

# *Design and Evaluation of a Peer-to-Peer MANET Crosslayer Approach: OneHopOverlay4MANET*

Mohammad Al Mojamed  
Computing Science and Mathematics  
University of Stirling  
Stirling, Scotland  
mma@cs.stir.ac.uk

Mario Kolberg  
Computing Science and Mathematics  
University of Stirling  
Stirling, Scotland  
mko@cs.stir.ac.uk

**Abstract:** Peer-to-Peer overlay networks can be deployed over Mobile Ad hoc Networks (MANET) to address content discovery issues. However, previous research has shown that deploying P2P systems straight over MANET do not exhibit satisfactory performance. Bandwidth limitation, limited resources and node mobility are some of the key constraints. OneHopOverlay4MANET exploits the synergies between MANET and P2P overlays through cross-layering. It combines Distributed Hash Table (DHT) based structured P2P overlays with MANET underlay routing protocols to achieve one logical hop between any pair of overlay nodes. In this paper, we present OneHopOverlay4MANET and evaluate its performance when combined with different underlay routing protocols. We evaluate OneHopOverlay4MANET with two proactive underlay (OLSR and BATMAN) and with three reactive underlay routing protocols (DSR, AODV and DYMO). Through simulation we show that the use of OLSR in OneHopOverlay4MANET yields the best performance. In addition, we compare the performance of the proposed system over OLSR to two recent structured P2P over MANET systems (MA-SP2P and E-SP2P) that adopted OLSR as the routing protocol. As simulation result shows, better performance can be achieved using OneHopOverlay4MANET.

**Keywords** (*OneHopOverlay4MANET, P2P, MANET, Cross-layering*)

## 1. INTRODUCTION

P2P networks are a distributed solution for content dissemination and support different applications such as file sharing, voice over IP (VoIP), and messaging without the use of centralised servers. They operate as overlay networks that allow higher-layer communication among peers. The established connections between peers in the overlay are underlay- independent. P2P architecture is mainly designed to operate in infrastructure (wired) networks. However, the rapid development in wireless communication technology and mobile computing has brought a need for the adoption of peer to peer network systems into the mobile field [1][2].

Mobile ad hoc networks (MANET) on the other hand are formed by mobile devices that communicate with each other

using wireless links that do not rely on any pre-existing infrastructure. In MANET, each individual node is regarded as a client and a server at the same time. In such networks, the mobile nodes collaborate with each other and forward network messages towards other nodes.

Many similarities between P2P networks and mobile ad hoc network can be identified. Both networks are self-organizing, decentralized, dynamic and have a changing topology. Consequently, both systems also face similar challenges, most notably the maintenance of connectivity in dynamic and decentralized networks. However, the challenges are amplified when P2P overlays are deployed on MANET underlays. This is a result of the lack of rich services provided in the IP routing infrastructure.

The biggest challenge when combining MANET with a DHT based overlay network is that each single logical hop in the overlay might be translated into a path in the underlay. Consequently, a logical hop results in multiple physical hops. Peers which are neighbours in the overlay may be separated by many physical hops in the underlay. When using multi-hop P2P overlays in such a setting, each overlay hop results in multiple hops in the underlay. Progressing through the overlay path to the last destination may well mean contacting some underlay nodes repeatedly and passing underlay nodes which are very close to the final destination node. Consequently, multi-hop overlays are not well suited to such systems. One-hop overlays are much better suited as they avoid these inefficient routing paths. OneHopOverlay4MANET is proposed to reach the destination in a one logical hop.

Figures 1 and 2 illustrate the basic problem. Figure 1 shows a network topology consisting of 9 nodes. Figure 2 shows an example overlay structure of the same network. Imagine an example where the node with physical ID d and logical ID 30020 tries to find a key that resides on the physical node c with logical ID 30080. In a typical multi hop overlay, the query will go through a number of overlay hops (See Figure 2 dashed path) to reach node c. However, looking at the underlay, node c can be accessed in a single physical hop since it is a neighbouring node to d in the underlay. Thus, one-hop overlays can benefit from efficient underlay routing and avoid crisscrossing the MANET network to reach a peer that might be close in the physical network.

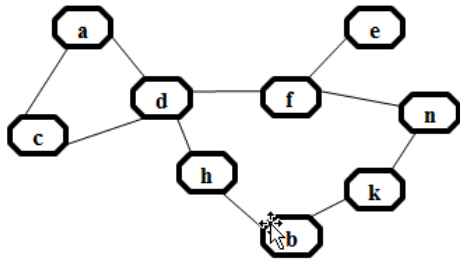


Fig. 1 A MANET Network with nine nodes

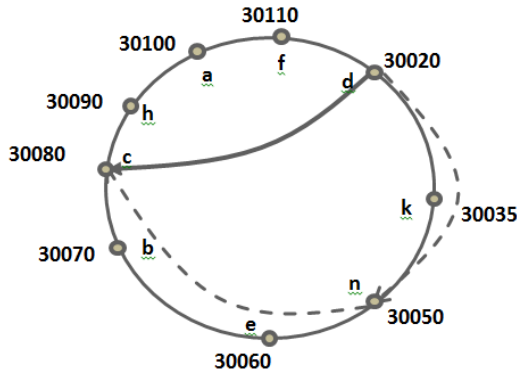


Fig. 2 A logical overlay over a MANET network

OneHopOverlay4MANET [3] employs a structured P2P overlay similar to EpiChord[4], a Distributed Hash Table (DHT) based P2P overlay network which can achieve lookups in a single hop. EpiChord was chosen as the overlay for OneHopOverlay4MANET because of its ability to achieve a one-hop overlay lookup performance. As is shown by Furness and Chowdhury et al [14,39,40], this performance stretches to large networks with thousands of nodes and high churn levels.

The approach presented in this paper is novel as previous work focused on pairing multi-hop overlays with MANET. OneHopOverlay4MANET is the first approach to combine an overlay with one-hop performance with MANET. Our approach uses cross-layering to exploit synergies between MANET and P2P network. The cross-layer channel is used to pass routing information between the underlay routing protocol and the overlay reducing the typical management traffic of deploying P2P over MANET. Therefore, the system is proposed to reduce overlay signalling traffic from nodes joining and leaving the network as well as traffic to maintain the overlay consistency. OneHopOverlay4MANET is designed to combine structured P2P overlays with MANET underlay routing protocols to achieve one hop lookups in the overlay and thus to reduce the message overhead traditionally introduced by combining MANET and P2P overlays. Consequently, OneHopOverlay4MANET significantly improves the performance of data storage and retrieval in MANET based networks.

This paper makes a number of contributions:

- Presenting the OneHopOverlay4MANET system,
- Demonstrating its operation when using different underlay MANET algorithms. OneHopOverlay4MANET is shown to work with two proactive underlay (OLSR[5] and BATMAN[6]) and with three reactive underlay routing protocols (DSR[7], AODV[8] and DYMO[9]).

- Evaluating its performance when combined with different underlay routing protocols. We show that when used with OLSR it achieves its best performance in terms of lookup hop count and maintenance traffic.
- Comparing the performance of OneHopOverlay4MANET over OLSR with two recent P2P over MANET systems (MA-SP2P[10] and E-SP2P[11]) that uses OLSR as the underlay routing protocol. We demonstrate that OneHopOverlay4MANET outperforms these systems under various conditions, incl. those as used in [4] and [5].

The rest of this paper is structured as follow: Section 2 presents a review of related work. In Section 3, OneHopOverlay4MANET and the adopted underlay routing protocols are introduced. The performance of OneHopOverlay4MANET when used with different underlay protocols is then evaluated in Section 4. In Section 5, we compare the performance of OneHopOverlay4MANET to other two proposed P2P systems for MANET. Finally, Section 6 concludes the paper.

## 2. BACKGROUND

### 2.1. RELATED WORK

Distributed Hash Tables (DHT) are typically employed by structured P2P to build and deploy scalable and distributed systems. A review and analysis of structured P2P overlays can be found in [12], [13] and [14]. In order to retrieve a key from a network, multiple logical hops are required by most of the DHT based overlay. However, the lookup can be resolved in a single logical hop in systems such as OneHop[15] EpiChord and D1HT[16]. Our approach OneHopOverlay4MANET employs a structure similar to EpiChord and Chord[17]. EpiChord is a DHT algorithm that can achieve  $O(1)$  hop lookup performance with intensive lookups compared to the  $O(\log N)$  hop performance offered in many multi-hop overlay systems. EpiChord is based on the Chord DHT which builds a one-dimensional circular space. A unique ID is assigned to each peer and a key is under responsibility of the peer whose ID most closely follows the key. EpiChord maintains a list of the  $k$  succeeding peers, a list of the  $k$  preceding peers and a cache of peers. EpiChord peers rely on routing table update information in lookup response messages to update their routing tables. Therefore an entry is added to the cache whenever a peer learns of another peer that does not exist in its cache. In addition, an entry gets deleted from the cache when its associated timer expires.

There are a number of systems in the literature that combine P2P overlays with MANET. Some of these systems have extended or modified existing P2P systems to suit MANET. For instance, M-CAN [18] and M-Chord[19] deploy CAN and Chord respectively over MANET. Grouping and registration are the used mechanisms to improve the performance. In M-CAN, nodes send their requests to their super-node, which in turn forwards the request to the destination node using the CAN algorithm. A hierarchical structure is used in M-Chord and ordinary peers register with super-peers. Chord routing is used between super-peers. A node must register with one or more super nodes according to the data it stores. Each node sends the request to its super

node, which in turn forwards the request on to the destination using standard Chord.

E-SP2P[11] and MA-SP2P[10] construct a minimum spanning tree (MST) to build an overlay that better matches the physical underlay. In E-SP2P, each peer maintains a disjoint portion of the ID space. A root peer is used to control the relationship between neighbours. When two peers establish a neighbour relationship, the root peer is used as a reference point to nominate one of them to be responsible for maintaining the relationship. The peer closer to the root peer will send probe messages to its neighbouring peer. For joining the network, a peer constructs a graph that consists of itself, its direct neighbours and 2 logical hops away peers. The minimum spanning tree is then constructed from the graph. The range of the file ID space of the direct neighbour peer is then split between the direct neighbour and the joining one. MA-SP2P is an extension of E-SP2P. MA-SP2P is similar to its predecessor but eliminates the use of root peers. It distributes the ID space in a way that the part of the ID space at peer peer (which can be non-contiguous) has to be consecutive with the peer's immediate connected neighbours. The upper end of the ID space at the peer points at the neighbour that has the part of the ID space that has the next greater IDs. The lower end of the peer's ID space part points to the peer's neighbour that has the ID space part with the next lower IDs. 3DO [20] is another system that builds a minimum spanning tree to understand the physical topology. It is based to work on OLSR as the underlay routing protocol. When calculating logical peers' IDs, It takes into account the relationships between neighbouring peers. Peers view the logical address space as a 3D rectangular coordinate system when computing logical IDs.

MADPastry [21], CrossROAD[22] and Ekta[23] have adopted Pastry to build P2P overlays in MANET. MADPastry integrates Pastry[24] with the reactive MANET protocol AODV. To consider local proximity, random land-marking is used where a set of nodes in the same physical cluster share a common overlay ID. CrossROAD [22] adopted Pastry over OLSR. Each CrossROAD node maintains a global services table that stores all the provided services. Ekta [23] integrates Pastry with DSR at the routing layer. Ekta uses an overhearing policy to update its information. When traffic passes through a node, it is used to update the node's routing information. Cramer and Fuhrmann [25] investigate the performance of Chord in MANET using different underlay routing protocols. Their conclusion was that the problem of deploying Chord over MANET is not the incurred overhead but Chord's pessimistic timeout and failover strategy. Castro et al[26] has investigated the performance of BAMBOO[27], a structured P2P algorithm based on Pastry, when deployed over multi-hop networks. They concluded that the aggressiveness of standard BAMBOO management in multi-hop networks causes network inconsistencies. MANETChordGNP[28] uses the Global Network Positioning (GNP) system to allow peers to detect their positions. It integrates modified Chord that uses GNP with AODV. Enhanced Backtracking Chord[29] modifies Chord to suit MANET by using retransmission and path selection.

## **2.2. MANET Routing**

Routing protocols for MANET can generally be divided into unicast, multicast and geocast approaches. For OneHopOverlay4MANET, the unicast routing protocols are of interest. Unicast routing approaches can be further divided into proactive, reactive and hybrid routing protocols [30][31]. Reactive routing protocols work on demand and discover a route to a destination as required. Once discovered, the route is maintained until no longer needed or becoming unavailable. On the other hand, in proactive routing approaches each node maintains routing information to the complete network in advance. In the following we provide a short description of the MANET routing protocols that were used with OneHopOverlay4MANET. A summary of these protocols is presented in Table 1.

### **2.2.1. OLSR**

Optimized Link State Routing OLSR[5] is a link state based proactive routing protocol. OLSR maintains routes to all other nodes in advance and make them available for use when needed. In order to maintain the routing table, OLSR diffuses partial link state information to all other nodes in the network. OLSR reduces the typical flooding of link state routing through the use of Multipoint Relays (MPR) which are used to retransmit control messages. Once a node has elected its MPRs, it notifies these nodes about being elected as MPRs. Each node maintains a list of who has chosen it as MPR. Such list is called Multipoint Relay Selector. The relaying node uses this list to decide whether to retransmit a received control message or not. In doing so, it only retransmits control messages if the sender was in its Multipoint Relay Selectors. OLSR uses hello and topology control messages which are sent regularly to construct routing tables. Topology control messages on the other hand contain partial link state information with the purpose of diffusing link state information in the network. Topology Control messages are generated by MPR and get forwarded only by MPR nodes to minimizing control overhead.

### **2.2.2. BATMAN**

Better Approach To Mobile Ad-hoc Networking BATMAN[6] is developed by Freifunk Community [32]. It relies on the distribution of the route knowledge among network nodes. Each node in a route only determines the best next hop to be used in order to reach the destination. BATMAN is designed to be a proactive underlay. However, it is rather different from link state routing and distance vector routing as it does not calculate or discover complete routing paths. Rather, the main strategy of BATMAN is to proactively maintain routing tables in the nodes that store the next hop to be used for every destination node. Every BATMAN node broadcast an originator message OGMs (like hello messages) regularly so it can be detected by its neighbours. Received messages will be rebroadcast by those neighbours. OGMs will be flooded repeatedly in the network until they get received by every node, or lost, or their Time To Live expires.

### **2.2.3. DSR**

The Dynamic Source Routing DSR[7] is a reactive routing protocol. It uses the source routing technique where the sender needs to include the complete route to the destination in the packet. DSR has two main operations, route discovery and route maintenance.

In route discovery, when a node wants to send data to another node, it broadcasts RREQ messages to its neighbours. Nodes that receive RREQ messages check their cache for a route to the desired destination. In the case no route is found, it will retransmit the RREQ message adding itself to the record route. If the destination node is found (or a cached route in an intermediate node), a RREP message will be issued. The target node checks its cache to find a route to the initiator of the route request. In the case, there is no route back to the originator; the node can initiate a route request to the originator of the initial route request piggybacking the route reply onto the message. In addition, the recorded route in the Route request can be reversed and used as source route for the route reply if bidirectional communication is in place. The source node will cache the fetched route to use it in the future. In route maintenance, if a link was found to be broken, the cache entry to that node will be removed and a RERR message will be sent to the source to allow it to update its cache

#### 2.2.4. AODV

Ad hoc On-demand Distance Vector AODV[8] is a reactive routing protocol. Unlike DSR, AODV does not use source routing. When a node requires a route to a destination node, it starts the discovery process by broadcasting a RREQ message. The source node includes some parameters in the RREQ message, such as the sequence number of the destination and a unique RREQ ID. Upon receiving a RREQ message, an AODV node checks its routing table if it knows of any fresh routes to the destination. If there is no valid route to the destination, it will broadcast the RREQ message to its neighbours. During processing a RREQ message, nodes store

a route back to the initiator of the RREQ. When the RREQ message reaches a node that knows a valid route to the destination or is the destination itself, a RREP message is generated and sent as unicast to the source node. Hello messages are used to detect neighbours' connectivity when maintaining active routes. Each node that participates in an active route sends Hello messages periodically to its neighbours. In addition, each node expects to receive frequent Hello message from its neighbouring nodes. If Hello messages stop arriving from a neighbour, the node assumes a link failure to the corresponding neighbour has occurred. A RERR message is then used to distribute the topology changes.

#### 2.2.5. DYMO

Dynamic MANET On-demand DYMO[9] is a reactive routing protocol that extends the functionalities of AODV. It constructs unicast routes between DYMO nodes in an on demand fashion. Like AODV, DYMO does not employ source routing. DYMO uses two main operations for routing: Route Discovery and Route Management.

In route discovery, DYMO nodes disseminate route request message to find a route. When an intermediate node receives a route request, it records the route to the originator of the route request message, and then appends information about itself to the request and forwards it to its neighbours. In addition to the information about the requested destination, each node on the path of the route request will receive information about all intermediate nodes. Once the target destination receives the route request, it replies with a unicast route reply message using routing information from the received route request. In route management, a DYMO node updates the timeout of its stored routes every time a packet is forwarded successfully. A DYMO node monitors its links over which data is transmitted. If it receives a packet to forward to a destination to which it has no route, it sends a route error to the source node.

TABLE 1: Comparison of the used MANET underlay protocols

Protocol	Route availability	Use of flooding	Route metrics	Source route (complete route)	Messages	Multiple route
OLSR	In advance	Yes through MPR only	Shortest Path	No	Hello message, TC message	No
BATMAN	In advance	Yes	Best next hop	No	Originator message OGM	No
DSR	On demand	No	Shortest path	Yes	RREQ, RREP, RERR	Yes
AODV	On demand	No	Freshest route or shortest path	No	RREQ, RREP, RERR	No
DYMO	On demand	No	Shortest path	No	RREQ, RREP, RERR	No

### 3. ONEHOPOVERLAY4MANET

#### 3.1. Overview

Unlike previous approaches, OneHopOverlay4MANET combines a one-hop structured P2P overlay (EpiChord) with MANET protocols using cross-layering. The use of cross-layering reduces the typical overhead traffic of P2P protocols where the underlay routing information is used to build the

overlay. Thus maintenance traffic overhead at the P2P overlay is greatly reduced. Figure 3 depicts the architecture of the proposed system. At any given time each OneHopOverlay4MANET peer should maintain logical routing information to every peer in the system. This will enable peers to fetch keys in one logical step. This would yield a routing complexity of  $O(1)$  for the overlay routing. As our experimentation (see Section 4) confirms, such an approach is well feasible with a MANET network. As our simulation results confirm, the incurred overhead in such a system is manageable, mainly due to the cross-layering approach being adopted.

A OneHopOverlay4MANET node assigns itself a unique identifier by hashing its own IP address. A cryptographic hash function like SHA-1[12] ensures that there will not be collision of peers ID. A one dimensional circular address space is used in OneHopOverlay4MANET as the case in EpiChord and Chord. A key is stored on the node that most closely follows the key which is known as the successor (Figure 4). In a similar way to EpiChord and Chord, OneHopOverlay4MANET maintains lists of neighbouring nodes that succeed and precede a node. OneHopOverlay4MANET maintains four successor peers in the successor list and four predecessor peers in the predecessor list to guarantee consistency of the network. In addition, both lists will help the performance of OneHopOverlay4MANET in the rare cases when it fails to resolve a key lookup in the first logical hop. OneHopOverlay4MANET also maintains a cache that stores information on all peers in the network. The main source of information for populating this cache is the use of available routing information from the underlay protocol through cross-layering.

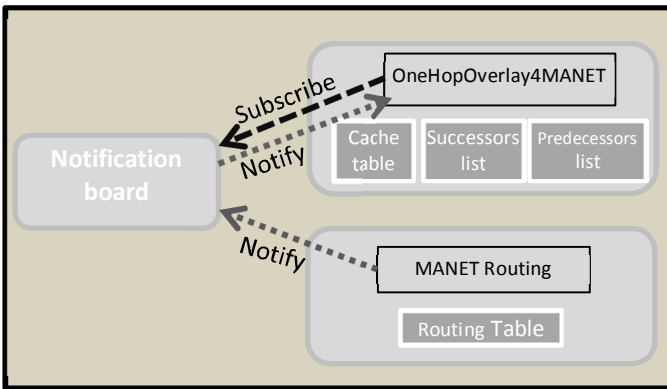


Fig. 3 OneHopOverlay4MANET System Overview

### 3.2. Cross Layering

OneHopOverlay4MANET adopts an approach of feeding routing information to the overlay from the MANET layer through cross-layering as the main strategy to learn routing entries. Each peer updates its successor list, predecessor list and cache table by using information from the MANET protocol. This approach reduces the typical overhead from employing one hop overlay systems as routing updates from underlay are forwarded to overlay which in turn can scale down its own update mechanisms. As a result of optimizing the underlay routing information, OneHopOverlay4MANET can build logical routing tables that enable it to solve a key

lookup in a single logical hop. For the cross-layering approach, OneHopOverlay4MANET follows the manager based method of cross-layering [33] to optimize the network layer's routing information transfer. The manager method stipulates that a channel can be created allowing to share data between some or all of the layers in the protocol stack. With this approach, no changes are required to the structure of the protocol stack. However, the functions of the protocols need to be adapted to allow passing shared information through the deployed channel. As is shown in Figure 3, OneHopOverlay4MANET implements sharing information between the Application Layer (where the proposed system OneHopOverlay4MANET resides) and the Network Layer (where MANET routing protocol operate).

A notification board is used as the cross-layering channel that manages sharing the information between the application and the network layers. During setup, OneHopOverlay4MANET peers need to subscribe to the notification board to receive notifications of changes that occur at the MANET layer.

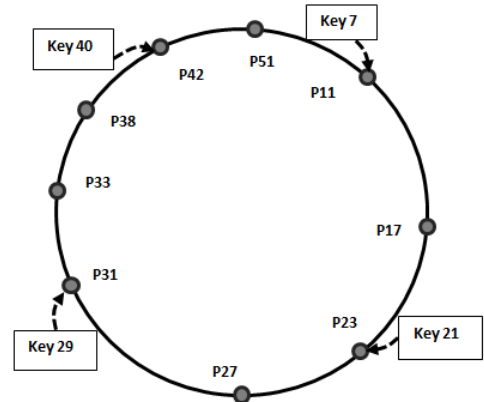


Fig. 4 Circular address space

During operation, the underlay routing protocol notifies the notification board whenever changes occur in its routing table. The notification board passes the updates to the overlay. The overlay then uses this information to update its view of the network. Notifications contain information from of the routing table of the used MANET protocol. Thus adopted MANET protocols need to be modified to enable it to send its routing table information updates to the notification board. The passed information is the IP addresses of the nodes that a node knows about. This includes information on nodes that can be reached in a single physical hop and nodes that can be reached in multiple physical hops.

### 3.3. Joining the Overlay

In order to be part of the overlay, OneHopOverlay4MANET peers needs to join the overlay. In our system, we consider two methods for joining the network when mobile nodes first arrive. These two methods are:

- Joining the network through the use of join messages which are typically used by P2P systems.
- Joining the network through optimizing underlay routing notification of existing peers in the network.

The use of the underlay information can be sufficient to instantiate the overlay. This is possible provided that the used protocol at the underlay is a proactive routing protocol. Proactivity of protocols means that it builds and populates nodes' routing tables without other nodes issuing some form of request messages to trigger routing table updates. Consequently, the overlay gets information from the underlay updates through the cross layer channel.

However, this approach is not possible when combining OneHopOverlay4MANET with reactive underlays which update routing information on demand. Such a combination would result in having a reactive overlay with a reactive underlay. Both layers will wait for a request from the other. This requires the use of joining messages propagated by the overlay.

Join messages are used by the OneHopOverlay4MANET overlay with two purposes. Firstly, they allow peers to join the overlay network. When a peer sends a join message, it will receive a response with a copy of its immediate neighbor's cache. Join messages also generate demand on a reactive underlay to form links and update routing information. Once a reactive underlay started, its routing tables will grow gradually. Through the cross-layering mechanism, OneHopOverlay4MANET will then be updated with routing information updates and build up its knowledge about other peers.

When a joining peer receives a join response or notification from the underlay, it will hash each received Id and store it along with the current time. OneHopOverlay4MANET uses the received routing information to populate the following list:

- Successor List: contains the four immediate peers that follow this peer in the circular address space.
- Predecessor List: contains the four peers that are immediately preceding this peer in the circular address space.
- Cache Table: stores peers that this node knows about, but which are not included in the other two lists.

(a) shows a mobile network topology that consists of 14 nodes. The network has been operating for a while and each mobile node already knows routes to a number of other nodes. The corresponding logical address space is presented in (c). The IDs depicted next to nodes inside the circle are the underlay IDs. The corresponding logical IDs (peer ID) are those shown outside the circular logical space. The underlay routing table for Node 3 and the logical routing table (successor list, predecessor list and cache) for the corresponding overlay peer are shown in (b) and (d) respectively.

Mobile Node 14 is a new node that wants to join the network. In this example, we assume Node 3 with logical ID 1003 is the bootstrapping peer. In order to join the overlay, node 14 with logical ID 1014 needs to contact Peer 1003. As Node 14 just started, its routing table is still empty. Therefore, it sends a Route Request message to discover a route to node 3. Once the route to Node 3 is known, Node 14 sends the Join message towards Node 3 along the discovered route.

According to the deployed cross-layering method, the underlay routing protocol sends a notification of its routing table contents to the logical overlay. The underlay routing agent will use the cross-layer channel to convey this notification. Upon receiving the notification, Peer 1014 will place each entry in the relevant logical table (successor list, predecessor list or the cache). In addition, Peer 1003 will respond to the Join message with its content of the successor list, predecessor list and the cache. The response message will be sent to Peer 1014 using the MANET path. Note, however, that this message does not make use of the cross-layer communication, as the message exchange is between two overlay peers. Finally, Peer 1014 stores the received table entries in the relevant tables. As a result, Peer 1014 starts with an up-to-date view of the network. As the network lives on, the underlay will learn of new and updated routes due to route discovery processes which are initiated by overlay requests. As a consequence, the overlay gets notified of these new paths and can build up a better view of the network.

Figure 6 shows the physical topology of the network (a) and its corresponding logical overlay (c) alongside with the Node 14 underlay routing table (b) and its corresponding logical routing tables (Peer 1014) (d). The contents of the logical routing tables are learned from the notification board and the response to the Join request. The entries for Node 6 and Node 3 were learned through the underlay notification and passed up to the overlay.

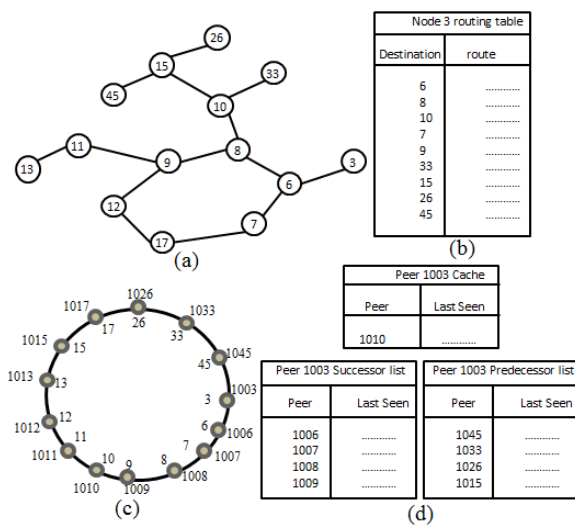


Fig. 5 Joining process with a reactive underlay.

In order to illustrate the joining process with a reactive underlay routing, we present the following example. Figure 5

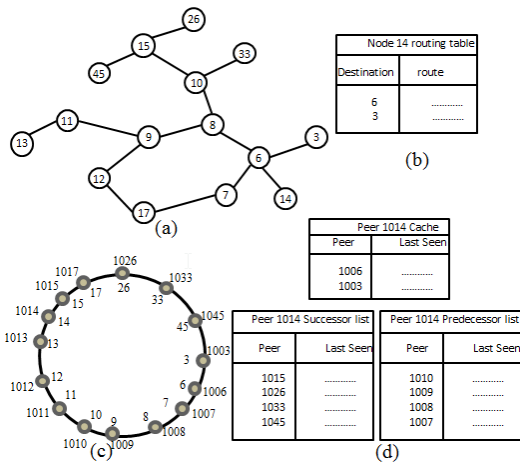


Fig. 6 The network after Node 14 joined the overlay.

The rest of the entries were learned via the response to the joining message where peer 1003 replies with the contents of its cache, successor list and predecessor list. As can be seen from the figure, the logical routing tables for Peer 1014 do not exactly reflect the actual logical neighbours at the beginning. However, by learning new table entries from the underlay, the logical view of Peer 1014 will improve with more information becoming available to it.

Unlike with reactive underlays, when the underlay is proactive, OneHopOverlay4MANET does not need to issue Join messages to join the overlay. Instead the underlay routing entries can be passed up to the logical overlay at the same node through the established cross-layer channel to allow the participating peer to build an initial view of the logical overlay.

In order to illustrate the joining process for OneHopOverlay4MANET over a proactive underlay we use the example depicted in Figure 7. This shows the MANET topology (a), its overlay (c), the underlay routing table for node 51 (b) and the overlay routing tables for the same node with peer ID 10051. As before, we assume the network has been running for a while and each node has built up its routing table. As the used underlay routing protocol is proactive, Node 51 has an entry in its routing table for every mobile node in the network. Consequently, the overlay routing tables of Peer 10051 are populated with the information received from the underlay (see Figure 7 (d)). Node 56 is assumed to be a new node that wishes to join the overlay. In doing so, Node 56 will communicate with the nodes in the network and start building its own routing table.

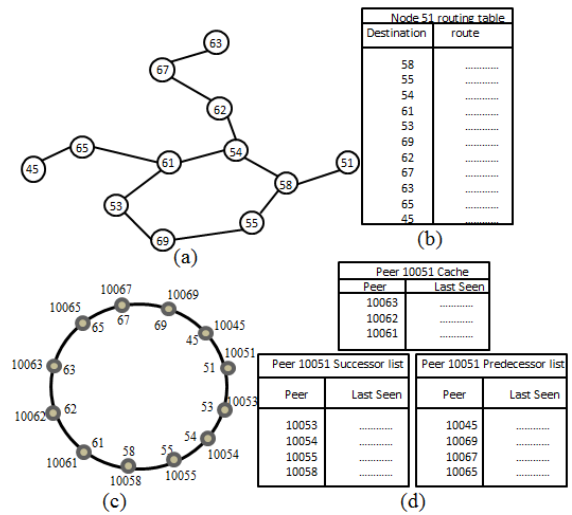


Fig. 7 Joining process over proactive underlay

In this case, logical peer 1056 will not send a Join message. It waits to get the information from the underlay. Once node 56 populates its routing table, it will immediately pass the routing entries up toward the overlay peer via the cross-layer channel. On receiving the update from the underlay, peer 10056 calculates its logical successors, predecessors. Information on any other nodes will be placed in its cache. Figure 8 shows the network after the joining process for this node is complete. The routing table for Node 56 is presented in (b) and the logical tables for the peer are shown in (d). The information of having added the new peer will be propagated to the rest of the overlay peers as other peers' underlay nodes discover the existence of node 56. Upon the discovery of the new node, each MANET node will notify its peer of the new node which in turn recalculates its logical routing tables to include the additional peer. Furthermore, the new participating peer can be also discovered by peers when a node replies to a key lookup request. For example, when a peer, which has already discovered the new peer, receives a key lookup request it may include information on the newly joined peer in its reply to the key lookup request. Both of these processes will occur in parallel in the network.

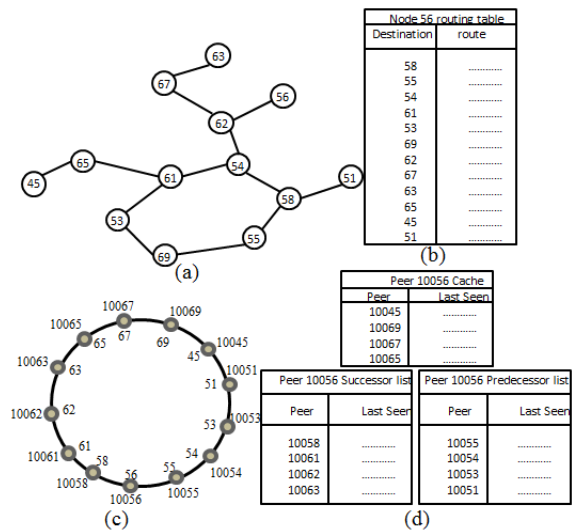


Fig. 8 Node 56 after joining the overlay

### 3.4. Lookups

EpiChord supports parallel lookups to increase its chance of finding the key with the first hop reducing lookup latency. In addition, the parallelism is also used to help the system to learn new routing table entries and thus to update the cache. However, OneHopOverlay4MANET does not make use of this technique and only sends lookup requests to single destination. The main reason for sending single lookup requests is to reduce unnecessary network load in a resource constraint MANET network. Since OneHopOverlay4MANET receives routing updates through its use of cross-layering from the MANET layer, its overlay routing tables are well populated and consequently, as we show in Section 4, it achieves lookup success within a single hop without parallel lookups. Thus, there is no need for parallel lookup requests.

To locate a file, a OneHopOverlay4MANET peer hashes the value of the file and then consults its routing table to find the best logical Id that follows this key. As a result of having up-to-date logical routing tables, OneHopOverlay4MANET can usually find the peer which is responsible for the queried key with the first logical hop. OneHopOverlay4MANET then sends the lookup query to that peer which in turns replies with location of the queried file on the overlay. Similar to EpiChord, the used lookup algorithm with OneHopOverlay4MANET uses an iterative approach where the queried peer will respond with its best knowledge of the queried key without forwarding the lookup to other nodes. When a peer receives a lookup request, its answer will be one of the following:

- If it is the immediate successor of the key, it responds with the value of the key and information about its successor and predecessor.
- If it is a predecessor of the looked up key, it responds with information about its own successor, information about the node succeeding and the two nodes preceding the best node that may hold the key.
- If it is a successor of the looked up key, it will respond with information about its own predecessor, and information about the node succeeding and the two nodes preceding the best node that holds the key.

Clearly, each logical hop in the overlay is translated to a physical path in the underlay that may consist of number of physical hops. Failing to solve a lookup with the first logical hop results in OneHopOverlay4MANET following distributed hash table routing and going through the circular address space until it finds the requested key. This significantly increases the traffic in the network and may result in network criss-crossing at the MANET layer. Thus OneHopOverlay4MANET strives to reach a destination in a single logical hop avoiding inefficient routing.

### 3.5. Management

Clearly, to a large part, content updates to a peer's logical cache, successor list and predecessor list depend on updates received from the underlay and thus on the used underlay routing protocol. Each entry has an associated timer and gets

flushed from the cache when the timer expired. When the underlay routing table gets updated or gets new entries, it sends an update to the notification board with the contents of its routing table. The notification board in turn passes the notifications to the corresponding peer. Once a peer receives the update, it hashes the IP addresses to get their corresponding logical ID. They are then added to its tables if they are not already present, in which case only the time to live value will be updated.

For each received entry, OneHopOverlay4MANET checks if the logical ID of the received peer falls between itself and the last entry of its successor list. If it does so, that means the new peer should be placed in the successor list. OneHopOverlay4MANET then recalculates the new successor list to include the new entry. The same procedure is carried out for the predecessor list. If the new node falls outside both lists, it gets stored in the cache. As a result of maintaining the logical address space by each peer, the typical  $O(\log N)$  lookup performance is guaranteed for overlay routing. Stabilization mechanisms as are used by EpiChord to maintain the consistent view of the overlay are not employed by OneHopOverlay4MANET as its routing tables are highly up-to-date from the underlay updates. Thus the stabilization mechanism is unnecessary on our system and would only increase the maintenance traffic.

EpiChord divides the logical address space into small slices as a way to keep the overly consistent. Each peer has to maintain certain number of entry per each slice. Again, OneHopOverlay4MANET does not employ this mechanism due to its accurate routing tables based on underlay updates. Again, this reduces overheads significantly.

## 4. PERFORMANCE EVALUATION USING DIFFERENT UNDERLAY PROTOCOLS

For the experimentation, we implemented OneHopOverlay4MANET in a network simulator which includes a packet level simulator together with implementations of the MANET and overlay protocols. This allowed us to configure and parameterize all aspects of OneHopOverlay4MANET, be at the network layer for the MANET protocols or the application layer for the P2P overlay algorithms. We used the discrete event simulation system OMNet++[34], the communication network simulation package INET-MANET[35] together with Oversim [36] for the P2P overlay model. As the current Oversim framework is built on an older version of INET framework we had to link a newer version of INET (INETMANET) to Oversim that provides additional support for mobile ad hoc network protocols.

### 4.1. Simulation Setup and Performance Metrics

Table 2 details the parameters for the simulated scenarios. All simulated scenarios were repeated ten times and the results shown in the graphs are the averages of the repetitions. The used mobility model is Random Way Point Model which is commonly used for simulating ad hoc networks. For each of the simulated scenarios, the network is given about 60 seconds to stabilize. After the 60 seconds, the measurements start to be taken. Lookups for file Ids were introduced using two frequencies: 60s and 10s.



TABLE 2. SIMULATION CONFIGURATIONS

Simulator	OMNeT++
Underlay routing protocol	OLSR, BATMAN, DSR, AODV, DYMO.
Topology size	1000m x 1000m
Propagation model	Two ray ground
Number of mobile nodes	40, 60, 80, 100, 120, 140.
Mobility model	Random way point
Node speed	1m/s, 2m/s, 3m/s, 4m/s, 5m/s.
Measurement time	1000 seconds
Transmission range	250 m
Network stabilization time	60 seconds
Simulation repetitions	10
MAC Layer	IEEE 802.11
Bandwidth	2MB

OneHopOverlay4MANET was evaluated using different underlay routing protocols (OLSR, BATMAN, DSR, AODV and DYMO). Cross-layering was implemented with each underlay protocol. We investigated the lookup success ratio, file discovery delay and traffic load of OneHopOverlay4MANET with different configuration scenarios of the network. The size of the network was increased up to 140 nodes and node mobility varied from 1 m/s to 5 m/s.

The following performance metrics are evaluated from the conducted simulations:

- Lookup success ratio: the ratio of the number of resolved file lookup queries to the total number of initiated file lookup queries.
- Average File discovery delay: the average amount of time that was required in order to solve a lookup query. It starts from sending the key lookup until the time when a peer received the answer to that lookup. This reflects the ability of the system to retrieve a shared key on the network.
- Hop Counts: the average number of logical hops that were required in order to solve a key lookup.
- Network Traffic Load: The total number of packets transmitted from the routing layer in the network over the period of simulation.

#### 4.2. Experimental results

We evaluate the performance of OneHopOverlay4MANET over five different underlay routing protocols. Figures 9 and 10 depict the performance of OneHopOverlay4MANET with a node speed of 1 m/s and 3m/s, respectively, for networks ranging from 40 to 140 nodes. File lookups were issued every 60 seconds.

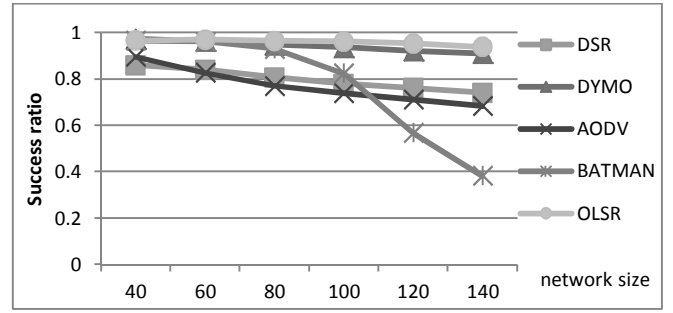


Fig. 9 1m/s Speed, 60s Lookup Frequency

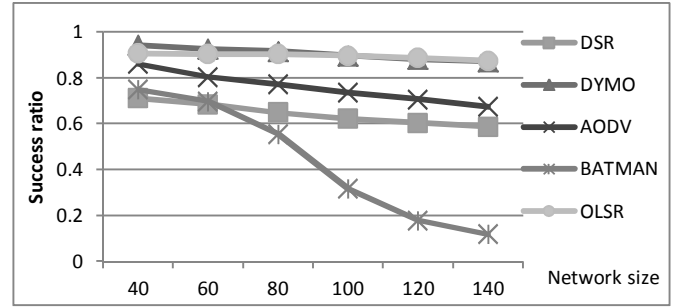


Fig. 10 3m/s Speed, 60s Lookup Frequency

In a slow moving network, OneHopOverlay4MANET achieves success ratio +90% when OneHopOverlay4MANET is used with OLSR or DYMO. However, when used with BATMAN the performance deteriorates drastically beyond a network size of 80 nodes. DSR and AODV achieve a performance of about 70%. As nodes move at a faster speed (Fig. 9) the performance decreases somewhat to about 80% success rate for OLSR and DYMO and 60% for AODV and DSR. The performance with BATMAN drops to below 20% success rate of lookups which is virtually unusable.

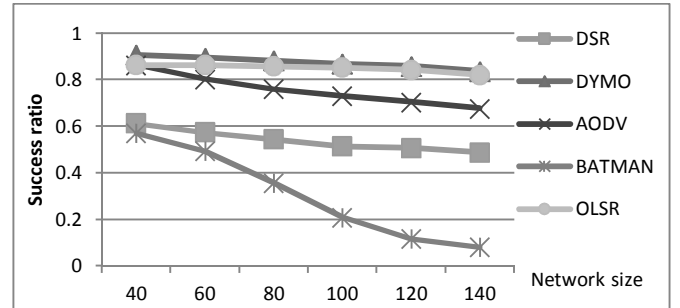
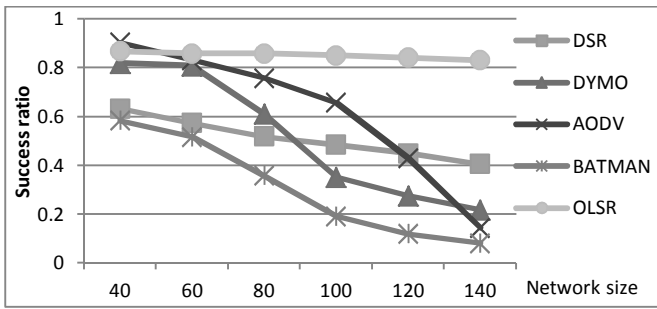


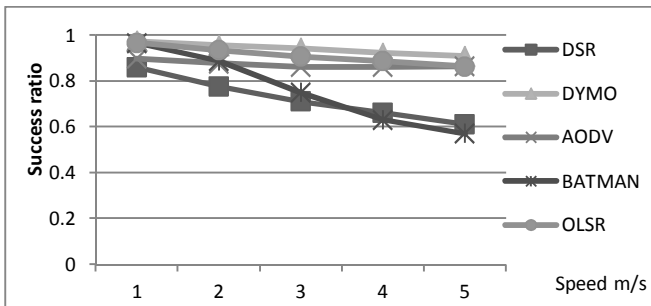
Fig. 11 5m/s Speed, 60s Lookup Frequency

Increasing the node speed to 5m/s leads to a further slight decrease in lookup performance when using DSR (Fig. 11). The systems based on OLSR, DYMO and AODV largely maintain their performance levels. Clearly, higher node mobility causes changes in the network topology. Hence, additional route maintenance traffic will be generated to solve overlay lookups. Furthermore, an increasing level of packet loss will occur resulting in poorer success ratio.



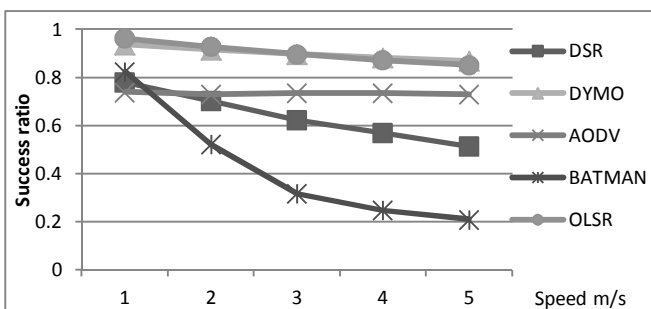
**Fig. 12** 5m/s Speed, 10s Lookup Frequency

Additionally, when lookups are being sent more frequently (every 10s, Fig. 12), the performance of OneHopOverlay4MANET decreases for all underlays except for the case when it is used with OLSR. OneHopOverlay4MANET over OLSR maintains a success rate of about 85%. The reason for the deterioration of OneHopOverlay4MANET performance over DYMO and AODV is that they are reactive protocols and update their routes on demand rather than actively.



**Fig. 13** 40 nodes, 60s Lookup Frequency

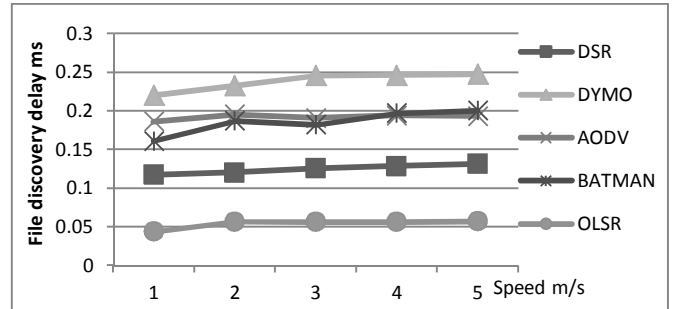
With a higher demand from the overlay (sending more frequent file lookups), an increased number of routes discover requests will be incurred. This is especially true in the context of high node mobility which leads to routes breaking. The resulting route error messages and route discovery traffic causes the network to overload and leads to poor performance.



**Fig. 14** 100 nodes, 60s Lookup Frequency.

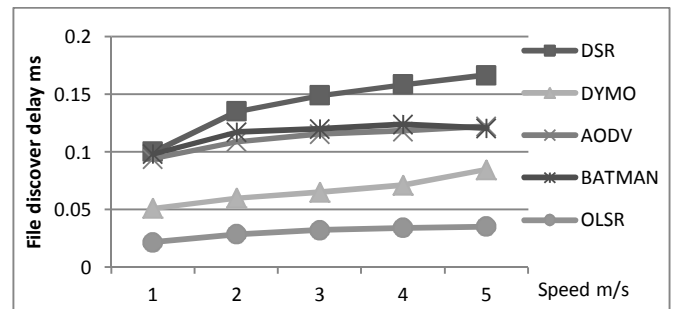
Figures 13 and 14 show the impact of node mobility on the performance of the system on networks consisting of 40 and 100 nodes with a 60s lookup interval. As can be seen, OLSR and DYMO based OneHopOverlay4MANET can maintain its high performance. For the smaller network, the AODV based system also maintains a competitive performance. However, for the larger network, the performance of the system based on AODV drastically deteriorates. When OneHopOverlay4MANET was designed, it was expected to

work best with proactive underlays. However, the results for the BATMAN based system disappoint. The performance of OneHopOverlay4MANET over BATMAN deteriorates with increasing node mobility and provides the worst performance amongst the tested systems overall. The very poor performance is due to the flooding approach that BATMAN employs. Every BATMAN node frequently broadcasts originator messages (like hello message) to allow other nodes to calculate the best next hop to every single destination in the network. Those packets are rebroadcasted until they have been received by the whole network. Consequently, with an increasing network size or node mobility speed, a large number of originator messages are flooded in the network. This leads to collisions and inefficient routing.

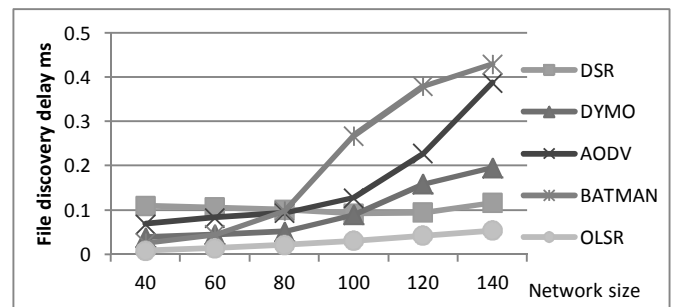


**Fig. 15** 80 nodes, 60s Lookup Frequency

On the other hand, OLSR as the other proactive underlay protocol which has been used with OneHopOverlay4MANET is showing a very strong performance, often the best performance in the experimentation. As is shown in Figure 12, the OLSR based system achieves a lookup success ratio of over 80%. The reason for the strong performance of the OLSR based system is its multipoint relay strategy avoiding flooding the network. OLSR uses a set of selected nodes to retransmit control messages.



**Fig. 16** 80 nodes, 10s Lookup Frequency



**Fig. 17** 1m/s Speed, 10s Lookup Frequency

Figures 15-17 depict the lookup latency in milliseconds for OneHopOverlay4MANET using various MANET routing protocols. Figure 15 and Figure 16 show that the OLSR based system achieves the lowest latency. It takes about twice as long for the next best system (DSR based) when lookups are sent every 60s (Figure 15). The latency is positively affected by an increase in the lookup frequency. The reason for the improved performance is that the additional lookups lead to the underlay routing tables being more up-to-date. This is especially true for reactive underlay based systems. In other words, the routing tables will have more fresh routes to other nodes since there is more demand from the overlay. Consequently, less time will be required to discover a key (Figures 15 and 16).

Both the network size and the node speed affect the lookup latency. Figure 17 shows how the latency increases when the size of the network increases. For networks with less than 80 nodes, the lookup latency is less than 100 milliseconds. However, as the network size grows the latency of the BATMAN and AODV based systems deteriorate significantly.

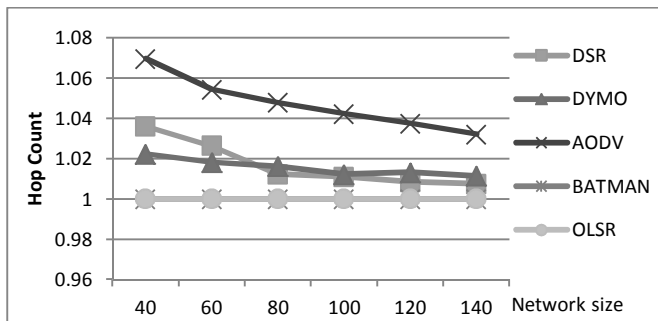


Fig. 18 3m/s Speed, 60s Lookup Frequency

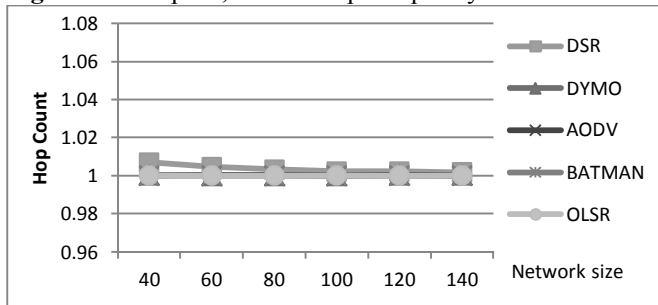


Fig.19 3m/s Speed, 10s Lookup Frequency

The average number of logical hops that were required in order to solve key lookups is presented in Figures 18 and 19. As shown, OneHopOverlay4MANET require almost one overlay hop to retrieve an object from the network when combined over various underlay routing protocols. Both the network size and lookup intensity have positive effect on the average hop count. When the number of mobile nodes increases, more lookups are sent in the network and more entries in underlay routing tables are achieved. This means that overlay routing tables gets more updates through the underlay and the responses messages of lookups. Therefore, better logical hop count is achieved. Similarly, the hop count gets closer to one when lookups are sent more frequent as Figure 19 shows.

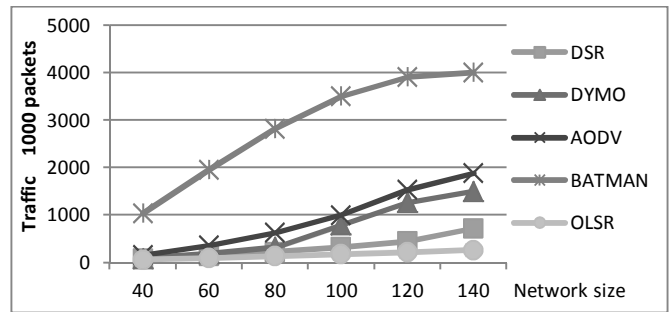


Fig. 20 1m/s Speed, 10s Lookup Frequency

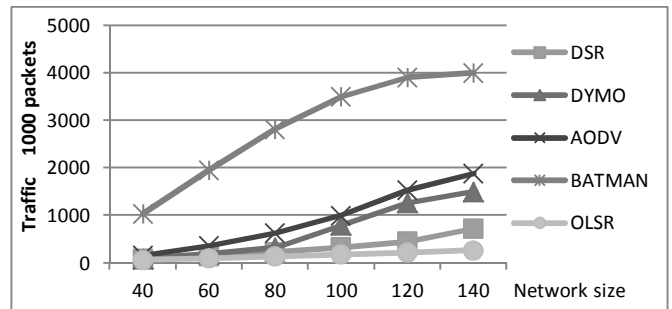


Fig. 21 5m/s Speed, 60s Lookup Frequency

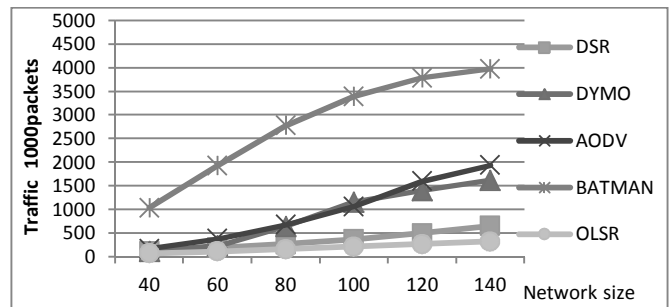


Fig. 22 5m/s Speed, 10s Lookup Frequency

Figures 20, 21 and 22 depict total amount of traffic that was generated in the network. This traffic includes both MANET and overlay traffic. The traffic with most of the underlay protocols is less than 500k packets with a lookup interval of 60s (see Figure 21). However, with a higher lookup frequency (every 10s), the only protocol that maintains about the same amount traffic is OLSR (see Figure 22). When deploying OneHopOverlay4MANET, DYMO is the most competitive protocol to OLSR with file lookup frequency of 60s. However, at a lookup frequency of 10s, DYMO's performance deteriorates and its total traffic volume reaches as much as three times that of OLSR in a network with 140 nodes (see Figures 21 and 22). On the other hand, the BATMAN based system generates the highest amount of traffic as a result of its flooding mechanism. As a result of omitting the use of Hello messages, DYMO generates less traffic than its predecessor AODV.

## 5. COMPARATIVE EVALUATION WITH ANOTHER SYSTEM

As the results in the previous section show, OneHopOverlay4MANET achieves its best performance when combined with OLSR. In this section, we carry out a comparative performance evaluation of OneHopOverlay4MANET over OLSR against two recently

published structured P2P over MANET (MA-SP2P [5] and E-SP2P [6]) that adopted OLSR as the underlay routing protocol.

### 5.1. Simulation Setup and Performance Metrics

The used parameters are listed in Table 3. The presented results are averages over 10 repetitions. Random churn is used with nodes randomly joining and leaving the network. Lookups were randomly initiated for 100 random keys. As in Section 4, lookup latency and traffic load were metrics of interest. In addition, we monitor (Fail Ratio) the ratio of the number of unresolved file lookups for files that exist in the network to the total initiated file lookups. The peers ratios in the network (number of peers in the overlay to the total number of mobile nodes in the network) were 10%, 20%, 30%, 40% and 50% of the overall network size (100 nodes). These parameters were chosen to match the parameters used in [5] and [6] and thus our results are comparable

### 5.2. Experimental Results

The performance of OneHopOverlay4MANET was compared to MASP2P and E-SP2P. The authors of MA-SP2P and E-SP2P have previously presented that their proposal outperforms Modified Chord[37] and P2P-WANT[38]. For clarity in the graphs we do not reproduce these results here.

TABLE 3: SIMULATION CONFIGURATION

Underlay protocol	OLSR
Topology size	1000m x 1000m
Propagation model	Two ray ground
Number of nodes	100
Peer ratio	10%, 20%, 30%, 40%, 50%.
Mobility model	Random way point
Node speed	0.4m/s, 0.8m/s, 1.2m/s, 1.6m/s
Measurement time	1000 seconds
Transmission range	250 m
Network stabilization	60 seconds
Simulation repetitions	10
MAC Layer	IEEE 802.11
Bandwidth	2MB
OLSR Hello Interval	3 seconds
OLSR Topology Control Interval	6 seconds.

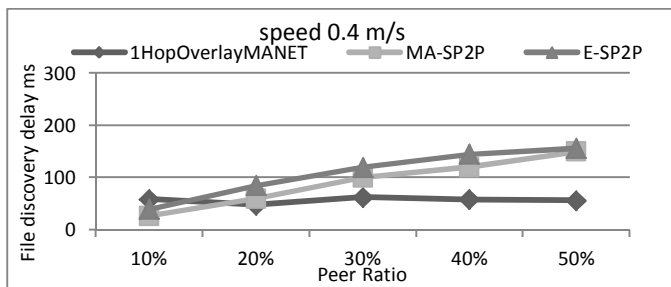


Fig. 23 (a) File discovery delay at 0.4 m/s node speed

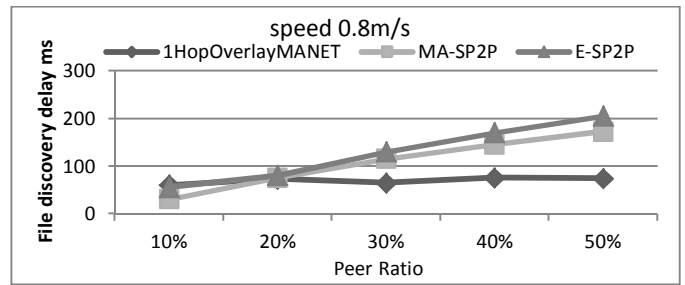


Fig. 23 (b) File discovery delay at 0.8 m/s node speed

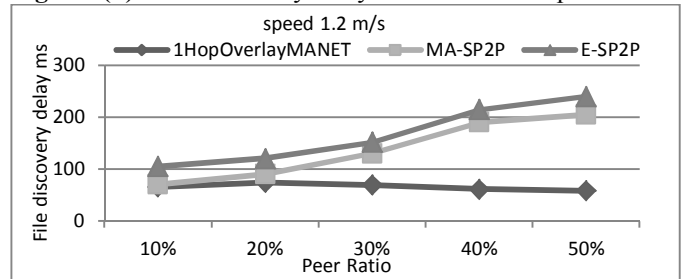


Fig. 23 (c) File discovery delay at 1.2 m/s node speed

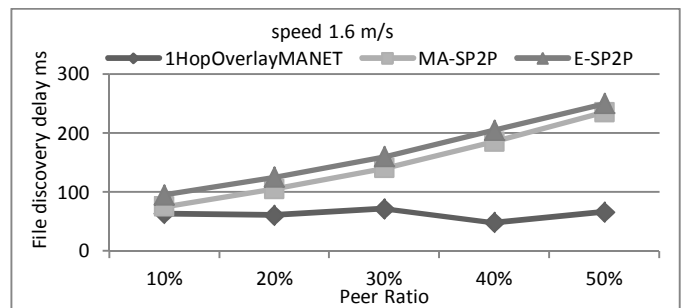


Fig. 23 (d) File discovery delay at 1.6 m/s node speeds

The key discovery delay for all the systems with different peer ratio over different speed are shown in Figures 23a-d. OneHopOverlay4MANET exhibits a latency of about 70ms across all four scenarios even when the ratio of participating peers increases to 50%. In contrast, the latency for E-SP2P and MA-SP2P for 50% peer ratio increases markedly to about three or four times the latency experienced at 10% peer ratio. For all simulated scenarios across different speeds and peer ratios, OneHopOverlay4MANET manages to achieve an equal or better latency than the other two systems. Albeit to a lesser extent than peer ratio, the performance of both E-SP2P and MA-SP2P is affected by increased node velocity. As the Figures show, the discovery delay with 50% peer ratio network at 0.4m/s node mobility of about 150ms increases to about 250ms for 50% peer ratio at 1.6 m/s node mobility.

OneHopOverlay4MANET achieves better results in term of lookup latency when compared to MA-SP2P and E-SP2P. The reason for the improved performance is that file lookups are almost completely resolved in a single hop. Since the overlay can route a look up to a destination in a single logical hop, less time is required to find the queried object. On the other hand, MA-SP2P and E-SP2P need multiple overlay hops resulting in an inefficient underlay lookup path.

The generated traffic by all three systems is shown in Figures 24a-d. As a consequence of optimizing the underlay routing information, the generated traffic by OneHopOverlay4MANET is the lowest amount across all the different scenarios. Across an increasing node velocity,

OneHopOverlay4MANET traffic varies only slightly, with about 75k at 0.4m/s up to about 90k at 1.6m/s being required with 50% peers. This is a result of exploiting the synergies between the underlay and overlay allowing OneHopOverlay4MANET to cut the P2P management traffic. As Figures 24a-d show, MA-SP2P and E-SP2P traffic increases with larger overlay sizes. A reason for this is that when increasing the peer ratio, more traffic is required to maintain consistency of the logical overlay. Moreover, with increasing velocity, an increasing load will be incurred in the networks. Nonetheless, the generated extra traffic by OneHopOverlay4MANET is at a much lower level than the

additional traffic required by MA-SP2P and E-SP2P. MA-SP2P and E-SP2P generate more than 140k at 50% ratio and 0.4 m/s speed, and about 200k at 50% node ratio and 1.6 m/s speed. This is about double what OneHopOverlay4MANET needs. Clearly, OneHopOverlay4MANET can handle the required logical routing updates which are required due to increased speed much better than E-SP2P and MA-SP2P. The reason for this is that with MA-SP2P and E-SP2P peers may move away from their logical neighbour peers. As a consequence, peers need to maintaining minimum spanning trees and form new relationships with new neighbouring peers. Hence, more traffic will be introduced.

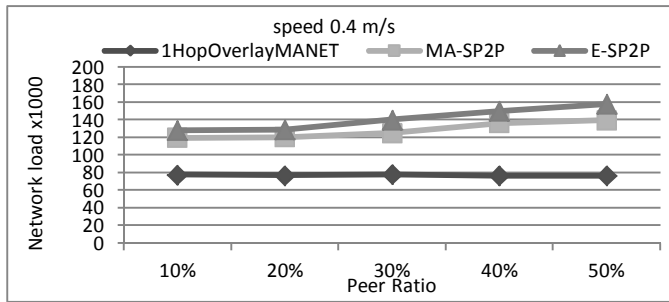


Fig. 24 (a) Network Load at 0.4m/s node speed

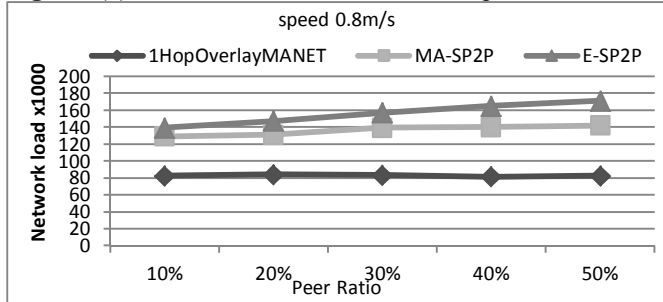


Fig. 24 (b) Network Load at 0.8m/s node speed

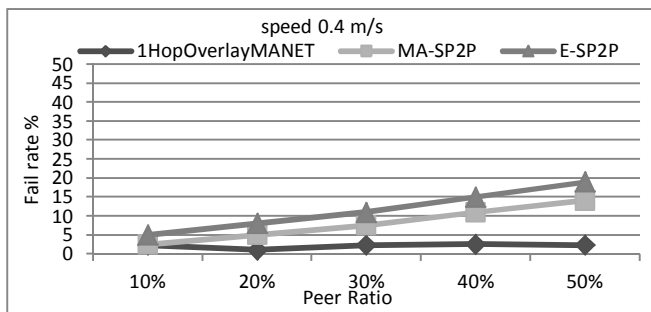


Fig. 25 (a) Fail rate at 0.4 m/s node speed

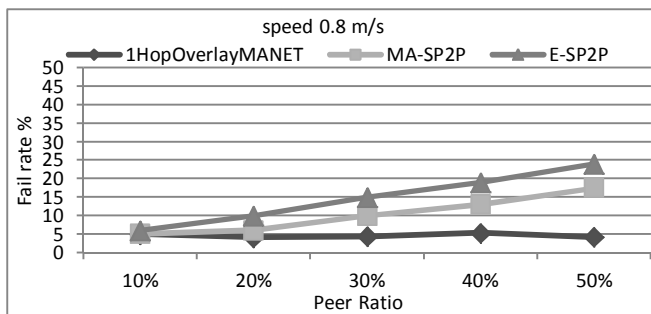


Fig. 25 (b) Fail rate at 0.8 m/s node speed

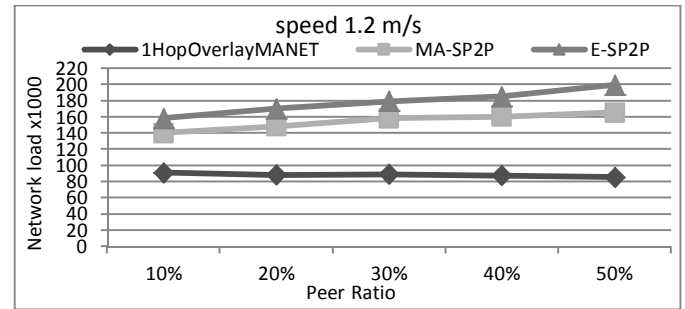


Fig. 24 (c) Network Load at 1.2m/s node speed

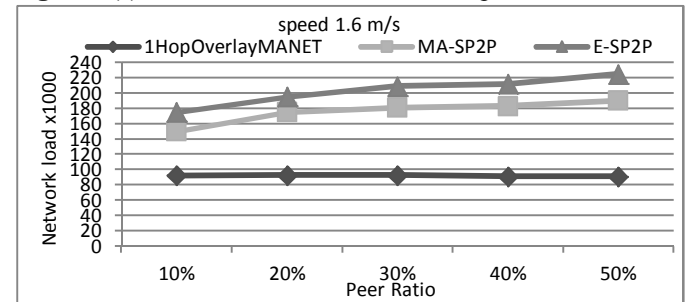


Fig. 24 (d) Network Load at 1.6 m/s node speed

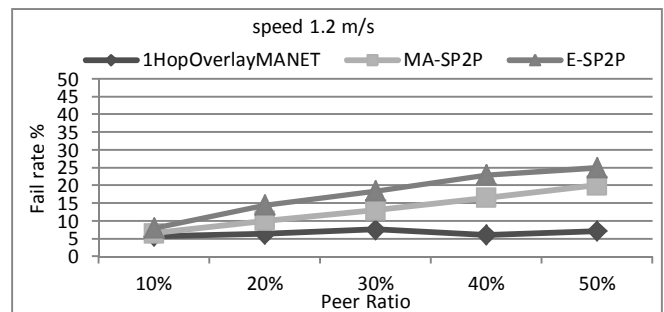


Fig. 25 (c) Fail rate at 1.2 m/s node speed

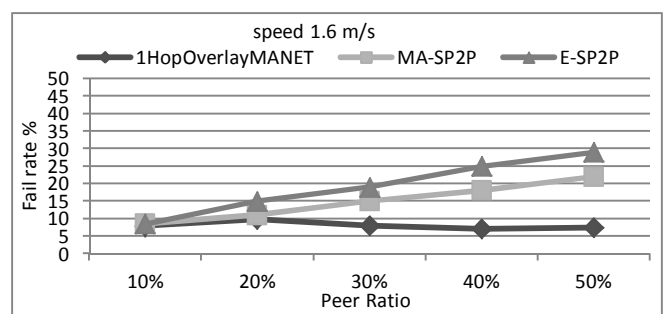


Fig. 25 (d): Fail rate at 1.6 m/s node speed

Figures 25a-d show the lookup failure rate. As the graphs indicate, the performance of MA-SP2P and E-SP2P decreases when the ratio of participating peer increases. A likely reason for this is as the peer ratio increases more traffic is created to maintain the overlay consistency. This results in an increased number of collisions and lost packets. Hence, the performance of the overlay routing is compromised. In addition, node mobility speed influences the efficiency of both systems. This can be seen as the speed is increased to 1.6 m/s the success rate drops to under 80%. One possible reason for this drop is the frequent topology changes which cause inconsistency in the overlay routing. These would require peers to maintain the minimum spanning trees and build new neighbour relationships. Furthermore, OneHopOverlay4MANET performs better than both systems especially with the higher peer ratios. The main reason for this is that any increase in peer ratio would not generate noticeable extra traffic since each OneHopOverlay4MANET peers optimize the underlay routing tables to maintain the links in the overlay. Even with the highest velocity, OneHopOverlay4MANET manages to resolve more than 90% of the lookups.

## 6. CONCLUSION

Straight deployment of structured P2P overlays over MANET is not an efficient solution to address content discovery in mobile ad hoc networks. Our approach OneHopOverlay4MANET optimizes the synergy between ad hoc routing protocols and structured P2P systems. Through cross-layering, OneHopOverlay4MANET manages to build a one hop P2P system over MANET that reduces typical P2P management traffic and thus drastically reduces the incurred overhead on the MANET. We evaluate OneHopOverlay4MANET through simulation using different underlay protocols (OLSR, BATMAN, DYMO, AODV, and DSR). In a large network, OneHopOverlay4MANET achieves its best performance of a single hop when combined with OLSR and DYMO. However, with an increased lookup frequency in larger networks, the performance of OneHopOverlay4MANET based on DYMO deteriorates whereas OneHopOverlay4MANET combined with OLSR manages to maintain its lookup success rate. Interestingly, the DYMO based system achieves the best performance when compared with the reactive underlays DSR and AODV. This is because of DYMO's routing strategy which specifies that each node in the Route Request path stores a route for every node in that path in its routing table. Consequently, an individual Route Request would result in a number of nodes adding table entries. This improves DYMO's routing performance and this is also reflected in the overlay performance.

Moreover, through simulation we compared OneHopOverlay4MANET combined with OLSR to two recent structured P2P systems for MANET (MA-SP2P and E-SP2P) which have separately been shown to outperform Modified Chord and P2P-WANT. MA-SP2P and E-SP2P adopt the same underlay (OLSR) protocol. We demonstrate that OneHopOverlay4MANET achieves a consistently better performance in terms of lookup latency, network traffic load and lookup fail rate than these systems.

## REFERENCES

- [1] S. A. Abid, M. Othman, and N. Shah, "A Survey on DHT-Based Routing for Large-Scale Mobile," *ACM Comput. Surv.*, vol. 47, no. 2, pp. 1–46, 2014.
- [2] F. Chowdhury and M. Kolberg, "A Survey of Peer-to-Peer Solutions in Mobile and Cellular Networks," in *PGNet*, 2012.
- [3] M. Al Mojamed and M. Kolberg, "OnehopMANET: One-hop Structured P2P over Mobile ad hoc Networks," *Next Gener. Mob. Apps, Serv. Technol. IEEE(NGMAST), 2014 Eighth Int. Conf.*, pp. 159–163, 2014.
- [4] B. Leong, B. Liskov, and E. D. Demaine, "EpiChord: parallelizing the chord lookup algorithm with reactive routing state management," *Proceedings. 2004 12th IEEE Int. Conf. Networks (ICON 2004) (IEEE Cat. No.04EX955)*, pp. 270–276, 2004.
- [5] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," *RFC:3626*-<http://www.ietf.org/rfc/rfc3626.txt>, pp. 1–75, 2003.
- [6] A. Neumann, C. Aichele, M. Lindner, and S. Wunderlich, "Better approach to mobile ad-hoc networking (BATMAN)," *IETF Draft. Oct.*, pp. 1–24, 2008.
- [7] D. Johnson, Y. Hu, and D. Maltz, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4 (February 2007)," *RFC 4728 - URL* <http://tools.ietf.org/html/rfc4728>, 2007.
- [8] S. Das, C. Perkins, and E. Belding-Royer, "Ad hoc on-demand distance vector (AODV) routing," <http://www.ietf.org/rfc/rfc3561.txt>, 2003.
- [9] C. Perkins, S. Ratliff, and J. Dowdell, "Dynamic MANET On-demand Routing draft-ietf-manet-dymo-," 2013.
- [10] N. Shah, D. Qian, and R. Wang, "MANET adaptive structured P2P overlay," *Peer-to-Peer Netw. Appl.*, vol. 5, no. 2, pp. 143–160, Nov. 2012.
- [11] N. Shah and D. Qian, "An efficient structured P2P overlay over MANET," in *Proceedings of the Ninth ACM International Workshop on Data Engineering for Wireless and Mobile Access - MobiDE '10*, 2010, pp. 57–64.
- [12] J. Li and J. Stribling, "A performance vs. cost framework for evaluating DHT design tradeoffs under churn," *INFOCOM 2005. 24th ...*, 2005.
- [13] I. Baumgart and B. Heep, "Fast but economical: A simulative comparison of structured peer-to-peer systems," *Proc. 8th Euro-NF Conf. Next Gener. Internet NGI 2012*, pp. 87–94, Jun. 2012.
- [14] F. Chowdhury and M. Kolberg, "Performance evaluation of structured Peer-to-Peer Overlays for Use on Mobile Networks," in *6th IEEE conference on Developments in eSystems Engineering (DESE)*, 2013.
- [15] A. Gupta, B. Liskov, and R. Rodrigues, "Efficient Routing for Peer-to-Peer Overlays.," in *Proceedings of the 1st Symposium on Networked Systems Design and Implementation NSDI*, 2004.
- [16] L. R. Monnerat and C. L. Amorim, "D1HT: A Distributed One Hop Hash Table □ Federal University of Rio de Janeiro," in *Parallel and Distributed Processing Symposium*, 2006.
- [17] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. B. Y., "Chord: A Scalable Peer-to-peer Lookup Service for Internet," in *SIGCOMM*, 2001, pp. 149–160.
- [18] G. Peng, S. Li, H. Jin, and T. Ma, "M-CAN: a lookup protocol for mobile peer-to-peer environment," in *Parallel Architectures, Algorithms and Networks*, 2004, pp. 544–549.
- [19] M. Li, E. Chen, and P. Sheu, "A chord-based novel mobile peer-to-peer file sharing protocol," *Front. WWW Res. Dev.*, pp. 806–811, Springer Berlin Heidelberg, 2006.

- [20] S. a. Abid, M. Othman, and N. Shah, "3D P2P overlay over MANETs," *Comput. Networks*, vol. 64, pp. 89–111, May 2014.
- [21] T. Zahn and J. Schiller, "MADPastry: A DHT substrate for practicably sized MANETs," *Proc. ASWN*, 2005.
- [22] F. Delmastro, "From Pastry to CrossROAD: CROSS-layer Ring Overlay for AD hoc networks," in *Third IEEE International Conference on Pervasive Computing and Communications Workshops*, 2005.
- [23] H. Pucha, S. Das, and Y. Hu, "Ekta: An efficient dht substrate for distributed applications in mobile ad hoc networks," in *the sixth IEEE workshop on mobile computing systems and applications (WMCSA)*, 2004, no. Wmcsa.
- [24] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Middleware 2001*, vol. 2218, Springer Berlin Heidelberg, 2001, pp. 329–350.
- [25] C. Cramer and T. Fuhrmann, "Performance evaluation of chord in mobile ad hoc networks," in *1st international workshop on Decentralized resource sharing in mobile computing and networking*, 2006, pp. 48–53.
- [26] M. C. Castro, E. Villanueva, I. Ruiz, S. Sargento, and A. J. Kassler, "Performance Evaluation of Structured P2P over Wireless Multi-hop Networks," *Second Int. Conf. Sens. Technol. Appl. (sensorcomm 2008)*, pp. 796–801, 2008.
- [27] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz, "Handling churn in a DHT," in *Proceedings of the USENIX Annual Technical Conference*, 2004, no. June.
- [28] S. G. Fantar and H. Youssef, "Locality-aware Chord over Mobile Ad Hoc Networks," in *Global Information Infrastructure Symposium*, 2009, pp. 1–6.
- [29] M. Thaalbi, N. Tabbane, T. Bejaoui, and A. Meddahi, "Enhanced Backtracking Chord protocol for mobile Ad hoc networks," in *International Conference on Communications and Information Technology (ICCIT)*, 2012, pp. 191–195.
- [30] R. R. Roy, *Handbook of mobile ad hoc networks for mobility models*. Boston, MA: Springer US, 2011.
- [31] A. Hinds, M. Ngulube, S. Zhu, and H. Al-Aqrabi, "A Review of Routing Protocols for Mobile Ad-Hoc NETWORKS (MANET)," *Int. J. Inf. Educ. Technol.*, vol. 3, no. 1, pp. 1–5, 2013.
- [32] "Better Approach To Mobile Ad hoc Networking (B.A.T.M.A.N.)." [Online]. Available: <http://www.open-mesh.org/projects/open-mesh/wiki>. [Accessed: 01-Jan-2015].
- [33] B. Fu, Y. Xiao, H. Deng, and H. Zeng, "A survey of cross-layer designs in wireless networks," *Commun. Surv. Tutorials, IEEE*, vol. 16, no. 1, pp. 110–126, 2014.
- [34] "OMNET++ network simulator." 2014.
- [35] "INET FRAMEWORK." [Online]. Available: <http://inet.omnetpp.org/>.
- [36] "OverSim." [Online]. Available: <http://www.oversim.org/>.
- [37] D. N. da Hora, D. F. Macedo, L. B. Oliveira, I. G. Siqueira, A. a. F. Loureiro, J. M. Nogueira, and G. Pujolle, "Enhancing peer-to-peer content discovery techniques over mobile ad hoc networks," *Comput. Commun.*, vol. 32, no. 13–14, pp. 1445–1459, Aug. 2009.
- [38] H. Sözer, M. Tekkalmaz, and I. Korpeoglu, "A peer-to-peer file search and download protocol for wireless ad-hoc networks," *Comput. Commun.*, vol. 32, no. 1, pp. 41–50, Elsevier, Jan. 2009.
- [39] Furness, J., Chowdhury, F., Kolberg, M.; An Evaluation of EpiChord in OverSim, 5th Intl. Conf. on Networks & Communications, Lecture Notes in Electrical Engineering, Meghanathan, N.; Nagamalai, D.; Rajasekaran, S. (Eds.), Springer, Vol. 284, 2014.
- [40] Chowdhury, F., Kolberg, M.; Performance Evaluation of EpiChord under High Churn, 8th ACM workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks (PM2HW2N '13), 2013.