hop	Cost	Is Base Peer	1-hop neighbors of Neighbor	Cost to 1-hop Neighbor	Is Base Peer of Neighbor
0	{1|1|1}-0	2	Yes	{65|-65|65}-1	3	No
				{64|64|64}-1	4	No
				{-255|1|1}-2	4	No
				{257|1|1}-1	4	No
3	{1|513|1}-3	2	No	-	-	-
	{513|257|1}-3	2	No	-	-	-

Routing Table of Node 'P_u' with LID {1\|513\|1}-3

Dimension ID	Next Hop	Cost	Is Base Peer	1-hop neighbors of Neighbor	Cost to 1-hop Neighbor	Is Base Peer of Neighbor
0	-	-	-	-	-	-
3	{1|257|1}-3	2	Yes	{513|257|1}-3	4	No
				{1|1|1}-0	4	Yes

Fig. 7. Peer-routing table for peers P_j, P_s and P_u.

For example, peer P_t acts as a PAP and stores file f's index information, i.e., P_u's LID and UID, and file f's key k. To replicate file f's index information, P_t selects a peer from its 1-hop neighboring peers as the SAP that has the second closest SD to key k. P_s has SD_s, i.e., 589 that is second closest to k. So, P_s acts as the SAP for P_t, and P_t replicates its index information of files at P_s. P_s becomes active in case P_t moves or fails.

6. Modeling and analysis using formal methods

We briefly describe HLPN, SMT-Lib, and Z3 below for better understanding of the reader.

6.1. High level petri nets

Petri nets model the system graphically and mathematically, and can be applied to a range of systems that are distributed, parallel, concurrent, non-deterministic, stochastic, or asynchronous [43]. We have used a variant of conventional petri net called high-level petri nets (HLPN).

Definition 1 (*HLPN*). [43]. A HLPN is a 7-tuple $N = (P, T, F, \varphi, R, L, M_0)$, where a set of places is denoted by P. T refers to the set of transitions such that $P \cap T = \varnothing$. F denotes a flow relation such that $F \subseteq (P \times T) \cup (T \cup P)$. φ maps places P to data types. R is a set of rules for transitions. L is a set of labels of F and M_0 represents the initial marking. (P, T, F) provides information about the structure of the net and (φ, R, L) provides static semantics, which means that the information does not change in the system.

In HLPN, places can have tokens of different types, and can be a cross product of two or more types. The pre-conditions must hold for any transition to be enabled. Moreover, the variables from the incoming flows are used to enable a certain transition. Similarly, the post-conditions use variables from outgoing flows for transition firing.

6.2. SMT-Lib and Z3

SMT-Lib is used for checking the satisfiability of formulae over the theories under consideration [44]. SMT-Lib provides a common input platform and benchmarking framework that helps in the evaluation of a systems. SMT has been used in many fields including deductive software verification. We use the Z3 solver of SMT-Lib, which is a theorem prover developed at Microsoft Research. Z3 is an automated satisfiability checker. In addition, Z3 determines whether the set of formulas is satisfiable in the built-in theories of SMT-Lib. Readers are encouraged to read [45] for use of SMT-Lib in the verification process.

6.3. Formal analysis and verification

The verification process checks for the correctness of the system. Bounded model checking verifies whether for any input parameters, the system terminates in a finite number of states. In bounded model checking: (a) a description of the system states properties or rules of the system; (b) the system is represented by a model; and (c) a verification tool is used to check whether or not the model satisfies the specified properties. In this paper, we use bounded model checking to verify 3DO.

The HLPN model for 3DO is shown in Fig. 8.

The first step in the development of the petri net model is to identify data types, places and mappings of data types to places. Data types and mappings are shown in Tables 2

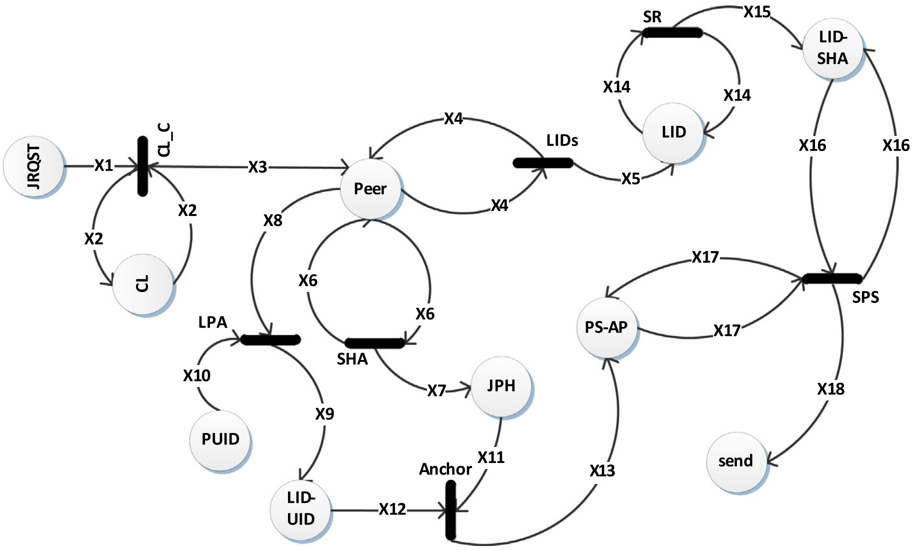


Fig. 8. HLPN model for 3DO.

and 3, respectively. In the HLPN model, all the rectangular black boxes are transitions and belong to set *T*. The circles are places and belong to set *P*.

Table 2 describes the data types and their definitions for 3DO. These data types are used in the HPLN of 3DO.

Table 3 is a mapping table for the HPLN model of 3DO. It describes the places and their corresponding mappings.

The working of 3DO is discussed in Section 5. In this section, we define formulas/rules that maps to transitions.

Table 2
Data types for the HLPN model.

Data type	Description
LID	A number representing the logical ID of a peer
PList	A list containing the LID of neighboring peers and the LIDs of neighboring peers of a neighboring peer
IP	A number representing the IP address of a peer
Hvalue	A string representing the hash value
LI	The LID of a joining peer
L2	The LID of primary anchor peer
L3	The LID of secondary anchor peer
HSA	The hash value of the LID of anchor peer
M	A string representing the message to be sent

Table 3
Places and mappings used in the HLPN model.

Place	Mapping
φ (*CL*)	$\mathbb{P}$(PList)
φ (Peer)	$\mathbb{P}$(PList × LID)
φ (LID–UID)	$\mathbb{P}$ (LID × UID)
φ (PUID)	$\mathbb{P}$ (UID)
φ (JPH)	$\mathbb{P}$ (Hvalue)
φ (PS-AP)	$\mathbb{P}$ (L1 × L2 × L3 × Hvalue × UID)
φ (LID)	$\mathbb{P}$ (LID)
φ (LID-SHA)	$\mathbb{P}$ (LID × HSA)
φ (Send)	$\mathbb{P}$ (M × UID)

The system starts with a joining peer sending a JRQST message and looking for neighboring peers, if there exist any. The following formula maps to the aforesaid transition:

$$R(CL\_C) = \forall x_1 \in X_1, \quad \forall x_2 \in X_2, \quad \forall x_3 \in X_3 | \\ X_3' = X_3 \cup \{x_2\} \tag{6}$$

After analyzing the response(s) from the neighboring peers, the joining peer determines and sets its LID according to the peer joining process explained in Section 5.1. The mathematical rule for the transition LIDs is as follows:

$$\begin{aligned} &R(LIDs) = \forall x_4 \in X_4, \quad \forall x_5 \in X_5 | \\ &x_4[1] = NULL \wedge x_4[2] := LID - Root() \wedge \\ &X_4' = X_4 \cup \{x_4[1], x_4[2]\} \wedge \\ &x_4[1] \neq NULL \wedge x_4[2] := LID - Set() \wedge \\ &X_4' = X_4 \cup \{x_4[1], x_4[2]\} \wedge X_5' = X_5 \cup \{x_4[2]\} \end{aligned} \tag{7}$$

The selection of a primary anchor peer requires generation of the hash value of peer's UID. The following transition and rule illustrates the process:

$$\begin{aligned} &R(SHA) = \forall x_6 \in X_6, \quad \forall x_7 \in X_7 | \\ &x_7 := Hsh(x_6[2]) \wedge \\ &X_7' = X_7 \cup \{x_7\} \end{aligned} \tag{8}$$

The logical and physical addresses are placed at place LID-IP by the transition LPA for model checking purposes. The rule over transition LPA highlights the process is the following:

$$\begin{aligned} &R(LPA) = \forall x_8 \in X_8, \quad \forall x_9 \in X_9, \quad \forall x_{10} \in X_{10} | \\ &x_9[1] := x_8[2] \wedge x_9[2] := x_{10} \wedge \\ &X_9' = X_9 \cup \{x_9[1], x_9[2]\} \end{aligned} \tag{9}$$

The process of anchor peer selection (primary and secondary) is represented by the following transition and associated formula:

$$R(Anchor) = \forall x_{11} \in X_{11}, \quad \forall x_{12} \in X_{12}, \quad \forall x_{13} \in X_{13}| \\ x_{13}[1] := x_{12}[1] \wedge x_{13}[4] := x_{11} \wedge x_{13}[5] := x_{12}[2] \wedge \\ X'_{13} = X_{13} \cup \{x_{13}[1], x_{13}[2], x_{13}[3], x_{13}[4], x_{13}[5]\} \tag{10}$$

Before sending any message, the peer requests the LID of the destination peer from the destination's primary anchor peer as described in the routing process that is explained earlier in Sections 5.3 and 5.4. The procedure is captured by the following rule:

$$R(SR) = \forall x_{14} \in X_{14}, \quad \forall x_{15} \in X_{15}| \\ x_{15}[1] := x_{14} \wedge x_{15}[2] := hash(x_{14}) \wedge \\ X'_{15} = X_{15} \cup \{x_{15}[1], x_{15}[2]\} \tag{11}$$

At the end, a peer acquires the corresponding LID-IP pair of the destination peer and sends the file retrieval message to the received LID of the destination peer. The procedure is represented by the following rule:

$$R(SPS) = \forall x_{16} \in X_{16}, \quad \forall x_{17} \in X_{17}, \quad \forall x_{18} \in X_{18}| \\ \exists x_{17}[1] : x_{16}[1] = x_{17}[1] \wedge x_{18}[2] = x_{17}[5] \wedge \\ X'_{18} = X_{18} \cup \{x_{18}[1], x_{18}[3]\} \tag{12}$$

6.4. Verification property

The aim of verification was to ensure that 3DO works according to the specifications and produces correct results. The following properties are verified:

- The LIDs calculated by the peers are according to the specifications and peers place themselves in the dimension according to the process.
- Requests for the LID of a destination peer are sent to the proper primary anchor peer and the primary anchor peer returns a valid and corresponding LID according to the received UID.

The above model was translated to SMT-Lib and verified using the Z3 solver. The solver showed that the model executes according to the specified properties. The Z3 solver took 0.05 s to execute 3DO.

7. Simulation results

We analyze the performance of the proposed scheme in NS-2 (version 2.35) [27] by using the standard values for both the physical and the link layers to simulate IEEE 802.11. The simulation parameters are illustrated in Table 4.

The proposed protocol is compared with MA-SP2P [22], which is a competitive approach for P2P overlays over MANETs. We performed ten runs per scenario. The upper and lower bars in the graphs show the margin of error of the mean estimates at 95% confidence interval.

We use OLSR [15] as the underlying routing protocol in our approach. The mobility scenarios are created according to the RandomWayPoint mobility model using Bonnmotion2 to ensure that the physical network is connected. The peers share ten unique files. File discovery is randomly initiated for 100 random files by the peers in the network. We study the performance of overlay maintenance and file-discovery by varying several parameters including the peer ratio. The peer ratio is defined as the ratio of the number of peers in the P2P overlay network to the total number of nodes in the physical network.

Table 4
Simulation parameters.

Input parameters	Value
Transmission range	50 m
Playground size	[1000 * 1000 m]
Data rate	[100 pps]
Simulation time	500 s
Start of data transmission	[70,300]
End of data transmission	[250,499]
Node speed	[0.5 m/s to 2 m/s]
Start of node failure	100 s
Mobility model	Random way-point
Radio propagation model	TwoRayGround Model
Traffic model	Random Traffic pattern
No. of file retries	2

For performance comparison, we choose the following parameters with varying network size.

- *Path-stretch ratio:* The ratio of the path length available in the P2P network to the length of the shortest path in the physical network between two peers. It describes the ability of the protocol to find the shortest possible route to a destination peer in the network. The path-stretch ratio equals to 1 means that the protocol takes the shortest route.
- *Average file discovery delay:* The average time elapsed from the moment when the file-look-up query is issued to the moment when the first reply is received.
- *Routing overhead:* The total number of control overhead packets used by the protocol to perform routing of query data packets.
- *False-negative (FN) ratio (%):* The ratio between the number of unresolved lookup queries for the destination in the physically connected network to the total number of initiated lookup queries.

A key difference between MANETs and fixed networks is node mobility that requires routes to be constantly updated. The same is true for queries in P2P overlay networks that expect the destination to be reached at a given time. We have evaluated the impact of the physical topology changes in 3DO and MA-SP2P. We study the mobility by varying the average node speed from 0.5 m/s, to 1.0 m/s, to 1.5 m/s, to 2.0 m/s.

7.1. Routing overhead

Fig. 9 shows that the routing traffic overhead for both approaches increases with the increase in the node speed and the peer ratio in the network for the following two reasons.

First, the frequent change in network topology due to the increase in node speed produces more traffic for

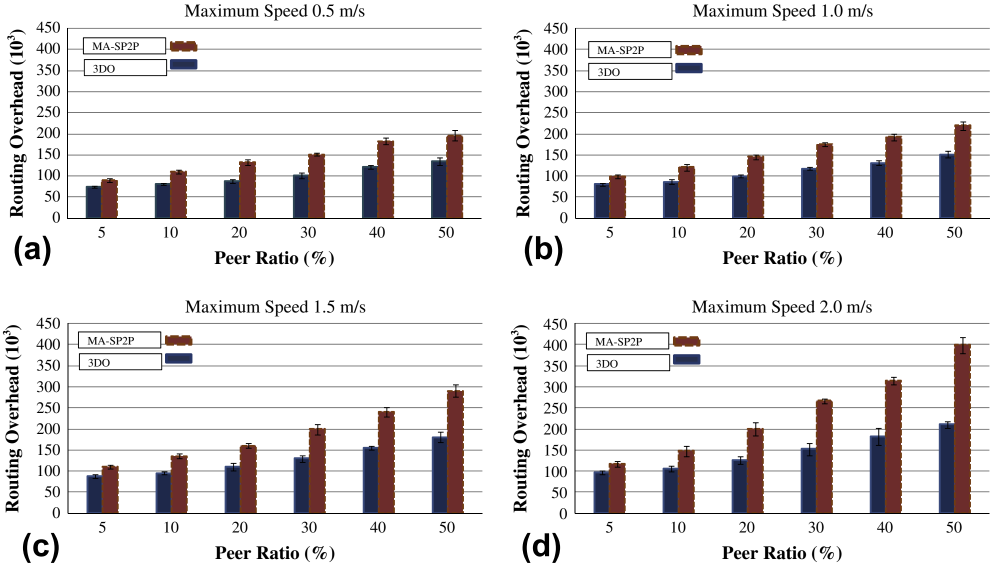


Fig. 9. Routing overhead as a function of peer ratio.

readjusting the overlay topology to match the physical topology. Peers out of range or broken routes increase the routing overhead. This effect is quite pronounced for both 3DO and MA-SP2P. However, the effect of an increase in node speed on the routing overhead is more in MA-SP2P compared to 3DO. This is due to the inflexible tree-like structure used to distribute the LID space portion between peers. In MA-SP2P, a parent peer retrieves the LID space portion of its child peer in case the child peer moves to some other place, generating extensive traffic overhead. However, in 3DO, the LID space portion of each peer is implicit and does not require any explicit mechanism to retrieve the LID space portion in case a peer moves. Moreover, the replication strategy used by 3DO, in case the primary anchor peer moves/fails, effectively reduces the overhead, especially, in case the node speed increases. The percentage improvement of 3DO over MA-SP2P in terms of the routing overhead is more at 2 m/s (Fig. 9(d)) than at 0.5 m/s (Fig. 9(a)) and it is increasing with an increase in the peer ratio. This shows the effectiveness of the proposed protocol with respect to the increase in the node speed and the peer ratio.

Second, increasing the number of peers in the network introduces more traffic in maintaining connectivity among the peers. *Third*, more traffic is needed to maintain the index information at the peers as the number of peers increases. Fig. 9 shows that 3DO is better in terms of the routing overhead because 3DO builds and maintains an overlay that better matches the physical network, eliminating redundant long links. Moreover, it introduces the replication strategy in case the anchor peer moves/fails, which effectively reduces the overhead. On the other hand, in MA-SP2P, peers frequently exchange information about their LID space portion and the index of the stored files when their distance to the root-peer changes due to node mobility, resulting in heavy network traffic. Secondly, a peer *P* (except the root-peer) has at least one directly connected neighboring peer, say P1, such that P1 is closer to the root-peer than *P* and *P* stores the portion of the identifier space higher than P1's identifier space. Peer *P* might not be a neighbor in the overlay network to all of its physical adjacent peers, resulting in an mismatch between the overlay and the physical network.

7.2. Average file discovery delay

Fig. 10 illustrates the average file discovery delay for MA-SP2P and 3DO.

The average file-discovery delay for both protocols increases with an increase in the peer ratio and node speed. This is caused by the heavier routing traffic, more contention due to the increase in peer ratio, and frequent change of network topology. But, MA-SP2P has a larger average file discovery delay compared to 3DO. The reasons are that: (i) MA-SP2P has a higher false-negative ratio than 3DO, which shows that, in MA-SP2P, some accesses to files never happen because of the false-negative results in locating the file; (ii) the resulting topological mismatch between the overlay and the physical network due to its tree-based overlay network prevents MA-SP2P from being efficient in the file lookup process; (iii) the LID space distribution among the peers is inconsistent in MA-SP2P and the peers are not placed in a proper overlay structure (like ring, chord, and multi-dimensional spaces), which causes the file-lookup query to experience larger delay in the physical network; and (iv) in MA-SP2P, a peer shares a consecutive LID space portion with all of its neighboring peers. A parent peer retrieves the LID space portion of its child peer in case the child peer moves to some other place, introducing extra control traffic overhead in the network. Fig. 10(d) illustrates that the effect of this increases with the increase in node speed.

In 3DO: (i) the 3D-Overlay avoids the topological mismatch between the overlay and the physical network due to its flexible 3D structure and design choices when assigning a LID to a peer, resulting in an efficient file

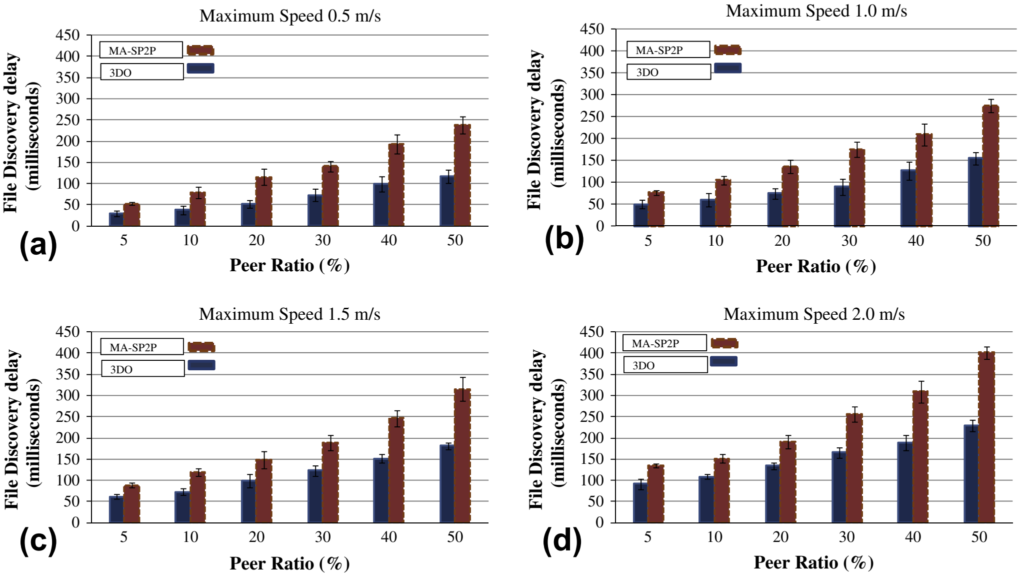


Fig. 10. Average file discovery delay as a function of peer ratio.

lookup process as illustrated in Fig. 10. Fig. 10 shows that the file discovery delay of 3DO is low compared to MA-SP2P. The false-negative ratio in 3DO is less, which ensures maximum accesses to the files and confirms the results in Fig. 10; (ii) a peer *P* in the 3D-Overlay maintains its adjacent/non-adjacent, common neighbor, and intra-neighbor relationships with its neighboring peers by assigning different LIDs and utilizing the 3D-Overlay. Consequently, in 3DO, a file-lookup query from a peer is always forwarded closer to the destination peer in both the overlay and the physical network and experiences a shorter route to the destination peer. This reduces the lookup delay; and (iii) the LID space portion recovery in 3DO is implicit because the LID of a peer determines the index information to be stored at the peer. No explicit recovery mechanism is used in 3DO for the retrieval of the LID space portion, which reduces the control overhead, resulting in a low file discovery delay. The improvement percentage of 3DO over MA-SP2P in terms of the file discovery delay is more at a node speed of 2 m/s as shown in Fig. 10(d) compared to the file discovery delay at a node speed of 0.5 m/s as shown in Fig. 10(a), which establishes the efficiency of 3DO with increasing node speed and peer ratio.

7.3. False-negative ratio

Fig. 11 illustrates the false-negative (FN) ratio of both MA-SP2P and 3DO.

The results show that the false-negative increases as the peer ratio and the node speed increase. But, the FN ratio of MA-SP2P is higher compared to 3DO. Increasing the peer ratio causes more routing traffic, therefore, more serious contention and packet loss problem in the network. MA-SP2P generates more routing traffic with an increase in the peer ratio because of the mismatch between the overlay and the physical network. Similarly, the network topology changes more frequently due to an increase in the node speed, resulting in a larger delay caused by contention in accessing the information and by an increase in the number of packet collision in the network. Node mobility increases the losses of packets sent. This occurs because more routes are invalid, and hence fewer messages are forwarded. The effect is quite pronounced for both 3DO and MA-SP2P. However, the effect of an increase in node speed on the false-negative ratio is more in MA-SP2P compared to 3DO. This is due to the simple and efficient lookup process, and flexible overlay structure of 3DO that avoids the topological mismatch between the overlay and physical topology, reducing the overhead on the control and data planes, which increases the successful accesses to the files as shown in Fig. 11. Moreover, the replication strategy of 3DO helps to reduce the traffic overhead in case a primary anchor peer moves/fails, resulting in a low false-negative ratio.

In MA-SP2P, peers frequently exchange information about their LID spaces and indices of stored files when their distance to the root-peer changes due to node mobility, resulting in heavy network traffic that leads to more packet collisions in the network and increases the FN ratio. Moreover, an explicit mechanism is introduced in MA-SP2P to retrieve the LID space portion of its child peer in case the child peer moves/fails, resulting in inconsistent index information that increases the false-negative ratio for MA-SP2P. There is no replication strategy in MA-SP2P to provide consistent and up-to-date index information about files in case a primary anchor peer moves/fails.

7.4. Path-stretch ratio

Reducing the path-stretch ratio improves the network performance by reducing redundant transmissions in the network. For example, for 1 vs 1.4 ratio, if the shortest available path is 6 hops, then 1:1.4 would be equivalent to 6:8.4. This means that the path with a path-stretch value 1.4 is almost 3 hops longer. This is equivalent to 3 extra transmissions in the network, which in turn increases the file discovery delay and the routing overhead.

Fig. 12 plots the path-stretch ratio of MA-SP2P and 3DO against the peer ratio.

In 3DO, the average path-stretch ratio is lower compared to MA-SP2P for all peer ratios. The path-stretch ratio of 3DO stays slightly above the shortest path, but this increase is reasonable and the mean value stays below 1.2. The slight increase of path-stretch ratio in 3DO results when a joining peer, for instance *P*, comes in contact with two non-adjacent neighboring peers (say, P1 and P2) with different dim values and there exists no common neighboring peer, so that *P* gets an LID using the available dimensions of either P1 or P2, depending on which one is closer in terms of hop distance. In this case, the LID of *P* would show only its relative position in the 3D-Overlay with respect to that neighboring peer from which it gets its LID. This can cause a slight mismatch problem in 3DO. However, this situation occurs less frequently in 3DO and its impact is less as shown by the simulation results in Fig. 12.

The path-stretch ratio of MA-SP2P as shown in Fig. 12 is high compared to 3DO, because of the tree-based structure of MA-SP2P. On the other hand, the 3D-overlay exactly maps the physical intra-neighbors relationship of peers with its 1-hop neighboring peers in terms of their LIDs.

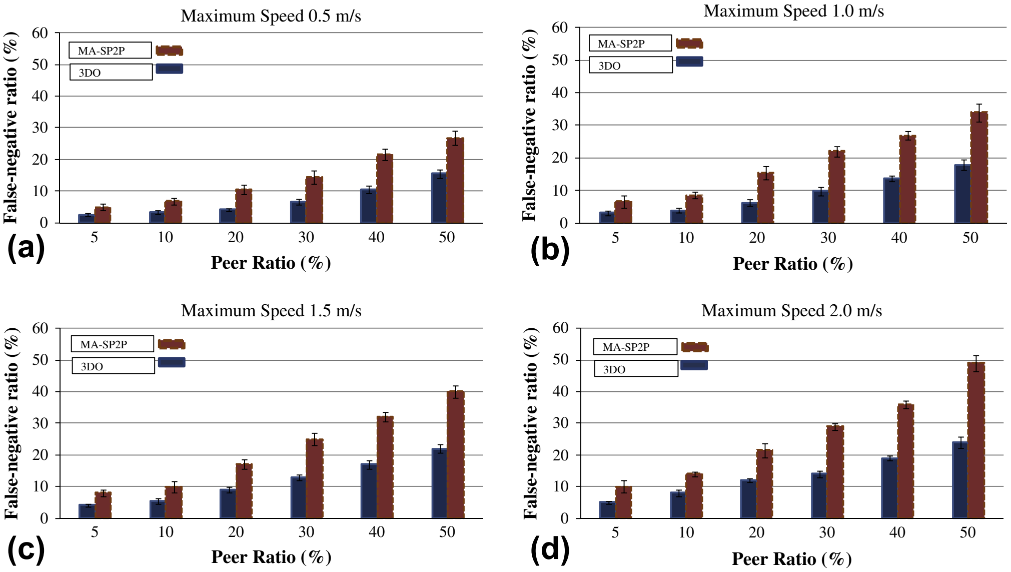


Fig. 11. False-negative ratio as a function of the peer ratio.

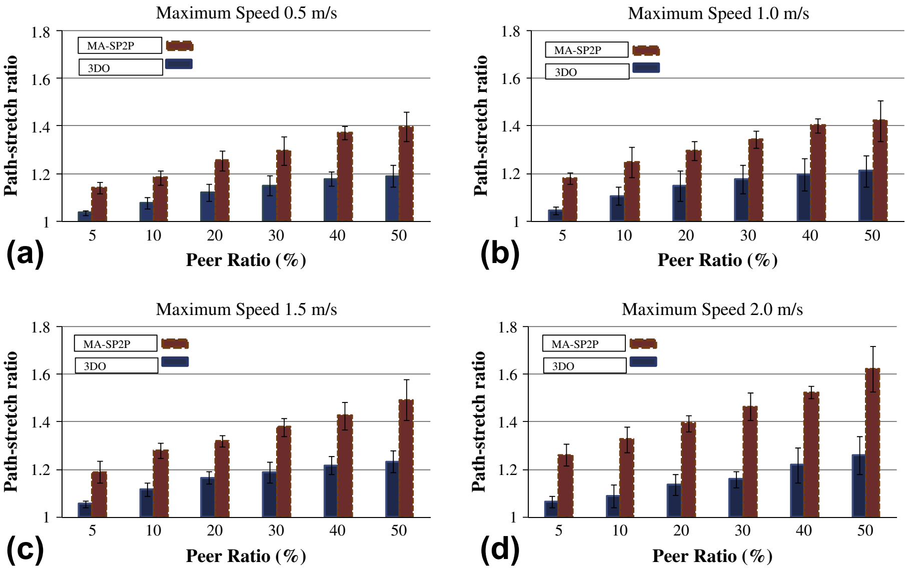


Fig. 12. Path-stretch ratio as a function of the peer ratio.

Moreover, it ensures: (i) the neighboring peers of a peer in the overlay network are adjacent in the physical network and (ii) a peer in its local 3D-Overlay is logically close to all its physically adjacent neighboring peers. This avoids long routes and redundant traffic overhead, and decreases the end-to-end, which establishes the results reported in Fig. 12.

The path-stretch of MA-SP2P increases with an increase in the peer ratio. By increasing the peer ratio, the topology of MA-SP2P scales up and the path length between the requesting peer and the source peer also increases. However, we can learn from Fig. 12 that the average path-stretch of 3DO is slightly affected by the peer ratio. But the value does not always equal one. This means that sometimes the path between the requesting peer and the source peer is not the shortest one in the physical network. In our view, this is due to the MPR (multi-point relay) selection in the OLSR routing.

8. Conclusions and future work

This paper presents a 3-dimensional overlay for P2P over MANETs called 3DO. Our approach constructs an efficient 3D-Overlay over a MANET with a topology matching the physical network. Using this structure of interconnection among peers, we design a new overlay routing algorithm to distribute, manage, and share file information among the peers. 3DO takes into account the physical intra-neighbor peer relationship of a peer by exploiting a 3D-Overlay. A performance simulation has been conducted using scenarios with relatively high peer ratio and low node speed. The simulation results show that our approach outperforms MA-SP2P in terms of routing overhead, average file-discovery delay, false-negative ratio, and average path-stretch ratio and proves to be effective in avoiding mismatch between the overlay network and the physical network. Scenarios with sparse peers and high node speed will be considered in our further experiments to investigate the applicability of our approach. An analytical comparison of our approach with other approaches will also be conducted in our future work. We also plan to address other issues such as load balancing, P2P network partition, user anonymity, and free-riding in our future research.

References

[1] I. Stoica, R. Morris, D. Liben-Nowell, D.R. Karger, M.F. Kaashoek, F. Dabek, H. Balakrishnan, Chord: a scalable peer-to-peer lookup protocol for Internet applications, IEEE/ACM Trans. Network. 11 (2003) 17–32.

[2] A. Rowstron, P. Druschel, Pastry: scalable, decentralized object location, and routing for large-scale peer-to-peer systems, in: Proceedings of Middleware, November 12–16, Springer, Berlin Heidelberg, Germany, 2001, pp. 329–350.

[3] B. Pourebrahimi, K. Bertels, S. Vassiliadis, A survey of peer-to-peer networks, in: Proceedings of the 16th Annual Workshop on Circuits, Systems and Signal Processing, 2005.

[4] R. Matei, A. Iamnitchi, I. Foster, Mapping the Gnutella network, IEEE Inter. Comput. 6 (2002) 50–57.

[5] E. Meshkova, J. Riihijärvi, M. Petrova, P. Mähönen, A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks, Comp. Netw. 52 (2008) 2097–2128.

[6] L.B. Oliveira, I.G. Siqueira, A.A. Loureiro, On the performance of ad hoc routing protocols under a peer-to-peer application, J. Paral. Distrib. Comput. 65 (2005) 1337–1347.

[7] D.N. da Hora, D.F. Macedo, L.B. Oliveira, I.G. Siqueira, A.A. Loureiro, J.M. Nogueira, G. Pujolle, Enhancing peer-to-peer content discovery techniques over mobile ad hoc networks, Comp. Commun. 32 (2009) 1445–1459.

[8] R. Kummer, P. Kropf, P. Felber, Distributed lookup in structured peer-to-peer ad-hoc networks, in: Proceedings of the 2006 Confederated international conference on the Move to Meaningful Internet Systems: CoopIS, DOA, GADA, and ODBASE – Volume Part II, Springer-Verlag, Montpellier, France, 2006, pp. 1541–1554.

[9] M. Li, E. Chen, P.C. Sheu, A chord-based novel mobile peer-to-peer file sharing protocol, in: Frontiers of WWW Research and Development-APWeb, Springer, Berlin Heidelberg, 2006, pp. 806–811.

[10] R.-H. Hwang, C.C. Hoh, Cross-layer design of P2P file sharing over mobile ad hoc networks, Telecommun. Syst. 42 (2009) 47–61.

[11] H. Sözer, M. Tekkalmaz, I. Korpeoglu, A peer-to-peer file search and download protocol for wireless ad-hoc networks, Comp. Commun. 32 (2009) 41–50.

[12] U. Lee, J.-S. Park, S.-H. Lee, W.W. Ro, G. Pau, M. Gerla, Efficient peer-to-peer file sharing using network coding in MANET, J. Commun. Netw. 10 (2008) 422–429.

[13] M. Shin, W.A. Arbaugh, Efficient peer-to-peer lookup in multi-hop wireless networks, Trans. Inter. Inform. Syst. 3 (2009) 5–25.

[14] N. Shah, D. Qian, An efficient structured P2P overlay over MANET, in: Proceedings of the 9th ACM International Workshop on Data Engineering for Wireless and Mobile Access, ACM, Indianapolis, IN, 2010, pp. 57–64.

[15] T. Clausen, P. Jacquet, C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, L. Viennot, Optimized link state routing protocol (OLSR), (2003) RFC 3626.

[16] S.R. Das, E.M. Belding-Royer, C.E. Perkins, Ad hoc on-demand distance vector (AODV) routing, (2003) RFC 3561.

[17] J. Deng, S. Zuyev, On search sets of expanding ring search in wireless networks, Ad Hoc Netw. 6 (2008) 1168–1181.

[18] D.F. Macedo, A.L. dos Santos, J.M. Nogueira, G. Pujolle, Fuzzy-based load self-configuration in mobile P2P services, Comp. Netw. 55 (2011) 1834–1848.

[19] J.C. Liang, J.C. Chen, T. Zhang, An adaptive low-overhead resource discovery protocol for mobile ad-hoc networks, Wirel. Netw. 17 (2011) 437–452.

[20] S.B. Lee, S.H.Y. Wong, K.W. Lee, S. Lu, Content management in a mobile ad hoc network: beyond opportunistic strategy, Int. J. Commun. Netw. Distrib. Syst. 10 (2013) 123–145.

[21] J. Hassan, S. Jha, Optimising expanding ring search for multi-hop wireless networks, in: Proceedings of the IEEE Global Telecommunications Conference, Dallas, TX, November 2004, pp. 1061–1065.

[22] N. Shah, D. Qian, R. Wang, MANET adaptive structured P2P overlay, Peer-to-Peer Network. Appl. 5 (2012) 143–160.

[23] D. Shepard, A two-dimensional interpolation function for irregularly-spaced data, in: Proceedings of the 23rd ACM National Conference, 1968, pp. 517–524.

[24] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker, A scalable content-addressable network, ACM SIGCOMM Comp. Commun. Rev. 31 (4) (2001) 161–172.

[25] M. Caesar, M. Castro, E.B. Nightingale, G. O'Shea, A. Rowstron, Virtual ring routing: network routing inspired by DHTs, ACM SIGCOMM Comp. Commun. Rev. 36 (4) (2006) 351–362.

[26] S.A. Abid, M. Othman, N. Shah, Exploiting 3D structure for scalable routing in MANETs, IEEE Commun. Lett. 17 (11) (2013) 2056–2059.

[27] K. Fall, K. Varadhan, The NS Manual (Formerly NS Notes and Documentation), The VINT Project <http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf>.

[28] M.C. Castro, A.J. Kassler, C.-F. Chiasserini, C. Casetti, I. Korpeoglu, Peer-to-Peer overlay in mobile ad-hoc networks, in: Handbook of Peer-to-Peer Networking, Springer, 2010, pp. 1045–1080.

[29] Ritchie S. King, Building a Subversive Grassroots Network, IEEE Spectrum, 2011 <http://spectrum.ieee.org/telecom/internet/building-a-subversive-grassroots-network>.

[30] A. Reynolds, J. King, S. Meinrath, T. Gideon, The commotion wireless project, in: Proceedings of the 6th ACM Workshop on Challenged Networks, Las Vegas, NV, 2011, pp. 1–2.

[31] Mobile phones could soon be helping in the aftermath of disasters by becoming an ad-hoc message passing network, Daily Mail Reporter, Mail Online, London, August 2011 <http://www.dailymail.co.uk/sciencetech/article-2024585/Mobile-phones-used-disasters-pass-emergency-messages.html>.

[32] Mobiles become emergency data net, EmDiv Portal, Brazil, August 2011 <http://emdiv.com.br/en/world/technology/33052-mobiles-become-emergency-data-net.html>.

[33] M. López-Nores, J. García-Duque, J.J. Pazos-Arias, Y. Blanco-Fernández, M. Ramos-Cabrer, A. Gil-Solla, R.P. Díaz-Redondo, A. Fernández-Vilas, KEPPAN: knowledge exploitation for proactively-planned ad-hoc networks, J. Netw. Comp. Appl. 32 (2009) 1194–1209.

[34] B. Distl, G. Csucs, S. Trifunovic, F. Legendre, C. Anastasiades, Extending the reach of online social networks to opportunistic networks with PodNet, in: Proceedings of the Second International Workshop on Mobile Opportunistic Networking, ACM, Pisa, Italy, February 2010, pp. 179–181.

[35] N. Shah, D. Qian, Cross-layer design for merging of unstructured P2P networks over MANET, in: Proceedings of the 5th International Conference on Ubiquitous Information Technologies and Applications, Sanya, China, 2010, pp. 1–7.

[36] N. Shah, D. Qian, An efficient unstructured P2P overlay over MANET using underlying proactive routing, in: Proceedings of the 7th International Conference on Mobile Ad-Hoc and Sensor Networks, Beijing, China, 2011, pp. 248–255.

[37] N. Shah, D. Qian, Context-aware routing for peer-to-peer network on MANETs, in: Proceedings of the International Conference on Networking, Architecture, and Storage, Hunan, China, 2009, pp. 135–139.

[38] H. Shen, Z. Li, K. Chen, A scalable and mobility-resilient data search system for large-scale mobile wireless networks, IEEE Trans. Paral. Distrib. Syst. (2013). 1-1.

[39] M. Fanelli, L. Foschini, A. Corradi, A. Boukerche, Self-adaptive context data distribution with quality guarantees in mobile P2P networks, IEEE J. Select. Areas Commun. 31 (2013) 115–131.

[40] H. Li, K. Bok, J. Yoo, Mobile P2P social network using location and profile, in: Y.-H. Han, D.-S. Park, W. Jia, S.-S. Yeo (Eds.), Ubiquitous Information Technologies and Applications, Springer, 2013, pp. 333–339.

[41] J.-L. Kuo, C.-H. Shih, C.-Y. Ho, Y.-C. Chen, A cross-layer middleware for context-aware cooperative application on mobile ad hoc peer-to-peer network, J. Syst. Softw. (2013).

[42] E. Papapetrou, P. Vassiliadis, E. Rova, A. Zarras, Cross-layer routing for peer database querying over mobile ad hoc networks, Comp. Netw. 56 (2012) 504–520.

[43] T. Murata, Petri nets: properties, analysis and applications, Proc. IEEE 77 (1989) 541–580.

[44] L. de Moura, N. Bjørner, Satisfiability modulo theories: an appetizer, in: M.V.M. Oliveira, J. Woodcock (Eds.), Formal Methods: Foundations and Applications, Springer, Berlin Heidelberg, 2009, pp. 23–36.

[45] S.U.R. Malik, S.K. Srinivasan, S.U. Khan, L. Wang, A methodology for OSPF routing protocol verification, in: Proceedings of the 12th International Conference on Scalable Computing and Communications, Changzhou, China, December 2012.

[46] M. Gerla, C. Lindemann, A. Rowstron, P2P MANET's-new research issues, in: M. Gerla, C. Lindemann, A. Rowstron (Eds.), Perspectives Workshop: Peer-to-Peer Mobile Ad Hoc Networks – New Research Issues, Dagstuhl Seminar Proceedings, vol. 05152, Dagstuhl, Germany, 2005 <http://drops.dagstuhl.de/opus/volltexte/2005/213>.

[47] A. Klemm, C. Lindemann, O.P. Waldhorst, A special-purpose peer-to-peer file sharing system for mobile ad hoc networks, in: Proceedings of the Vehicular Technology Conference, 2003, pp. 2758–2763.

[48] A. Duran, C.-C. Shen, Mobile ad hoc P2P file sharing, in: Proceedings of the Wireless Communications and Networking Conference, 2004, pp. 114–119.

[49] A. Abada, L. Cui, C. Huang, H.-H. Chen, A novel path selection and recovery mechanism for MANETs P2P file sharing applications, in: Proceedings of the Wireless Communications and Networking Conference, 2007, pp. 3472–3477.

[50] C. Lindemann, O.P. Waldhorst, A distributed search service for peer-to-peer file sharing in mobile applications, in: Proceedings of the 2nd International Conference of Peer-to-Peer Computing, Linköping, Sweden, September 2002, pp. 73–80.

[51] M. Conti, E. Gregori, G. Turi, A cross-layer optimization of gnutella for mobile ad hoc networks, in: Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Urbana–Champaign, IL, 2005, pp. 343–354.

[52] N. Shah, D. Qian, An efficient overlay for unstructured P2P file sharing over MANET using underlying cluster-based routing, Trans. Inter. Inform. Syst. 4 (2010) 799–818.

Shahbaz Akhtar Abid received his MCS degree from Bahuddin Zakariya University, Pakistan. Afterward, he received MS(CS) degree in Data Communication Networks and Distributed Systems from COMSATS Institute of Information Technology, Pakistan. He enjoyed the fully funded performance based scholarship during his MS(CS) from the same institute. He is currently doing his Ph.D. under Bright Spark Scholarship Program (BSP). He is attached with Faculty of Computer Science and Information Technology, University of Malaya, Malaysia. His research interests include routing issues in self-organized networks and distributed systems.

Mazliza Othman received her B.Sc. in Computer Science from Universiti Kebangsaan Malaysia. Later she received a M.Sc. in Data Communication Networks and Distributed Systems and Ph.D. from University College London. She is a Senior Lecturer at the Faculty of Computer Science & Information Technology at Universiti Malaya. Her research interest includes pervasive computing and distributed systems. She is the author of "Principles of Mobile Computing & Communications".

Nadir Shah is working as an Assistant Professor at COMSATS Institute of Information Technology, Pakistan. He received Ph.D. from Sino-German Joint Software Institute, Beihang University, Beijing, China. He earned M.S. in Computer Science from International Islamic University, Islamabad, Pakistan. He earned B.Sc. and M.Sc. in Computer Science from Peshawar University, Peshawar, Pakistan. He was a Lecturer at Department of Computer Science, COMSATS Institute of Information Technology, Abbottabad, Pakistan between August 2007 and June 2008. His current research interests include mobile ad hoc networks, peer-to-peer networks, and delay/disruption tolerant networks.