A ONE HOP OVERLAY SYSTEM FOR MOBILE AD HOC NETWORKS

MOHAMMAD AL MOJAMED

DEPARTMENT OF COMPUTING SCIENCE AND MATHEMATICS

Doctor of Philosophy

Institute of Computing Science and Mathematics
University of Stirling

March 2016

DECLARATION

I, Mohammad Al mojamed, hereby declare that the work in this thesis is original and has been composed by myself, except where reference is made to other works, and has not been submitted for examination for any other degree at this university or any other learning institutions.

Stirling, March 2016

Mohammad Al mojamed

ABSTRACT

Peer-to-Peer (P2P) overlays were initially proposed for use with wired networks. However, the very rapid proliferation of wireless communication technology has prompted a need for adoption of P2P systems in mobile networks too. There are many common characteristics between P2P overlay networks and Mobile Ad-hoc Networks (MANET). Self-organization, decentralization, a dynamic nature and changing topology are the most commonly shared features. Furthermore, when used together, the two approaches complement each other. P2P overlays provide data storage/retrieval functionality and MANET provides wireless connectivity between clients without depending on any pre-existing infrastructure. P2P overlay networks can be deployed over MANET to address content discovery issues. However, previous research has shown that deploying P2P systems straight over MANET does not exhibit satisfactory performance. Bandwidth limitation, limited resources and node mobility are some of the key constraints.

This thesis proposes a novel approach, OneHopOverlay4MANET, to exploit 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. OneHopOverlay4MANET constructs a cross-layer channel to permit direct exchange of routing information between the Application layer, where the overlay operates, and the MANET underlay layer. Consequently, underlay routing information can be shared and used by the overlay. Thus, OneHopOverlay4MANET reduces the typical management traffic when deploying traditional P2P systems over MANET. Moreover, as a result of building one hop overlay, OneHopOverlay4MANET can eliminate the mismatching issue between overlay and underlay and hence resolve key lookups in a short time, enhancing the performance of the overlay.

In this thesis, we present OneHopOverlay4MANET and evaluate its performance when combined with different underlay routing protocols. OneHopOverlay4MANET has been combined with two proactive underlays (OLSR and BATMAN) and with three reactive underlay routing protocols (DSR, AODV and DYMO). In addition, the performance of the proposed system over OLSR has been compared to two recent structured P2P over MANET systems (MA-SP2P and E-SP2P) that adopted OLSR as the routing protocol. The results show that better performance can be achieved using OneHopOverlay4MANET.

ACKNOWLEDGMENTS

In the name of Allah, the Most Gracious and the Most Merciful. I am extremely thankful to Almighty ALLAH for his blessings and providing me with the ability to carry out this research, without which none of my work would have been done.

This work has been achieved due to the help and support of many people. I would like to express my deepest gratitude to all of them.
My sincerest gratitude and deepest appreciation goes to my supervisor Dr. Mario Kolberg for his guidance throughout my PhD study, without which this thesis could not have been produced in the current form. His insightful comments, constructive criticism and suggestions have given me invaluable guidance in various stages of my work. He has been very helpful and supportive for all these years. I greatly appreciate his care and concern throughout my PhD study.
My thanks and appreciation also go to to my second supervisor, Prof. Evan Magill, for his valuable comments, guidance and suggestions.

I would also like to express my thanks to my parents for their support, advice and encouragement throughout my research. I would also like to thank my beloved wife for her endless patience, continuous encouragement, and support. Finally, my loves to my daughters Zainah and Layan who have always pushing me forward through their love.

LIST OF PUBLICATIONS

During the period of this research, the following papers have been published:

- Mohammad Al mojamed and Mario Kolberg. "Structured Peer-to-Peer overlay deployment on MANET: A survey", *Computer Networks*, Elsevier, Vol. 96, pp. 29-47, February 2016.(Materials from this paper is included within this thesis in chapters two and three).
- Mohammad Al mojamed and Mario Kolberg. "Design and evaluation of a peer-to-peer MANET crosslayer approach: OneHopOverlay4MANET", *Peer-to-Peer Networking and Applications*, Springer US, pp. 1-18, October 2015. (Materials from this paper is included within this thesis in chapters four, five and six).
- Mohammad Al mojamed and Mario Kolberg. "Performance evaluation of OnehopMANET", *in Science and Information Conference* (SAI). London, UK: IEEE, July 2015 pp. 1028-1032.(Materials from this paper is included within this thesis in chapter six).
- Mohammad Al mojamed and Mario Kolberg. "OnehopMANET: One-Hop Structured P2P over Mobile Ad Hoc Networks", *in the Eighth International Conference on Next Generation Mobile Apps, Services and Technologies* (NGMAST), Oxford, UK: IEEE, September 2014 pp. 159-163.(Materials from this paper is included within this thesis in chapter four).

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OFF ACRONYMS

P2P Peer To Peer

TTL Time To Live

DHT Distributed Hash Table

PRR Plaxton, Rajaraman and Richa

XOR Exclusive OR

EDRA Event Detection and Report Algorithm

MANET Mobile Ad hoc Networks

IEEE The Institute of Electrical and Electronic Engineers

OLSR Optimized Link State Routing

DSDV Destination-Sequenced Distance-Vector

STAR Source Tree Adaptive Routing

WRP Wireless Routing Protocol

QOLSR Quality Optimized Link State Routing

QoS Quality of Service

AODV Ad hoc On Demand Distance Vector

DSR Dynamic Source Routing

ZRP Zone Routing Protocol

HARP Hybrid Ad Hoc Routing Protocol

LANMAR Landmark Routing

DDR Distributed Dynamic Routing

AOMDV Ad hoc on-demand multipath distance vector

MP-DSR Multi-path Dynamic Source Routing

AMRoute Ad hoc Multicast Routing

CAMP Core-Assisted Mesh Protocol

ODMRP On-demand multicasting routing protocol

EHMRP Efficient Hybrid Multicast Routing Protocol

GPS The Global Positioning System

BGR Blind Geographic Routing

ALARM Adaptive Location Aided Routing Protocol-Mines

LBM Location-based Multicast

GeoGRID Geocasting based on GRID

DYMO Dynamic MANET On-demand

BATMAN Better Approach To Mobile Ad-hoc Networking

MPRs Multipoint Relays

OGMs Originator Messages

RREQ Route Request

RREP Route Reply

RERR Route Error

OTcl Object extension Tool command language

NS-2 Network Simulator 2

NED NEtwork Discription

MST Minimum Spanning Tree

MMOG Massively Multi-player On-line Games

CNT Chord Neighbour Table

MPI Multi-Level Peer Index

GNP Global Network Positioning System

EMC Enhanced Mobile Chord

EBC Enhanced Backtracking Chord

RIGS Ring Interval Graph Search

PNS Proximity Neighbour Selection

RIG Ring Interval Graph

VRR Virtual Ring Routing

SSR Scalable Source Routing

ATR Augmented Tree-based Routing

ITR Indirect Tree-based Routing

AIR Automatic Incremental Routing

LDAG Labeled Directed Acyclic Graph

E-SP2P Efficient Structured P2P Overlay over MANET

MA-SP2P MANET adaptive structured P2P overlay

1

INTRODUCTION

1.1 BACKGROUND

Mobile Ad hoc Networks (MANET) are a collection of autonomous mobile nodes, which communicate wirelessly without any need for pre-existing network infrastructure. In addition, MANETs do not rely on centralized control. MANETs are self-configuring and nodes form a dynamic topology. As nodes move, they can organize themselves on the fly and hence the topology of MANETs may be subject to frequent and unpredictable changes. MANETs can be utilized in campus and conference environments, on railways, remote areas without communication infrastructure, or areas where the communication infrastructure is down due to natural disaster or political tensions.

P2P networks are self-organizing, adaptable, and highly scalable. P2P networks are offered as an overlay network on top of a physical network. The overlay is responsible for storing and locating services. The overlay provides a peer with the address of another peer that has a copy of the desired content. The peer can then fetch the content through an out of band connection. P2P overlays pose a low level of entry as no major server component needs to be in place. P2P networks can grow or shrink with demand and are highly adaptable.

P2P networking paradigms have gained substantial popularity as they support a wide range of applications without the use of centralized servers. P2P networking systems are usually implemented as an overlay network that allows for higher-layers communication among participating peers. The established connections between peers are logical and underlay-independent. The initial proposed P2P systems were mainly targeted at operating in wired networks. However, the very rapid proliferation of wireless communication technology has prompted a need for the adoption of P2P systems in mobile networks too.

There are many common characteristics between P2P overlay networks and MANETs. Self-organization, decentralization, dynamic operation and changing topology are the most important shared features. Crucially, co-implementing MANET with a P2P overlay on top of physical infrastructure addresses the two issues in MANETs: it will help routing decisions in the underlying network and it will allow for data storage. In a MANET, nodes may store content locally, or transmit it to a central node for storage. Both approaches pose significant problems: locally stored data is difficult for other nodes to locate, and the central data server requires a nominated node to be maintained and be available. P2P overlay networks have an in-built routing approach to store and locate data items. Another issue in MANETs is that the networking layer is inadequate to support sophisticated applications. The mobility of users makes routing challenging. In such networks, routing is usually restricted to level 2 routing based on MAC addresses. Thus P2P overlays complement MANETs well.

P2P overlay networks can be deployed over Mobile Ad hoc Networks (MANET) to address content discovery issues. P2P systems can also be used in MANETs to manage data storage and retrieval. However, previous research has shown that deploying P2P systems straight over MANETs do not exhibit satisfactory performance. Bandwidth limitation, limited resources and node mobility are some of the key constraints.

In this thesis, OneHopOverlay4MANET is proposed as a P2P system for mobile ad hoc networks. The proposed system exploits the synergies between MANETs and P2P overlays through cross-layering. It combines a Distributed Hash Table (DHT) based structured P2P overlay with MANET underlay routing protocols to achieve one logical hop between any pair of overlay nodes.

1.2 THESIS STATEMENT

Similarities between P2P networks and mobile ad hoc networks can be identified. Both networks are self-organized, decentralized, dynamic and have a changing topology. Consequently, both networks face similar challenges such as maintaining connectivity in a dynamic network. Both networks involve searching the network, but at different layers. MANET routing technology aims to provide a route from a node to a specific destination at the Network layer, whereas the goal of P2P is to manage searching and retrieving data items within the network at the Application layer. This thesis investigates the feasibility of deploying P2P overlays for data storage and retrieval over mobile ad hoc networks. It investigates possible synergies between these networks to build an efficient P2P overlay over MANETs.

The biggest challenge when combining MANETs with a DHT based overlay network is that each single logical hop in the overlay may be translated into a path in the underlay. Consequently, a logical hop results in multiple physical hops. Peers which are neighbors 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 final 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. Therefore unlike previous systems, this thesis proposes a novel one hop overlay system for MANETs that avoids criss crossing MANET when looking for a data item and optimizes the similarities between these networks to reduce P2P maintenance traffic.

1.3 AIMS AND OBJECTIVES

This research project aims to investigate the possibility and applicability of building a one hop overlay system over MANETs. The aim of such a system is to avoid the criss crossing issue when logical routing is carried out over physical routing in MANETs, thus, building an efficient overlay system that can retrieve shared keys within the network in less time. The following are the objectives of this thesis:

- This thesis investigates how different P2P overlays cope over MANETs with different network sizes and loads. In particular, the performance of existing multi-hop overlays is investigated and compared to a one-hop overlay when deployed as they are over MANETs.
- This thesis identifies obstacles when deploying one-hop overlay over MANETs and hence considers such obstacles when designing the new system for use in a constrained network such as MANETs.
- Design a one hop overlay system for MANETs that optimizes the synergies between MANETs and P2P networks. Optimization of synergy throughout this thesis means to make the most use of the exiting underlay routing information when building overlay routing.
- This thesis also investigates the use of a Cross-Layering mechanism to share underlay routing information with the overlay system in order to exploit the already available MANET routing information at the underlay.
- This thesis investigates possible reductions in the management traffic of the overlay through optimizing available routing data at the underlay layer. The reduction of such traffic will contribute toward an efficient P2P system over MANETs since MANETs suffers from limited bandwidth availability.
- This thesis investigates the suitability of such a one-hop overlay system by deploying it over different underlay routing protocols with different

network sizes, network loads and level of velocity, establishing what type of underlay is most suited to such a system.

- This thesis evaluates the performance of the proposed system considering various MANETs setting and scenarios. In addition, it also compares the performance of the proposed system to existing P2P MANETs systems.

1.4 CONTRIBUTIONS

The contributions of this thesis are highlighted in the following list:

- In order to understand the adoption of P2P technology into mobile ad hoc network, a survey of the use of P2P in MANETs was carried out. The survey has identified existing solutions that have adopted P2P for data storage and retrieval in MANETs. In addition, it has also identified the use of P2P techniques to support MANETs routing. Regarding the use of P2P for data dissemination and management, the survey has identified which type of P2P systems were used and where the gaps exist.

- The work in this thesis has deployed several P2P systems over MANETs. For the first time, it has evaluated the straight deployment of EpiChord in MANETs environments. This thesis has deployed a traditional one-hop overlay system over MANETs.

- A one-hop overlay system for MANETs, OneHopOverlay4MANET, was designed. The proposed system aims to achieve one logical hop in MANETs when searching for shared items. Thus, OneHopOverlay4MANET avoids criss crossing MANET when solving key requests.

- Full optimization of the synergies between MANET and P2P was considered in the designed system, OneHopOverlay4MANET. It builds a cross-layer channel between MANET underlay routing protocols and the overlay to allow routing information exchange between Network Layers. The underlay routing information is then used to build the overlay routing tables. Hence, this reduces typical one-hop overlay maintenance traffic.

- The one-hop overlay system for MANETs presented in this thesis was combined with different types of underlay routing protocols. OneHopOverlay4MANET was combined with proactive underlay (OLSR and BATMAN) and with reactive underlay (AODV, DYMO and DSR). The cross-layer channel was built with all the aforementioned routing protocols.

- The proposed system was comprehensively assessed using OMNeT++ as the simulation tool. It was evaluated in different MANET scenarios considering different velocity and network sizes. The performance of OneHopOverlay4MANET over OLSR was also compared with two recent P2P over MANET systems (MA-SP2P and E-SP2P) that use OLSR as the underlay routing protocol. OneHopOverlay4MANET outperforms these systems under various conditions.

1.5 THESIS STRUCTURE

This thesis consists of seven chapters. It is structured as follow:

- **Chapter 1:** Chapter 1 is an introduction to the whole thesis. It presents the thesis statement, objectives of the research and the contributions.

- **Chapter 2:** Chapter 2 introduces P2P networks. It describes structured and unstructured P2P systems. After that, the chapter introduces a number of well known structured P2P systems. Chapter 2 then introduces MANET networks. It provides general classification of the existing MANET routing protocols and then introduces some of the unicast MANET routing protocols which are of particular interest for this thesis. In addition, this chapter covers the available performance evaluation methodologies, and providing more focus on the chosen methodology.

- **Chapter 3:** Chapter 3 reviews the state of the art of P2P technology within mobile ad hoc networks. It starts by identifying the main challenges when deploying P2P over MANETs. The chapter then presents three approaches for deploying P2P systems over MANETs, and then surveys the existing works in the literature. It splits them according to the usage of Distributed Hash Table. After introducing the related works, this chapter discusses them with regard to the adopted overlay systems, the use of clustering and registration, consideration of physical proximity, deployment strategies and how they were evaluated.

- **Chapter 4:** Chapter 4 presents a novel system, OneHopOverlay4MANET, that builds a one logical hop overlay for MANETs. The chapter starts by experimenting with the mismatching issue when deploying P2P overlay over MANETs using simulation tools. The result shows that deploying one-hop overlay is a promising solution and can solve the mismatching issue in P2P MANETs. The novel system, OneHopOverlay4MANET, is then presented to achieve one logical hop in MANETs with minimum cost optimizing synergies between the two technologies. The system exploits synergies between P2P and MANET aiming for minimum cost for the operation of the system. OneHopOverlay4MANET is then evaluated and compared with EpiChord and Chord layered over MANETs. The result shows that the proposed system achieves better performance than traditional one-hop P2P overlay when deployed without cross-layer optimizations over MANETs.

- **Chapter 5:** Chapter 5 provides wider exterminations of the proposed system with other underlays. It introduces the required different joining procedures. Firstly, it presents the joining process when OneHopOverlay4MANET is combined with a reactive underlay such as AODV. Secondly, the chapter introduces the joining process over a proactive underlay such as OLSR.
  The chapter then provides more evaluation of the proposed system. It presents an evaluation of OneHopOverlay4MANET over two MANET proactive routing protocols and three MANET reactive routing protocols. A

cross-layering mechanism was implemented with all the adopted underlays. The purpose of this evaluation is to identify the most suitable underlay for the proposed system in different environments.

- **Chapter 6:** Chapter 6 presents further evaluation of OneHopOverlay4MANET. It presents a performance evaluation of OneHopOverlay4MANET against two P2P systems that were specifically proposed for MANET networks. The chapter introduces both systems (E-SP2P and MA-SP2P) which are based on OLSR as the MANET underlay routing protocol. The proposed system is also combined with OLSR as the underlay routing protocol in this evaluation. Chapter 6 then presents the evaluation which shows an improvement over both systems, where the proposed system manages to resolve key lookups in less time, generating less traffic and achieving a higher success rate.

- **Chapter 7:** chapter 7 concludes the thesis. It starts by giving a summary of the whole thesis and then revisits the objectives of the thesis and discusses how they were addressed. Finally, this chapter addresses the limitations of the system and potential future research direction.

2

P2P AND MANET NETWORKS

This chapter discusses the background of both P2P technology as well as mobile ad hoc technology. The P2P section gives an overview of structured and unstructured P2P systems and then introduces some well known P2P protocols. Secondly, this chapter gives a background to MANET networks. It also provides a broad classification of the existing technology in MANET. Additionally, it covers some well known routing protocols in this area. After giving backgrounds for both P2P and MANET, this chapter introduces the evaluation methodology for this research, and related simulation tools.

2.1 P2P NETWORKS

Peer To Peer (P2P) networks have attracted considerable interest recently. The P2P communication paradigm enables peers to access distributed resources and services without the need for a centralized server [1, 2, 3]. P2P networks are self-organizing, adaptable, and highly scalable. A P2P network is offered as an overlay network on top of a physical network. The overlay is responsible for storing and locating services. The overlay provides a peer with the address of another peer that has a copy of the desired content. The peer can then fetch the content through an out of band connection. P2P overlays pose a low level of entry as no major server component needs to be in place. P2P networks can grow or shrink with demand and are highly adaptable.

P2P systems can be classified broadly into three main categories: Centralized P2P systems, decentralized P2P systems, and hybrid P2P systems [4]. The decentralized systems form by far the largest category.

Centralized P2P systems use one or more servers in the P2P network to allow peers to locate the desired contents. Hence, the central servers maintain the addresses of peers and the contents or services they provide. Napster [5] is the prime example of a centralized P2P system. The major drawback of centralized P2P systems is that the server may be susceptible to a single point of failure. Furthermore, as the number of peers grows, the central server may become a bottleneck.

In decentralized P2P systems, all of the participating peers are equal in terms of their rights and responsibilities that they have towards the network. As a result of having no central server, there is no single point of failure or bottleneck in the overlay. Such networks can be described as self-organizing and symmetric.

Hybrid P2P structures are designed to include the advantages of both centralized and decentralized P2P systems. Hybrid P2P overlays introduce a dynamic server notion to exploit the advantages of centralized and decentralized algorithms. In Hybrid P2P networks, more powerful peers are selected to act as dynamic servers (often referred to as super peers) serving other peers. In such networks, a peer can locate contents using decentralized lookup strategies or the centralized super-peers. Examples of hybrid P2P protocols are Gnutella 0.6 [6] and JXTA [7].

In terms of the overlay topology, decentralized P2P systems can be subdivided into structured and unstructured P2P networks. They differ in the way that they forward the query messages. Overviews of both categories are provided below.

2.1.1 *Unstructured P2P*

Early P2P systems used an unstructured topology since they do not follow any particular structure when forming the network. There is no relationship between the stored contents and peers in the overlay. In other words, mapping between services and peers does not exist. Thus searches are based on Random Walk and

Flooding techniques since it is difficult to foresee the location of the queried service, and the only way to find contents is through searching part of, or the entire network, to locate the desired object. As a result, many unstructured systems use broadcasting to locate services. This floods the network with query messages leading to a very high traffic volume. This can be reduced by using a Time To Live (TTL) approach. Each receiving peer decreases the TTL counter by one before forwarding the query. When the TTL reaches zero, the query message does not get forwarded again. Examples of unstructured overlays are Freenet [8] and the original Gnutella 0.4 [9].

In unstructured P2P, no guarantee can be given that a data item which exists in the network can be found by a search without searching the whole network. These issues are addressed by introducing structure in the organization of the nodes in the overlay. These overlays are referred to as structured P2P overlays.

2.1.2 *Structured P2P*

Structured P2P networks [10, 11] have been devised to address the shortcomings of unstructured systems, namely to provide efficient search strategies that guarantee to reach the destination within a small number of hops. Some systems organize peers as a multidimensional grid (e.g. CAN [12]), a ring (e.g. Chord [13]), or as a mesh (e.g. Pastry [14]). Structured P2P overlays impose a mapping between a node identifier and the identifier of a data item. Most commonly, such a mapping is achieved through the use of hash functions on the data and node identifiers. Files are located on the peers with the closest hashed node Ids to the hashed file ID. Thus in structured P2P overlays, a data lookup query can be directed towards a particular peer which is responsible for the content required. As a consequence, the searching in the network is much more efficient even if the number of the participating nodes is large. Most structured P2P overlays offer a complexity of O(log n) for lookups, where n is the number of participant nodes [11]. However, some systems achieve O(1) lookup complexity. This difference in performance is dependent on routing table size and updating mechanisms. A large routing table which is kept up-to-date through frequent refreshes supports an excellent lookup performance, but comes at the cost of increased bandwidth requirements for the update messages. In addition, an increased level of node churn (nodes joining and leaving the network) requires additional maintenance traffic to keep the data in the nodes' routing tables current. Consequently, structured overlays can be categorized according to their lookup complexity. Approaches can be split into multi-hop, variable-hop and single-hop overlays.

Multi-hop overlays require multiple hops to deliver a lookup request from source to destination. Well known protocols using this approach are Chord and Pastry. Multi-hop approaches can be further divided into constant degree and logarithmic degree overlays. Logarithmic degree overlays reduce the lookup space in half with each hop. Hence a lookup would require at most O (log n) hops, where n is the number of the participating nodes. An example algorithm for this category is Chord. Constant degree overlays require a fixed number of hops to route a message from the originator to the destination. CAN and Cycloid [15] are examples of constant degree overlays. Sections 2.1.2.1, 2.1.2.2 and 2.1.2.3 introduce Chord, Pastry and Kademlia, which are well known multi-hop P2P systems.

On the other hand, in one-hop overlays each participating node holds a complete routing table for the whole overlay. If this routing table is fully up-to-date (which is difficult and costly), only a single hop in the overlay is needed to achieve the routing of a message from a source to a destination. Examples of such algorithms are OneHop [16, 17], EpiChord [18] and D1HT [19]. Section 2.1.2.4 and section 2.1.2.5 present D1HT and EpiChord respectively.

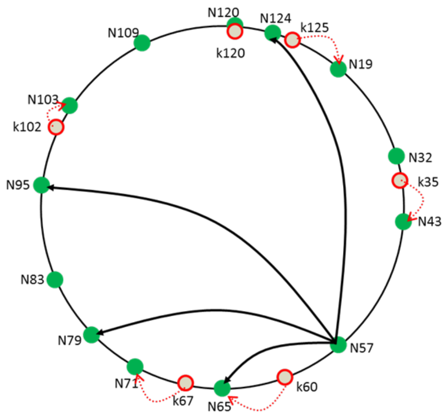
Variable hops overlay takes into account the available bandwidth and adjusts their routing table update mechanism accordingly. During times when bandwidth is at a premium, the overlay reduces routing table updates and employs a multi-hop routing technique. However, when bandwidth is more plentiful, the overlay increases its routing table update frequency and employs a one-hop lookup mechanism. Examples of variable hops overlays are Accordion [20] and Chameleon [21]. Accordion is discussed in sections 2.1.2.6.

2.1.2.1 *Chord*

Chord is one of the most influential structured P2P system, and was developed at MIT by Stoica et al [13]. It is one of the most widely known structured P2P systems. It is also one of the first P2P systems that uses a Distributed Hash Table (DHT). Chord uses a one-way consistent hash function to associate a node and content item with an identifier. An m-bit length identifier is assigned for each item and node using a hashing function such as SHA-1. The basic functionality of Chord is to order nodes' identifiers and items' keys in identifier circle modulo 2^m . The address space is a circle of number ranging from 0 to $2^m - 1$. Each key is assigned to a node whose identifier is greater than or equal to k. This means a node is responsible for content items whose keys are less or equal to a node's identifier.

Figure 2.1 shows a Chord ring with 7 bits identifier. The address space of this Chord ring varies from 0 to $(2^7 - 1)$ identifiers. K35 is assigned to the closest node clockwise, which is N43. K120 is assigned to N120, as the item's key is equal to the node identifier. Moreover, it can be seen that the successor of k125 is N19 since the circular structure modulo 127 Ids.

For the purpose of efficient lookup, each Chord node maintains a table called a finger table. The purpose of the finger table is to point to other nodes in the Chord ring. A node's finger table can contain many entries as the used bit identifier (m) when initializing the Chord ring. Each row i in the finger table holds information about a node whose identifier succeeds or is equal to $n + 2^{i-1}$, where $1 \leqslant i \leqslant m$. Figure 2.1 shows the finger table for node N57. The second row in the finger table identifies N65 as the successor of N57 + 2^{2-1}. Similarly, the fifth entry identifies N79 as the successor of N57 + 2^{5-1}.



Finger Table of N57		
Index	Target ID	Successor
0	N57+2^{1-1}	N65
1	N57+2^{2-1}	N65
2	N57+2^{3-1}	N65
3	N57+2^{4-1}	N65
4	N57+2^{5-1}	N79
5	N57+2^{6-1}	N95
6	N57+2^{7-1}	N124

Figure 2.1: Chord ring with 7 bits identifier

The stored information in the finger table is used by the Chord protocol to efficiently find a key in the network. When a node receives a key lookup request, it firstly checks whether the key exists between itself and its successor or not. If the key does exist between itself and its successor, the queried node forwards the lookup request to its successor as the answer of the lookup. If the key does not fall between the queried node and its successor, the queried node searches its finger table to find the closest predecessor of k on the identifier circle. It then forwards the lookup request to the closest node if recursive routing is used. The same procedure will be followed recursively until the lookup is solved.

2.1.2.2 *Pastry*

Pastry [14] is one of the earliest distributed hash table systems developed for P2P content distribution. Each Pastry peer is assigned a random 128 bit identifier by, for example, hashing its address. The random identifier is then used to indicate the peer's position in a Chord-like circular address space. In addition to the circular address space architecture, Pastry adopts the Plaxton, Rajaraman and Richa (PRR) [22] tree scheme for locating desired objects within the overlay. The PRR scheme operates the longest matching prefix routing approach. Pastry also introduces network locality or proximity to minimize the distance between a query initiator and a queried peer. This is considered in terms of number of hops via different zone.

In terms of routing tables, each Pastry peer maintains a routing table, neighborhood set and leaf set. The routing table for a peer p consists of [$\log_2^b N$] rows and each row has $2^b - 1$ entries, where b is a configuration parameter with a value of 4 typically. The entries at row i have to share the first i digits of their identifiers with the current peer p's identifier. Figure 2.2 shows an example of a Pastry peer routing table.

Another table that Pastry peer maintains is the leaf set, which stores L/2 peers with numerically closest larger identifiers, and L/2 peers with numerically closest smaller identifiers. The typical value of L is 2^b . The neighborhood set keeps the identifiers and the IP addresses of L peers that are closest to the current peer with regard to their proximity.

Routing table for peer 20130210				
Row 0	02110012	11101312	2	12011121
Row 1	0	21012132	21100113	22011210
Row 2	20210113	1	20211131	20212311
Row 3	20102103	20103120	20112013	3
Row 4	0	20131031	20131101	20132021
Row 5	20130013	20130123	2	
Row 6	20130201	1		
Row 7	0			

Figure 2.2: Pastry routing table for peer 20130210 where b=2. The corresponding digit of the peer identifier is displayed in the filled cell in each row. The shared common prefix in each individual cell is highlighted.

In order to route a query in the overlay, a peer checks its leaf set first to find whether the queried key identifier falls within a range of identifiers in the leaf set. If no closer peer is found, the peer consults its routing table and follows the prefix matching scheme to forward the query. According to this scheme, the peer forwards the query to a peer that has a common prefix with the queried key by at least one more digit than current peer.

2.1.2.3 *Kademlia*

Kademlia [23] is another DHT based P2P system. It builds a 160-bits address space. Keys and peers are assigned a 160 bit identifier from the address space using the hash function. In a similar way to Chord and Pastry, object items in Kademlia are stored at the closest peer in the logical space according to their identifiers.

In order to determine the distance between peer identifiers, Kademlia adopts the notion of Exclusive OR (XOR). Kademlia organizes peers in a tree-like topology according to their bitwise XOR. In such a topology, differences in the higher order bits of an identifier are more important compared to the difference in the lower order bits. This is because the differences at higher order bits constitute a larger distance in the topology. Figure 2.3 shows a Kademlia binary tree for peer 0011...

Each Kademlia peer maintains lists for each bit in their identifier. These lists are called buckets. For example, if the length of the identifier is 160 bits, a peer maintains 160 lists. Each list presents a distance to a specific peer or sub tree of peers. Moreover, each bucket may store multiple entries for a particular sub tree of address space up to k, where k is a configuration parameter. The contents of the k-bucket itself are sorted by last seen time.

For finding a desired key, a Kademlia peer first finds the closest k peers in its binary tree. It then picks L of the closest k-bucket peers and sends the query to them, where L is the concurrency parameter. In the second step, the initiator picks another set of peers, which has learned about them in the first step, to be the destination of the query. Thus, each lookup step will reduce the searched space by half.

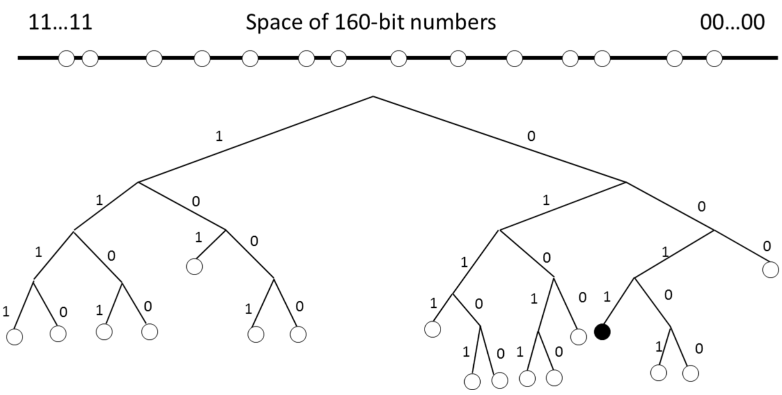


Figure 2.3: Kademlia address space with a binary tree showing peer 0011... in a black circle.

2.1.2.4 *D1HT*

D1HT [19] builds a ring topology as Chord to organize keys and peers. Peers are assigned identifiers by hashing their IP addresses or their user names. In a similar way, key identifiers are calculated as the cryptographic hash SHA-1 of the key value. After calculating a key identifier, it is assigned to the closest succeeding peer clockwise in the ring topology.

In order to achieve O(1) performance, a D1HT peer maintains a routing table that stores the addresses of all participating peers in the network. This would enable a key lookup to be solved from the first hop in the logical overlay.

To maintain a full and up-to-date routing table, D1HT uses the Event Detection and Report Algorithm (EDRA) for event dissemination in the whole system. EDRA is proposed in order to spread notifications within the whole network in logarithmic time and using low bandwidth alongside a load balancing mechanism.

In order to disseminate joining and leaving information, each D1HT peer P sends propagation messages regularly at every θ second intervals. These messages can be as much as ρ , where ρ is $\log_2(N)$, θ is a configuration parameter and N is the number of participating peers in the network. A TTL counter (ℓ) is assigned for each propagation message. ℓ can be a value in the range 0 to ρ-1. A propagation message is addressed to the 2^{ℓ}th successor of the initiator peer P. In addition, the current peer P includes all received notifications from its predecessors in its propagation messages, providing that their TTL's counter are still valid and were received within the last θ time interval. Figure 2.4 shows how events propagate in a D1HT P2P network.

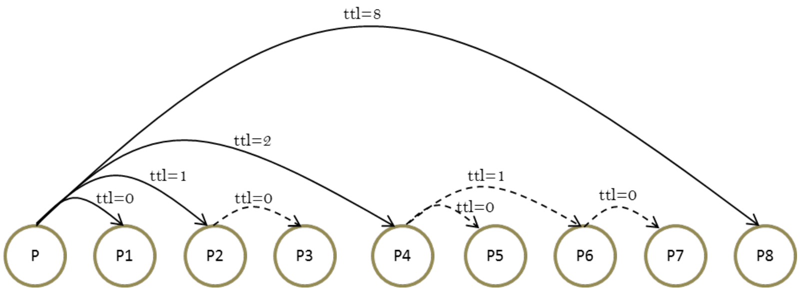


Figure 2.4: Event propagation in a D1HT system using the EDRA algorithm with 9 peers and $\rho = 4$.

2.1.2.5 *EpiChord*

EpiChord is based on Chord with the aim of improving the efficiency of lookup performance. This was achieved through using parallelism when sending key lookups and through caching more routing information. It is capable of achieving O (1) hop performance within intensive workload network [24, 25, 26]. However, in non- intensive lookups network, it guarantees at least O (log n) hop performance, similar to the performance of Chord.

In a similar way to Chord, it assigns each data object an m bits length identifier using a hashing function such as SHA-1. It builds a circular address space modulo 2^m where peers and objects are ordered according to their identifiers. A data object is stored at its successor peer. A successor peer is a peer which comes next in the circle space in a clockwise direction. A predecessor peer on the other hand is the one that comes next in the anti-clockwise direction.

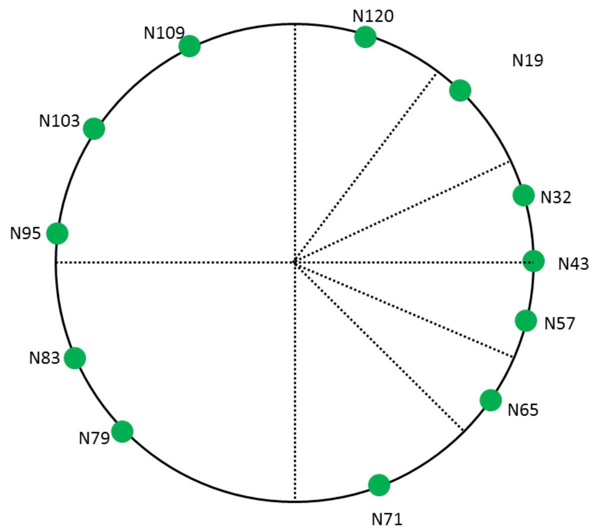


Figure 2.5: Splitting the address space into small slices for N43

In order to deliver better performance, EpiChord maintains a list of k peers succeeding and a list of k peers preceding. Additionally, each peer maintains a cache. The cache gets populated through observing lookups traffic. In addition, each EpiChord peer divides the circle into two sections and each half of the circle is divided into small slices as figure 2.5 shows. For each slice, each peer has to maintain at least $j/1-\gamma$ entries in its routing table. j is a predefined parameter and γ is the probability that a routing entry is out of date. In order to calculate the probability of a routing entry being out of date, each peer has to monitor the number of sent messages to that entry and how many of them were timed out. The probability is then calculated as the number of sent messages/ the number of timed out messages.

2.1.2.6 *Accordion*

Accordion [20] is a variable hop system that adjusts its routing mechanism to achieve best performance. The performance of Accordion is heavily dependent on only a single parameter, the Network bandwidth budget. The overlay structure of Accordion is quite similar to Chord's structure. It assigns an m-bit identifier for each peer and data item through the use of consistent hashing. Data items are placed on the closest identifier on the logical ring. Furthermore, an Accordion peer also maintains a list similar to Chord's Successor list.

For key lookup, an Accordion peer sends a lookup query to the peer whose identifier most closely precedes the required key. If the contacted peer is unable to fulfil the lookup, it consults its routing table and forwards the lookup following the mentioned procedure until the preceding peer of the key is reached. A lookup response by then should be sent directly to the original initiator of the key lookup.

The main strategy of Accordion is to emit parallel lookups and explore multiple paths. These parallel lookups can be as much as the available bandwidth budget. Parallelism is adopted to reduce lookup latency and increase the chance of retrieving a key in less logical hops. In addition, parallelism is also optimized to allow peers to learn new entries and disseminate routing information, thus, increasing the chance of finding a key by closer to one hop performance. However, this is subject to high bandwidth budget availability. The routing information can be spread between participating peers through acknowledgement messages. The acknowledgement messages are sent by targeted next hops when routing a key lookup. A next hop on the key query route should reply to the sender of the query with the acknowledgement message to acknowledge the reception of the query and to pass on some routing information. It should include in such message a list of its neighbors that falls between itself and the required key.

2.2 MANET NETWORKS

Mobile Ad hoc Networks (MANET) can be defined as a collection of autonomous mobile nodes, which communicate wirelessly without any need for pre-existing network infrastructure. In addition, MANET do not rely on centralized control. Each node participating in MANET is not only regarded as an end system but also as a router relaying messages to other participating nodes. The Institute of Electrical and Electronic Engineers (IEEE) 802.11 standard is the standard used in these networks.

MANET are self-configuring networks and nodes form a dynamic topology. As the mobile nodes move, they can organize themselves on the fly and hence the topology of MANET may be subject to frequent and unpredictable changes. MANET can be utilized in campus and conference environments, on railways, remote areas without communication infrastructure, or areas where the communication infrastructure is down due to natural disaster or political tensions [27]. The number of users in each scenario can range from about 10 nodes in emergency settings, to hundreds of people in a campus or conference setting, to thousands of participants in political unrest or military settings.

A number of surveys of MANET routing have been published. Hong et al. [28] is the oldest but still provides a good overview of the basic concepts. More recent (and more detailed) surveys have been published by Boukerche et al. [29] as well as Alotaibi and Mukerjee [30]. Each of these surveys adopts a slightly different categorization of the approaches. However, this thesis will concentrate on the major classes of MANET routing approaches.

2.2.1 *Classification*

Traditionally, MANET routing protocols have been divided into reactive and proactive systems. However, more recently, additional categories have been introduced. These include hybrid approaches (using some elements of reactive and proactive systems), hierarchical techniques, geographical routing systems as well as multi-path, multicast and geocast systems have been introduced. The following subsections briefly introduce each category and highlight key approaches.

2.2.1.1 *Proactive Routing*

In proactive routing [31] each node builds and maintains routing information to all other nodes in the network. The information is stored in tables in the nodes and maintained by exchanging information with other nodes. In proactive routing, a node can consult its own routing table to get a route from itself toward the final destination. This is possible due to the fact that each routing agent maintains a full routing table to all network nodes.

Optimized Link State Routing (OLSR) [32] is a well known example of a proactive routing protocol. Other examples include Destination-Sequenced Distance-Vector (DSDV) [33], Source Tree Adaptive Routing (STAR) [34], Wireless Routing Protocol (WRP) [35], and Quality Optimized Link State Routing (QOLSR) [36]. Such approaches create a substantial amount of background maintenance traffic to keep all the routing tables up to date. Furthermore, participating mobile nodes are required to maintain their routing table entries even if they are not being used. Also the routing tables in the nodes grow as the network size increases. Besides these scalability issues however, proactive approaches have many desirable properties such as low latency route access and Quality of Service (QoS) path support and monitoring. Proactive routing is most suitable for applications which require a low message latency and which have a high message throughput.

2.2.1.2 *Reactive Routing*

Reactive (or on-demand) routing protocols establish a route to the destination node only when there is a demand for it. Once a route is needed, the source node initiates a route discovery procedure to find a route to the destination. The route is then maintained by the source node until it is no longer required or the route becomes unavailable. As a result of node mobility, active routing entries may become invalid. Therefore, reactive routing approaches need to operate route maintenance.

With reactive routing, the route discovery process occurs much more frequently than in proactive routing and the latency for sending a message to a destination is considerably larger due to the initial route discovery process. However, reactive approaches can substantially reduce the control overhead in the network if the frequency of required route discoveries is relatively low. They are best suited for networks with low and medium traffic. In such environments they can be more scalable to larger network sizes than proactive approaches [37]. Common examples of reactive routing protocols are Ad hoc On Demand Distance Vector (AODV) [38] and Dynamic Source Routing (DSR) [39, 40].

2.2.1.3 *Hybrid Routing*

Hybrid routing approaches use aspects of both proactive and reactive routing [41]. The advantages of both routing mechanisms are combined to overcome their weaknesses, such as the traffic load issue in proactive routing, and the latency issue in reactive routing.

Commonly, proactive techniques are used to fetch routes to close neighboring nodes, while reactive approaches are used to locate nodes further away. Hybrid approaches reduce the control traffic required by proactive approaches and also reduce the message latency of reactive approaches. The performance of this type of network depends on the distribution of two approaches on network nodes. Hybrid approaches can also be implemented using a hierarchical network architecture. Zone Routing Protocol (ZRP) [42, 43] and Hybrid Ad Hoc Routing Protocol (HARP) [44] are examples of hybrid routing.

2.2.1.4 *Hierarchical Routing*

Hierarchical routing approaches organize nodes into groups called zones or clusters, with each group having a head and a gateway. Some surveys differentiate hierarchical and cluster approaches (cluster approaches employ two hierarchy levels whereas hierarchical approaches may support more than two levels). While the head is responsible for maintaining connectivity between the nodes within the cluster, the gateway manages traffic to and from other clusters. Ordinary nodes can only communicate with their own cluster head (and other nodes within their cluster). Nodes within a cluster are usually either directly connected with the head, or within very few hops. Only gateway nodes can communicate beyond the boundaries of a cluster.

Hierarchical routing may implement hybrid routing (as described above). For instance, proactive routing may be used within clusters while reactive routing is used for outer cluster communication. This approach cuts down the maintenance traffic in the network, increasing scalability. Hierarchical approaches tend to work well for high-density networks with low node mobility since their generated traffic can be low. In networks with high node mobility, hierarchical approaches can exhibit low performance due to required changes in the clusters' organization. Head nodes and gateways pose sensitive points of failure. Furthermore, head nodes should be selected carefully since they may impose a bottleneck issue. In addition, they may result in large power consumption, especially when the group size is large. Landmark Routing (LANMAR) [45] and Distributed Dynamic Routing (DDR) [46] are examples of routing protocols in this category.

2.2.1.5 *Multipath Routing*

Multipath routing protocols create multiple distinct paths between a pair of source and destination nodes. Such approaches make better use of the network's resources and are more tolerant to network failures than traditional single path approaches. Multipath routing can also make use of the aggregate bandwidth available on parallel paths to achieve better throughput. This is especially useful for high-bandwidth transmissions.

Multipath approaches can avoid network bottlenecks and achieve redundancy in the message transmission to the destination node. They can balance network load by diverting traffic to use a different path when a particular path is overloaded. Moreover, such routing protocols can also decrease the required time to recover from failure through using alternative routes. However, all these advantages come at a cost: a potentially substantially increased complexity in the route discovery process as disjoint routes between pairs of nodes are required. Ad hoc on-demand multipath distance vector (AOMDV) [47] and Multi-path Dynamic

Source Routing (MP-DSR) [48] are extensions to AODV and DSR respectively to support multipath routing.

2.2.1.6 *Multicast Routing*

Multicast routing approaches transmit a packet to a set of destination nodes. They can be classified according to how they construct the distribution routes between group members. There are tree-based, mesh-based, and group forwarding based approaches, as well as hybrid multicast protocols [37].

Most MANET multicast approaches are tree-based, creating a tree from the source to the destinations. An example of tree based multi-casting is Ad hoc Multicast Routing (AMRoute) [49]. A mesh-based network consists of a set of interconnected nodes between which a message can be sent. It is seen as a more robust alternative to tree-based approaches since backup paths are available in the event of a path failure. However, this comes at a cost of higher traffic to maintain the mesh structure. Core-Assisted Mesh Protocol (CAMP) [50] is an example of a mesh-based approach.

In group forwarding-based approaches, a set of nodes are chosen to be the forwarding nodes responsible for forwarding of multicast messages. Such approaches are also termed core based approaches and may take into account the state of the underlying network. Core nodes may be high performance nodes or nodes with additional resources. On-demand multicasting routing protocol (ODMRP) [51] is a group forwarding multicast protocol . Hybrid multicast approaches combine both tree-based and mesh-based techniques. Efficient Hybrid Multicast Routing Protocol (EHMRP) [52] is an example of this type of multicast routing protocols.

2.2.1.7 *Geographical Routing*

In geographical routing approaches, a sender uses the geographical location of a destination to route the message. Thus the node's location is used for routing rather than its network address. The main advantage of these systems is that nodes do not need full knowledge of the network topology. However, each node needs to be aware of its own location, and a source node needs to know the location of the destination node. The Global Positioning System (GPS) is typically used to work out a nodes' location. However, sometimes, a node is unable to establish its exact location (for instance inside buildings). Thus the assumption that each node knows its location may not be achievable in certain conditions.

There are three different routing strategies in geographical routing. Single path, multi path and flooding strategies. The first strategy sends a single copy of the message from the source node to the destination. On the contrary, the flooding approach adopts a broadcasting mechanism where it floods the network with the same message. Multi path strategy introduces a balance between the previous two strategies. It sends multiple copies of the same message using different paths toward the destination node. Blind Geographic Routing (BGR) [53] and Adaptive Location Aided Routing Protocol-Mines (ALARM) [54] are examples of protocols that use geographical information.

2.2.1.8 *Geocast Routing*

Geocast approaches are sending a message from a source node to a group of destination nodes based on their geographic location. Thus, in geocast routing, the physical location of destination nodes determines group membership, that is, nodes in a specific geographic location belong to the same group.

Geocast approaches are a merger of geographic routing, where routing is based on the geographical location of nodes, and multicast techniques, where a message is routed from a single source to multiple destinations. Destinations may be defined as different shapes, such as point, circle or polygon. Location-based Multicast (LBM) [55] and Geocasting based on GRID (GeoGRID) [56] are examples of geocast routing protocols.

2.2.2 *MANET Routing Protocols*

In the previous section, classification of MANET routing protocols was introduced. MANET routing protocols can also be classified according to how they transfer messages in the network. There are unicast, multicast, broadcast and geocast routing protocols [57]. For this thesis, the unicast (Point-to-Point) are of interest. Unicast routing approaches can be further divided into proactive, reactive and hybrid routing protocols [58]. The well known MANET routing protocols fall under reactive and proactive categories. In this section, we introduce five underlay routing protocols for MANET which are relevant to the work in this thesis. AODV, DSR and Dynamic MANET On-demand (DYMO) are reactive routing protocols. For proactive underlay, OLSR and Better Approach To Mobile Ad-hoc Networking (BATMAN) are discussed.

2.2.2.1 *OLSR*

Optimized Link State Routing OLSR [32] 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 optimizes link state routing to build a topology that allows each node to have routes to all other network nodes. In a pure link state routing, each node announces all of its links to its neighbors and floods such link state across the entire network. However, OLSR reduces the typical flooding of link state routing by the use of Multipoint Relays (MPRs). MPRs are a set of selected nodes which are used by a node to retransmit control messages. As a result of having MPRs, the control messages are retransmitted only through them, thus minimizing the overhead from flooding. In order to elect MPRs, a node collects information about nearby nodes which are at most 2-hops away. It uses the collected information to find the smallest number of Multipoint Relay nodes so that all 2-hop neighboring nodes can be reached through at least one relay. Figure 2.6 shows an example of MPRs nodes (N1 and N3) for node N0. 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 one of its MPRs. This list is called a 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, hence, it is an MPR node for the sender. In the case when the sender is not listed in the Multipoint Relay of the relaying node, the relaying node processes the received control message but does not forward that message, to minimize flooding.

Hop by hop forwarding is used in OLSR. In hop by hop routing, a node in the route of transmitted packet can forward the packet according to its routing information. Thus, the most recent information can be used when forwarding a packet. Moreover, when a destination moves due to mobility, its traffic can be delivered to it.

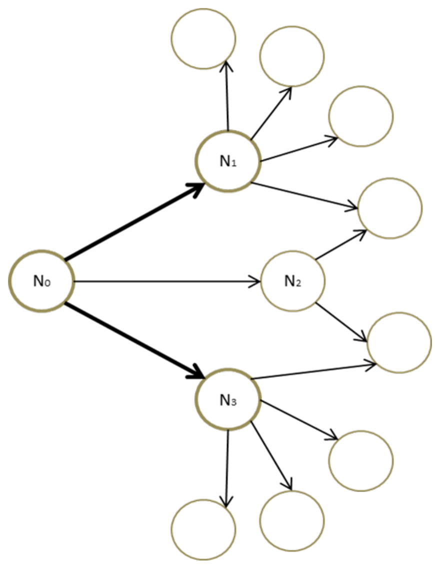


Figure 2.6: N1 and N3 were selected as Multipoint Relays for N0.

OLSR uses Hello and Topology Control messages which are sent regularly to construct routing tables. Hello messages contain information about a node's neighbors and link status and gets broadcast to neighbors. 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 MPRs and get forwarded only by MPRs nodes to minimize control overhead.

2.2.2.2 *BATMAN*

Better Approach To Mobile Ad-hoc Networking BATMAN [59] is developed by Freifunk Community [60]. 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 pro-actively maintain routing tables in the nodes that store the best next hop to be used for every destination node. It does not calculate the complete route to a destination, but only the next hop. Each BATMAN node maintains an entry in its routing table for each known node in the network. It maintains an entry for each originator of Originator Messages (OGMs) (like Hello messages) that has been received.

Every BATMAN node broadcasts OGMs regularly so it can be detected by its neighbors. Received messages will be rebroadcast by those neighbors. OGMs will be flooded repeatedly in the network until they get received by every node in the network, lost, or their TTL has expired. In order to determine the best next hop to a destination, a BATMAN node counts how many OGMs were received from the destination and logs which of its neighbors has relayed them. This is used as a metric in BATMAN to estimate the quality of a route or link. This information is then used to determine the best next hop towards that destination. Sequence numbers for OGMs are also used to distinguish new OGMs from duplicated ones.

2.2.2.3 *DSR*

Dynamic Source Routing DSR [39] is a reactive routing protocol that works on demand. It uses the source routing technique where the sender needs to include the complete route from the source node 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 a Route Request (RREQ) message to its neighbors. A RREQ message contains a list to record the addresses of the intermediate nodes that this message has passed through. Nodes that receive a RREQ message check their routing cache to find if it knows a route to the desired destination. In the case where no route is found, it will retransmit the RREQ message as a local broadcast, adding itself to the record route. Figure 2.7 shows the route discovery procedure for DSR. If the destination node is found (or a cached route in an intermediate node), a Route Reply (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 of no route back to the originator being found, the target node can initiate a route request to the originator of the initial route request piggybacking the RREP into the message. In addition, the recorded route in the Route request can be reversed and used as source route for the RREP if bidirectional communication is in place. The source node will cache the fetched route to use it in the future.

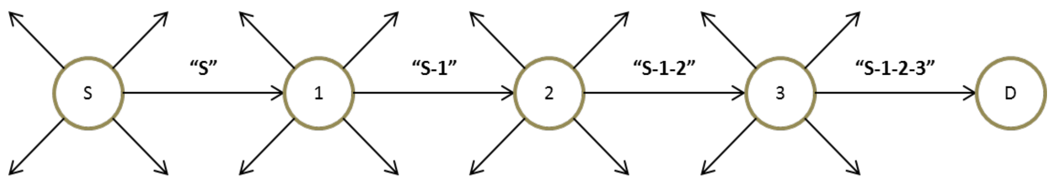


Figure 2.7: DSR route discovery mechanism, from source S to destination D

As a result of processing a RREQ, intermediate nodes can learn different routes. Therefore, they cache this routing information and operate multiple routes to any nodes in the network. Multiple routes can be used to react to route failure rapidly.

It is also claimed that caching would eliminate the need to perform a RREQ when a route failure happens, hence, avoiding overhead traffic that may result from extra RREQ.

In DSR, when an intermediate node forwards a message, according to the message's source route, the forwarding node is responsible for verifying that the next hop in the source route has received the forwarded message. If a link is found to be broken, the forwarding node deletes the entry to the next hop from its cache and a Route Error (RERR) message will be sent to the initiator of the message to allow it to update its cache. The RERR massage will include the address of the RERR initiator and the address of the node that cannot be reached. When the initiator receives the RERR message, it consults its routing cache to find an alternative route to the destination. If no route is found, a new RREQ will be sent.

DSR uses some mechanisms to improve the overall performance. Salvaging is one of the mechanisms. With salvaging, an intermediate node can consult its cache to figure out a new route to a destination when a scheduled route in a packet is broken. Complimentary repair is another mechanism. In this technique, a source node, which has received an RERR message, will send the received RERR in the next RREQ to allow other nodes to update their caches.

2.2.2.4 *AODV*

Ad hoc On-demand Distance Vector (AODV) [38] is one of the most well known MANET protocols. It is a reactive routing protocol that discovers routes when needed. Unlike DSR, AODV does not use source routing where the initiator includes the complete route to the destination. Each AODV node maintains a routing table that stores a single entry (single path) to a known destination. An entry in the routing table stores the next hop only to the destination.

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 addresses of the source and destination, the most recent known sequence number of the destination, and a unique RREQ ID. The sequence number is used to decide the freshness of a used route and the RREQ ID allows other nodes to check if they have already received this request. This provides a loop-free operation. TTL is also used to reduce the traffic in the network. A RREQ begins with a small number of TTL and it then can be increased if destination is not found. Upon receiving a RREQ message, an AODV node checks its routing table to find out if it knows of any fresh routes to the destination. If there is no route to the destination, it will broadcast the RREQ message to its neighbors. During the processing of 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. Each mobile node that happens to be along the path of the RREP message updates its routing table with the sequence number of the destination. Figure 2.8 shows how the source node S discovers the route to node D.

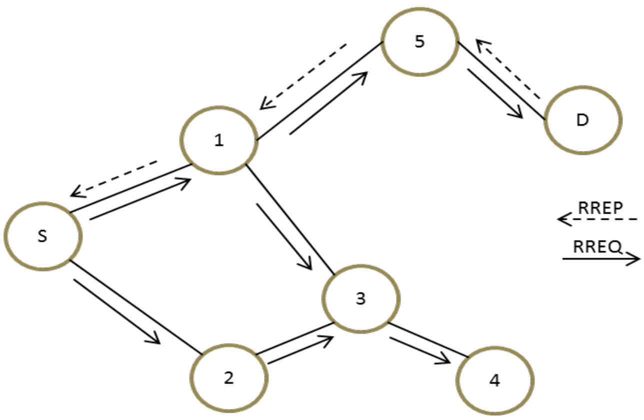


Figure 2.8: AODV route request and route reply propagation, from source S to destination D

Hello messages are used by AODV nodes to detect neighbors' connectivity. Thus, active routes can be maintained. Each node expects to receive frequent Hello messages from its neighboring nodes. If Hello messages stop arriving from a neighbor, the node assumes a link failure to the corresponding neighbor has occurred. A RERR message is then used to distribute the topology changes. If the source node receives the RERR message and still requires the route, it needs to start a new RREQ message to find a valid route.

2.2.2.5 *DYMO*

Dynamic MANET On-demand (DYMO) [61] is a reactive routing protocol that extends the functionalities of AODV. It constructs unicast routes between DYMO nodes in an on demand fashion. DYMO adopts some previous proven strategies from previous routing protocols such as AODV. An example is the use of sequence numbers to operate a loop-free routing protocol. Similar to AODV, DYMO does not employ source routing. Each DYMO node builds a routing table that maintains known destinations. Each entry in the routing table stores the next hop to the associated destination. It does not store the complete route to the destination. In a different way from AODV, DYMO does not only receive information about the final target node when discovering the route but also can receive information about all intermediate nodes alongside the discovered route. Therefore, it creates an entry in its routing table for each of the intermediate nodes specifying the next hop towards that node. Figure 2.9 shows how routing information can be disseminated within the network in DYMO when discovering the route between source and destination.

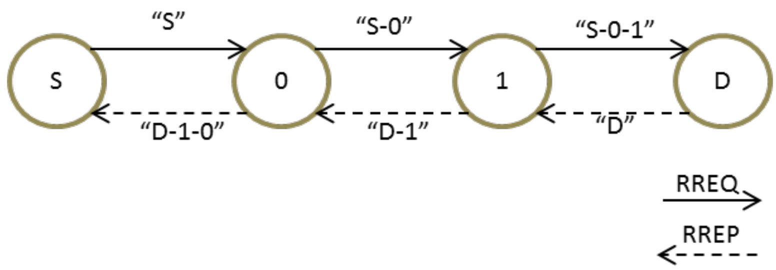


Figure 2.9: Dissemination of routing information in DYMO when discovering the route between source S and destination D

DYMO uses two main operations; Route discovery and Route management, to achieve routing in an ad hoc network. In route discovery, DYMO nodes disseminate a RREQ message throughout the network to find a route to a destination. When an intermediate node receives a RREQ, it records the route to the originator of this RREQ message, and then appends information about itself to the request and forwards it to its neighbors. 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 RREQ, it replies with a unicast RREP message that will be transmitted hop by hop to the originator. The target destination uses routing information from the received RREQ. In a similar way to processing RREQ messages, intermediate nodes receiving RREP messages, record the route to the originator of route reply, appends information about themselves to the message and then forward it along the reverse path.

In route management, a DYMO node updates the time-out 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 RERR to the source node. Upon receiving a RERR message, the source node should delete the corresponding entries and initiate a new RREQ to the target node if needed.

2.3 Performance evaluation methodology

There are three main methodologies which are typically used to evaluate the performance of a communication system. These are real experimental measurement, analytical/mathematical modelling, and the simulation technique.

Real experimental measurement involves using a test bed to carry out performance evaluation of the proposed system. A test bed usually consists of wireless devices that are configured to run the proposed protocol. The advantages of using such a technique is that it takes into account realistic conditions. This would provide better information about the system before deploying it in the real world.

However, constructing and maintaining a test bed is expensive, especially when evaluating the performance of the proposed system in larger networks. This is due to the required equipments, people and coding works. This, however, would require a team work.

Analytical modelling is based on a mathematical description of the proposed system using applied mathematical theories such as queuing and probability. In order to get insight about the system, numerical methods can then be applied to the model. Mathematical modelling suits simple and relatively small networks. Deriving such models for a dynamic network such as MANET may become more complicated. These type of networks require factors such as node mobility and signal propagation to be taken into account. These factors might be simplified by introducing some assumptions in the model. However, the simplified assumptions may affect the accuracy of the evaluation. Due to the complexity of mobility, there are limited uses of the analytical modelling technique [62].

The performance of the communication system can also be evaluated using simulation tools. In simulation, a network simulator is built to represent a computer network simulating the functionality of the network nodes and the communication channels. Simulation has advantages over other evaluation methodologies such as providing scalability, the results being repeatable, and complicated scenarios can be simulated. Simulation based evaluation also involves fewer assumptions compared to analytical modelling. In other words, it allows the designer to cover more details of the evaluated system. In comparison to the real experimental measurement, simulation provides a cheaper solution. It also makes it easier to investigate the performance of the proposed system under varying load and different network scenarios. Simulation based evaluation therefore is adopted in this thesis.

2.3.1 *Simulation Tools*

There are several network simulators that can be used to simulate mobile networks. NS [63], riverbed (known previously as OPNET) [64], Omnet++ [65] and

QualNet [66] are typically used to simulate MANETs. In the following, these simulators are introduced and their support for P2P protocols is also discussed.

Network Simulator 2 (NS-2) is a discrete event network simulator. It includes models to simulate wired and wireless communication protocols at different layers such as physical, link and network layers. It has been widely used in research. It uses C++ as the programming language. In addition to C++, it uses MIT's Object extension Tool command language (OTcl) to allow the user to specify the parameters to the C++ code. NS-2 has been replaced by its successor NS-3. NS-3 was built from scratch using C++ as well. It focuses on solving existing problems in the NS-2 simulator. It has a compatibility issue with NS-2 where it does not support all the models written for NS-2. Ns-3 uses C++ or python scripts to define simulations. Both simulators are open source and freely available for research, development and use. Both Network Simulators do not provide much support for P2P protocols. There are only two simulations for P2P systems available for BitTorrent and Gnutella in NS-2 and NS-3.

riverbed, formally known as OPNET, is a discrete event simulator that comprises of a suit of protocols and technologies written in C++. It supports wired and wireless communication systems, e.g. MAC protocols of IEEE 802.11 a, b and g. It is one of the most widely used network simulators and is also available as a commercial simulator. As a commercial simulator, riverbed provides excellent graphical support that allows a user to build a network model entities from the application layer to the physical layer. The graphical interfaces also allow the user to analyze and visualize simulation results. With regard to the riverbed' support for P2P networks, there is a framework called iP2P [67] to support P2P networks in OPNET. iP2P does only support one P2P system, ED2K [67] . Therefore, the work on P2P in riverbed is quite limited.

QualNet is another commercial network simulator. It is a discrete event simulator based on the GloMoSim simulator which was designed for MANETs.

QualNet is the commercial derivative of GloMoSim, which is no longer being developed. QualNet supports different ranges of both wired and wireless networks. QualNet provides a powerful graphical GUI that allows users to build their scenarios' components and set their specifications. It also provides excellent analysis tools for simulation results. QualNet uses C/C++ to implement new models. However, as it is commercial, it is not widely used in research. Regarding the support for P2P overlay, QualNet does not provide any implementation of P2P protocols.

OMNeT++ is an extensible, modular, component-based C++ simulation library and framework for building network simulators. OMNet++ is not a simulator itself but provides requirements with basic tools to allow developers to build their simulators. Frameworks are independent projects that provide specific support for specific environments. There are for example frameworks that support wireless networks, internet protocols, performance modelling, P2P overlays, etc. OMNeT++ is open source software and available free of charge for academic use. It offers extensive graphical support for user. Simple modules are written in C++ and assembled to form larger components. Components can be modelled using a High-Level language called NEtwork Discription (NED), which functions in a similar way to OTcl in NS-2. NED allows developers to declare simple models, connect them to each other, and to assemble them as a compound component. OMNeT++ adopts message passing as the way to communicate with other modules.

In terms of available support for P2P networks, the OMNeT++ simulation environment has the Overlay Simulation Framework (OverSim) [68]. OverSim provides simulation models for several structured and unstructured P2P overlays. The simulator is developed by the Institute of Telematics, Karlsruhe Institute of Technology (KIT). On the other hand, there is the INET framework [69] for OMNeT++ simulation environment. INET provides researchers with an open source library for simulating protocols, agents and other models, including wired and wireless communication networks. It has implementations of the Physical, Link, Network, Transport and Application layer. The Framework is actively maintained by an OMNeT++ team and utilizes community member contributions. The INET framework has been used as the base for several other frameworks including OverSim. When OverSim was developed, it adopted the INET framework as the base to provide the underlay simulation environment. Therefore OMNeT++ has been chosen for performance evaluation in this thesis. It has been chosen for the following reasons:

- Open source software.
- Free license for academic and educational environments.
- OMNeT++ has a good manual and good community support and discussion forums.
- Supports several mobile ad hoc routing protocols.
- Supports several P2P overlay protocols.
- Existing components can be modified easily.

2.3.2 *Validation*

Simulation models can be validated using different techniques [70]. One of these techniques is the comparison with the simulation results of other models of the same algorithms. With this approach, results from a previous study are compared with results achieved using the model to be validated. This requires parameters to be set to the same values in both simulations.
To validate the chosen simulation environment including the INET framework with implementations of AODV, DSR and OLSR protocols, we have attempted to reproduce results reported in papers [71] and [72]. These papers were chosen because they include detailed list of simulation parameters and their values. However, even these papers do not specify a complete list of parameters, Consequently, some parameters were assumed, and hence this may be a reason for

slight discrepancies in the results of the two sets of the simulations.

For AODV and DSR, simulation parameters were set to match those reported in [71]. Simulations were performed in an area of 500 m x 500 m, with 50 mobile nodes, using the random way point mobility model and moving at random speeds between 0 and 20 m \s with 0, 10, 20, 40 and 100 s as the pause time. The total simulation time was 100 s. Two scenarios were simulated to consider different numbers of traffic sources in the network. These scenarios were first a network of 50 nodes, including 30 nodes as traffic sources, and the second scenario was a network of 50 nodes, including 40 nodes as traffic sources. Source nodes were set to transmit 4 packets of 512 bytes every one second.

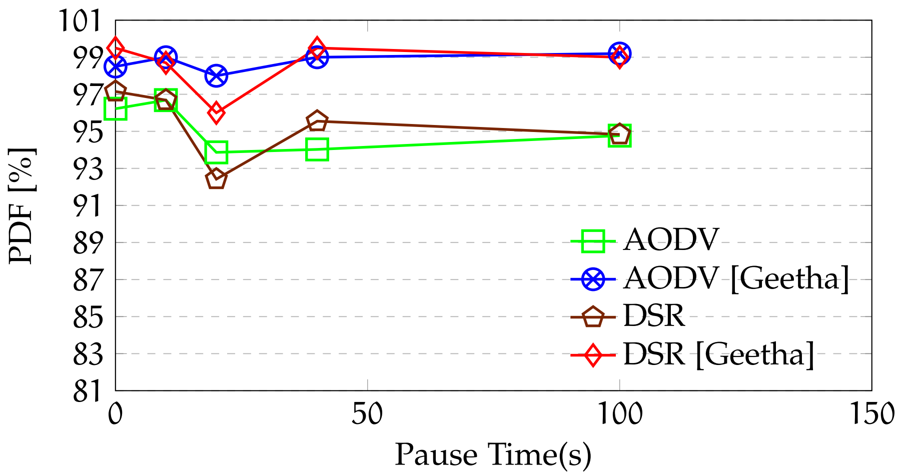


Figure 2.10: Packet delivery fraction with 30 node sources

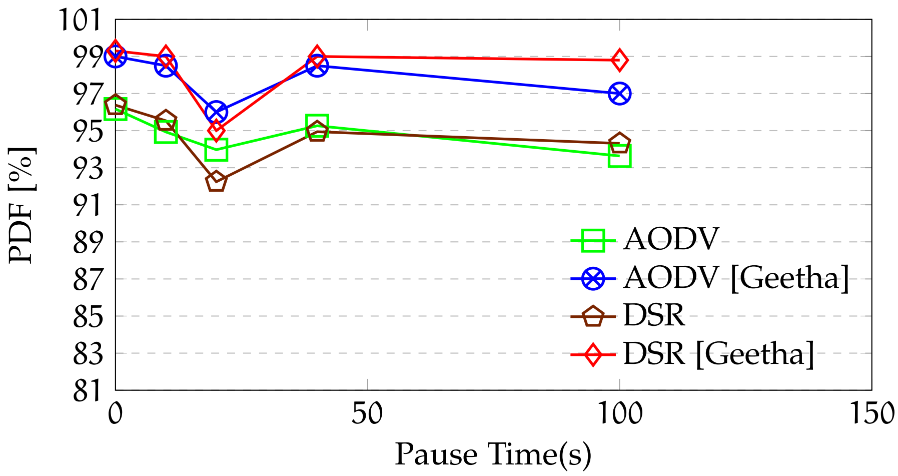


Figure 2.11: Packet delivery fraction with 40 node sources

Figure 2.10 and 2.11 show the packet delivery fraction for AODV and DSR with different number of mobile sources for INET models and results reported by Geetha [71]. As can be seen, the results are not identical. However, similarities between the performance can be noted. This can be seen when looking at the trend of packet delivery fraction versus pause time, with different numbers of sources where both models show a similar trend.

For OLSR, simulation parameters were set to match those in [72]. The simulated area was 1000 m x 1000 m, containing 60 nodes. CBR traffic was issued by 10 of the 60 nodes with a packet size of 128 bytes. For mobility, random way point was the used mobility model with no pause time. Different node speeds were simulated; 0, 10, 20, 30, 40 and 50 m\s were the maximum node speeds. The simulation for each scenario was 100 s.

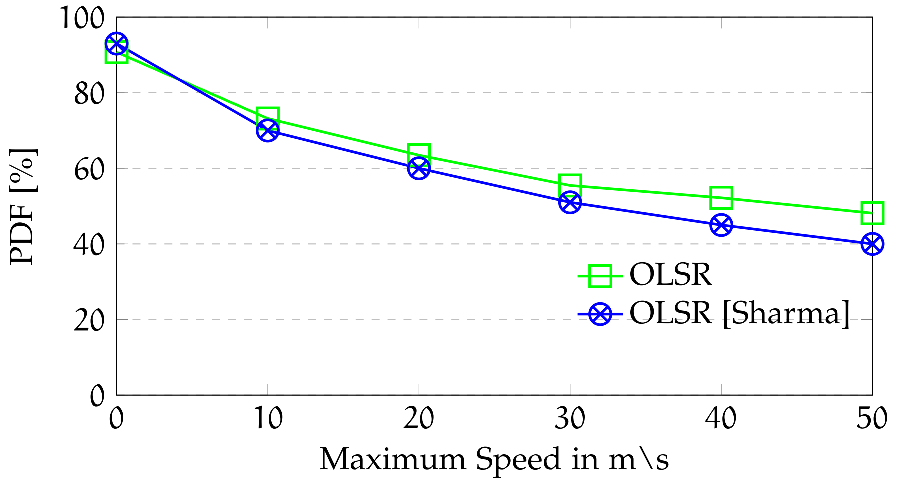


Figure 2.12: Packet delivery

Figure 2.12 shows the packet delivery ratio for the OLSR INET model and the results reported by Sharma [72] as a function of maximum node speeds. As the graph depicts, the results are not identical. However, the graph shows a similar performance for both two models. The performance of both models decreases when the maximum node speed increases, and both models show the same trend. With regard to the overlay framework OverSim model, the framework was validated in several works. EpiChord was validated in [25]. Chord and Pastry were also validated in [73].

2.4 SUMMARY

This chapter introduced P2P networks. Decentralized P2P systems can be classified according to the overlay topology into structured and unstructured systems. Since the focus of this thesis is on structured systems, the chapter has addressed this type of P2P and introduced a number of well known systems in the literature.

It has also introduced the second part of this thesis; MANET networks. It shows that routing in such networks can be classified into reactive, proactive, hybrid, hierarchical, multipath, multicast, and geographical, as well as geocast routings. Furthermore, it has shown that MANET routing can also be classified according to how network members exchange their messages. It then investigate the well known protocols in the literature that adopt unicast routing and are relevant to this research.

After discussing both networks, this chapter has investigated possible evaluation methodologies for P2P and MANET systems. The adopted methodology, the simulation technique, was then discussed in depth, and where possible simulation tools were introduced.

3

P2P OVER MANET

This chapter discusses the state of the art for P2P networks in mobile ad hoc networks. It starts by introducing the common challenges for deploying P2P over MANET. It then introduces the approaches for such deployment. The chapter then surveys the existing works in the literature. Furthermore, it classifies them according to the purpose of using distributed hash table technology. They are classified mainly in two categories; systems that provide P2P overlay over MANET and those which adopt a distributed hash table to improve the performance of MANET routing. The former is further classified according to the underlay used; proactive-based, reactive-based, GPS-based and underlay independent systems. The latter is further classified according to the structure of the adopted P2P technique. This chapter then introduces a discussion of the existing works regarding the different criteria.

3.1 CHALLENGES

Deploying P2P overlay networks on MANETs faces a number of challenging issues [74] [75] . The following list summarizes the key challenges:

- Limited bandwidth: This is one of the most fundamental challenges when deploying P2P over MANET. Current P2P overlay algorithms can be quite wasteful with bandwidth. However, MANET bandwidth resources are more constrained when compared with a wired IP infrastructure. As a result, deploying standard P2P algorithms will incur high maintenance overhead and hence are unsuitable for MANET.

- Logical overlay maintenance: In order to maintain their routing tables, DHT protocols periodically send maintenance requests and responses to learn about unavailable peers. This traffic duplicates route discovery in MANET and hence is undesirable.

- Physical topology changes: By their nature, MANET nodes are mobile which can break links between nodes and hence has an impact on the overlay. Thus, in order to have a consistent overlay, the P2P system needs to be informed about the changes in the underlay. Clearly, this incurs a cost.

- Routing stretch: In a system combining P2P overlay and MANET underlay, routing is implemented both at the physical (MANET) layer and the logical (P2P) layer. Crucially, each hop in the overlay corresponds to a path in the underlay. Furthermore, two nodes which are close in the overlay, may be far apart in the MANET, and nodes which are physically close may be multiple logical hops apart. Routing stretch is the ratio between the logical and the optimal physical path. As a result of this a lookup message may require a number of logical hops in order to access a physically close service. Hence minimizing routing stretch is one of the most critical challenges.

- Network resiliency: When a peer fails in the structured P2P overlay, the overlay will adapt and discover an alternative path. The same mechanism can be applied at the MANET layer when used in conjunction with an overlay. However, operating such mechanisms in MANET is expensive. Thus network resiliency is an important issue.

- Query propagation: Due to the fact that a direct hop in the overlay translates to a potential long path in the underlay, a high frequency of overlay messages may lead to path congestion in the underlay. Clearly this should be reduced as much as possible. Approaches include limiting the query range through using TTL and using smart relaying (the relaying node waits for a period before rebroadcasting the message).

- Battery power: As most P2P overlay algorithms were designed for a wired environment, they do not take into account the constraints imposed by limited battery power availability. However, when using such algorithms with MANET, the number of messages needs to be minimized to conserve battery power.
- Infrastructure-less: the lack of infrastructure with MANET makes it difficult for some P2P protocols to be adopted for use with MANET. For instance, CAN relies on using static landmarks when assigning logical Ids.

3.2 DEPLOYMENT APPROACHES

MANET and P2P overlays traditionally operate at different layers in the protocol stack. P2P overlays reside and operate at the application layer, while MANET protocols are concerned with connectivity between nodes and work at the network layer. In order to deploy P2P overlays over MANET there are three possibilities:

3.2.1 *Legacy design*

The most straightforward design is to build a P2P overlay on top of the network layer. However, simply adding a P2P overlay on top of MANET is not a functional solution. Many of the issues outlined above will make such a system almost unworkable. Such a system would be a highly inefficient solution as both layers will operate their own routing algorithms to maintain connectivity between logical and physical neighbors. This results in considerable redundancy and very poor performance. Figure 3.1 depicts this approach. Such a system does not exploit any synergies which exist between the overlay and underlay routing. Still a number of systems following this approach have been experimented with. Backtracking Chord [76] uses modified Chord with AODV.

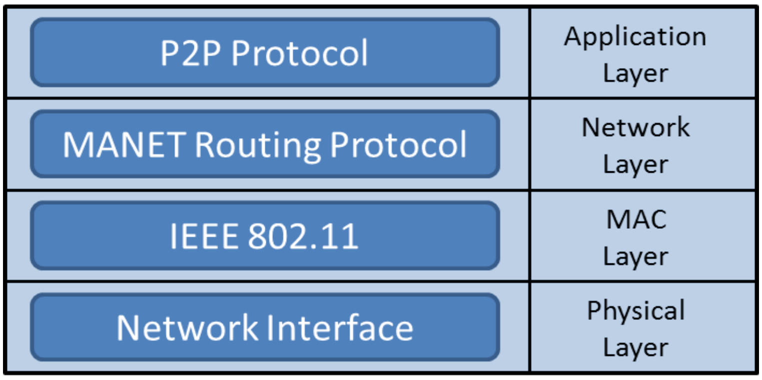


Figure 3.1: Straight layering design of P2P overlay over MANET

3.2.2 *Cross-Layer design*

In order to improve the performance offered by the layered design, P2P overlays need to know about the status of the underlay network. This would help reduce the maintenance overhead and thus result in a better overall performance. On the other hand, MANET algorithms should also be aware of the P2P overlay in order to maintain the appropriate connections with relevant neighboring overlay nodes.

This type of approach is termed a Cross-Layer design. However, this design violates the layered architecture since it allows sharing information among non-adjacent layers. Information at the network layer is made available to the application layer and vice versa. However, permitting such an exchange of information has the potential to reduce the incurred overhead considerably. Figure 3.2 depicts the cross-layer design. CrossROAD [77] is a well-known example that follows this approach by allowing Pastry to communicate with OLSR.

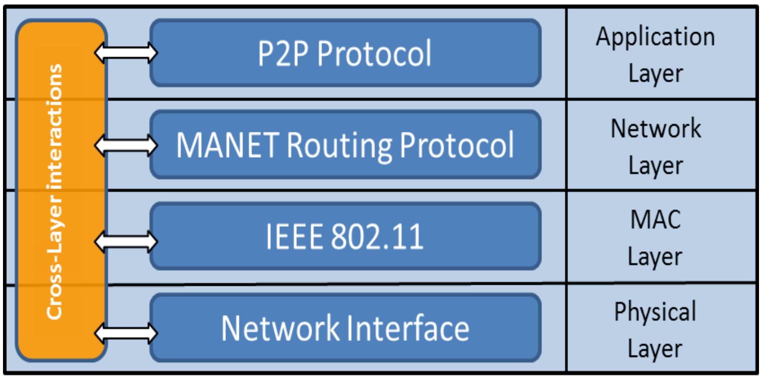


Figure 3.2: Cross-layer design

3.2.3 *Integrated Design*

The third possibility to deploy P2P over MANET is to integrate P2P algorithms directly with the network layer. Clearly, such a close integration eliminates redundant routing between the two layers at the same host. However, while this is another form of violating the layered architecture, it leads to an improved performance. Figure 3.3 shows this design option. Ekta [78] is an example that belongs to this category. Ekta integrates Pastry with DSR into one structure to exploit the synergies between the two algorithms for an improved routing performance.

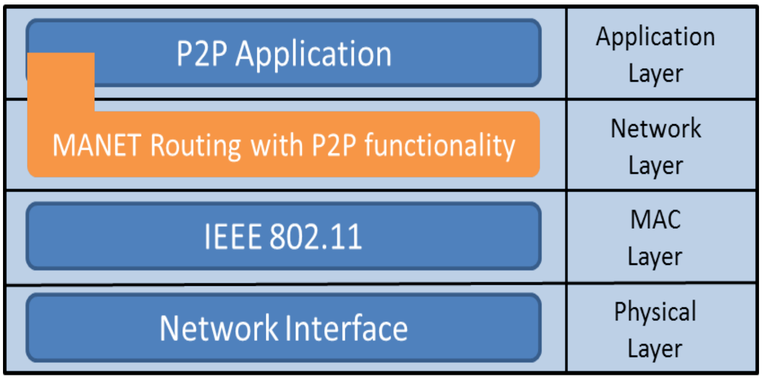


Figure 3.3: Integrated design

P2P technologies in MANET networks can be categorized in a number of different ways. Firstly, they can be categorized according to their integration design (see Section 3.2). Secondly, they can be categorized according to the type of P2P overlay used and thirdly they may be categorized according to the purpose of employing a distributed hash table.

If the used P2P overlay is used for categorization, at the most basic level, approaches which use unstructured and structured overlays can be distinguished [79]. Structured overlay approaches can be further divided into subcategories according to the hop-count performance of the overlay algorithm (one-hop, multi-hop, variable hop).

Considering classification according to the usage of a DHT in MANET, two main groups of systems can be distinguished. There are systems that have used the DHT to enhance the performance of MANET routing. Others have used the DHT in a similar way to its use in traditional P2P networks, to support data storage and retrieval. Systems that have used P2P algorithms to enhance MANET routing can be subdivided according to the structure of the adopted P2P technique. There are ring-based systems, tree-based systems and systems built on other structures. The data storage and retrieval systems can be divided according to the type of underlay network used. Systems can then be classified as proactive underlay based systems, reactive underlay based systems, GPS-based systems, and finally underlay independent systems. Some systems which use a GPS-based approach may also employ a proactive or reactive underlay. Such an approach would be classed as a GPS-based technique as this is a more specific feature than the type of underlay used. Thus approaches which are listed as proactive or reactive-based do not fit in another category, whereas GPS based systems may also be classed as proactive or reactive systems. The tables in the following sections provide further details.

This categorization is used to structure the existing systems. Section 3.3.1 looks at approaches that adopted P2P functionality to support data storage and retrieval in MANETs. It introduces all the aforementioned four subcategories; proactive underlay based systems, reactive underlay based systems, GPS based systems and underlay independent systems. Section 3.3.2 covers the second main category; approaches that have optimized P2P mechanism to improve routing in MANET networks with its three sub-categories; ring-based systems, tree-based systems and systems built on other structures . Figure 3.4 shows the main classifications of structured P2P over MANET.

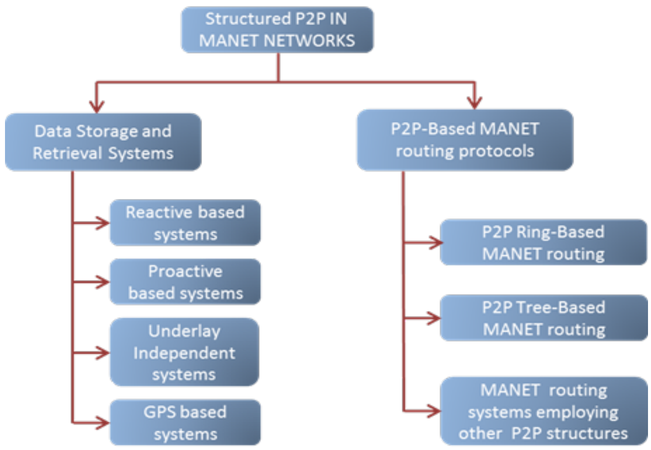


Figure 3.4: Structured P2P approaches in MANETs

3.3.1 *Data storage and retrieval systems over MANET*

3.3.1.1 *Proactive-based systems*

Proactive routing systems maintain routing information to other nodes in the network in advance of their use. The proactive underlay permits optimization of the P2P overlay as the underlay provides complete information of the routes in the mobile network. Commonly, the generated traffic in pro-active based networks is also less than systems using a reactive underlay. This is due to the fact that P2P overlays frequently send messages between any two nodes in the network. That means in a reactive system, route discovery requests are very common. Indeed they are so common that reactive systems lose their advantage over proactive systems of lower bandwidth consumption. This is especially true in systems with high node mobility. For instance, Shah et al. [80] and Abid et al. [81] claim that with a proactive underlay, a lookup issued by a peer is immediately transmitted to the target node using the shortest underlay path. This helps eliminate route discovery overhead used by reactive protocols. So, this type of underlay would offer a lower delay when discovering other peers in the network. In addition, proactive underlay may suit networks with high mobility and intensive usage since reactive ones may overwhelm the network in such setting. Abid et al. [81] use OLSR to calculate the weight of the links between peers. The approach builds a graph at each node and calculates the weight of each link in terms of the physical hops involved. Systems that use proactive underlays are CrossROAD [77], Shah and Qian [82], MA-SP2P [80] , P2P MMOG over MANET [83] , Chord in Mobile ad hoc network [84] , 3DO [81] and ROBUST [85, 86]. All of these systems have used OLSR as the underlay routing protocol. Table 3.1 summarizes proactive-based systems.

CrossROAD [77] has adapted Pastry to operate over OLSR. It inherits all the basic functionality from Pastry. In CrossROAD, each node maintains a global services table that stores all the services provided by all the nodes in the network. It uses an External Data Sharing Module (network status [87]) to allow cross-layering between the overlay and network layers. Services are updated through cross layering with the underlay routing protocols where the proactive flooding of OLSR is optimized to serve the Service Discovery protocol. When a CrossROAD peer wants to join the overlay, it needs to subscribe first with the desired service from the service repository. It then can retrieve information about peers that provide the service. Upon receiving the information, it builds its view of the overlay. CrossROAD peer maintains a logical routing table organized similarly to the prefix based routing employed by Pastry.

Shah and Qian, MA-SP2P and 3DO adopt a similar structure. They construct a Minimum Spanning Tree (MST) to build an overlay that better matches the physical underlay. Each peer builds a graph that includes the peer itself and nodes which are up to 2 logical hops away. The graph is then used to construct the MST. Shah and Qian [82] use a root peer to connect all peers in the network and to be used as a reference point. As a reference point, the root peer will be consulted whenever a pair of peers builds a neighbor relationship. The root peer announces one peer to be in charge of maintaining the relationship. The chosen peer to maintain the relationship will sends frequent probe messages to its neighbor. Each peer maintains a disjoint portion of the ID space. The ID space is distributed between peers so that a peer P must have a connected neighbor peer P1 where the lower end value of P's portion of the ID space is greater than upper end value of P1's portion. Each peer is required to maintain a routing table that stores information about the root and neighbor peers including their disjoint portion of the ID space. For routing, a query is sent to one of the directly connected neighbors which has a closer portion of the logical space. A similar procedure will be repeated until the destination is found.

MA-SP2P [80] is an extension of the previous system. It does not use a root peer. Each peer in MA-SP2P maintains a disjoint portion of the ID space that may be non-consecutive. A Data item or key is stored on a peer P if the hash value of the key falls in the P's portion of the ID space. MA-SP2P distributes the ID space in a way that the Lower End (LE) value of a portion at peer P points to neighbor peer P1 that has a portion of the ID space lower than LE. In a similar way, the Upper End (UE) of a contiguous portion at P should point to a neighbor P2 that has a portion with a greater value of ID space. MA-SP2P peers use frequent probe messages which are exchanged between directly connected neighbors to maintain the overlay.

3DO [81] attempts to solve the mismatch problem between the underlay and the overlay topology through interpreting the exact physical relationships between nodes at the logical level. It considers the relationships between neighboring peers when computing a peer's logical ID. Peers view the logical address space as a 3D rectangular coordinate system when computing logical Ids. Each 3DO peer is assigned a transient logical identifier regarding its relative location in the 3D overlay. A logical identifier takes a three tuple $\{x \mid y \mid z\}$ form. For each axis, the identifier ranges from 1 to $\pm 2^m - 1$. Probe messages are exchanged between neighbors periodically to maintain the overlay. 3DO assigns weight to the links between peers using the number of physical hops between them. This information can be achieved through the underlay protocol (OLSR). When a peer joins the overlay, it builds the MST using neighboring peers' information.

P2P MMOG [83] targets Massively Multi-player On-line Games (MMOG) over MANET. It uses clustering and landmarking. It organizes peers in a hierarchical structure where it builds two levels of DHT; local DHT and remote DHT. It divides the network into clusters where peers share common overlay ID. Each cluster has a head that exchanges information about its cluster with neighboring cluster heads. A global DHT is used to locate the cluster heads following a prefix strategy as in Pastry. The local cluster is further organized as a virtual ring and uses a local DHT to maintain local peers. Each peer in the network is assigned a hierarchical identifier. The hierarchical identifier is used for routing purposes. Each peer in the system maintains two overlay routing tables; a Remote DHT table and a Local DHT table. However, cluster heads also maintain an extra routing table called a Hierarchical link table to maintain connectivity with cluster heads.

Cramer and Fuhrmann [84] implement Chord over a number of underlays including a proactive system to investigate the issues of deploying P2P over MANET. This is a straight deployment of P2P Chord as it is over MANET. Their conclusion was that the poor performance of Chord over MANET is not a result of congestion. It is a result of the protocol failover strategy where it assumes the destination has left the network when a packet gets lost.

ROBUST by Millar et al. [85, 86] is based on the structured P2P system Bamboo [88]. However, it distinguishes itself from Bamboo by introducing hierarchical clustering of the peers. Peers are clustered depending on their proximity in the underlay. Each cluster has a super peer, which is in charge of its cluster members. In addition, cluster super peers are also responsible for cluster maintenance. Each cluster is restricted to a specific size and all of its peer members have to be one logical hop away from the cluster super peer. Each ROBUST peer is assigned a logical identifier depending on the ID of the cluster super peer with which the joining peer is a member. Beacon packets are sent periodically from the super peer to all of its members. Member peers are expected to forward these packets with TTL=1 to allow other peers from different clusters to detect if they are moving closer to this super peer. A ROBUST peer has to get a new ID if it finds a new cluster super peer is closer than its current super peer.

3.3.1.2 *Reactive-based systems*

This group of approaches have proposed P2P systems for MANET using reactive MANET underlays. MANET routing protocols employed include AODV and DSR. Both protocols work on demand and discover routes as they are needed. Therefore, low traffic is generated when there is less demand from the overlay. However, a structured overlay usually requires maintenance traffic which may cause more routing traffic at the underlay. Thus, in scenarios where there are high mobility and high overlay maintenance traffic, the network may be overwhelmed by the reactive routing traffic. An obvious reason for this is the route discovery and maintenance procedures of the MANET reactive routing protocols. Since this type of underlay works on demand, it is expected that the overlay messages would encounter higher delay until successfully routed. This will affect the performance of the overlay. However, in the case of less node mobility with a low maintenance overlay system, the reactive underlay routing has the advantage of releasing low traffic in MANET and hence better performance of the overlay. AODV is claimed to give the best efficiency when deploying P2P over MANET as

Table 3.1: Proactive-based systems

System name, authors and reference	P2P system	MANET routing	Deployment method	Mobility model and node speed	Simulation/Evaluation
CrossROAD, Delmastro, [77], 2005	Pastry	OLSR	Cross-layer	No	Testbed was used to evaluate the performance of CrossROAD using 8 nodes.
Shah and Qian [82], 2010	Structured P2P, MST	OLSR	Layered	Random Way Point, Speed: 0.4, 0.8, 1.2, and 1.6 m\s.	Compared to Da Hora et al [89]. 100 nodes.
MA-SP2P, Shah et al, [80], 2012	Structured P2P, MST	OLSR	Cross-layer	Random Way Point, Speed: 0.4, 0.8, 1.2, and 1.6 m\s.	Compared to modified Chord over AODV, Shah and Qian [82] and Hashline [90]. 100 nodes.
P2P MMOG over MANET, Yu and Vuong [83], 2011	Structured hierarchical overlay	OLSR	Integration	No	No evaluation.
Cramer and Fuhrmann, [84], 2006	Chord	OLSR	Layered	Random Direction Model, Speed: 0, 2, and 5 m\s	For comparison, other flood-based protocol was implemented. It was also compared to Chord layered over DSR and AODV.
3DO, Abid et al. [81], 2014	3D rectangular coordinate system	OLSR	Cross-layer	Random Way Point, Speed: 0.5, 1, 1.5 and 2 m\s.	Compared to MA-SP2P .
ROBUST, Millar et al. [85] [86], 2012	Bamboo	OLSR	Layered	Speed: 1 m\s	Compared to Bamboo over MANET, size of network varied from 20 to 70 with only up to 50% of them are moving.

claimed by [89]. Systems that use reactive MANET routing protocols are Hashline [90], Backtracking and Redundant Chord [76], MRDP [91], Bamboo/AODV [92], Convergence Chord [93], Chord in mobile ad hoc network [84] and da Hora et al.[89]. Table 3.2 lists these systems.

Hashline [90] builds a one dimensional logical space like a line and hashes keys to their corresponding points on the line. The responsibility of the keys is distributed between peers in the logical line structure with a Hashline peer being in charge of a segment of the line. To determine the logical route between nodes, Hashline optimizes a tree-structure to help in accessing a required peer. This structure is built from the graph that represents the connectivity in the physical topology. Thus, adjacent peers in the overlay are also close to each other in the underlay. When a new peer joins the overlay, it receives a portion of the Hashline from its closest peer in the physical underlay. Afterwards, the joining peer takes responsibility of this part of the Hashline. Furthermore, the joining peer considers the peer, from which it receives part of the Hashline, as its parent in the logical tree structure. On the other hand, the joining peer will be added as one of the parent's children. In order to find a key, a peer sends the request to its parent. The parent checks if it has the requested key or one of its children has it. Otherwise, the request is forwarded to the parent's parent after adding the current parent to the route list of the request.

Backtracking and Redundant Chord [76] has introduced some modifications to Chord to enhance its performance in MANET. It couples the modified versions of Chord over AODV as a MANET underlay. It proposed Backtracking and Redundant Chord. The former sets a timer for a query search. If it times out before receiving any reply, a new query is sent to another successor. The new recipient successor is the preceding peer in the Chord finger table. The time out value is set to t where t is a number between 0 and log N, and N is the number of peers in the network. However, Backtracking Chord may increase the success rate of Chord in MANET but at a cost of a higher delay to achieve the lookup response. Redundant Chord sends r queries to r successors concurrently, aiming to reduce the latency when looking up an object in the network. The number of r is set between 0 and log N as well. However, redundant queries may overload the network.

MRDP [91] integrates a DHT with AODV at the network layer. The authors claim that the DHT is only used for caching information and not for MANET routing purposes. MRDP forms nodes into dynamic clusters. It utilizes the DHT to cache information and search for resources in each cluster. When the desired object is not found in the same cluster, broadcasting starting from the originator is used to locate it elsewhere. Once the resource is found through broadcasting, the originator of the request sends (resource publish) to its cluster head. The aim of the (resource publish) is to cache information about the resource within the local cluster. MRDP requires the existence of a geometric relation between the cluster head and members. Cluster members should be physically close to the cluster head. They are n-hops away from the cluster head. In addition, cluster heads are selected according to some grades. An example of the grades is the link quality of the mobile node. So, nodes with high quality link should be chosen as cluster heads. In networks with slow mobility, the grade can be the degree of the node. Thus, nodes with a large number of neighbors should be selected as the cluster head. For cluster maintenance, MRDP adopts a proactive strategy, where each node should broadcast periodic Hello-like messages with TTL =1. Such messages contain information about the node itself, such as network grade, its neighbors and its cluster heads.

Bamboo/AODV [92] layers Bamboo over AODV. The authors evaluated different settings to find a good balance between overlay management traffic and network congestion. Bamboo operates a proactive management strategy to maintain connectivity between overlay peers regardless of overlay changes. Peers exchange periodic maintenance messages. The developers of Bamboo/AODV examined three types of management; No management, standard management and custom management. The latter was the recommended management type for Bamboo when deployed over MANET. It is claimed that with the custom setting, the protocol can achieve its best performance in term of overhead traffic and lookup efficiency. In the custom setting, the frequency of local routing and global routing update were almost doubled and the data storage update was tripled compared to the standard management.

Convergence Chord [93] was proposed to solve the convergence problem for separated Chord rings in MANET. It exploits routing layer information through cross layering to detect any neighboring node that belongs to another Chord ring. In addition to Chord routing tables, it maintains another table to detect ring separation, called a Chord Neighbour Table (CNT). The CNT stores information about physical neighbors. It gets updated from the underlay routing protocol. The overlay peer then hashes the received IP addresses and stores them in the CNT giving them a value; 0 for a new neighbor that should be identified and 1 for an already identified neighbor. Convergence Chord has two phases. It first detects separated Chord overlays, and then converges them.

Cramer and Fuhrmann [84] implemented Chord over two different reactive underlays including DSR and AODV. Their aim was to investigate the issues of deploying P2P over MANET. They concluded that the problem of deploying Chord over MANET is not the resulting overhead from the overlay but the protocol's pessimistic time out and failover strategy.

Da Hora et al. [89] also modified Chord to suit MANET and layered the proposed system over AODV. They claimed that the poor performance of Chord over MANET is due to the loss of query messages. Similar to Redundant Chord [76], Da Hora et al. modified Chord so that the originator sends the query to the n closest entries in the finger table rather than just to a single peer. This modification applies only to the query initiating node. The objective of introducing redundancy is to have different request messages that may be transmitted via different paths in the underlay. Thus, it would increase the probability of the query being solved. The receiver of the query should follow the traditional Chord algorithm. Da Hora et al. concluded that there is a trade-off between success rate, energy consumption and the number of redundancy.

3.3.1.3 *GPS-based systems*

A number of P2P over MANET approaches consider nodes' locations using GPS and use this information in the routing algorithms. Multi-Level Peer Index (MPI) [94], MeshChord [95] , Kummer et al. [96] and MANETChordGNP [97] are solutions that exploit GPS information to improve overlay operation in MANET. Some use MANET routing protocols that already use location information, whereas others use reactive routing protocols and add such functionality to the systems. The authors of MeshChord claim that using location information when assigning the logical ID of a peer contributes toward reducing Chord traffic. They further claim that most of Chord's traffic occurs between a peer and its predecessor/successor peers. Thus, assigning physical neighbor nodes with a close-by logical ID will reduce the traffic in the network. In addition, the utilization of coordinate information is seen to help predict network distance, reducing routing stretch. However, employing a topology which is dependent on nodes' location causes issues when the nodes move. Node movement changes the neighbors of a node and hence requires node ID and topology changes which in turn may require moving data between nodes as the changed Ids of nodes no longer match the data key. Furthermore, GPS is not available universally and scenarios where mobile nodes are inside buildings are difficult. Table 3.3 presents an overview of these systems.

MPI [94] assumes that each mobile node knows its location through GPS. It divides the physical network into equal-sized logical grid cells, with each cell being further divided into smaller equal-sized children cells. Data objects are hashed into geographical coordinates with the assumption that the network

Table 3.2: Reactive-based systems

System name, authors and reference	P2P system	MANET routing	Deployment method	Mobility model and node speed	Simulation/Evaluation
Hashline, Sozer et al. [90], 2009	Structured one-dimensional space. Similar to CAN.	DSR	Cross-layer	Random Way Point, Speed: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1 m\s.	Compared to flooded approach. No. of nodes varies between 10 and 100.
Backtracking and Redundant Chord, Lee et al. [76], 2004	Chord	AODV	Layered	Random Way Point, Speed: 1 m\s.	Compared in a simulation environment with Chord. 1000 mobile nodes were used with 1 m transmission range.
MRDP, Liang et al. [91],2010	Clustering	AODV	Integrated	Semi-Markov Smoth, 2-5 m\s.	Compared to Ekta and VRR. 50-200 nodes were simulated.
Bamboo/AODV , Castro et al. [92],2008	Bamboo	AODV	Layered	No	Three configurations were evaluated; No management, standard management and custom management. Simulated 36 and 49 nodes. No comparison with other systems.
Convergence Chord Mei et al. [93], 2009	Chord	AODV	Cross-layer	No	Compared to conventional Chord over MANET. 30 nodes were simulated.
Chord in mobile ad hoc network, Cramer and Fuhrmann, [84], 2006	Chord	DSR and AODV	Layered	Random Direction Model, Speed: 0, 2, and 5 m\s	For comparison, other flood-based protocol was implemented and was also compared to Chord over OLSR.
da Hora et al. [89], 2009	Chord	AODV	Layered	Random Way Point, Speed: 0, 0.25, 0.5, 1, 2, 4, 8 m\s	Compared to Chord layered on AODV without modification. 50 nodes were used.

boundary is known. For data lookups, a node sends a request to the index node in its level first. If not resolved (data outside current cell), the index node forwards the request to the index node in the next level up. The process continues until the highest level cell is reached or the index entry for the required key is found. Once the index entry is found, a location lookup procedure is invoked at the cell where the index entry was found. The aim of the location lookup is to find the peer that is responsible for the peer ID of the source node. The reply to this request will be the precise location of the source node that stores the content. When a node moves from one grid cell to another, it deletes its old index information and obtains new index information from any node in the new grid cell.

MeshChord [95] has adopted Chord and uses non-location aided routing protocols at the network layer. However, location information was used in MeshChord to include awareness of physical proximity. MeshChord was proposed for mesh networks that have stationary nodes (upper tier architecture) and mobile clients (lower tier architecture). The upper tier architecture implements the DHT Chord to locate services within the network. Each stationary node takes responsibility for a set of physically close mobile nodes. In order to exploit locality information and assign a close logical ID for peers that are physically close to each other, MeshChord uses location awareness by using GPS coordinates of the peers to calculate their logical peer Ids. Thus, two underlay neighbors are likely to also be close to each other in the overlay. However, services are assigned Ids in a similar way to traditional Chord where the hash function is used to get the ID. Moreover, MeshChord adopts a cross-layering technique to speed up the process of key lookup. Each MeshChord peer is designed to pass up any received packet at the MAC layer even it was not destined to this peer. The packet is further processed by the application layer if it was a lookup request. Otherwise, it should be discarded. In order to find a service, a mobile client sends the key request to its upper peer (the stationary node) that is responsible for this client. Upon receiving the key request, the stationary node uses Chord protocol to find the key in the network. The reply has to be transmitted to the initiator of the request through the stationary node to which the initiator belongs.

The approach by Kummer et al. [96] constructs a DHT over a Location Aided Routing protocol where the position of each node is known. It builds a logical ring similar to Chord, and keys are stored on the closest Id on the logical space according to their hash value. However, the logical overlay is organized such that each peer maintains a minimalist logical overlay and ignores the logical long range neighbors. The long range neighbors are ignored to avoid high maintenance cost. For efficient lookup, it relies on physical neighbors to find long range neighbors. Routed lookups can be diverged from their original route if a shorter route was found by an intermediate peer. This is achieved when each intermediate peer checks its logical and physical neighbors to find a closer peer to the key. Peers get information about their physical neighbors by exchanging neighborhood information messages. Such messages are propagated by single broadcasting.

MANETChordGNP [97] has also adopted Chord. It uses a non-location aided routing protocol at the network layer. However, it uses location information to include awareness of physical proximity. MANETChordGNP considers physical locality through using the Global Network Positioning System (GNP). It modified Chord protocol to position peers on the ring according to their GNP coordinate instead of hash-based placement. In GNP, a set of hosts are chosen as landmarks to serve as references for other nodes that want to participate in the coordinate space. If one of the landmarks fails, the peer with the closest logical ID to the failing one in the MANETChordGNP overlay takes over the role and acts as a landmark. Each peer is assigned a temporary unique overlay ID and whenever a peer changes its position, it requires a new overlay ID. MANETChordGNP maintains Chord routing tables and standard ADOV routing tables to carry out logical and physical routing. For logical routing, a key lookup is destined to the closest logical peer in the overlay. In order to reach the next hop in the overlay, the underlay routing protocol is consulted to carry out the physical routing.

Table 3.3: GPS-based systems

System name, authors, reference	P2P system	MANET routing	Deployment method	Mobility model and node speed	Simulation/Evaluation
Multilevel Peer Index Lee and Sivasubramaniam [94],2004	Grid structure	GPSR	Cross-layer	Random Way Point, Speed : ranging from 0 to 20 m\s.	Simulated with csim simulator. Network size: 64 - 4096 nodes with average density of 4 nodes \175 m x 175 m. Compared to flooding approach.
MESH CHORD Burresi et al. [95], 2008	Chord	DSR	Cross-layer	No	Compared to Chord, Chord with location awareness (ChordLoc) and Chord with cross-layer design. 49-144 nodes were simulated.
Kummer et al. [96], 2006	Logical ring similar to Chord	LAR (Location Aided Routing)	-	No	Static scenarios for networks of 1000, 10000 and 100000 nodes. Evaluated using LAR (Location-Aided Routing protocol) with the positions of all nodes being known through GPS.
MANETChordGNP Fantar and Youssef [97], 2009	Chord	AODV	Integrated	Speed: 9 m\s.	Compared to Chord over AODV. Network size: 50 - 200 nodes.

3.3.1.4 *Underlay-Independent systems*

In this category, approaches which improve on the performance of overlay algorithms with respect to their use on mobile networks are introduced. M-CAN [98], M-Chord [99], Enhanced Mobile Chord (EMC) [100], Enhanced Backtracking Chord (EBC) [101], PNS-Chord [102] and Ring Interval Graph Search (RIGS) [103] do not work together with any specific MANET protocol, but aim to improve the performance of the overlay regardless of the underlying MANET protocol. These systems are presented in table 3.4.

M-CAN [98] and M-Chord [99] adopt the P2P overlay of CAN and Chord respectively. Both systems use grouping and registration mechanisms to improve the performance. M-CAN uses super-nodes to improve the performance of CAN in mobile networks. Super-nodes are nodes selected according to their capacity and connection reliability to manage a range of the ID space. Each group is restricted to manage a certain number of nodes. If the number of nodes in that group exceeds the maximum number, the group should be divided into two groups. A new super node for the new group is announced by the old super node. In order to join the M-CAN overlay, a node registers itself with super nodes according to the Ids of the shared files. Thus, a node may register with more than one super node. To find a key, peers send their requests to their super-node which in turn forwards the request to the destination super node using the CAN algorithm. The destination super node checks its local directory to check whether the destination node is registered with it or not. If the node was found, a reply with the destination address is sent to the request initiator. Otherwise, the request message gets broadcast to other groups by the destination. This procedure is repeated until the request is solved.

M-Chord uses a hierarchical structure and registering mechanism to allow normal peers to register with super-peers which are selected depending on their computing capability and their connectivity. Ordinary nodes register with super nodes according to the shared files, hence a peer may register with more than one super peer. A peer that does not have any file to share would be registered with a super node that has a low load. In addition, super nodes are also required to register with other super nodes and are required to store their information of shared files at their super nodes. The logical space is distributed between the super peers, where each super peer maintains a range of the logical space. The Chord protocol is used for routing between super-peers. In addition to maintaining a Chord routing table, each super peer has to maintain a directory of all the shared files registered on it. Each node sends the request to its super node, which in turn routes the request using standard Chord. Upon receiving the request, the destination super peer checks its directory of shared files to find the requested key. If the requested key is found, the destination super peer replies to the initiator with the address of the peer that is storing the file. In order to retrieve the file, a direct communication then takes places between the source peer and the peer storing the file.

EMC [100] modifies Chord to enhance its performance in MANET, where criteria such as delay, packet loss rate and jitter were taken into account. EMC includes a path selection mechanism based on a set of parameters such as application type and node stability. It uses periodic pinging to provide information about path latencies. Each EMC peer sends a periodic message to all of its successor peers in its finger table. Such ping messages are also optimized for updating the finger table. A peer can then compare the latencies to a threshold and uses the paths with latency less than the specified threshold. Exchanged ping messages are also used to determine another parameter called node stability. To calculate node stability of a particular peer, each peer keeps track of the number of received ping messages from that peer. The path cost can then be calculated as a combination of the node stability and latency.

Similarly EBC [101] builds a Chord-based overlay on MANET. It combines aspects of EMC and Backtracking Chord [76]. It uses the retransmission approach from

Backtracking Chord, which is claimed to enhance the success ratio of the P2P system. For each sent key request, the initiator sets a timer. If it is timed out without any reply, a new copy of the key request should be sent. For the purpose of retransmitting the request, the EBC peer first checks the same entry to that successor to find if there is a second best path. The second best path is then used for retransmitting the key request. Otherwise, the strategy of backtracking Chord would be used, where the key request is retransmitted to the next successor in the finger table. Moreover, EBC also adopts the path selection mechanism from EMC. Again, periodic pinging is used to provide information about the routes latencies. The typical Chord routing table is changed to store these new routing parameters. For each finger in the routing table, a peer maintains extra two fields; node stability and path delay.

PNS-Chord [102] has used Proximity Neighbour Selection (PNS) to enhance the performance of overlay in an ad hoc network. PNS-Chord uses PNS to mirror the physical proximity in the logical Chord overlay. In order to measure distance between physical nodes, the number of hops was used instead of latency when PNS is used with traditional internet. A node in PNS-Chord selects its logical neighbors considering their physical proximity, which helps to avoid routing stretch. In PNS-Chord, nodes should maintain information about their physical neighbor nodes. This is achieved by exchanging hello-like messages between nodes at the underlay routing level. Each node should broadcast a periodic hello message to its neighbors. In addition, a node should also append information about its currently known neighbors to the hello message. This allows a node to know about neighboring nodes that are 2 physical hops away. Such information is then provided to the overlay to build a logical structure that better matches the physical structure.

Ring Interval Graph (RIG) was used by RIGS. The approach presented in [103] builds an underlay-dependent DHT where one hop neighbors in the overlay are also one hop neighbors in the underlay topology. Node Ids are not assigned in a random way; but rather in a way that generates a RIG. For this a spanning tree of the entire network is needed. Upon constructing the spanning tree, each node is assigned an ID in an increasing order along the spanning tree. Once the Ids are assigned, a circular logical space is built for the overlay system as in Chord. A key item is hashed and assigned to the peer with equal or greater logical ID in the logical space. In order to route a key request to final destination, a peer needs only to know local routing information that would lead to a step closer to the destination. RIGS uses a continuous interval hash space [0,1) instead of the discrete set used by Chord {0,1, ... 2^m}. When a node joins or leaves the system, the RIG and the overlay needs to be reconstructed so the matching between the overlay and underlay is maintained.

3.3.2 *P2P-Based MANET routing protocol systems*

This group of approaches uses techniques from P2P overlays to improve MANET underlay routing protocols. These approaches do not use the DHT in the application layer, but at the network layer, to enhance the scalability of MANET routing. In this setting, the DHT is used to distribute MANET node location information throughout the network. This information would reflect the topological positions of the nodes. However, some of these systems support traditional P2P functionality like resource distribution and discovery, but at the network layer. Approaches falling into this category can be further divided according to the P2P structure used. There are ring-based systems, tree-based systems and systems based on other structures. Virtual Ring Routing (VRR) [104], Scalable Source Routing (SSR) [105], DPSR [106], Ekta [78], MADPastry [107] and MA-Chord [108] belong to the ring-based MANET routing systems (Table 3.5). Tribe [109], DART [110], Augmented Tree-based Routing (ATR) [111], Indirect Tree-based Routing (ITR) [112] and KDSR [113] are tree-based MANET routing approaches (Table 3.6). DHT-based functionality using Hypercube [114], Automatic Incremental Rout-

Table 3.4: Underlay-Independent systems

System name, authors and reference	P2P system	Simulation/Evaluation
M-CAN, Peng et al. [98], 2004	CAN	Compared to flooded requests protocol and centralized directory protocol. 50-500 peers were used in the simulation.
M-Chord, Li et al. [99], 2006	Chord	Comparison was done between normal Chord, super M-Chord nodes and ordinary M-Chord nodes. No. of nodes varies from 128 to 2048.
Enhanced Mobile Chord, Thaalbi et al. [100], 2011	Chord	Compared against native Chord and Backtracking Chord. Simulates 100 nodes in a dynamic network "by changing the churn rate" and 50-250 nodes in a static network.
Enhanced Backtracking Chord, Thaalbi et al. [101], 2012	Chord	Compared to EMC and Backtracking Chord in static network consists of 50-250 nodes and in a dynamic network "by changing the churn rate" that consists of 100 nodes.
PNS-Chord, Cramer and Fuhrmann [102], 2005	Chord	Simulated but not compared to other systems. 100-1000 nodes were used but no mobility model nor speed was simulated.
Ring Interval Graph Search, Shin and Arbaugh [103], 2009	Chord	Compared to Chord, VALLEYWALK (unstructured P2P for ad hoc) . 100 nodes were simulated in a static network.

ing (AIR) [115] and 3D-RP [116] are approaches which adopt various structures such as hypercube, acyclic graph and 3D rectangular structure (Table 3.7).

3.3.2.1 *P2P ring-based MANET routing systems*

VRR [104] uses location independent Ids. It organizes nodes into a virtual ring similarly to Chord and Pastry. VRR supports traditional point-to-point routing and DHT routing. VRR nodes maintain paths to their virtual neighbors and a set of physical neighbors. A VRR node maintains a virtual neighbor set. The virtual neighbor set is divided into two halves. The first half stores closest neighbors clockwise in the logical space. The second half is used to store the closest nodes anti-clockwise in the identifier space. Each routing entry in the virtual neighbor set records the address of the next physical node and the address of the virtual neighbor. A node also maintains a physical neighbor set to store nodes which are in the communication range of this node. Routing tables are updated in a proactive way even when there is no traffic going through the paths.

Similarly, SSR [105] creates a Chord-like ring at the network layer. SSR nodes maintain their physical neighbors, virtual successors, virtual predecessors, and some cached information. SSR was proposed to deliver indirect routing using DHT-like routing and direct routing between nodes. To route a packet to a destination, a node consults its routing tables to find the destination or a closer node to the destination. The route cache stores source routes to the virtual successors and predecessors. Each transmitted data packet has a source route to follow. The source does not necessarily lead to the final destination, but to a virtual node which is closer to the final destination. A source route can be altered by any intermediate virtual node if a short-cut source route is found. According to the results presented, this approach suits growing networks that are supported by some fixed wired nodes. It is most effective at low mobility.
DPSR [106] proposes a MANET source routing protocol that combines a MANET routing protocol and a P2P system. It integrates Pastry with the DSR routing protocol. It limits the number of source route that a node needs to maintain to O

(log N) and operate DHT. Each node is assigned an ID by hashing its address. DPSR changes the content of the routing table entry and leaf set of the original Pastry algorithm to store the source route to reach a destination instead of storing its IP address as in Pastry. DPSR routes a key request similarly to Pastry where a key is routed to the node with the closest logical ID. It also adopts the same routing table maintenance mechanism as Pastry where for each row in the routing table, DPSR picks a random entry from the row and asks that node for a copy of the corresponding routing table entry.

Ekta [78] and MADPastry [107] have also integrated Pastry with DSR and AODV routing protocols respectively at the network layer. Ekta is an enhancement of DPSR. It integrates the P2P system with the underlay protocol via one to one mapping between underlay address and logical ID. It builds a separate overlay at the network layer and does not use the DHT for unicast routing. Similarly to DPSR, Ekta modifies Pastry routing tables to store a source route for each entry instead of storing the IP address of the destination. Prefix based routing is used in Ekta to route a key to a node with a logical ID closer to the destination. Ekta inherits the functionality of DSR, such as route discovery and maintenance. For updating routing table entries, Ekta also optimizes overheard and forwarded packets to maintain fresh routes. MADPastry combines AODV and Pastry at the underlay to provide indirect routing for MANET. It has utilized random landmarking to group physically neighbor nodes in a cluster. A set of nodes in the same physical cluster share a common overlay ID prefix. MADPastry divides the space into landmark keys. A node can temporarily be the landmark and takes the responsibility for a set of landmark keys. In case of landmark node failure, a node in the same cluster with the closest ID to the failing one will take the responsibility for the landmark keys. Landmark nodes broadcast a periodic message to allow other nodes to measure their distance from the landmark nodes. In order to limit the traffic, such messages are forwarded only by nodes that belong to the initiator landmark node. If the node belongs to different cluster, it just stores the information and discards the message. Each node that receives the message should determine which landmark node is closer to register with it. MADPastry maintains three routing tables; an AODV routing table for underlay routing, a Pastry routing table and a leaf set for indirect routing.

MA-Chord [108] combines AODV and Chord at the network layer to provide

Table 3.5: P2P ring-based MANET routing protocol systems

System name, authors and reference	P2P system	Mobility model and node speed	Simulation/Evaluation
VRR , Caesar et al. [104], 2006	Structured DHT similar to Chord and Pastry	Random Way Point, Speed: randomly selected from 0-20 m\s	Compared to DSR, AODV and DSDV. 50 nodes were simulated with and without mobility. The system was also evaluated in a 802.11 a testbed containing 30 PCs and in a sensor network containing 67 nodes.
SSR, Fuhrmann et al. [105], 2006	Chord	Random Way Point, Speed: varied up to 4 m\s	Compared to AODV and DSR. In pure MANET, it was only compared to AODV (450 nodes). When the speed is higher than 2 m\s, SSR performs worse than AODV.
DPSR, Hu et al. [106], 2003	Pastry	No	No performance evaluation.
Ekta, Pucha et al. [78], 2004	Pastry	Random Way Point Model, Speed: uniformly distributed 1-19 m\s	50 nodes were used for simulation in 1500 m x 300 m. Compared to as it is Pastry layered over DSR.
MADPastry, Zahn and Schiller [107], 2005	Pastry	Random Way Point, Speed: 0.1, 0.6, 1.4, 2.5, 5.0 m\s	Compared to Gnutella style broadcast routing and to regular Pastry, without clustering, over AODV. 100 and 250 nodes were used with a node density of 100 nodes per square km.
MA-Chord, Meng and Ji [108], 2007	Chord	Random Way Point, Speed: 1.4 m\s	Simulated and compared to AODV. Node density of 100 nodes per square km.

DHT routing in MANET. It applies random landmarking and uses fixed landmark keys to exploit physical proximity. MA-Chord maintains an AODV routing table to serve physical routing. In addition, it also has a Chord routing table for DHT routing. The Chord routing table is maintained in a proactive way where each MA-Chord node sends periodic message to its successors and predecessors in the overlay. Other routing entries are updated through overhearing network traffic. Each node is assigned an overlay ID whenever it joins the overlay or moves to a different cluster. Each node publishes its own overlay ID with an overlay key node which works as a temporary address server. For routing purposes, a node needs first to retrieve the destination's overlay ID from the destination's address server. After retrieving the overlay ID, a node can then use it in MA-Chord overlay routing.

3.3.2.2 *P2P tree-based MANET routing systems*

Tribe [109] builds a tree-like logical structure of the physical topology. Neighboring nodes in the underlay topology are represented in a close logical region of the address space. Each Tribe node maintains information about itself, its immediate neighbors and any nodes under its responsibility. Each tribe node has three identifiers; universal ID, logical Tribe ID and relative ID. The relative address is topology dependant and changes when a node moves to another location in the network. When a node joins the network, one of its physical neighbors with the largest portion of the ID space, will share its logical space with the new node. The joining node then needs to register with an anchor node which is responsible for this logical ID. The anchor node stores information about the joining node such as its location (relative ID). To find a node, the source node needs to contact the destination's anchor node to get the location of the destination. It can find the anchor node for a destination by hashing the destination universal ID. Once the source node finds the location of the destination, it can communicate with it.

DART [110] builds a logical tree structure on top of an underlay to implement hierarchical routing. The constructed logical tree does not represent the exact physical links between nodes. However, neighbor nodes in the logical address space should be relatively close to each other in the physical topology. Each DART node has its routing address and a permanent identifier. Routing addresses may change due to node movement as they are assigned according to node position. Any branches of a leaf share a prefix of the routing address with its vertex leaf. A node maintains a routing table that stores an entry for each sibling using proactive distance-vector. DART uses a distributed lookup table to map an identifier to an address on the network. A DART node needs to register its (identifier, routing address) with another node that acts as an anchor node. To achieve this, a node hashes its identifier and the result is the anchor node where this node should store its (identifier, routing address) pair. In routing, a node can hash the destination identifier to find its anchor node where the destination's routing address is stored.

ATR [111] builds a similar structure to DART, where it builds a binary tree of L+1 levels (L being the number of address bits). In the tree structure, the set of nodes in the same sub tree level shares an address prefix. Thus the number of shared bits in nodes' addresses expresses the distance between them in the topology. ATR augments the tree structure to offer more than a single path between nodes. ATR maintains and explores all the possible paths through its neighbors. It uses periodic hello messages to build and update the routing table. A routing table for an ATR node stores multiple entries per each sibling to achieve multi path routing in the tree structure. Similarly to DART, DHT is used for address lookup. Each ATR node registers its network address and identifier with a node in the network depending on the hash value of its identifier. When a node wants to retrieve the network address of a destination, it hashes the identifier of the destination to find the node which is responsible for this destination. It can then get the destination's network address from that node.

ITR [112] extends ATR. ITR presents the identifier space as a tree structure to assure agreement between logical and physical structure. It is quite similar to its predecessor. However, it provides full functional P2P services where nodes can store resources and fetch them. It assigns each ITR node a location-dependant identifier. It divides the routing table in each node to as many sections as the number of siblings. An n-th section stores the next physical hop which can be used to deliver a packet to the level n sibling in the logical space. Nodes need to register their identifier with an anchor node periodically, as in ATR. The same procedure applies for the shared services, where a node is required to periodically send a pair of (resource identifier, storing peer identifier) to the node responsible about this service according to the hash function.

KDSR [113] integrates the functionality of Kademlia with DSR to provide indirect

Table 3.6: P2P tree-based MANET routing protocol systems

System name, authors and reference	P2P system	Mobility model and node speed	Simulation/Evaluation
Tribe, Viana et al. [109], 2004	Tree structure	-	Evaluated in simulation but was not compared to any other system. Simulates a 1000-10000 network size.
DART, Eriksson et al. [110], 2007	Tree structure	Random Way point, Random speed between 0.5 m\s and 5 m\s	Compared to AODV, DSR and DSDV with up to 800 nodes.
ATR, Caleffi et al. [111], 2007	Tree structure	Random Way point, random speed between 0.5 m\s and 5 m\s	Compared to DART, AODV and DSR in simulation.
ITR, Caleffi and Paura, [112], 2009	Structured DHT	Random Way Point, Speed: between 0 and 5 m\s.	Compared to MADPastry. No. of nodes: 50.
KDSR, Zhao et al. [113], 2009	Kademlia	Random way point, Random speed between 0 and 20 m\s.	Compared to DSR in a simulation with 50 nodes network size.

and direct routing in MANET. KDSR adopts the Kademlia XOR-based algorithm. It also builds a route cache and adopts the use of non-propagating route requests to minimize network traffic. Each KDSR node maintains a k-buckets routing table as in Kademlia. However, the content of each entry is modified to suit MANET. Each entry stores a four tuple that consists of node ID, IP address, XOR distance to the node, and the source route to reach the node in this entry. In addition, KDSR also maintains a cache route similar to DSR. KDSR optimizes the k-bucket update mechanism to learn new source route entries, which are stored in the cache route.

3.3.2.3 *MANET routing systems employing other P2P structures*

DHT-based functionality using Hypercube [114] constructs a hypercube-like address space to build a multi path unicast routing. It translates the physical network into a logical network through the use of node coordinates in a hypercube. Each node in the hypercube manages a portion of the address space which can be split with any joining node. For each dimension, a node in the hypercube must have coordinate 0 or 1 to cover all the possible combinations. For example, a node in the hypercube with logical (0,0,0) coordinate has links to nodes with coordinate (0,0,1) , (0,1,0) and (1,0,0). These nodes differ from (0,0,0) in 1 dimension only.

AIR [115] builds a Labeled Directed Acyclic Graph (LDAG). The LDAG structure is constructed with reference to an elected root node. AIR maintains a DHT to minimize the use of flooding in MANET. The DHT is used to provide mapping of a node identifier such as the IP address to prefix label. It adopts prefix based routing where a node fetches the prefix label of a destination node. Each AIR node hashes its identifier to get its prefix label and then announces its presence by sending a publish message to its anchor node, which is the node that best matches its prefix label. Each node in the network maintains a neighbor table, a two-hop neighbor table and a multicast group table. AIR uses hello messages as the control messages to exchange information between nodes.

3D-RP [116] builds a three-dimensional logical structure on top of the network topology. A 3D-RP node calculates its m-bit logical ID in form of a tuple $\{x \mid y \mid z\}$

where each one of the axis takes a value in the range 1 to $\pm 2^m$. A Hello messages are exchanged between one hop physical neighbors to maintain the 3D logical space. Hello message contains information about a node, such as logical ID, universal ID (IP address), logical space portion, and the current node's logical one hop neighbors. Using a hash function, a node can identify its anchor node by hashing its universal ID (IP). Once the anchor node is identified, the node can store its logical identifier and its universal identifier on the anchor node, thus allowing other nodes in the network to find its logical identifier. 3D-RP nodes route messages towards their logical neighbor that has the closest position in every tuple {x|y|z}.

Table 3.7: MANET routing protocol systems based on other P2P structures

System name, authors and reference	P2P system	Mobility model and node speed	Simulation/Evaluation
DHT-based Functionalities Using Hypercubes, Alvarez-Hamelin et al. [114], 2006	Hypercubes structure	-	-
AIR, Garcia-Luna-Aceves and Sampath [115], 2009	Labeled directed acyclic graph	Random way point	Compared to AODV and OLSR as unicast routing.
3D-RP, Abid et al. [116], 2013	3D rectangular structure	Random way point, Random speed between 0.4 m\s and 1 m\s.	Compared to M-DART in a simulation with 25-400 nodes.

3.4 DISCUSSION

3.4.1 *Adopted P2P system*

Considering the used overlay algorithms, we find that most of them are well known algorithms commonly used on wired networks. They were introduced in the MANET context to support either data dissemination or location distribution for MANET routing purposes. However, most of the proposed systems concluded that straight deployment of wired internet technology for P2P does not suit MANET networks. Therefore, most of the proposed systems have introduced modifications to the original P2P technology to suit MANET.

Chord, Pastry, CAN, Bamboo and Kademlia were used and modified to work in MANET. Specifically, Chord and Pastry were the most commonly used DHT protocols on MANET for data distribution in mobile ad hoc networks. Approximately half of the studied systems have used Chord or a system that builds a logical structure similar to Chord. Pastry was also used by well-known systems such as CrossROAD, DPSR, Ekta and MADPastry. Moreover, the tree structure of the logical overlay was also adopted by a number of proposed systems. RIGS, MPI, DART, ATR and ITR are examples of systems that build an overlay with a tree structure. They build such a structure at the network layer to support scalable MANET routing. However, such a tree structure typically creates substantial traffic, especially when considering node mobility, as this frequently requires reconstructing the overlay. Another issue with tree structures is the possibility of a disconnected sub set of the network in the event of node failure. This is due to single paths between nodes in the tree structure. To overcome this weakness, ATR modified tree structure to support multiple paths between nodes in the network. In addition to the aforementioned logical structures, some of the studied systems built 3D- rectangular and hypercube structures.

3.4.2 *Clustering and registration*

A number of systems make use of a clustering mechanism to structure the overlay. Examples of systems that follow this paradigm are M-Chord, M-CAN, MADPastry, MRDP, P2P MMOG, ROUBST and MA-Chord. Some of these systems employ a second mechanism to organize the cluster heads. For instance, M-Chord uses Chord among super peers, and M-CAN adopts CAN to organize cluster heads.

A drawback of using clustering is that it requires the cluster members to register with the cluster head. The registration can be done the first time a peer publishes a file, as in M-CAN and M-Chord. Each super-peer is allocated a range of Ids it is responsible for. When publishing a file, a peer calculates the ID of the file then registers with the super-peer that is responsible for this file's ID. This approach is independent of a node's physical location. Another form of registration requires peers to register each time a peer changes location and moves into the territory of a different cluster. Clearly, in networks with substantial node mobility such an approach creates significant overhead and would not be efficient. However, it may be advantageous in mostly static networks.

3.4.3 *Consideration of physical proximity*

Generally, constructing a P2P over MANET system which operates independently of the physical proximity of nodes leads to a longer path of messages at the MANET layer. This is due to the fact that nodes which are close in the constructed virtual network may be physically distant. So a hop in the overlay translates to a path in the underlay. Thus, building a P2P over MANET system considering the physical proximity between nodes, i.e. the overlay closely maps the physical topology, can substantially reduce traffic in the network. Examples of approaches that consider node proximity are MADPastry, Hashline, PNS-Chord, RIGS, MRDP, P2P MMOG, MA-Chord and MANETChordGNP.

Mapping the physical topology closely at the overlay can result in more traffic, especially when node mobility is high. In order to maintain the relationship between the overlay and the underlay, additional maintenance traffic is required. In order to consider physical proximity, some of the proposed systems use landmarking, clustering or building logical tree structures. However, with the presence of node mobility, it is more expensive to maintain the consistency of the network. This problem is exacerbated in systems that use logical Ids which are based on node locations.

On the other hand, some of the systems do not consider physical node proximity. They build overlay networks independent of the MANET physical topology. These approaches introduced some modifications to existing P2P algorithms to adapt to MANET networks. EMC, EBC, Backtracking Chord, Redundant Chord, M-CAN and CrossROAD are examples of such systems.

3.4.4 *Deployment strategy*

The first and easiest deployment strategy is to deploy the overlay directly on top of the existing MANET architecture without any direct communication between the layers. This design conforms to the TCP/IP layering architecture. The layered approach was used by Backtracking Chord, Redundant Chord and Bamboo/OADV. Backtracking Chord and Redundant Chord introduce some modifications to the P2P Chord protocol to improve the efficiency of the proposed system. However, systems that follow the layering approach run logical routing fully independent of the physical underlay routing. This type of deployment usually results in nodes that are logically close in the overlay being physically distant in the underlay. This substantially increases lookup traffic and hence results in a comparably poor performance of the system.

Another approach is the integration of the P2P protocol with the underlay routing protocol. Examples of systems that have adopted the integration approach are DPSR, Ekta (which was an enhancement of DPSR), MADPastry, Hashline, MRDP, VRR, SRR, Tribe, DART, ATR, ITR, 3D-RP and MA-Chord. Systems that follow this type of deployment have different purposes. Some of them were proposed to provide P2P functionality at the MANET routing layer. However, most of the systems that employed DHTs at the network layer aimed to improve the routing performance. Examples of these systems are VRR, Tribe and ITR.

However, it has been argued that since P2P overlays and MANET protocols have a number of commonalities, common information should be exploited to improve the performance of combined systems. This resulted in cross-layering, where P2P overlays operate at the application layer and MANET protocols operate at the network layer, with the two systems sharing information. To enable the exchange of such information, a violation of the TCP/IP architecture is needed. By permitting non-adjacent layers to communicate, significant performance improvements can be achieved due to the reduction of duplicate efforts in the two layers. For example, CrossROAD uses an external data sharing module which holds the shared information. Another example is MA-SP2P, where underlay routing information was used to identify the number of physical hops between logical peers.

3.4.5 *Network routing protocol*

With regard to the adopted underlay routing protocols, they generally belong to three MANET routing protocol categories, which are proactive, reactive, and location based routing protocols. However, some of the studied systems have proposed their systems as a MANET underlay routing protocol. The most popular category in the studied systems is the reactive routing protocol category, which establishes routes on demand. In particular, AODV and DSR seem to be very popular. They were used by P2P MANET systems that adopted the layering and cross-layering strategy. In addition, they were also integrated with DHT at the network layer in some systems to enhance MANET routing and provide direct and indirect routing in MANET. AODV was used by MADPastry, Backtracking Chord, Redundant Chord, MRDP, MA-Chord and MANETChordGNP. DSR was used by DPSR, Ekta, MeshChord, Hashline, KDSR and SSR.

On the other hand, OLSR, a proactive protocol, was used by some of the analyzed systems. Despite the fact that there are a number of proactive protocols, OLSR is the only proactive protocol that has been used for deploying P2P over MANET across the studied systems. Examples of systems which employ a proactive routing protocol (OLSR) are CrossROAD, MA-SP2P, 3D and ROBUST. Compared to proactive protocols, reactive systems result in a higher routing latency since they establish routes to other nodes on demand. However, perhaps rather unexpectedly, reactive systems can also result in a higher load on the system than proactive systems. This is especially true in more dynamic networks with high demand, where many route requests and replies propagate throughout the network [84].

Moreover, some of the reviewed systems have used routing protocols that optimize node location information through the use of GPS. The use of GPS information can help to predict network distance when routing, hence reducing routing stretch. However, building a logical ID that depends on the global position of a node requires increased overheads to maintain the consistency of the network when nodes move, requiring a new logical ID and reassigning contents to different nodes.

3.4.6 *Performance evaluation*

Most of the studied systems have used network simulation tools for evaluation. In order to evaluate their proposed systems, they usually simulate the system, and then compare it to other systems. The compared to system varies between another P2P over MANET solution, a MANET routing protocol or straight layering of normal P2P protocol over MANET. Another approach for evaluation that was adopted is to compare the efficiency of the proposed system to the mobile ad hoc routing protocol. This was quite popular for systems that introduced their solutions as a replacement for traditional MANET routing.

However, not all the studied systems have used the simulation tools to evaluate the proposed system. In fact some of them have used a testbed containing a small number of nodes for evaluation purposes, instead of using a simulation environment. For example, CrossROAD used eight nodes in a testbed experiment to analyze the performance of the system. Another example that was evaluated using a testbed is VRR, where 30 PCs were used in the performance evaluation. In addition, some of the proposed systems did not evaluate the performance of their solutions. For example, DPSR and P2P MMOG over MANET did not evaluate their systems.

A number of proposed systems have not included node mobility in their evaluation. For example, Kummer et al. [96] and PNS-Chord did not consider mobility. Clearly, simulating static networks does not reflect the performance of the system in MANET with node mobility. However, some of the proposed systems have studied the effect of churn, where nodes join and leave a static network and consider it as mobility in the network. Other systems consider mobility and they state how much velocity was considered when simulating their systems. Looking at the considered mobility we can find that most of the approaches that considered mobility, consider a node speed that is randomly selected from the range 0 m/s - 2 m/s. Few of the studied systems have considered a higher mobility of up to 20 m/s. Ekta and MPI choose the mobility speed randomly up to about 20 m/s. It appears that most of the simulation-evaluated systems use the Random Way Point mobility model to simulate mobile node movement within the simulated area.

3.4.7 *Conclusions*

Despite the fact that some systems adopt cross-layering, the resulting synergies between the two technologies need to be exploited more. Specifically, exploitation of the underlay routing protocol information when building the overlay needs to be enhanced. This would contribute towards building a less costly overlay. So far work has focused on using multi-hop overlays with MANET. Most MANET networks are characterized by a relatively small number of participating nodes which may be mobile. Consequently, investigating the combination of MANET with a one hop overlay could be worthwhile. Considering small to medium sized MANET networks, such a solution is not prohibitively expensive, especially when used on a proactive underlay.

3.5 SUMMARY

In summary, this chapter discussed the state of the art with regard to deploying P2P networks over mobile ad hoc networks. It introduced key challenges when undertaking such a deployment, and the approaches of carrying out such deployment were also introduced. These are straight layering of P2P technology over MANET, integration design and the cross-layering technique. The chapter then examined existing systems that optimized P2P functionality for MANET. Existing systems were classified in accordance with the use of the DHT. DHT was used for different purposes. It was used by some systems to enhance the performance of MANET routing protocols, and was also used to serve in a similar way as used in P2P over traditional internet. Systems that used DHT for the latter purpose were further divided according to underlay protocol used. These are proactive-based systems, reactive-based systems, GPS-based systems and systems that were independent of the underlay routing. The chapter then discussed different aspects of deploying P2P over MANET and how they were addressed by the studied systems.

4

TOWARDS ONE HOP OVERLAY FOR MANET

In this chapter, a novel system, OneHopOverlay4MANET, that builds a one logical hop overlay for MANETs is presented. First, experiments are performed on one of the most significant challenges faced when deploying a structured P2P overlay over MANETs; the mismatching issue. In addition, it is proposed that the deployment of a one logical hop overlay system can solve the mismatching issue between the logical overlay and the physical topology. Experiments were carried out to investigate the feasibility of the proposed solution. In doing so, multi hop overlay systems and another P2P system that can achieve one hop performance over MANETs are deployed. All the deployments of these systems comply with the IP stack structure. The result of these deployments was then discussed.
The novel system, OneHopOverlay4MANET, is then presented. The aim of the system is to achieve one logical hop performance for P2P over MANETs at minimum cost, optimizing the cross-layering mechanism to exploit synergies between the overlay and MANET, thus leading to less lookup latency when retrieving resources from the network and preventing criss crossing of the network when routing a resource request at the overlay. The proposed system, OneHopOverlay4MANET, is then described in detail. It is then evaluated compared to EpiChord and Chord when layered, as they are, over MANETs.

4.1 DEPLOYMENT OF TRADITIONAL P2P OVERLAY OVER MANET

4.1.1 *Mismatching issue between logical overlay and physical topology*

As discussed in the previous chapter, the most faced challenge when combining MANETs 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 virtually neighbors in the overlay are usually separated by many physical hops in the network. 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 aim to avoid such inefficient routing paths.

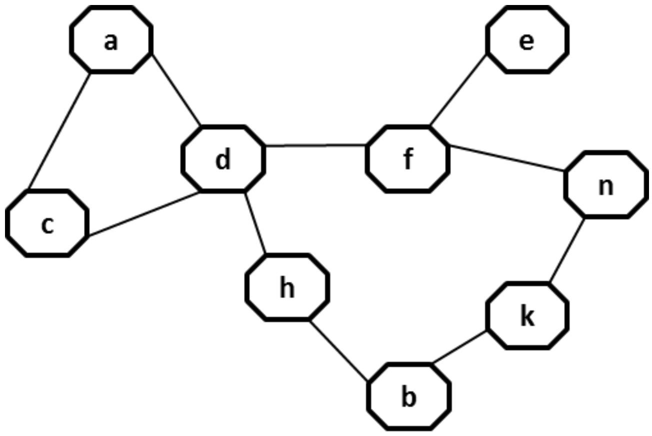


Figure 4.1: MANET Network

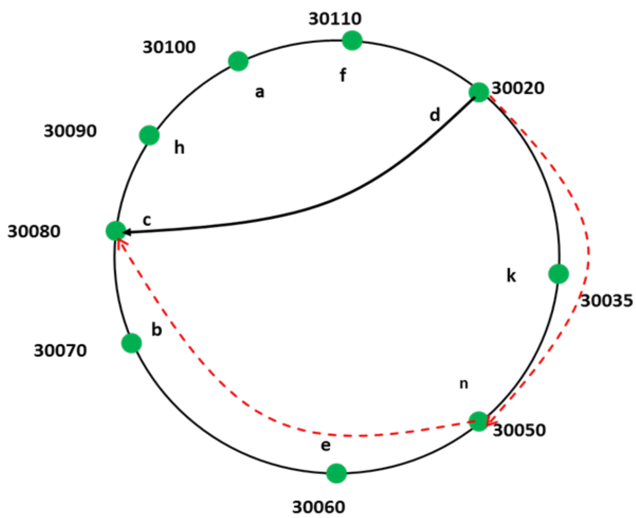


Figure 4.2: Logical Overlay over MANET Network

Figures 4.1 and 4.2 illustrate the mismatching issue between the logical overlay and the physical topology of MANET. Figure 4.1 shows a mobile ad hoc network topology consisting of 9 nodes. Figure 4.2 shows an example overlay structure of the same network. Imagine an example where the node with physical ID d and logical ID 30020 (See figure 4.2) wants to find a key that resides on node c with logical ID 30080. In a typical multi hop overlay, the query will go through a number of overlay hops (See figure 4.2 dashed path) to reach node c. However, looking at the physical topology (Figure 4.1), node c can be accessed in a single physical hop since it is a neighboring node to d in the underlay. Thus, one-hop overlays would avoid criss-crossing the MANET network when operating logical routing to reach a peer that might be close in the physical network, consequently reducing the required time to fetch the desired key. A one logical hop overlay system would push the logical overlay to understand the physical topology, avoiding criss-crossing MANET.

4.1.2 *Straight deployment of structured P2P systems over MANETs*

Many of the traditional structured P2P systems were deployed as they are over MANETs. The goals of these deployments were to investigate their efficiency when deployed over MANETs and to find a way to improve such performance. In addition, some of the work that did layer P2P systems over MANETs aimed to compare their performance when deployed as they are to their proposed systems' performance. The well known structured P2P system Chord for example was deployed over MANET by [84] to evaluate its performance in mobile ad hoc networks. Moreover, it was also deployed as it is over MANET by [76], [93], [89], [95], [97], [99], [100] and [103]. Another P2P system that was deployed over MANETs by different research is Pastry. It was deployed over MANETs as it is by [107], [78] and [77]. The Bamboo DHT P2P system was also deployed over an ad hoc network by [92] and [85].

Looking at these systems, it can be noticed that all of these systems are multi hop

P2P systems regarding the overlay lookup performance. To the best knowledge of the author, non of the existing works has investigated the performance of traditional one hop P2P systems over mobile ad hoc networks. A clear reason to avoid such an evaluation in MANETs is the expected heavy traffic that a traditional one hop P2P system may produce in a scarce network like MANET. In this chapter, a P2P system is deployed that can achieve one logical hop performance (EpiChord) over MANET to evaluate a one hop P2P system in the MANET environment. In addition, multi hop structured P2P systems (Chord, Pastry, and Kademlia) are deployed as they are over MANETs to compare the performance of one hop P2P system to the performance of multi hop P2P system when deployed over MANETs as they are.

As mentioned previously in section 3.2, P2P overlay systems can be deployed over MANETs using three different approaches. They can be layered as they are over MANETs, cross-layered with the adopted MANET underlay routing protocol, or integrated with the underlay routing protocol at the network layer. In order to evaluate the performance of P2P multi hop overlay systems and a one hop overlay system as they are over MANETs, four different P2P systems were layered over the used underlay routing protocol. Figure 4.3 shows the adopted straight layering design.

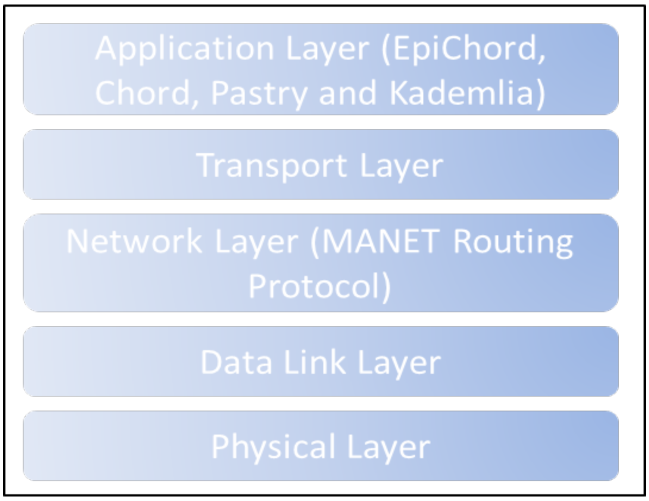


Figure 4.3: Straight layering for existing P2P systems over MANET

According to the layering approach, the layered systems (EpiChord, Chord, Pastry and Kademlia) which reside at the application layer, will operate their own logical routing independently from the MANET underlay routing protocol which operates its own routing at the network layer. The objective of such deployment is to investigate which logical overlay routing mechanism would suit MANETs better. Specifically, how the one hop overlay would behave when deployed in MANETs.

4.1.3 *Straight layering result*

In order to evaluate the performance of the aforementioned overlays over mobile ad hoc networks, EpiChord, Chord, Pastry and Kademlia overlays are simulated in an ad hoc environment using the Oversim package simulator and the INET package simulator for MANET routing protocols in the OMNet++ network simulation environment.

All the overlay protocols were simulated with the same setting using OLSR as the underlay routing protocol. The number of participating peers in the network was increased from 30 up to 90 in steps of 20 mobile nodes. Nodes were moving inside the simulation area (1000 m x 1000 m) at 5 m\s speed. For each simulated scenario, the network was left for about 120 seconds to stabilise. The simulation then runs for 600 seconds for each simulated scenario. Every peer in the overlay sends a file lookup query with a different frequency for different scenarios. Two different sets of lookup frequencies were used (30 s and 5 s). The Random Way Point was used as the mobility model. The performance of the deployed four overlays was investigated in terms of how many lookups were resolved (Success Rate), how much traffic was released in the network (Network Load), and finally, the average required logical hops in order to resolve a key lookup.

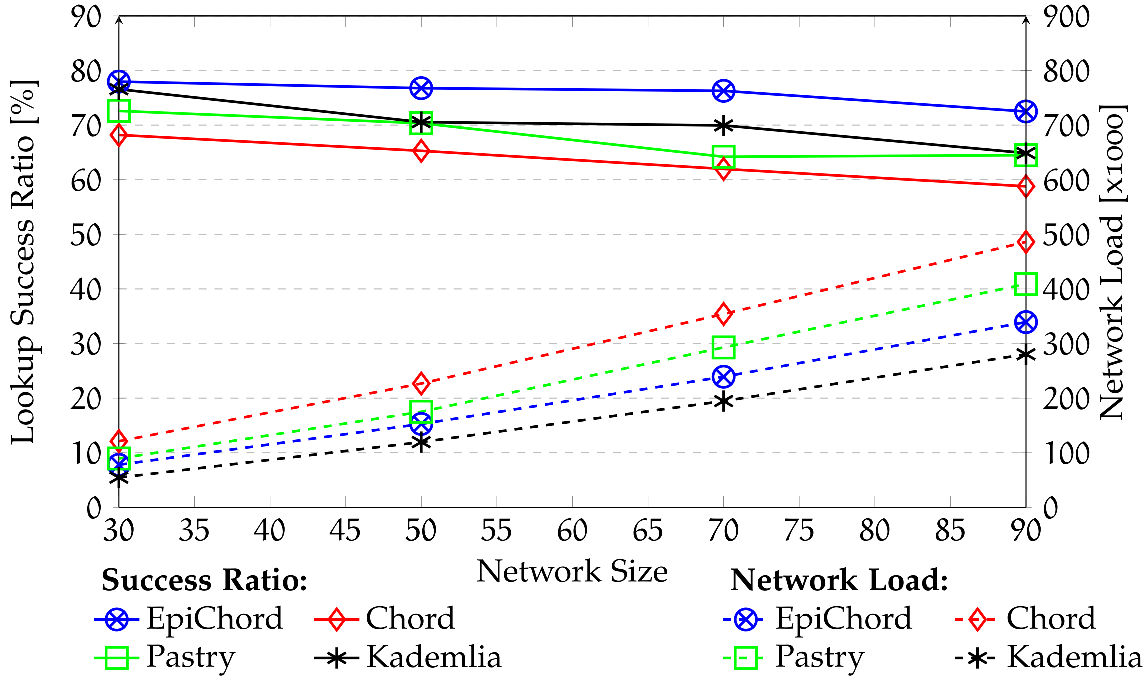


Figure 4.4: Success rate and network load for EpiChord, Chord, Pastry and Kademlia when layered over OLSR with 30 s Lookup frequency

The left y axis of figure 4.4 depicts the average success rate for key lookups for all the four overlays when each peer fires key lookups every 30 seconds. Additionally, it also shows on the right y axis the corresponding traffic that was released in the networks over the simulated time. As can be seen, the EpiChord overlay outperforms the rest of the simulated overlays as it manages to achieve a better success ratio in all the used network sizes. A reason for the good performance of EpiChord is that EpiChord uses a cache to store as many routing entries as possible about other mobile peers in the overlays. Thus, it builds a better understanding and consideration of the underlay topology. Consequently, this helps avoid the criss crossing issue when carrying out overlay lookup routing. Such issue might be faced by other overlays, causing worse success rate and more traffic. As the number of mobile nodes increases in the network, the performance of all the overlays deteriorates. This is expected when more traffic is released in MANET, as more collision and packet loss occurs, especially with the high node speed of 5 m\s.

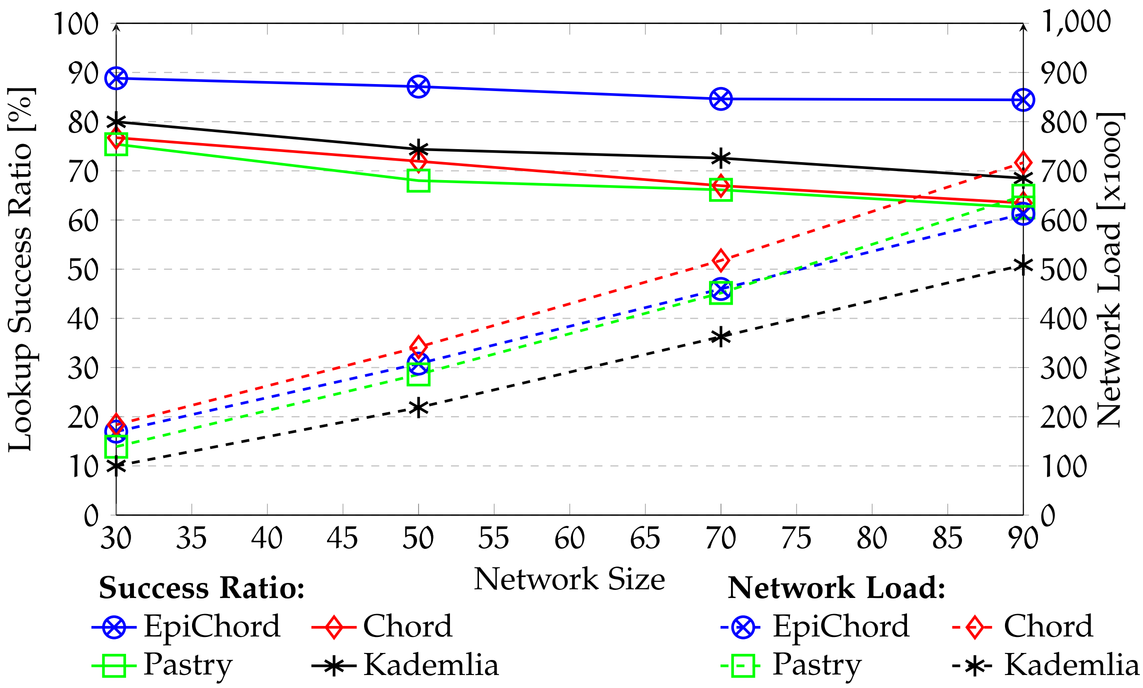


Figure 4.5: Success rate and network load for EpiChord, Chord, Pastry and Kademlia when layered over OLSR with 5 s lookup frequency

Figure 4.5 is another graph that plots the same metrics for the same networks but with more lookup queries being sent. Lookups were sent every 5 seconds by every individual peer to increase the possibility of EpiChord achieving one logical hop performance and hence comparing the performance of multi hop and one hop systems on MANETs. As a result of sending more lookups, the total released traffic for each system increases compared to figure 4.4 . This is also due to the required underlay routing traffic to satisfy the need of the overlay. In other words, with more lookups being sent, more traffic in the underlay will be released to route these lookups towards their final destinations. Due to utilizing the lookup response messages to exchange overlay routing information, EpiChord manages to achieve higher performance of up to 90% with a network consisting of 30 nodes compared to about 80% that was achieved by others in the same setting. The graph also shows improvements for Chord, Pastry and Kademlia performance with more frequent lookup queries with small network sizes. Despite the fact that EpiChord is more proactive than Chord, it produces less traffic in all the adopted scenarios. A reason for this is the ability for EpiChord to answer a key lookup in less logical hops. Hence, it can avoid criss crossing the MANET network to solve a key lookup and consequently releases less traffic. Even with 90 nodes network size, The graph shows that EpiChord manages to achieve the best performance. It can be claimed that this is the advantage of operating one logical hop performance. On the other hand, the performance of multi hop overlays deteriorate with larger network sizes. As a result of maintaining a larger routing table, the Kademlia overlay manages to achieve better performance compared to Chord. This is because Kademlia organizes its routing table in two dimensional space, whereas the entries of Chord routing tables are organized in one dimensional space [117]. Thus, Kademlia has a higher ability to solve key lookup in reasonably less logical hops as figure 4.6 shows. This consequently affect the total network load of Kademlia.

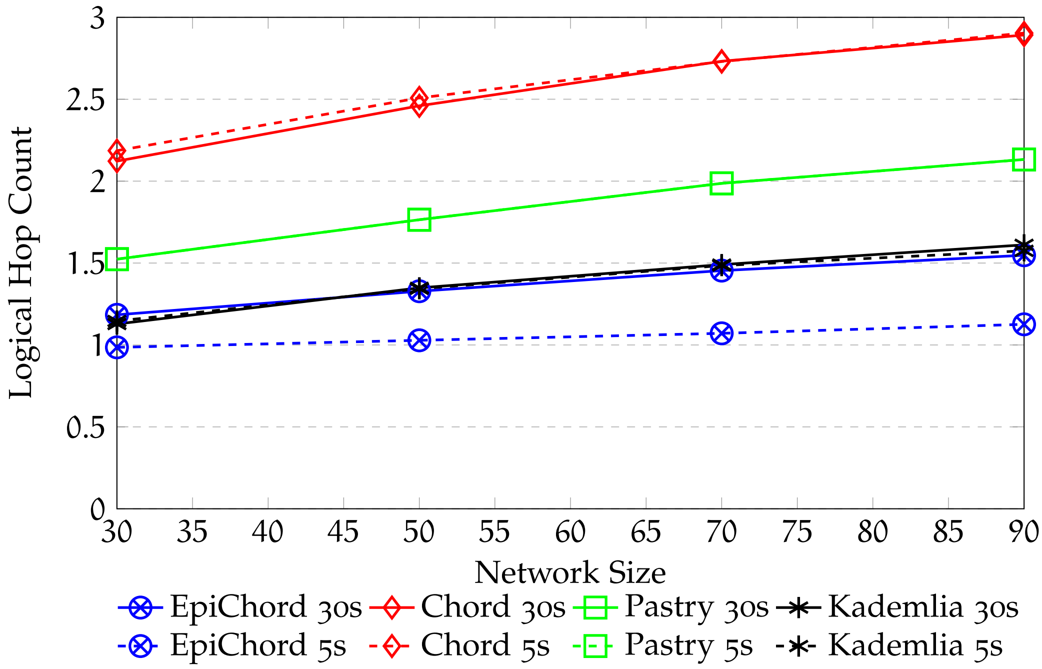


Figure 4.6: Logical path length for EpiChord, Chord, Pastry and Kademlia when layered over OLSR with 30 s and 5 s lookup frequencies

The average logical path length that was required by the overlays to resolve a key query is depicted in figure 4.6 for 30 seconds and 5 seconds lookup intervals. As the graph shows, EpiChord resolves lookups in less logical hops. Due to optimizing lookup responses, and when lookups are sent more frequently, EpiChord builds an overlay that resolves key lookups in almost a one logical hop, as the dashed line shows for EpiChord with 5 seconds lookup interval.

As discussed in section 4.1.1, building a P2P overlay over MANETs which gives one logical hop would contribute towards understanding the physical topology. Consequently, it would avoid the mismatching issue between the logical overlay and the physical topology. The above results of running multi hop and one hop overlays over MANETs prove that EpiChord achieves better performance when it operates a one hop overlay. On the other hand, other overlays tend to yield poor performance since they operate multi hop overlay, that may lead to criss crossing the network when solving a key lookup. In addition, the use of a one logical hop overlay does not only benefit the success ratio, but also contributes to preventing the generation of extra traffic in MANET when solving key lookups in the first logical hop.

Based on the experiment results of deploying one hop and multi hop overlays over MANETs, it is concluded that a one hop overlay system is a promising solution for data dissemination and discovery in mobile ad hoc networks. However, the expected amount of traffic that a one hop P2P system may produce is quite high, especially in a scarce network like MANET. Thus, such a system needs to be more optimized to minimize its traffic in MANETs and improve its performance as well. This is possible due to the fact that there exist similarities between P2P overlay networks and mobile ad hoc networks.

4.2 OneHopOverlay4MANET

The proposed system, OneHopOverlay4MANET, employs structured P2P similar to Chord [13] and EpiChord [18], a Distributed Hash Table (DHT) based P2P overlay network which can achieve lookups in a single hop with an intensive lookup environment. This is a novel approach, as previous work focused on pairing multi-hop overlays with MANETs.

OneHopOverlay4MANET uses cross-layering to optimize synergies between MANETs and P2P networks. Such optimization would cut down typical management traffic resulting from deploying traditional P2P over MANETs. A cross-layer channel is used to pass routing information between the adopted underlay routing protocol and OneHopOverlay4MANET, reducing the typical management traffic of deploying P2P over MANETs.

OneHopOverlay4MANET is designed to combine Structured P2P with the underlay routing protocol to achieve one logical hop in MANETs. Therefore, the system intends to reduce overlay signalling traffic from nodes joining and leaving the network as well as to reduced required traffic to maintain the overlay consistency. The proposed system is designed to solve lookups in short time while incurring less traffic in the network and providing highly accurate logical routing. Consequently, OneHopOverlay4MANET significantly improves the performance of data storage and retrieval in MANET based networks.

4.2.1 *Overview*

Unlike previous approaches, OneHopOverlay4MANET builds a one-hop structured P2P overlay system similar to the structure of EpiChord[18] and Chord[13] with MANET protocols using cross-layering. The optimization of cross-layering reduces the typical overhead traffic of P2P protocols where the underlay routing information is used to build the overlay and populate its routing tables. Maintenance traffic overhead at the P2P overlay is thus greatly reduced. Figure 4.7 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 a peer to fetch a key in one logical step. This would yield a routing complexity of O (1) for the overlay routing. This solution is feasible in a network such as a MANET. In other words, it is not so expensive to maintain a logical routing to all peers in a network that is characterized with a small number of nodes. In addition, the routing information is also provided for no cost since cross-layering is adopted. This is the opposite of the traditional one hop system deployed over the internet, where there are thousands of peers and more management traffic is required to maintain consistency. As experimentation confirms, such an approach is feasible with a MANET network. As simulation results confirm, the incurred overhead in such a system is manageable, mainly due to the cross-layering approach being adopted.

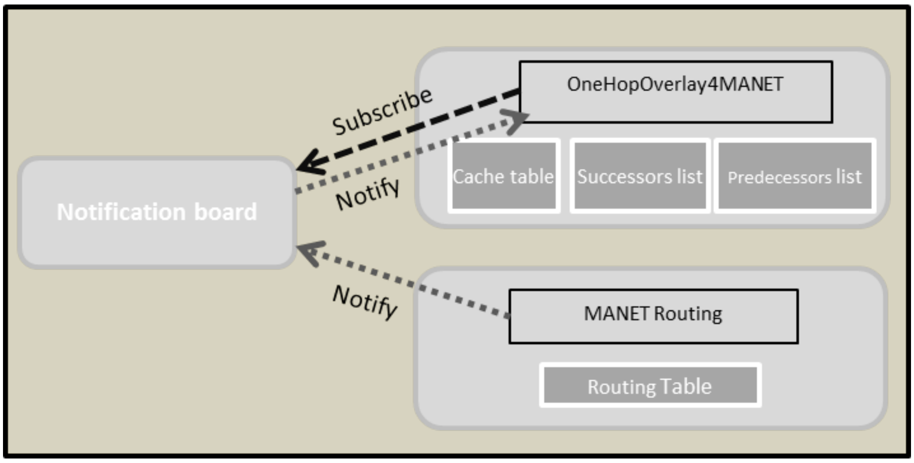


Figure 4.7: OneHopOverlay4MANET System Architecture

A OneHopOverlay4MANET node assigns itself a unique identifier by hashing its IP address. A cryptographic hash function like SHA-1 [118] ensures that there will not be a collision of Ids. A one dimensional circular address space is used in OneHopOverlay4MANET as is 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 4.8 shows that key29 is stored on the following peer, which is P31.

In a similar way to EpiChord and Chord, OneHopOverlay4MANET maintains lists of neighboring 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 routing protocol through cross-layering.

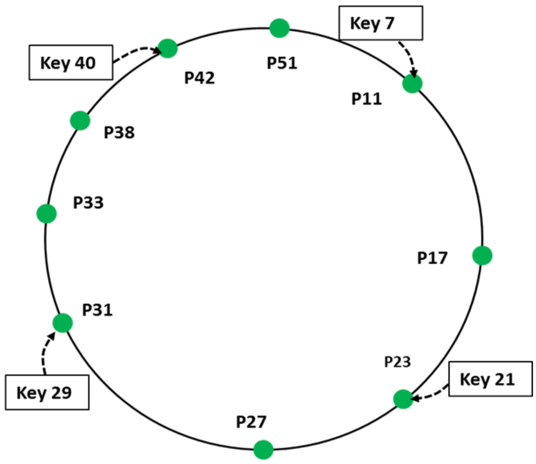


Figure 4.8: Circular address space

4.2.2 *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 underlay routing protocol.

OneHopOverlay4MANET uses cross layering to exchange information between underlay and overlay to reach one hop performance. This approach reduces the typical overhead from employing one hop overlay systems, as routing updates from the underlay are forwarded to the 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.

In terms of cross-layering methods, there are manager method and non-manager method. OneHopOverlay4MANET follows the manager method of cross-layering [119] to optimize the network layer's routing information transfer. The manager method stipulates that a channel can be created allowing for the sharing of data between some or all of the layers in the protocol stack. With the manager method of cross-layering, 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 Figure 4.9 shows, OneHopOverlay4MANET fallows the manager method to implement sharing information between the Application layer (where the proposed OneHopOverlay4MANET system resides) and the Network layer (where MANET routing protocol operates).

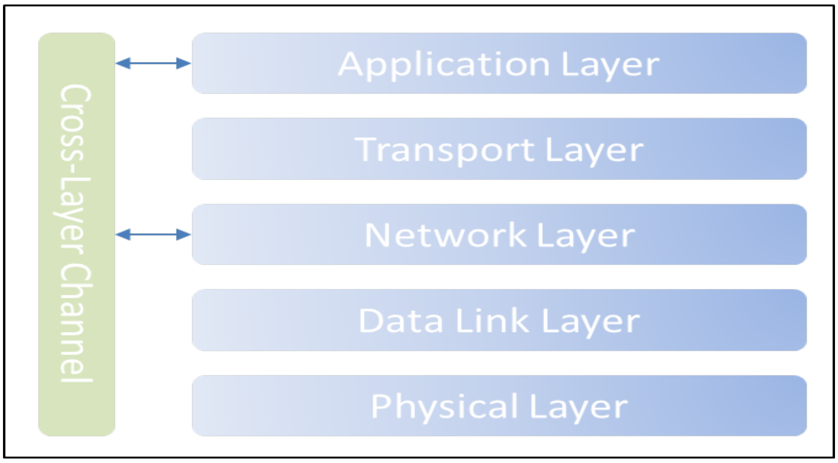


Figure 4.9: The manger method of cross-layering for OneHopOverlay4MANET

A notification board is used (see figure 4.7 and 4.10) as a channel that manages the sharing of information between the Application layer and the Network layer. During set-up, OneHopOverlay4MANET peers need to subscribe to the notification board to receive notifications of changes that may occur at the underlay layer. During operation, the underlay routing protocol notifies the notification board whenever changes occur in its routing table. The notification board in turn passes the update to the overlay. Such notifications will provide the overlay with up-to-date contents of the underlay routing protocol. The overlay then uses this information to update its view of the network.

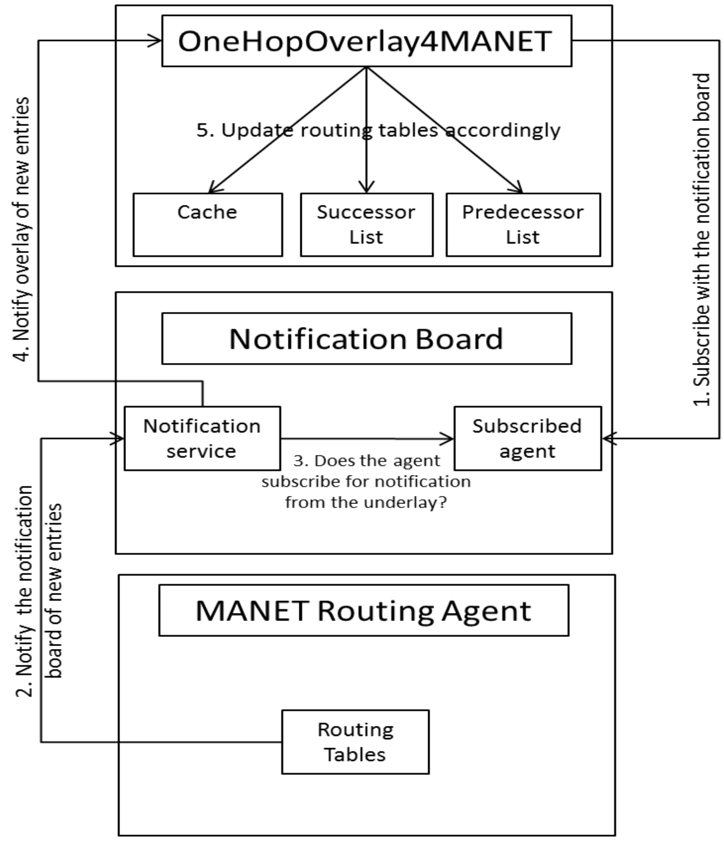


Figure 4.10: Notification board to pass notifications between MANET underlay and OneHopOverlay4MANET

Each OneHopOverlay4MANET peer is supposed to receive notifications from the underlay layer. Notifications contain information from the routing table of the MANET protocol used. Thus, adopted MANET protocols need to be modified to enable the sending of its routing table information updates to the notification board. The information passed is the IP addresses of the nodes that this node knows about. This includes information on nodes that can be reached in a single physical hop and those nodes that can be reached in multiple physical hops. This information will then be used by a OneHopOverlay4MANET peer to populate and update its view of the network.

4.2.3 *Joining The Overlay*

In order to be part of the overlay, a OneHopOverlay4MANET peer needs to join the overlay. In the OneHopOverlay4MANET system, two methods for joining the network are considered when a mobile node first arrives. 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 the 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 providing that the used protocol at the underlay is a proactive routing protocol. The reason for this is the proactive nature of those protocols, where they build and populate their routing table in advance. Proactive protocols build and populate 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. Such information can be used to instantiate the overlay.

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. As a consequence, the underlay routing protocol will not start since it works on demand and there is no demand from the overlay. This requires the use of joining messages propagated by the overlay. The Join message procedure is used by OneHopOverlay4MANET overlay with two purposes:

- The first is to allow a peer to join the overlay network. When a peer generates a Join Message, it receives a Response Message containing the contents of the sender's cache, Successor list and Predecessor list.

- The second purpose is to generate demand on the reactive underlay to form links and update routing information. Once the reactive underlay is started, its routing table will grow gradually. Through the cross-layering mechanism, OneHopOverlay4MANET will then be updated with the 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 lists:

- Successor List: This list contains the four immediate peers that follow this peer in the circular address space.
- Predecessor List: This contains the four peers that immediately precede this peer in the circular address space.
- Cache Table: This is used to store any peers that this peer knows about, but which are not included in the other two lists; Successor List and Predecessor List.

4.2.4 *Lookup*

EpiChord supports parallel lookups to increase its chance of finding a key with the first hop reducing lookup latency. In addition, parallelism is also used to help the system to learn new routing table entries and update the cache. However, OneHopOverlay4MANET does not make use of this technique and only sends lookup requests to a single destination.

The main reason for sending a single lookup request at a time is to reduce unnecessary network load on a resource constrained MANET network. Since adopting a parallel lookup technique would incur more traffic, OneHopOverlay4MANET sends a single lookup request at a time to reduce the bandwidth usage and reduce expected traffic from employing parallelism. Another reason for sending a single lookup is that since OneHopOverlay4MANET receives routing updates through its use of cross-layering from the MANET layer, its overlay routing tables are well populated and consequently, it achieves lookup success within a single hop without parallel lookups. Thus, there is no need for parallel lookup requests.

To locate a file, OneHopOverlay4MANET hashes the value of the file and then consults its own 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 from 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 in OneHopOverlay4MANET is an iterative lookup 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 value of the key and information about its successor and predecessor.
- If it is a predecessor of the looked up key, it will respond 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, information about the node succeeding and the two nodes preceding the best node that holds the key.

In terms of the number of required messages to solve a key lookup with regard to the logical overlay, OneHopOverlay4MANET requires 2 x logical hops in the overlay. Total messages includes both lookup request and response messages.

Clearly, each logical hop in the overlay is translated to a physical path that consists of number of physical hops. Failing to solve a lookup with the first logical hop would result in OneHopOverlay4MANET following distributed hash table routing and going through the circular address space until it finds the requested key. This would increase the traffic in the network and may result in the criss-crossing issue that was presented in section 4.1.1. Thus, OneHopOverlay4MANET attempts to reach a destination in a single logical hop that results in reducing the overall traffic and the required amount of time to find a key in the network.

OneHopOverlay4MANET also handles the rare case when the lookup is not resolved from the first logical hop. When the queried peer has no information about the queried object, it replies to the originator with its best knowledge about the queried object. The response is the best next logical hops to the queried key. This will decrease the distance to the destination peer on the one dimensional circular address space. After receiving the reply, the originator sends a further query until it finds the peer which is in charge of the queried object. However, due to optimizing the MANET underlay, OneHopOverlay4MANET rarely follows this case. In the majority, it has a highly accurate routing table that can find the key from the first logical hop.

4.2.5 *Management*

Clearly, to a large extent, 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 expires. When the underlay routing table gets updated or gets new entries, it sends a notification to the notification board with the routing information of its routing table. It sends the IP addresses of all of its routing entries. The notification board in turn passes such notifications to the corresponding peer if it has subscribed to it. Once a peer receives the update, it hashes the received 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 first 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 also carried out to find out if the new peer is in the Predecessor list. The Predecessor list needs to be recalculated to include the new logical ID if it falls in its range. If the new peer ID falls outside the successor and predecessor ranges, it will be stored in the cache. This ensures the correctness of the distributed routing at the overlay. As a result of maintaining the logical address space by each individual peer, the typical O (log N) lookup performance will be guaranteed for overlay routing.

Stabilization mechanisms are used by EpiChord to maintain a consistent view of the overlay. EpiChord runs the stabilization mechanisms frequently to fix any inconsistency in the network. However, OneHopOverlay4MANET does not use such stabilization for the following reasons:

- Running the stabilization protocol incurs more traffic in a scarce network like MANET.
- OneHopOverlay4MANET uses cross-layering to gets updates from the underlay which gives frequent image of the existing nodes in the network. In addition, OneHopOverlay4MANET uses the responses messages to the key lookups to exchange overlay information between peers. Therefore, OneHopOverlay4MANET can handle the consistency of the overlay without emitting any management traffic.

Consequently, using the underlay information significantly reduces the need for overlay maintenance traffic. Once a peer gets information through the cross layer channel, it calculates the logical Ids of other peers by hashing their IP addresses and then populating its tables. Besides the updates from the underlay, OneHopOverlay4MANET can also use lookup queries to update its routing tables. Each lookup query response contains some information from the queried peer. The need to use additional lookup messages to update peers' routing tables is however, much reduced in OneHopOverlay4MANET .

The logical address space is also divided into small slices by EpiChord as a way to keep the overlay consistent. Each peer has to maintain a certain number of entries per slice. This can be justified when deploying P2P over the traditional internet which may consist of thousands of nodes. Moreover, each agent peer in such a network will not care about physical routing nor have any information about the physical topology. Providing that a MANET network consists of small number of mobile nodes and each node is expected to participate in routing and forwarding traffic within the network, a mobile node will have some information about the topology of the network that can be exploited to feed the overlay without introducing overlay traffic in scarce MANETs. Therefore, OneHopOverlay4MANET does not divide the logical address space into slices to avoid excessive management traffic due to maintaining such cache invariants. Table 4.1 highlights differences between EpiChord and OneHopOverlay4MANET.

4.3 OneHopOverlay4MANET versus Chord and EpiChord

In order to evaluate the performance of OneHopOverlay4MANET, an implementation of the system was developed in OverSim [68] and INET [69]. Figure 4.11 shows a high level class diagram of the implemented system in OverSim. OverSim is a modular simulation framework package for the OMNeT++ simulation environment that supports different structured and unstructured P2P systems. Different types of underlays can be used with Oversim. This includes the INET underlay, which is a communication networks simulation package for

Table 4.1: OneHopOverlay4MANET and EpiChord features

System	EpiChord	OneHopOverlay4MANET
Management strategy	Proactive	Reactive
Use of cross layering over MANET	No	Yes
Use of join message	Yes	Only when based on reactive underlay
lookup approach	Parallel lookups	Single lookup
Stabilization mechanism	Yes	No
Splitting logical space into smaller slices	Yes, and maintains cache invariant per each slice	No splitting of logical space to avoid excessive traffic

the OMNeT++ simulation environment that supports several wired and wireless models including Mobile ad hoc network protocols.

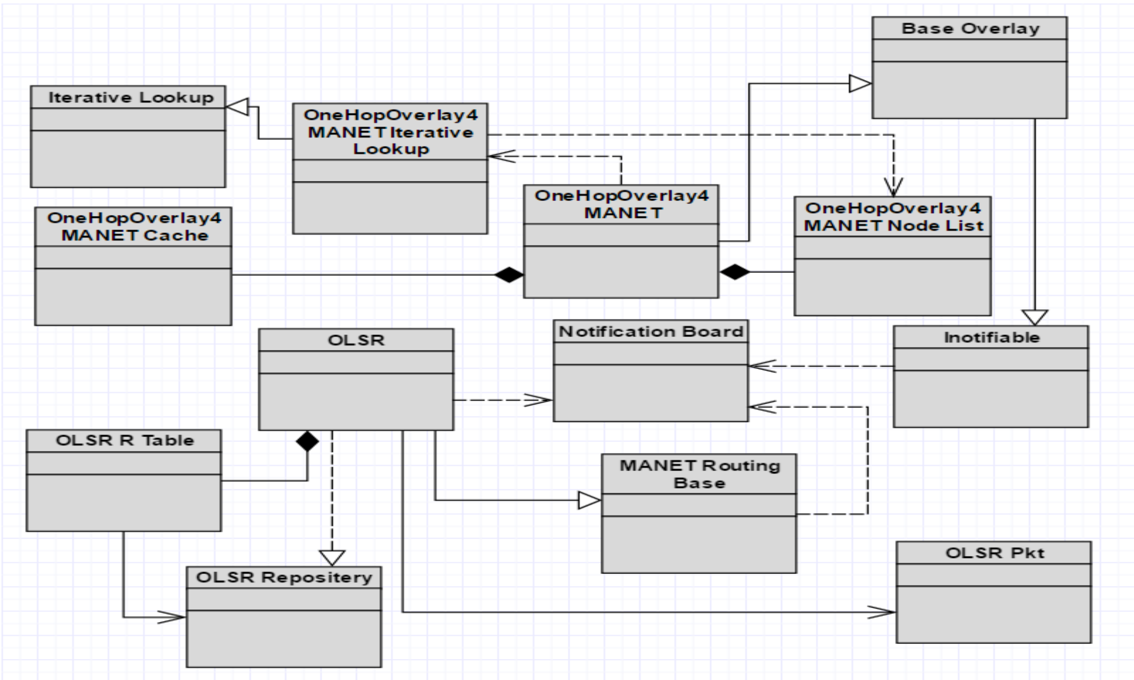


Figure 4.11: Class Diagram

In this section, an evaluation of the proposed system, OneHopOverlay4MANET, is conducted and its performance compared to EpiChord and Chord which build a similar logical address space. EpiChord and Chord were deployed as they are over MANETs without any use of cross-layering. The objective of such evaluation is to compare the performance of the three systems using the following metrics:

- Network Load: How much traffic each system generates in the network. This includes both overlay and underlay traffic.
- Logical Hop Count: The average required number of logical hops which a system needs to find a key in the network.
- Lookup Success Ratio: the average percentage of managing to resolve key lookups within the network for each evaluated system .

4.3.1 *Simulation Set up*

Table 4.2 shows the simulation parameters for the simulated scenarios. All the simulated scenarios were repeated five times and the result is the average of the repetitions. The number of participating mobile nodes in the network were increased from 30 up to 90 nodes. All nodes were moving at 5 m\s speed within a 1000 m X 1000 m simulation area. The mobility model used is the Random Way Point model which is commonly used for simulating an ad hoc network. According to the Random Way Point mobility model, when a node travels from one position to another, it pauses for a time, which is known as the pause time. In these simulations, the pause time is a random number between 1 and 50 seconds. For each of the simulated scenarios, a network is given about 120 seconds to stabilize. After 120 seconds, the measurements of the metrics begin over the simulated time.

The used underlay routing protocol is OLSR which is a proactive underlay. The performance of OneHopOverlay4MANET is compared with basic layered systems of EpiChord over OLSR (no cross-layer communication) and Chord layered over OLSR (no cross-layer communication). The frequency of firing lookup requests

Table 4.2: Simulation parameteres

Simulator	OMNeT++
Underlay routing protocol	OLSR
Topology size	1000 m x 1000 m
Number of mobile nodes	30, 50, 70, 90
Mobility model	Random Way Point
Node speed	5 m\s
Pause time	Random between 1 and 50 sec.
Measurement time	600 seconds
Transmission range	250 m
Network stabilization time	120 seconds
Lookup interval	30 s, 10 s, 5 s
Parallelism	3 for EpiChord only
Simulation repetition	5 times

were investigated with different values. Scenarios were simulated where the file lookups were sent every 30, 10, and 5 seconds as an intensive lookup scenario. The main reason for varying this parameter is to find out how much the network will be affected when sending lookups more frequently.

In addition, lookup intervals were varied to investigate how much benefit each system can get when decreasing lookup intervals, especially EpiChord, which uses file lookup replies as a policy to learn new entries and hence, builds a better view of the P2P overlay. The only system of the three systems that adopts parallelism is EpiChord. It sends parallel lookup requests to increase the chance of finding a key from the first logical step. The parallelism parameter was set to 3 parallel requests for each file lookup for EpiChord system.

4.3.2 *Lookup success rate*

Figures 4.12, 4.13 and 4.14 show the success ratio of solved lookup requests to the total sent key lookups for the three systems with three different lookup intervals (4.12: 30 s, 4.13: 10 s, 4.14: 5 s). In most cases, OneHopOverlay4MANET outperforms Chord and EpiChord in all three different scenarios. As can be seen OneHopOverlay4MANET maintains a 85%+ performance across all network sizes and lookup frequencies. Even with a low frequency of sending key lookups, OneHopOverlay4MANET manages to achieve similar results to those with higher lookup intervals. This means that unlike EpiChord, OneHopOverlay4MANET is not dependent on additional lookup messages (and the update information from response messages) to achieve a high degree of lookup success.

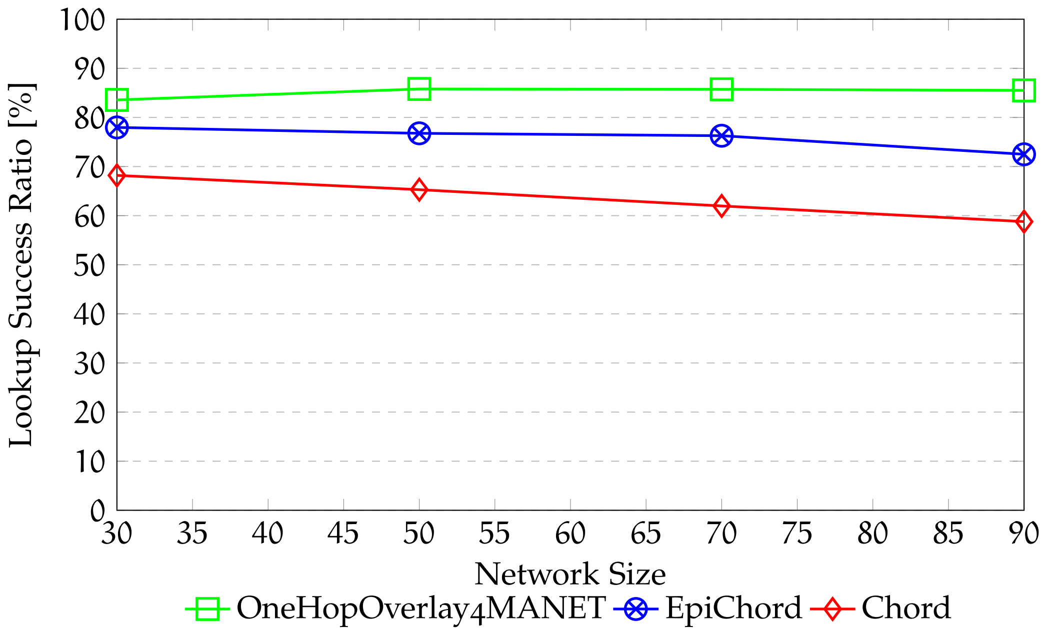


Figure 4.12: Success ratio for OneHopOverlay4MANET, EpiChord and Chord with lookup frequency of 30 s

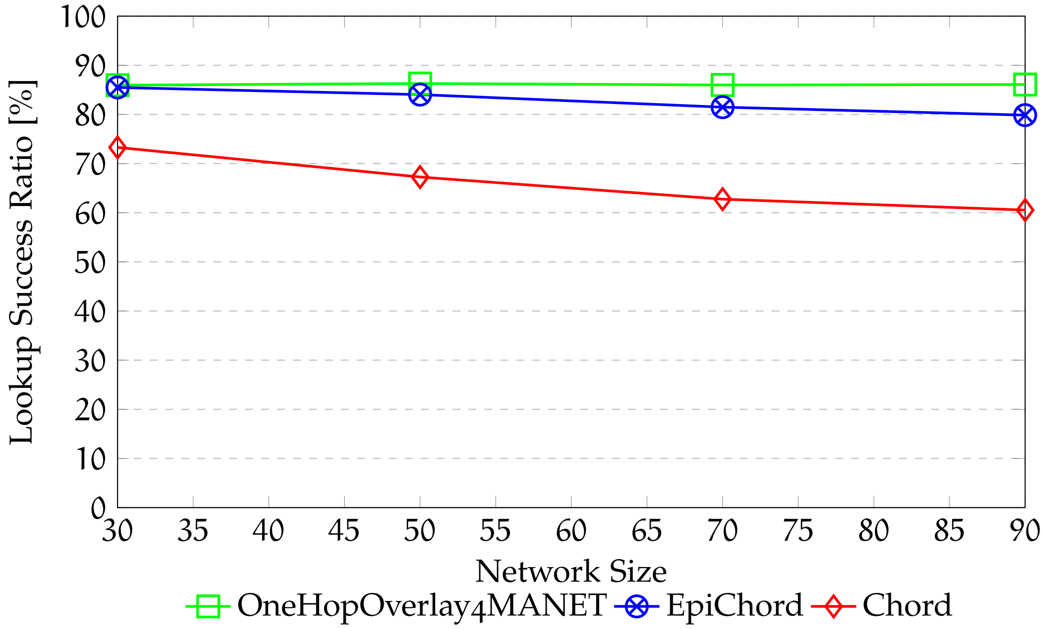


Figure 4.13: Success ratio for OneHopOverlay4MANET, EpiChord and Chord with lookup frequency of 10 s

The impact of the network size can also be seen in all three figures. It is clear that as the network size grows, the performance of layered EpiChord and layered Chord deteriorates. However, OneHopOverlay4MANET manages to maintain the best result even with large network sizes. An obvious reason for this is that even with larger network sizes, OneHopOverlay4MANET peers' logical routing is reactive and only utilizes the underlay routing information. This means that no more overlay maintenance traffic is required by OneHopOverlay4MANET to maintain the consistency of the overlay. Consequently, optimizing underlay routing knowledge helps OneHopOverlay4MANET to build and maintain better logical routing, which results in good performance.

On the other hand, EpiChord and Chord build their logical routing tables (EpiChord uses a Predecessor list and a Successor list, Chord uses a Finger table and Successor table) in a proactive manner through emitting overlay management traffic over MANETs. Each individual peer in either system has to maintain part of the logical space to guarantee distributed routing. When the network size increases, that means more maintenance is required by both systems, especially

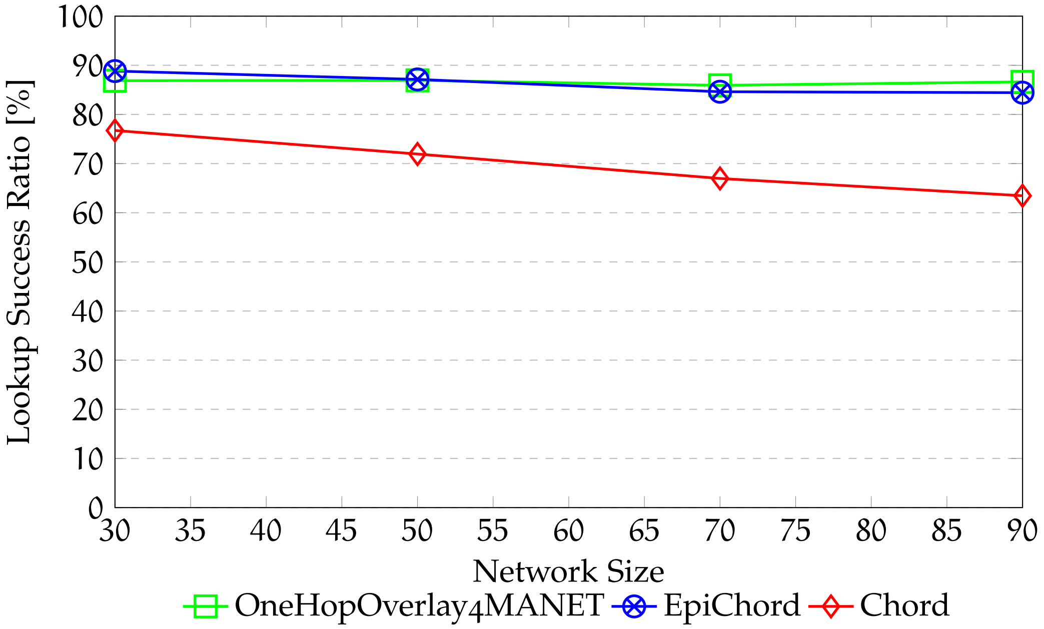


Figure 4.14: Success ratio for OneHopOverlay4MANET, EpiChord and Chord with lookup frequency of 5 s

in situations where nodes move at a high speed. Moreover, a logical neighbor in the overlay may be physically far away, thus resulting in the criss-crossing issue (see section 4.1.1) and yielding deteriorated performance. All these factors force the performance of both EpiChord and Chord to decrease as the network size increases.

However, when sending key lookups more frequently, EpiChord performance improves compared to scenarios when lookups are fired less frequently with large network size. This is because EpiChord uses the response messages to learn new entries and update its view of the overlay. Thereupon an EpiChord peer will have richer routing tables when key queries are sent more frequently. Thus, better performance is achieved with 5 s lookup frequency in comparison to when lookups are sent every 10 or 30 seconds.

OneHopOverlay4MANET achieves this performance due to optimizing the available routing information at the underlay to build logical routing tables that can solve file lookup queries accurately. The impact of decreasing lookup interval time can be seen on the performance of the basic EpiChord/OLSR layered system. It achieves a comparable performance to OneHopOverlay4MANET only with the highest lookup frequency of 5s (see figure 4.14) . Clearly this comes at the cost of increased traffic in the network, as can be seen in figures 4.15, 4.16 and 4.17.

4.3.3 *Network load*

Figures 4.15, 4.16 and 4.17 show the total traffic that was generated in the networks with 30, 10 and 5 seconds lookup frequencies and varying network size for OneHopOverlay4MANET, EpiChord and Chord. This traffic includes both overlay traffic and the mobile ad hoc routing protocol traffic. Since the overlay traffic causes more underlay traffic, the inclusion of the underlay routing protocol load was necessary to see the overall impact of deploying a P2P system over MANETs.

As expected, when the network size increases, the overall load increases as well. This is mainly due to the number of mobile nodes and their required routing traffic. However, the increments with EpiChord and Chord based network traffic are seen to be higher when the network size increases compared to the traffic generated by the OneHopOverlay4MANET based network. This can be seen clearly from the figures, where the OneHopOverlay4MANET total traffic is increased from 50 K packets with 30 node network size to about 210 K packets with 90 node network size and 30 s lookup frequency.

In comparison, EpiChord and Chord ended up releasing 350 K and over 450 K respectively with 90 nodes (see figure 4.15). This is an increase by approximately 160 K for OneHopOverlay4MANET and 270 K for EpiChord. This is obviously a result of the management strategies that EpiChord adopted to maintain a consistent overlay. Another reason for the larger amount of released traffic by the EpiChord based network is that in the scenario with a 30 second lookup interval, EpiChord does not manage to resolve key queries with the first logical hop (section 4.3.4, figure 4.19). Consequently, requiring more than one logical hop would cause more traffic where each logical step in the overlay is translated to a path in the underlay.

As the figures show, OneHopOverlay4MANET causes the least amount of traffic in all the simulated scenarios. It generates about half of the released traffic by the layered EpiChord/OLSR system when lookups are being sent more frequently (figures 4.16 and 4.17). This is due to optimizing the underlay information when building and maintaining the overlay. Such optimization of underlay routing information enables OneHopOverlay4MANET to build accurate logical routing. Thus, OneHopOverlay4MANET can resolve key lookups with the first logical hop and hence, reduce overall traffic.

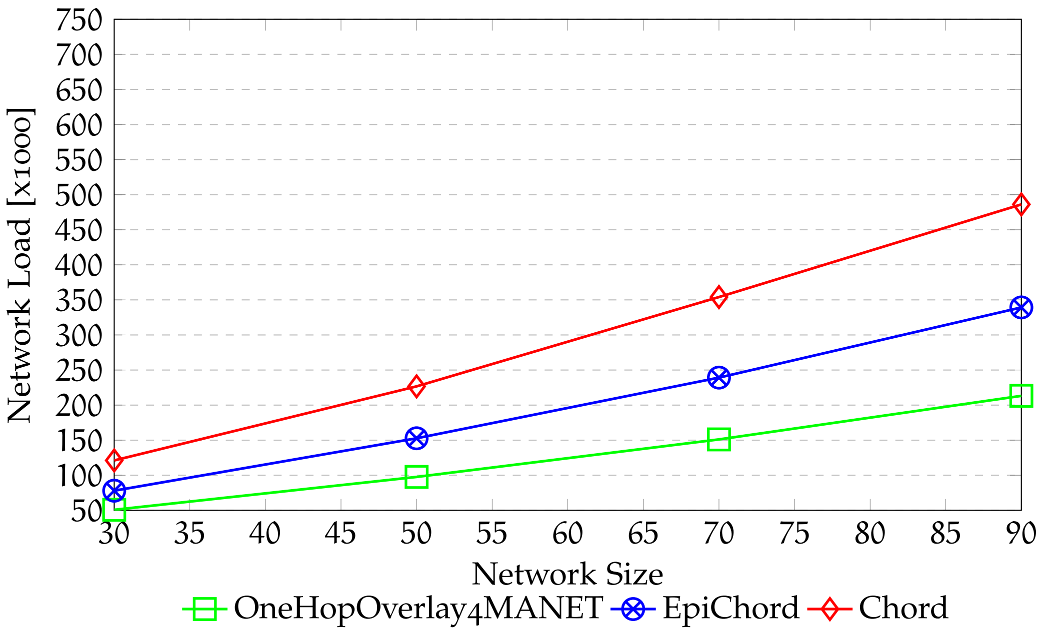


Figure 4.15: Network load for OneHopOverlay4MANET, EpiChord and Chord with lookup frequency of 30s

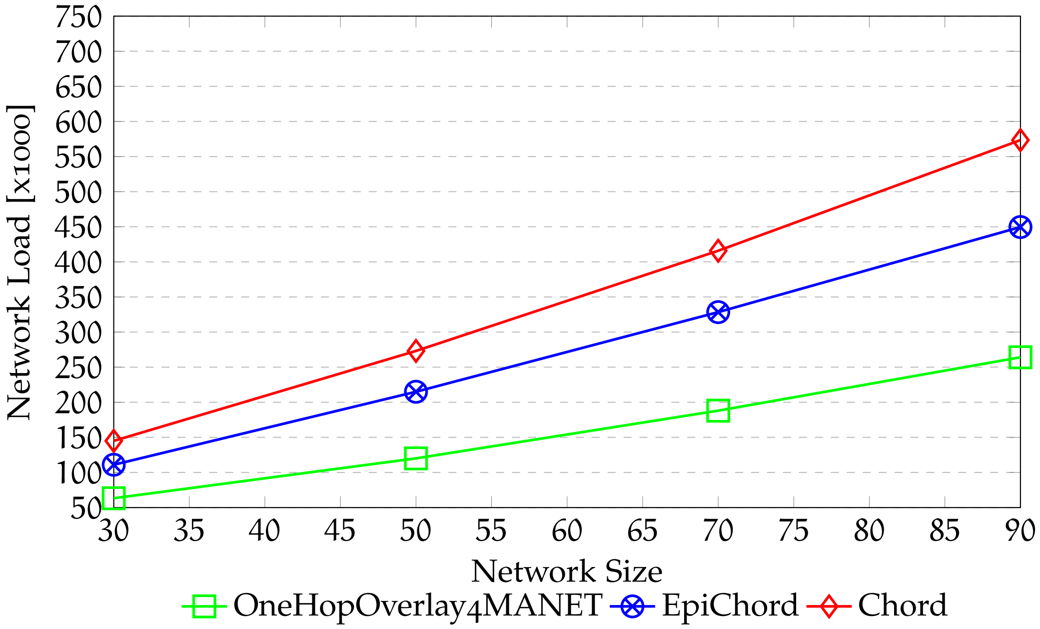


Figure 4.16: Network load for OneHopOverlay4MANET, EpiChord and Chord with lookup frequency of 10 s

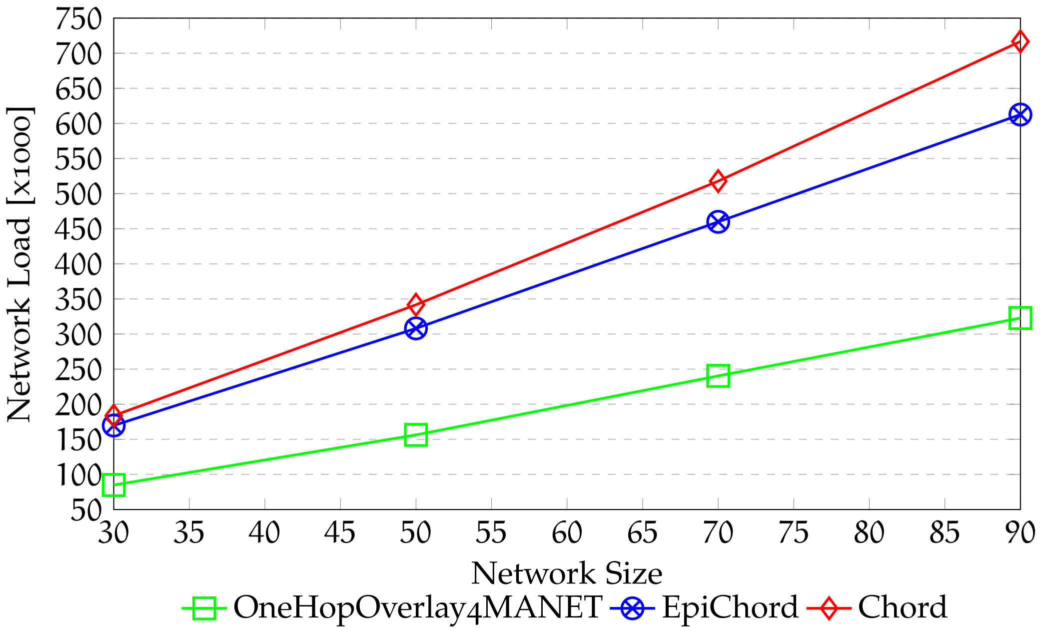


Figure 4.17: Network load for OneHopOverlay4MANET, EpiChord and Chord with lookup frequency of 5s

Looking at the three figures, it can be seen that when more key lookups are being sent, more traffic is released in the network. This is due to the increase in the number of lookups and hence the required underlay routing traffic to satisfy sent lookup queries. However, traffic increases in EpiChord and Chord networks are seen to be quite high compared to the increases in OneHopOverlay4MANET traffic when lookup are sent with small frequency. The reason for this is that for EpiChord and Chord, multiple logical hops might be contacted in order to resolve a key lookup. Hence, more traffic will be generated. In addition, with more frequent key lookups, more entries will be learnt, which means more traffic is required to maintain consistency.

Figure 4.18 shows the generated traffic by the three systems for a network consisting of 90 nodes when changing the lookup frequency interval. As it shows, OneHopOverlay4MANET is the most stabilized system. The extra traffic for sending lookup queries every 5 seconds is about 100 k compared to the traffic released with 30 seconds lookup frequency. However, EpiChord and Chord network traffic increases from 340 k and 485 k (with 30 second interval) to 610 k and 715 k (with 5 seconds interval) respectively.

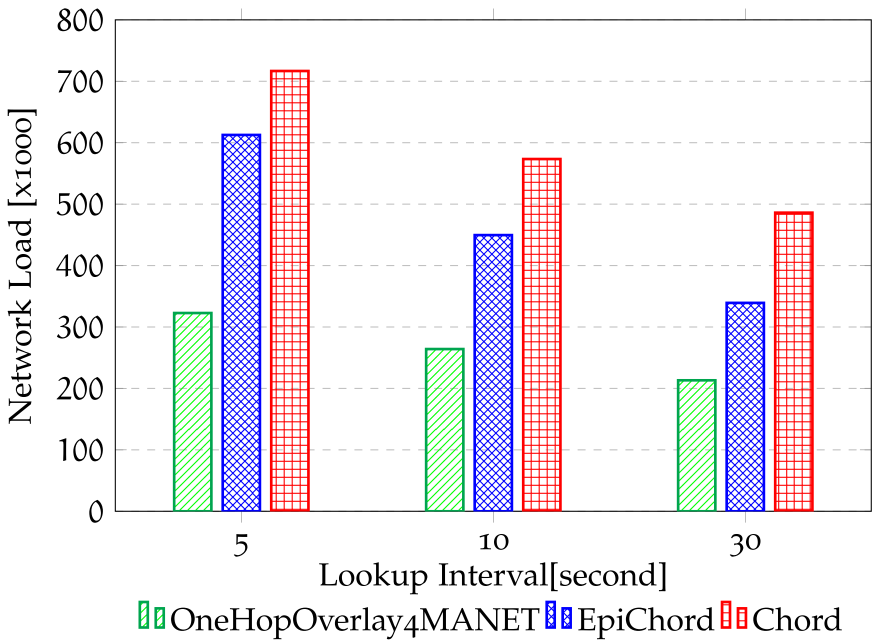


Figure 4.18: Network load for OneHopOverlay4MANET, EpiChord and Chord in a 90 nodes network size

4.3.4 *Average logical path length*

This section investigates the average required logical hops in order to solve a key lookup by each of the three systems. One of the aims of the proposed system is to achieve one logical hop for lookups avoiding inefficient lookup forwarding in the underlay. Figures 4.19, 4.20 and 4.21 show the logical hop count results for the three systems over different network sizes and with varying lookup intervals. OneHopOverlay4MANET shows a consistent hop count performance of 1, independent of the network size and lookup frequency parameters. This indicates that OneHopOverlay4MANET maintains highly accurate information in its routing tables.

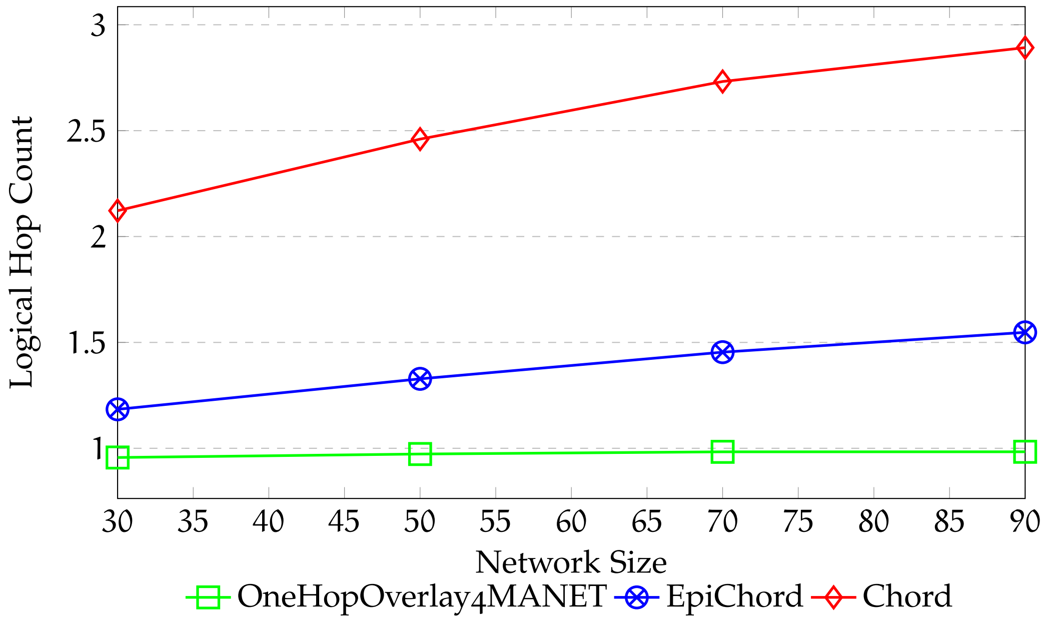


Figure 4.19: Logical path length for OneHopOverlay4MANET, EpiChord and Chord with lookup frequency 30 s

Interestingly, the two other systems show a declining performance as the network size increases. This is because each peer in both systems maintains a part of the logical address space. In order to solve a key query, the query may go through multiple logical peers, which results in an increase in the average logical path for both systems. This may involve contacting some mobile nodes at the underlay more than once when the logical routing clashes with the physical routing. This is a reason for the high amount of traffic that was produced by both systems (see figures 4.15, 4.16 and 4.17).

Figure 4.20 shows that the layered EpiChord/OLSR system can improve its performance with an increased lookup frequency. This is due to optimizing lookup response messages to exchange logical routing information, that helps improve the performance of EpiChord. However, even at the highest lookup rate (figure 4.21 for 5 s lookup interval), performance declines as the network size increases. Despite its 3 parallel lookups, it cannot outperform OneHopOverlay4MANET in any of the configurations.

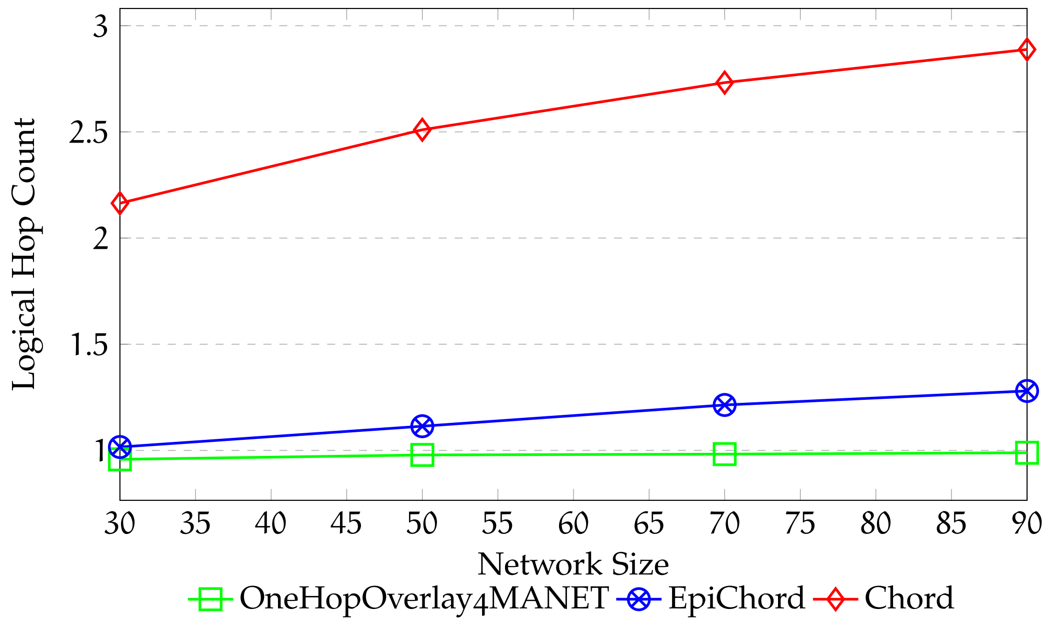


Figure 4.20: Logical path length for OneHopOverlay4MANET, EpiChord and Chord with lookup frequency 10 s

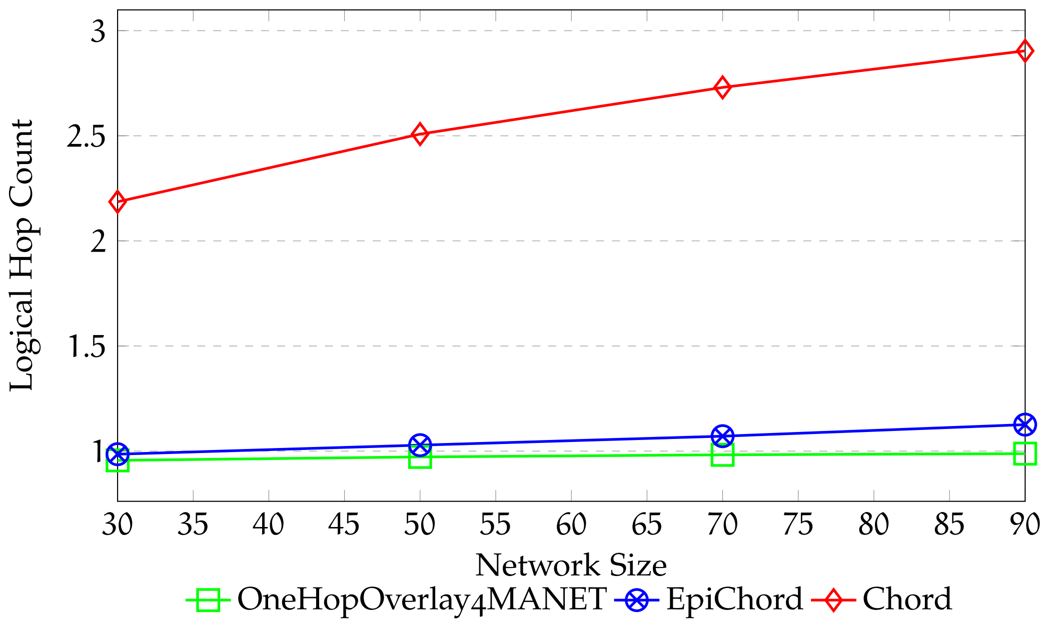


Figure 4.21: Logical path length for OneHopOverlay4MANET, EpiChord and Chord with lookup frequency 5 s

In conclusion, OneHopOverlay4MANET has achieved better performance compared to layered EpiChord and Chord. This was a result for several strategies that OneHopOverlay4MANET uses. Firstly, OneHopOverlay4MANET aims to reduce overlay management traffic in scarce network such as MANET. OneHopOverlay4MANET achieves this through the usage of underlay routing information. Consequently, less traffic is released in the network increasing the chance of sending traffic between sources and destinations successfully.

Secondly, OneHopOverlay4MANET builds one hop overly system which overcomes mismatching issue when P2P deployed over MANET. This however entitles OneHopOverlay4MANET to decrease the required time when answering lookups and decrease required traffic to satisfy a query. In addition, OneHopOverlay4MANET opts for sending single lookup compared to parallelism which is used by EpiChord. This was chosen by OneHopOverlay4MANET because of that OneHopOverlay4MANET maintains logical information about all peers in the network. So, OneHopOverlay4MANET does need to send parallel lookups.

Moreover, the usage of parallelism would cause more traffic in the network which conflicts with the original aim of OneHopOverlay4MANET.

4.4 SUMMARY

In this chapter the mismatching issue between the logical overlay and the physical topology was introduced, which is one of the most challenging issue when deploying structured P2P over mobile ad hoc networks. It was identified that the deployment of a one hop structured P2P system would contribute towards minimizing the effect of this issue. In other words, it would help to build a better understanding between the logical overlay and the physical topology. Therefore, it would yield a better performance in mobile ad hoc networks. This was experimented by deploying a multi hop overlay and one hop overlay systems over MANETs. The result showed that one hop overlay is a promising solution for the mismatching issue between the overlay and MANET topology.

As a consequence, in this chapter , a novel system entitled OneHopOverlay4MANET was proposed. This system was developed to achieve efficient data dissemination over mobile ad hoc networks using P2P paradigm. The proposed system fully optimized the nature of ad hoc networking to build efficient overlay routing. Since MANETs are a scarce Network, OneHopOverlay4MANET decreases the typical traffic of traditional P2P systems by deploying a cross-layer channel between the MANET routing protocol and the P2P overlay system to optimize the synergy between mobile ad hoc networks and P2P systems.

OneHopOverlay4MANET aims to overcome the mismatching between the logical overlay and the physical topology through building an overlay that achieves one logical hop performance. As a consequence of reaching the destination from the first logical hop, less delay will be required to solve key lookups in the overlay. Moreover, this would decrease the chance of criss crossing the physical underlay when routing a request at the overlay, hence, reducing the overall released traffic in MANET.

A simulator for the proposed system was implemented as a P2P system in the OMNeT++ environment and evaluated against two P2P systems with a similar structure. The results show that OneHopOverlay4MANET manages to achieve its aim of routing a key lookup in a one logical step. It also manages to scale down the typical traffic for traditional P2P systems through the use of cross-layering with the underlay routing protocol. In addition, it also achieves better performance in terms of the ability to route key lookups successfully.

5

ONEHOPOVERLAY4MANET REACTIVE OR PROACTIVE UNDERLAY

The previous chapter introduced the proposed system with OLSR as the key underlay exhibiting many similarities with one hop overlays. In this chapter, wider experimentation using other underlays is presented. The discussion also includes the different required joining approaches. Firstly, the joining process when OneHopOverlay4MANET is combined with a reactive underlay is introduced, followed by the joining process over a proactive underlay. In order to evaluate the performance of the proposed system with different MANET underlays, pairing of OneHopOverlay4MANET over three reactive underlays (DSR, AODV and DYMO) and two proactive underlays (OLSR and BATMAN) are discussed.

5.1 ONEHOPOVERLAY4MANET OVER REACTIVE MANET

5.1.1 *Reactive MANET*

Reactive MANET protocols are the type of MANET routing protocols that work on demand. Routes to destinations are only discovered when there is a demand for them. This type of protocol typically floods the network with route request messages starting with neighboring nodes when routes are needed. As a result of node mobility, active routing entries may become invalid. Once a route is discovered, a route maintenance mechanism can be used to maintain the existence of established routes.

There are many MANET routing protocols that fall in this category of routing. Dynamic Source routing (DSR) [39] is one of the most widely known protocols in the literature that works on demand. It uses route discovery and route maintenance mechanisms. Ad hoc On-Demand Distance Vector (AODV) [38] and Dynamic MANET On-Demand Routing (DYMO) [61] are other examples that operate only on demand. Both protocols adopt route discovery and maintenance to discover and maintain existing routes.

5.1.2 *Overlay Joining over Reactive MANET*

As discussed in the previous chapter (section 4.2.3), OneHopOverlay4MANET uses join messages to allow a new arriving peer to join the overlay. In addition, such messages trigger the reactive underlay to populate its routing tables. Consequently, underlay routing entries are used to populate the overlay routing tables.

In order to illustrate the joining process with a reactive underlay routing, the following example is presented. Figure 5.1 (a) shows a mobile network topology that consists of 14 nodes. It is assumed that the network has been operating for a while and each mobile node already knows some routes to other nodes. The corresponding logical address space is presented in 5.1(c). The Ids depicted next to nodes inside the circle are the underlay Ids. The corresponding logical Ids (peer Ids) 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 5.1(b) and (d) respectively.

Mobile node 14 is a new node that wants to join the existing network (figure 5.2). In this example, node 3 with logical ID 1003 is assumed to be the bootstrapping peer. In order to join the overlay, node 14 with logical ID 1014 needs to contact peer 1003. As node 14 has just joined, its routing table is still empty. Therefore, it sends a Route Request to find a route to node 3 to which the join message should be sent. Once the route to node 3 is discovered, node 14 will send the join message toward node 3 through the discovered route.

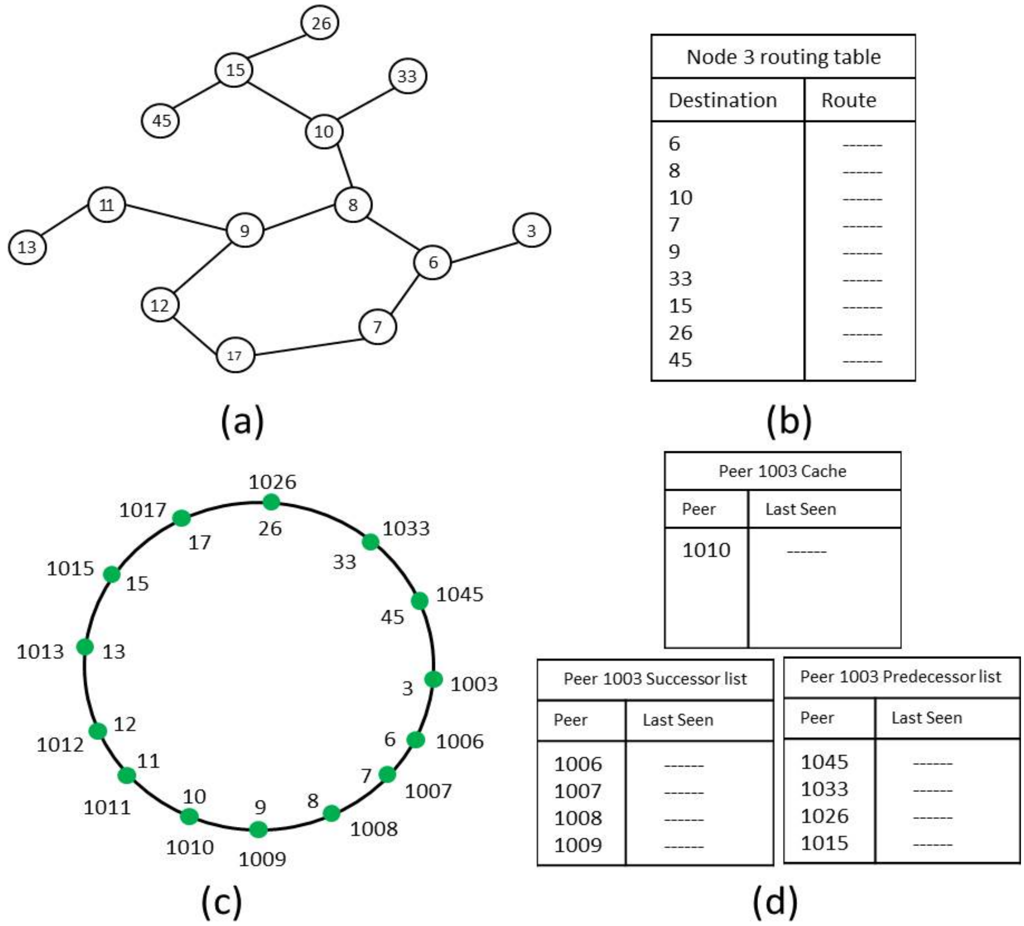


Figure 5.1: OneHopOverlay4MANET joining process over reactive MANET underlay

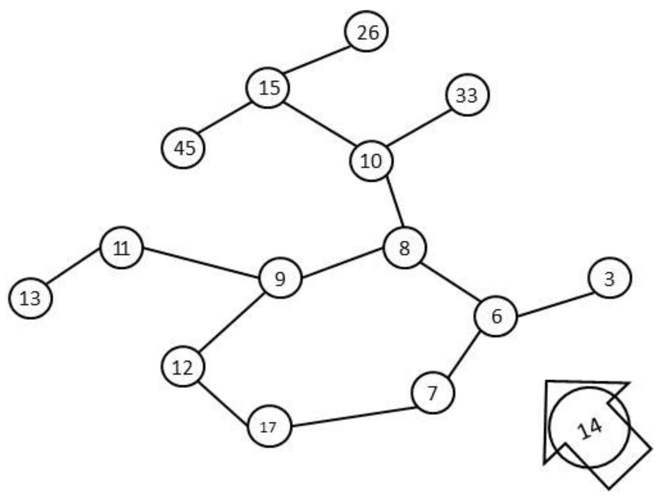
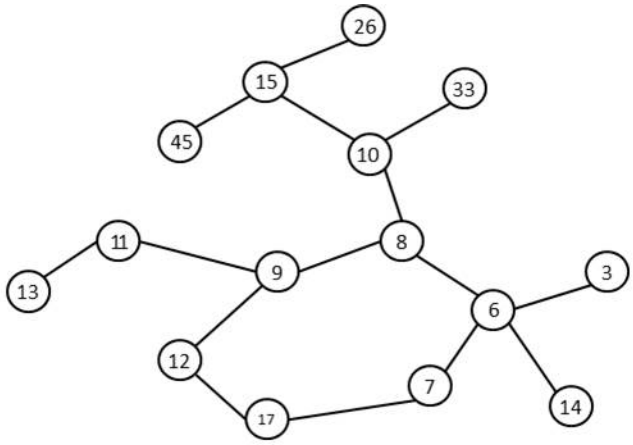


Figure 5.2: Node 14 to join the overlay

According to the deployed cross-layering method, the underlay routing protocol sends a notification of its routing table contents to the logical overlay as long as it has any route. The underlay routing agent will use the created 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 established MANET route. Note, however, that this message does not make use of the cross-layer communication, as the message exchange is between two overlay peers. Thus, the Response Message will be delivered to peer 1014 following the IP layer stack and not using the cross-layer channel.

The peer agent 1014 in turn, stores each individual received entry in the relevant table according to its address on the circular logical address space. As a result, peer 1014 starts with a good view of the network. Moreover, as the network continues to operate, the underlay routing agent will learn of new and updated routes. This is due to route discovery processes which are initiated by overlay requests. The new and updated routes are also learned due to the route maintenance mechanism which is operated by the reactive underlay protocols to handle physical node mobility and link breakage. As a consequence, the overlay gets notified of these new routing entries and can build up a better view of the network.

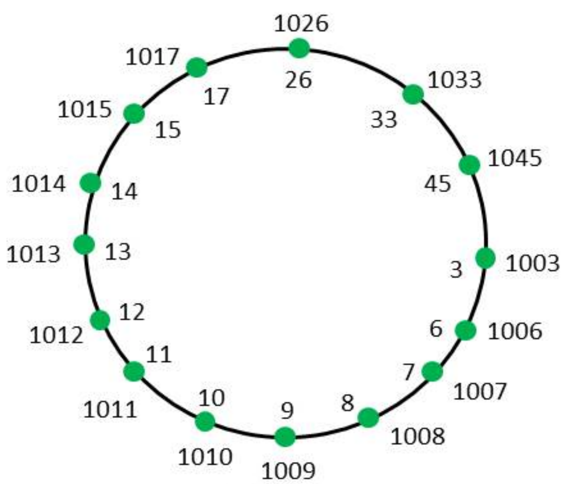
Figure 5.3 shows the physical topology of the network (a) and its corresponding logical overlay (c) along with the node 14 underlay routing table (b) and its corresponding logical routing tables for peer 1014 (d). The contents of the logical routing tables are learned from the notification board and the response to the Join request message. The entries for node 6 and node 3 were learned through the underlay notification and passed up through the created cross-layer channel between the underlay and 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 figures, the logical routing tables for peer 1014 do not exactly reflect the actual logical neighbors at the beginning. However, due to mobility and learning new entries at the underlay routing agent, the logical view of peer 1014 will improve with more information becoming available to it.



(a)

Node 14 routing table	
Destination	Route
6	------
3	------

(b)



(c)

Peer 1014 Cache	
Peer	Last Seen
1006	------
1003	------

Peer 1014 Successor list	
Peer	Last Seen
1015	------
1026	------
1033	------
1045	------

Peer 1014 Predecessor list	
Peer	Last Seen
1010	------
1009	------
1008	------
1007	------

(d)

Figure 5.3: The network after node 14 joined the overlay with node 14 (peer 1014) physical and logical routing tables

5.2.1 *Proactive MANET*

Proactive or table driven routing protocols for MANET are characterized by maintaining routes by each individual node to every single destination in the network. They propagate routing information throughout the network to keep routing information up to date and consistent. So there is no need for any demand on the proactive protocols to run. Routes to every destination in the network are ready in advance of their use.

Optimized Link State Routing OLSR [32] and Better Approach To Mobile Ad-hoc Networking BATMAN [59] are examples of protocols that operate proactive routing mechanisms. Each OLSR routing node diffuses partial link state information to the rest of the network. This allows other nodes to update their routing information and maintains the consistency of the network. It uses Multi Point Relay MPR nodes as the ones who retransmit control messages traffic to decrease the retransmission of such traffic. BATMAN on the other hand follows a different strategy to meet the need for maintaining routing information to every node in the network. It maintains a routing table that stores for each single destination in the network the best next hop. Each node transmits a control message regularly to allow itself to be detected. This message gets flooded throughout the network until it gets received by every node.

5.2.2 *Overlay Joining over Proactive MANET*

As discussed previously in section 4.2.3, the underlay routing information can be sufficient to instantiate the overlay providing that the used underlay is a proactive underlay where routing information is available in advance. When the adopted underlay is proactive, OneHopOverlay4MANET does not need to emit a Join Message to join the overlay. Instead the underlay routing entries at the routing layer 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 up its first view of the logical overlay.

In order to illustrate the joining process for OneHopOverlay4MANET over proactive underlay the following example is used. Figure 5.4 shows MANET topology (a), its overlay (c), the underlay routing table for node 51 (b) and the logical routing tables for the same node with peer ID 10051. In this scenario, it is assumed that the network has been running for a while and each node has built up its routing table. As the underlay routing protocol used in this example 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 through the cross-layer channel (see figure 5.4 (d)).

Node 56 is assumed to be a new arriving mobile node that wants to join the overlay (see figure 5.5). When approaching the physical topology of MANET, Node 56 will communicate with the nodes in the network and start building its own routing table.

In this case, the logical peer with ID 10056 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 logical overlay peer 10056 through the established cross-layer channel. On receiving the update from the underlay, peer 10056 will calculate its logical successors and its logical predecessors. After calculating the Successors and the Predecessors lists, peer 1056 then will place any other nodes in its cache. Figure 5.6 shows the network after the joining process for node 56 (peer 10056) is complete. The routing table for node 56 is presented in figure 5.6(b) and the logical tables for the peer are shown in figure 5.6(d).

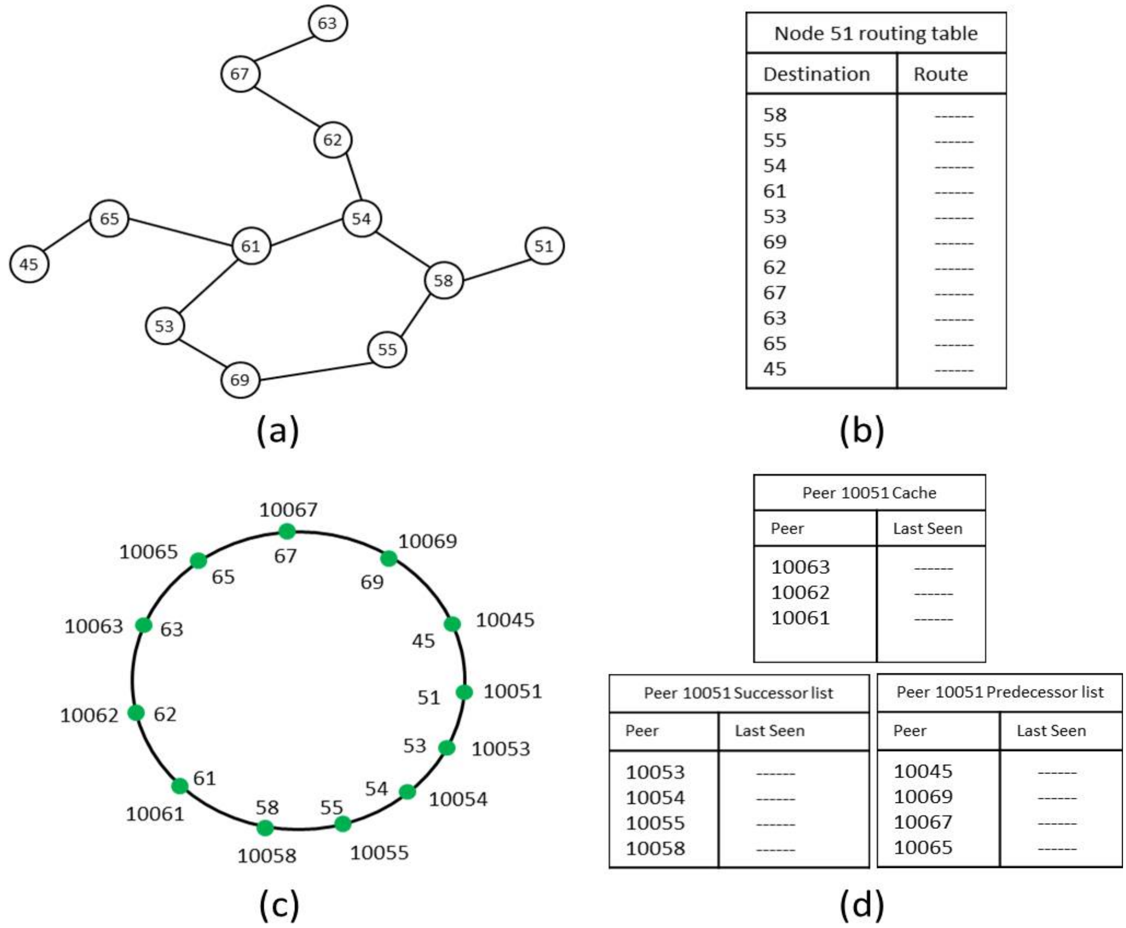


Figure 5.4: OneHopOverlay4MANET joining process over proactive MANET underlay

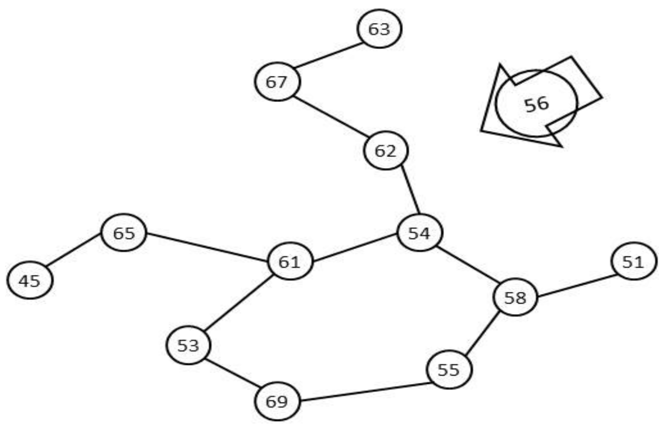


Figure 5.5: Node 56 to join the overlay

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 this node which in turn will recalculate its logical routing tables to include the new joining peer. Furthermore, the new participating peer can be also discovered by other overlay 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 to allow the rest of the overlay to discover recently joining peers.

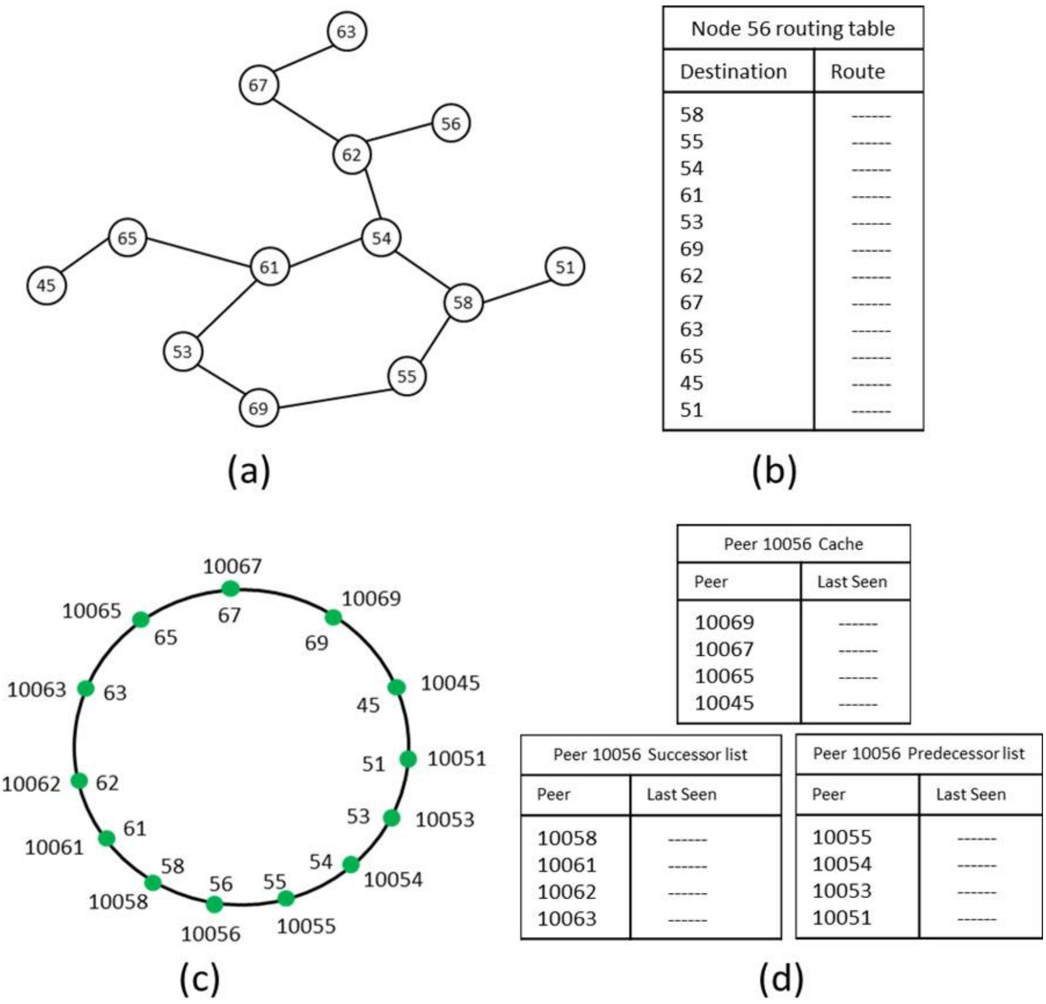


Figure 5.6: The network after node 56 joined the overlay with node 56 (peer 10056) physical and logical routing tables

5.3 EVALUATION FOR ONEHOPOVERLAY4MANET OVER PROACTIVE AND REACTIVE MANET

In this section the performance of OneHopOverlay4MANET over different underlay routing protocols is evaluated. The performance is examined over two different categories of routing in MANET; reactive MANET routing and proactive MANET routing. AODV, DSR and DYMO are the reactive routing protocols used, and OLSR and BATMAN are the proactive underlays used.

The Cross-layering mechanism was implemented with each network layer routing protocol. All the used underlay routing protocols were modified in order to support such cross-layering. As stated before, the cross-layer channel is used to pass the underlay routing information to the OneHopOverlay4MANET overlay. Underlay protocols implementations were modified to reflect that whenever the routing table for each routing protocol is updated, the new image of this routing table will be shared with the overlay. This was achieved using the notification mechanism where notifications are being sent to the overlay through the cross-layer channel. The following metrics were investigated to compare the performance of OneHopOverlay4MANET over various underlay routing agents:

- Lookup success ratio: the ratio of the number of resolved file lookup queries for files that exist in the network to the total initiated file lookup queries in the network.

- Average File Discovery Delay: the average amount of time required in order to solve a lookup query. It starts from sending the key lookup until the time when a peer receives the answer of that lookup. This reflects the ability of the system to retrieve a shared key on the network.

- Logical Hop Count: the average number of logical hops in the overlay 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 simulation period. The traffic load includes the P2P overlay traffic and the used network layer routing traffic.

5.3.1 *Simulation Set up*

Table 5.1 shows the used parameters for the simulated scenarios. All simulated scenarios were repeated ten times and the results shown in the graphs are the average of the repetitions. The used mobility model is the Random Way Point model which is one of the most popular models and 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 of the metrics starts to be taken. Lookups for shared files are regularly initiated by peers. 60 seconds is used as an average lookup frequency for relaxed networks. When simulating intensive lookups, 10 seconds is used as the interval between lookups.

The number of the participating nodes in the network was varied. A 40 node network was the smallest simulated network. Then, the number of nodes was increased to 60, 80, 100, 120 and up to 140 to investigate the performance when increasing network size. In addition, the effect of node velocity was also taken into account by increasing the node speed from 1 m\s (a slow moving node) to 5 m\s in steps of 1. The size of the simulation area was set to 1000 m x 1000 m.

5.3.2 *Success Ratio*

The performance of OneHopOverlay4MANET is evaluated over five different underlay routing protocols. Figure 5.7 and 5.8 depict the performance of OneHopOverlay4MANET with a node speed of 1 and 3 m\s respectively for network

Table 5.1: Simulation parameteres

Simulator	OMNeT++
Underlay routing protocols	OLSR, BATMAN, DSR, AODV, DYMO.
Topology size	1000 m x 1000 m.
Number of mobile nodes	40, 60, 80, 100, 120, 140.
Mobility model	Random Way Point Model
Speed	1 m\s, 2 m\s, 3 m\s, 4 m\s, 5 m\s.
Measurement time	1000 seconds.
Transmission range	250 m.
Network stabilization time	60 seconds.
Lookup Frequency	60 s, 10 s
MAC Layer	IEEE 802.11.
Simulation repetition	10 times.

sizes ranging from 40 to 140 nodes. In both graphs, the file lookups were issued every 60 s.

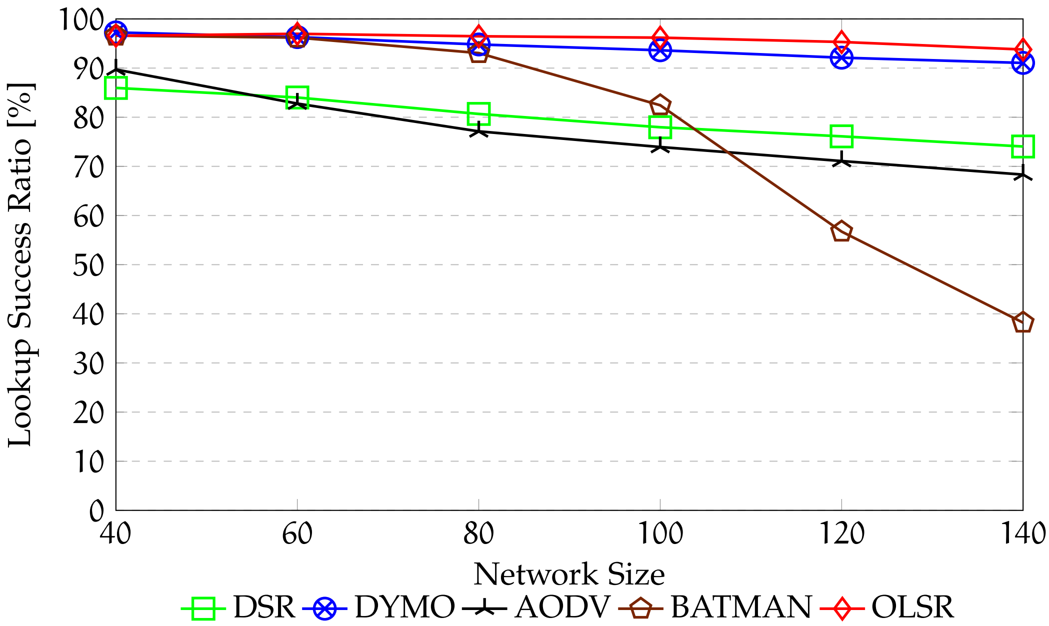


Figure 5.7: Lookup success rate for 1 m\s speed network and 60 s lookup frequency

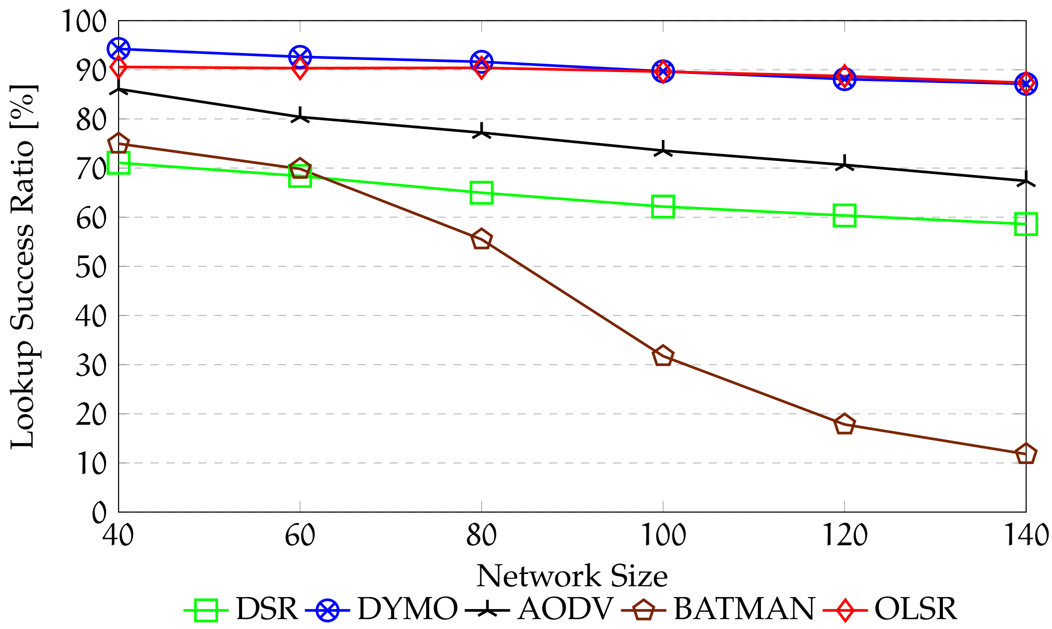


Figure 5.8: Lookup success rate of 3 m\s speed network and 60 s lookup frequency

In a slow moving network, OneHopOverlay4MANET achieves a 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. The DSR and AODV based systems achieve a performance of about 70%. As nodes move at a faster speed (figure 5.8; Speed 3 m\s) the performance decreases somewhat to less than 90% success rate for OLSR and DYMO with 100+ network size. The performance also decreases to less than 70% and 60% for AODV and DSR respectively with 140 nodes network size. The performance with BATMAN drops to below 20% success rate of solving key lookups, which is virtually unusable.

Clearly, an obvious reason for the deterioration is that the underlay routing agents are insufficient when nodes are moving at a higher speed. This causes more routing traffic, resulting in more packet collision and loss . For BATMAN, this can be seen as the cost of adopting the flooding mechanism to meet the requirement of each node maintaining the best next hop for all the nodes in the network. The flooding mechanism overloads the network and leads to poor underlay routing. As a consequence, such an inefficient routing is reflected in the overall performance of the overlay when based on BATMAN.

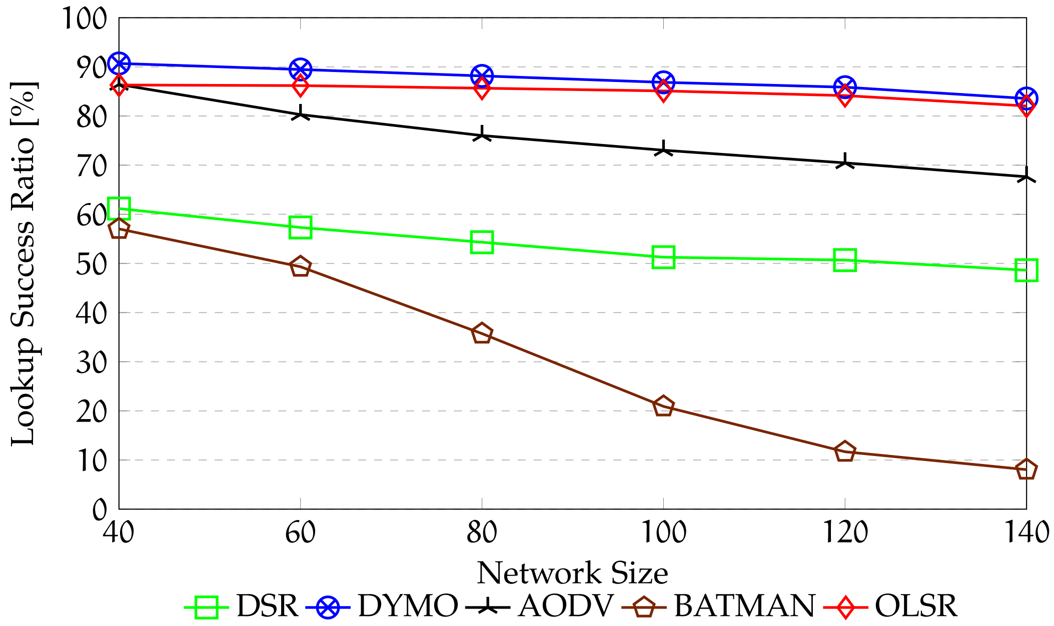


Figure 5.9: Success rate with 5 m\s speed network and 60 s lookup frequency

Increasing node speed to 5 m\s leads to a further slight decrease in lookup performance when using DSR (figure 5.9). The systems based on OLSR, DYMO and AODV largely maintain their performance levels. DSR's reaction to link breakage can be seen as a reason for the further decrease in the performance of the DSR based system. As nodes move faster, the probability of link failure becomes higher. When link failure happens, DSR does not send new route discovery immediately. However, DSR opts for trying out all the cached routes which are likely to be stale. Thus, the performance of DSR with higher mobility is affected.

Clearly, higher node mobility causes changes in the network topology. Hence, existing underlay routing table entries would be affected. Thus, additional route maintenance traffic and route discovery traffic will be generated to solve overlay lookups. Furthermore, an increasing level of packet loss will occur, resulting in poorer underlay routing efficiency. Consequently, the overlay, OneHopOverlay4MANET, performance will be affected.

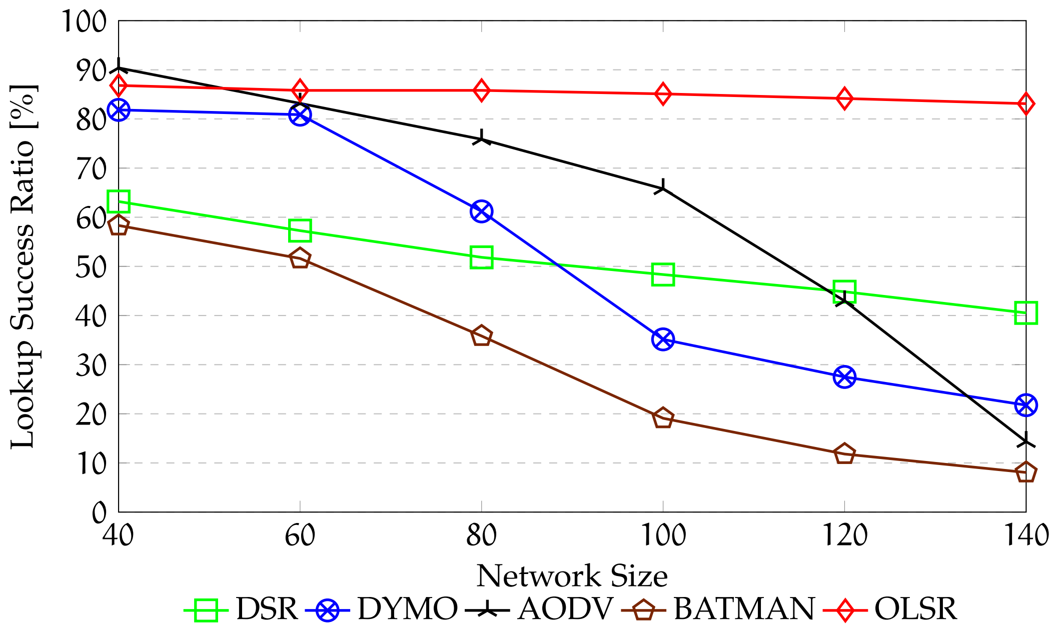


Figure 5.10: Success rate with 5 m\s speed network and 10 s lookup frequency

Additionally, when lookups are sent more frequently (every 10 s, figure 5.10) with a higher speed of 5 m\s, 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%, similar to the achieved result when key lookup queries were sent every 60 s (figure 5.9).

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 pro-actively. With a higher demand from the overlay (sending more frequent file lookups), more routes would be needed, meaning an increase of route discovery 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 cause the network to overload and lead to less efficient underlay routing, which in turn affects the overall performance of the overlay.

As the figure shows, the impact of lookup frequency with high mobility is seen to be stronger on the DYMO based system compared to the AODV based system. This is a result of maintaining more routing entries by DYMO where it stores an entry for each of the intermediate nodes alongside the discovered path. Hence, with higher mobility, more routing entries would be affected compared to AODV.

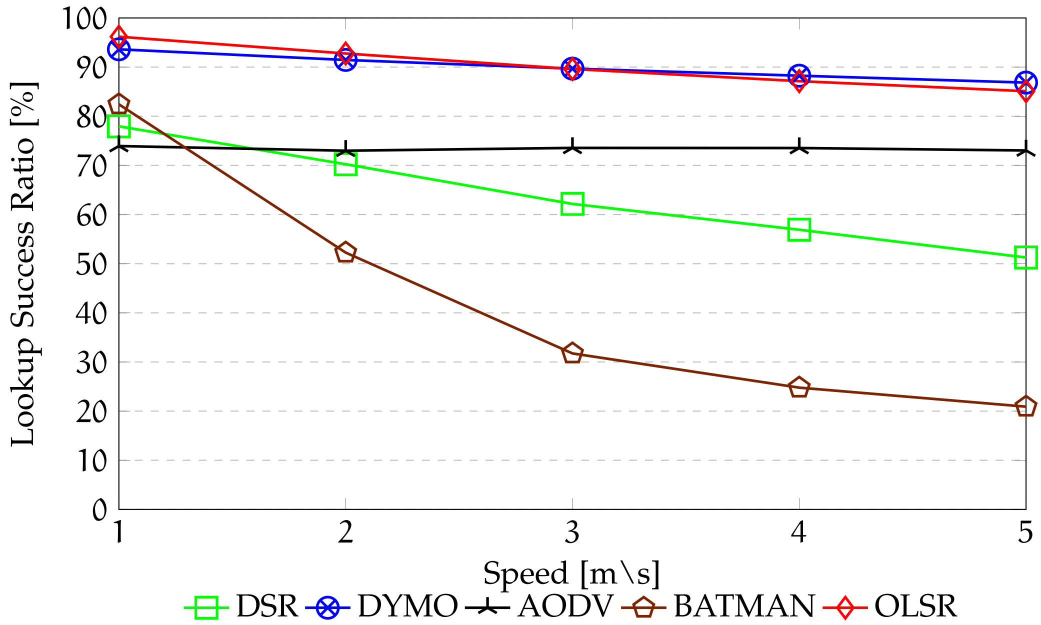


Figure 5.11: Success rate in a network consisting of 100 nodes with 60 s lookup interval

Figures 5.11 and 5.12 show the impact of node mobility on the performance of the system on networks consisting of 40 and 100 nodes with a 60 s lookup interval. As can be seen, OLSR and DYMO based OneHopOverlay4MANET can maintain their high performance. For the smaller network, the AODV based system also maintains a competitive performance where it manages to resolve about 90% of the sent lookup queries. However, for the larger network (100 nodes, figure 5.11), the performance of the system based on AODV yields poorer performance.

When OneHopOverlay4MANET was designed, it was expected to work better with proactive underlays due to the strong similarities between them. However, the results for the BATMAN based system disappoint. The performance of One-HopOverlay4MANET over BATMAN deteriorates with increasing node mobility and provides the worst performance amongst the tested systems overall. The good performance of the BATMAN based system is only achieved when the network size is 40 nodes and node velocity is 2 m\s or less (figure 5.12).

The very poor performance is due to the flooding approach that BATMAN employs. Every BATMAN node frequently broadcasts Originator Messages (like Hello messages) to allow other nodes to calculate the best next hop to every single destination in the network. Those packets are rebroadcast 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 across the network. This leads to collisions and inefficient routing which is reflected in the shown performance of the BATMAN based overlay.

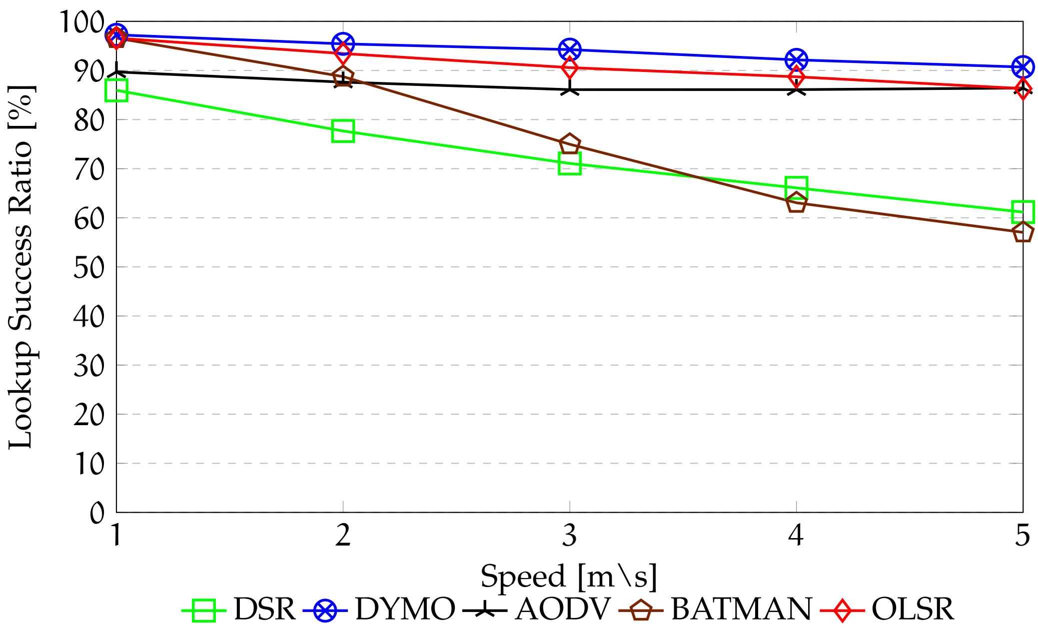


Figure 5.12: Success rate in a network consisting of 40 nodes with 60 s lookup interval

On the other hand, OLSR, as the other proactive underlay protocol which has been used with OneHopOverlay4MANET, shows a very strong and stable performance, often the best performance in the experimentations. As is shown in figure 5.10, the OLSR based system achieves a lookup success ratio of over 80% even with intensive lookup and high mobility cases. 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. Thus, it avoids flooding the network.

5.3.3 *File discovery delay and logical hop count*

Figures 5.13, 5.14 and 5.15 depict the lookup latency in seconds for OneHopOverlay4MANET using various MANET routing protocols. Figure 5.13 and 5.14 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 60 s (figure 5.13).

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 and recent routes to other nodes since there is more demand for routes from the overlay. Consequently, less time will be required to resolve a key lookup (figure 5.13 and figure 5.14).

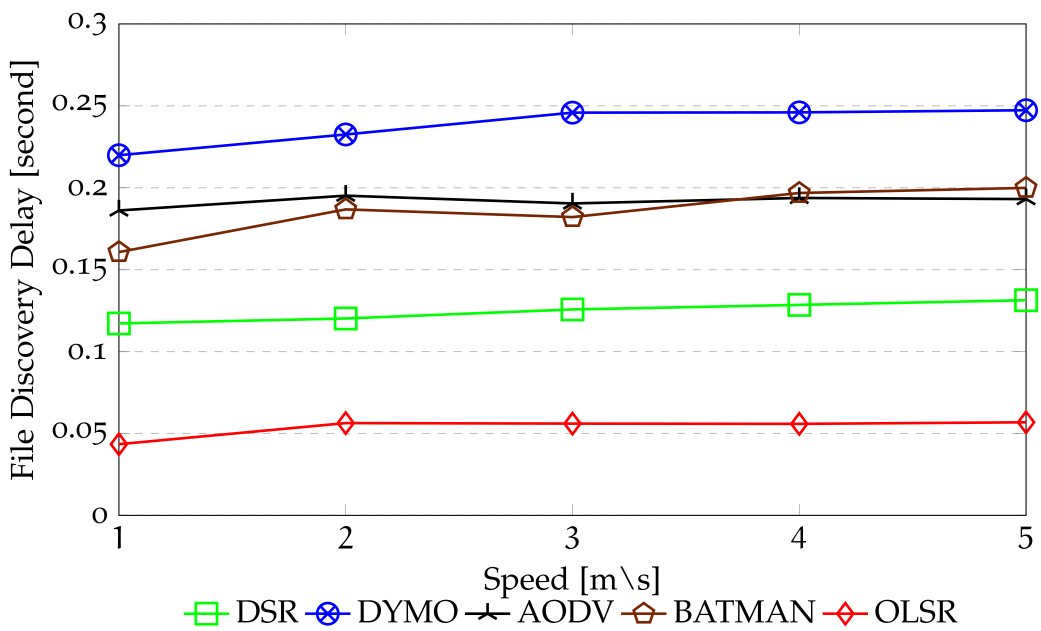


Figure 5.13: Discovery delay in 80 nodes network size and 60 s lookup frequency

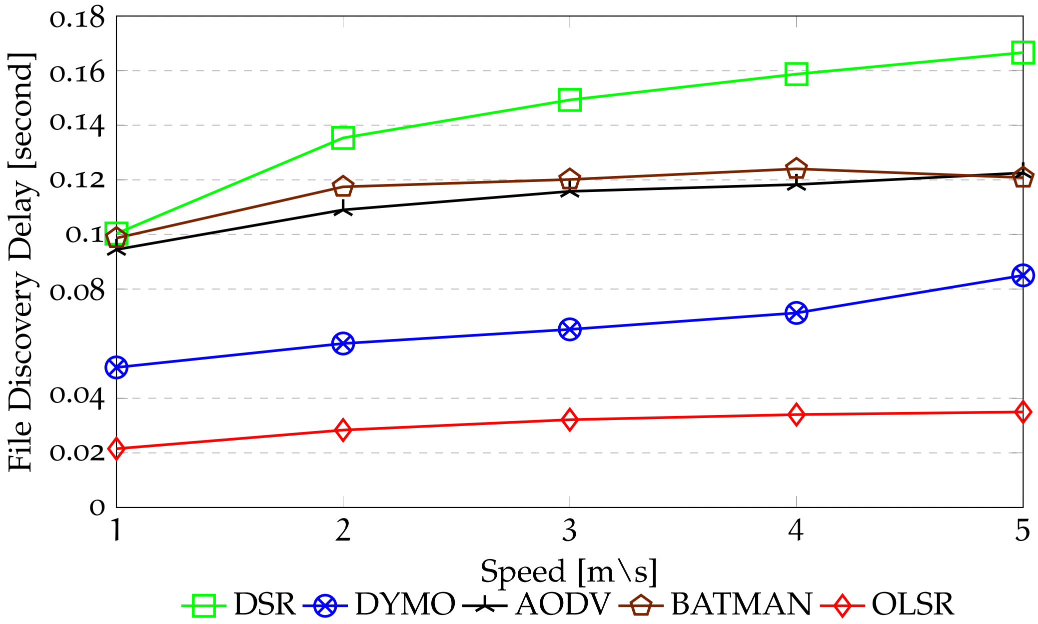


Figure 5.14: Discovery delay in 80 nodes network size and 10 s lookup frequency

The effect of sending key lookups more frequently can be seen in Figure 5.14 where lookups were sent every 10 seconds. The results show that the discovery delay is decreased as a result of sending more lookups when OneHopOverlay4MANET was combined with the different underlay apart from the DSR underlay.

As figure 5.14 shows, the DSR based system tends to yield the worst result in terms of discovery delay with more frequent key lookups. This can be seen as a result of the DSR routing strategy, where it caches a route or more for a destination in its cache and does not use any regular messages such as Hello messages to detect any link breakage. This would result in having stale entries in the DSR cache. Consequently, new Route Discovery might be required, resulting in more delay for solving overlay lookups. The same situation happens for the DSR based system when nodes speed increases, as figure 5.14 shows. High mobility causes link breakage and results in stale entries. The node will then need to discover a route again and this causes a higher delay for solving key lookups.

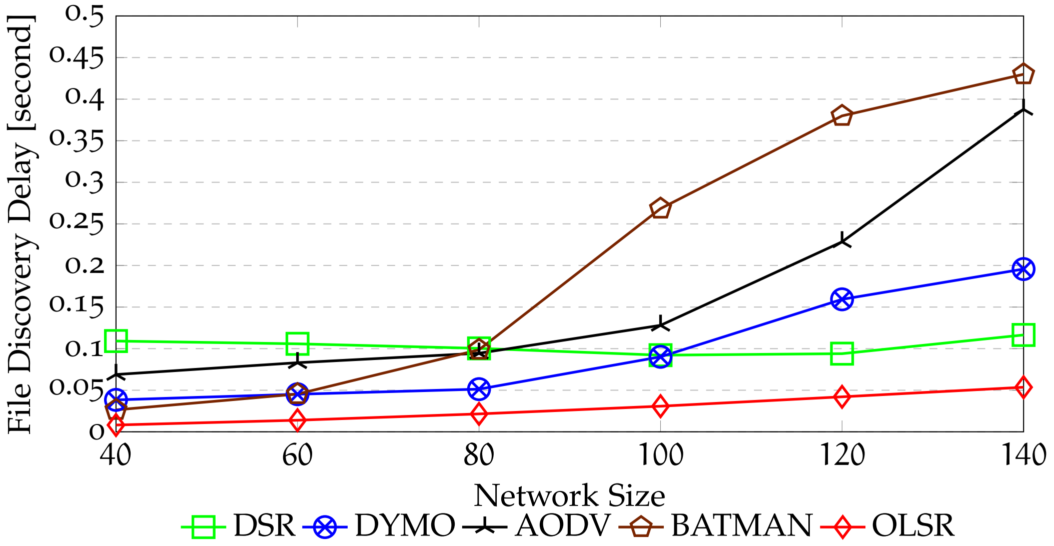


Figure 5.15: Discovery delay with 1 m\s speed and 10 s lookup frequency

Both the network size and the node speed affect the lookup latency. Figure 5.15 shows how the latency increases when the size of the network increases. For networks with less than 80 nodes, the lookup latency is around 100 milliseconds or less. However, as the network size grows, the latency of the BATMAN, DYMO and AODV based systems deteriorate significantly. The poor performance of the BATMAN based system is due to the overall poor performance of the BATMAN underlay routing when the network size increases as was shown in section 5.3.2. The DSR based system shows stable performance when the size of the network increases, compared to the AODV and DYMO based systems. This is a result of employing the multi path strategy by DSR where it employs caching more than a single route to a destination. However, AODV and DYMO stores only one route for a destination. Hence, if the entry is invalid, a new route discovery would be issued, resulting in higher delay.

The OLSR based system manages to achieve the least delay where it can solve overlay file lookups in an average of 50 milliseconds, and less even with a larger network consisting of 140 nodes. Clearly, this is due to efficient OLSR routing. OLSR is a proactive underlay protocol that maintains routing entries for every destination in the network prior to their use. This would contribute towards solving overlay key lookups in a minimum time.

The average number of logical hops that were required in order to solve key lookups is presented in figures 5.16 and 5.17. As can be seen, OneHopOverlay4MANET requires almost one hop regarding the logical overlay to retrieve an object from the network when combined over various underlay routing protocols. Both the network size and lookup intensity have a 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 the overlay routing tables get more updates through the underlay and the response messages of key lookups. Therefore, better logical hop count is achieved. Similarly, the hop count gets closer to one when lookups are sent more frequently as figure 5.17 shows.

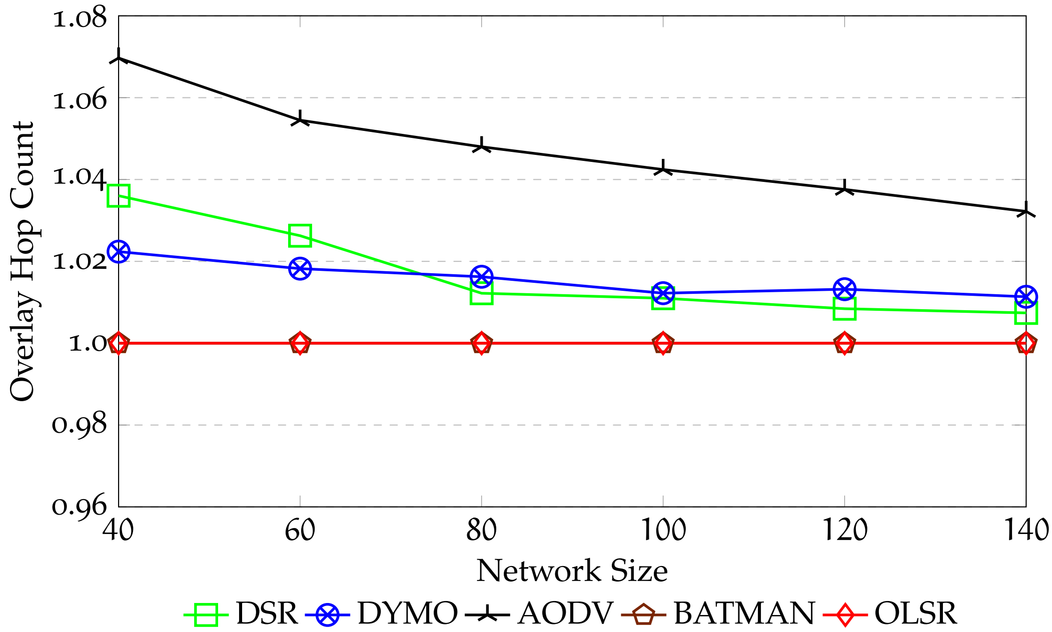


Figure 5.16: Logical hop count with 3 m\s speed and 60 s lookup frequency

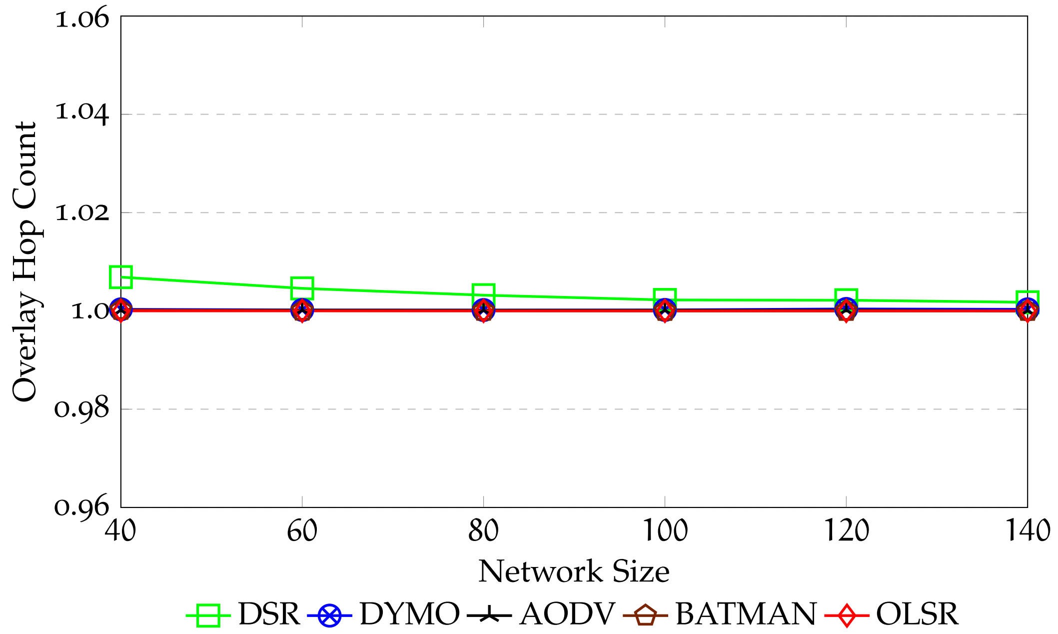


Figure 5.17: Logical hop count with 3 m\s speed and 10 s lookup frequency

5.3.4 *Network Load*

Figures 5.18, 5.19 and 5.20 depict the total amount of traffic that was generated in the network. This traffic includes both MANET and overlay traffic. Figure 5.18 presents the total load for a network that has mobile nodes moving at 1 m \s speed and lookups sent every 10 seconds. As is expected, when the network size grows, the total load in the network grows as well. However, the increase in the total load is seen to be remarkable when the network consists of 100 nodes and more for the AODV and DYMO based OneHopOverlay4MANET compared to DSR and OLSR based OneHopOverlay4MANET.

Despite the fact that OLSR is a proactive underlay, it manages to generate the least amount of traffic compared to others. The reason for this is that OLSR selects some nodes only to forward the control messages and hence avoids flooding the network. Therefore, it can manage to route packets under intensive demand without introducing much routing traffic. The DSR based system is the runner up system, and produced the second least amount of traffic. DSR purely works

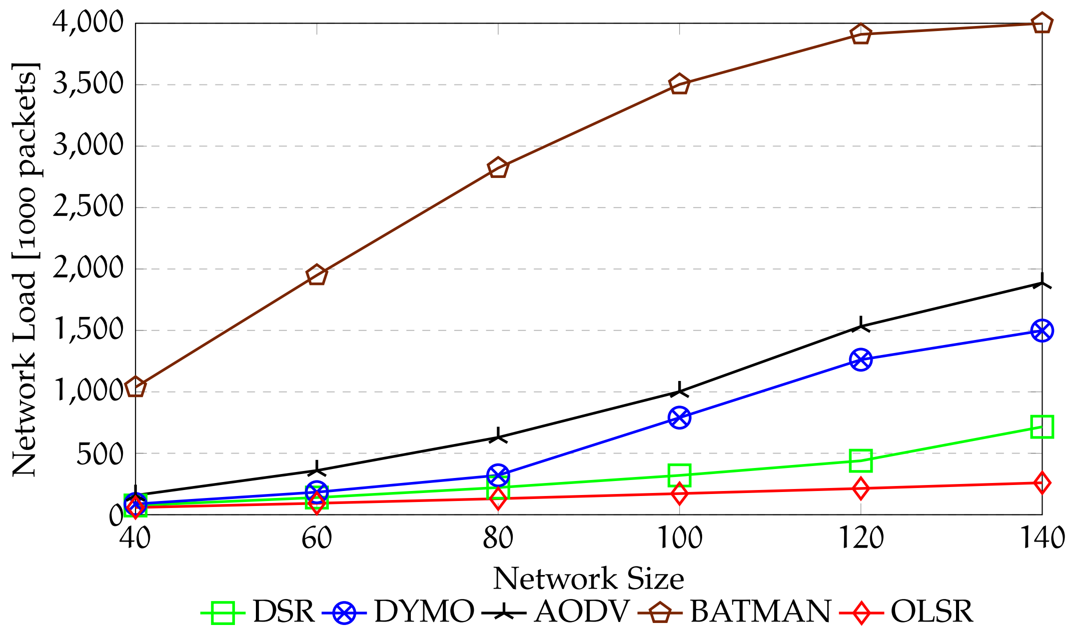


Figure 5.18: Network load with 1 m\s speed and 10 s lookup frequency

on demand and does not exchange any periodic messages to maintain its cache entries. In addition, DSR operates multiple path routing to a destination through caching, which contributes toward generating less traffic. Therefore, its generated traffic is less than the generated load by DYMO and AODV, which use Hello messages to detect any route failure, and hence maintains more accurate routing table entries. It can be seen as an advantage of DSR as it does not overload the network. However, this comes at the cost of having stale entries in its cache, which will result in undertaking route discovery and hence high file discovery delay, as seen in section 5.3.3 figure 5.14.

The total generated traffic with most of the adopted underlay protocols is less than 500 k packets with a lookup interval of 60 s and 5 m \s speed (figure 5.19). However, with a more intensive lookup network (every 10 s), the only system that manages to generate a similar amount of traffic (less than 500 k) is the OLSR based system (figure 5.20).

As the result indicates, the DYMO based OneHopOverlay4MANET is the most competitive system to the OLSR based one with file lookup frequency of 60

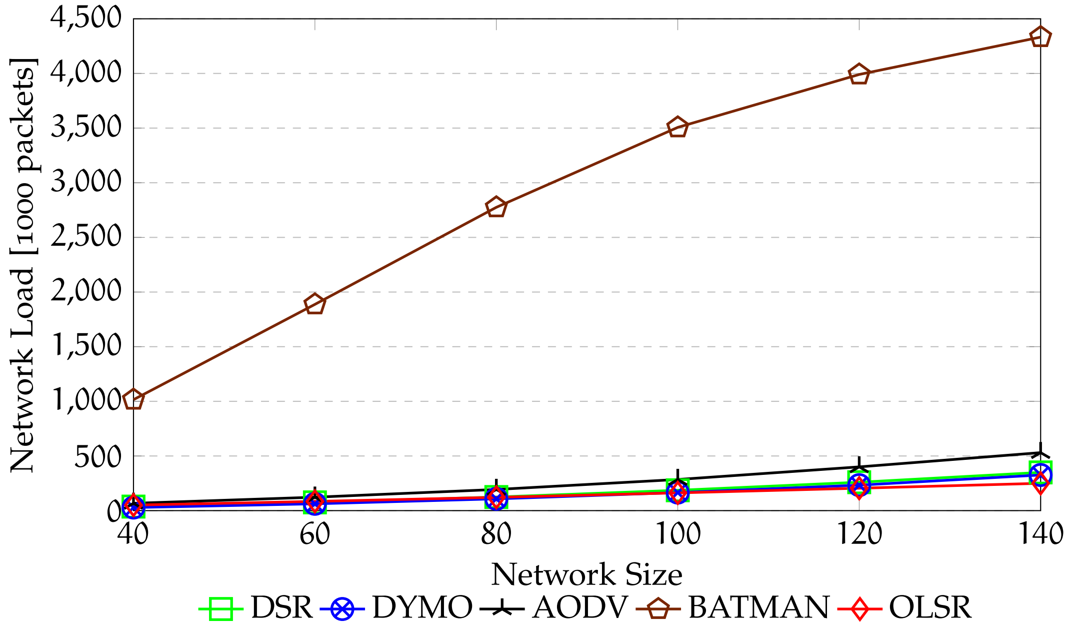


Figure 5.19: Network load with 5 m\s speed and 60 s lookup frequency

s. However, at a lookup frequency of 10 s, DYMO's performance deteriorates and its total traffic volume reaches as much as three times that of OLSR in a network with 140 nodes (figure 5.20). Thus, DYMO underlay routing efficiency is affected due to the high amount of traffic that may overwhelm the network. As a consequence, this would affect the efficiency of routing key lookup queries by the DYMO based OneHopOverlay4MANET and leads to a poor performance in terms of success rate as seen in section 5.3.2 figure 5.10.

On the other hand, the BATMAN based system generates the highest amount of traffic as a result of its flooding mechanism, where each mobile node needs to transmit regular Hello messages. These Hello messages are required to be rebroadcast throughout the whole network. With high mobility and large network size the BATMAN network gets flooded and generates remarkably traffic. This consequently affects the performance of the underlay routing and leads to poor OneHopOverlay4MANET success rate of solving key lookups (figure 5.9).

As figure 5.20 shows, the DYMO based system manages to generate less traffic than the traffic generated by the AODV based system. This is the result of exclud-

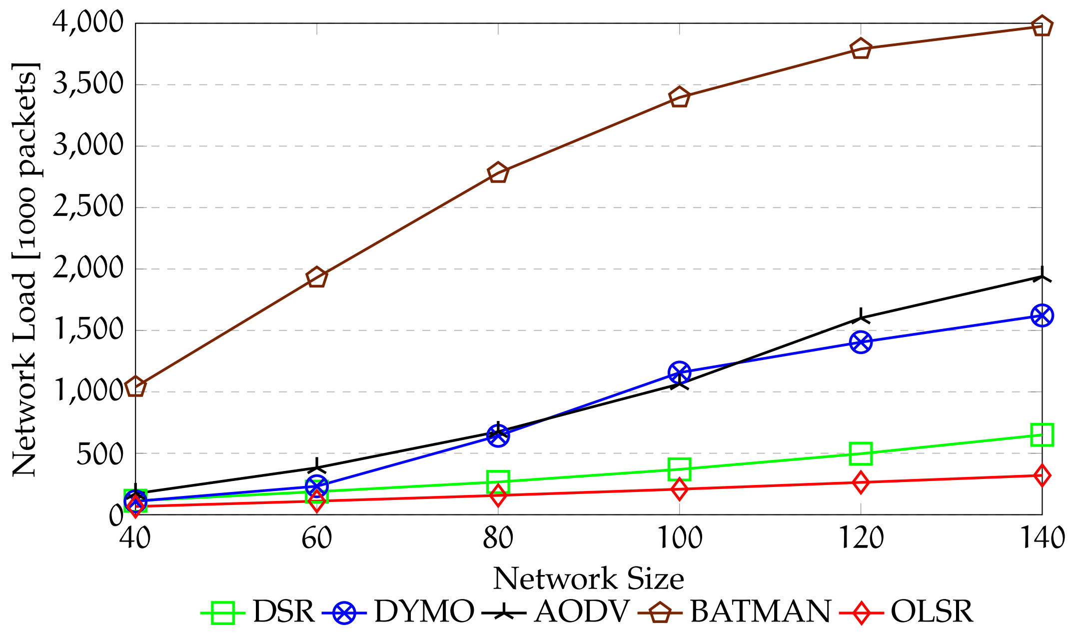


Figure 5.20: Network load with 5 m\s speed and 10 s lookup frequency

ing the use of Hello message that AODV uses to maintain existing routes. With a high demand from the overlay, more routes will be learnt by the ADOV routing agent, which requires more traffic to maintain them.

5.4 SUMMARY

In this chapter, the joining process for OneHopOverlay4MANET was illustrated. Two different procedures were used depending on the adopted underlay routing protocol. Join messages were used when OneHopOverlay4MANET was cross-layered with reactive underlay to allow a peer to join the overlay and push the reactive underlay to start. On the other hand, it was shown that with a proactive underlay, the underlay routing information can be sufficient to instantiate the overlay and such information can be used to enable a peer to join the overlay.

The proposed system was then evaluated with different types of underlay routing protocols. Cross-layering was implemented with five underlay routing protocols; proactive underlay (OLSR and BATMAN) and reactive underlay (DSR, AODV and DYMO). The best combination was achieved when OneHopOverlay4MANET was cross-layered with OLSR. Regarding the reactive underlays, the DYMO based OneHopOverlay4MANET gave a competitive performance when the overlay demand was not intensive. Moreover, the AODV based system gives a reasonable performance when the network size was small and its best performance is with more static networks. However, all the reactive underlays tend to overload the network when there is an intensive demand from the overlay or in cases when the node velocity is high and network size is large.

6 PERFORMANCE EVALUATION OF ONEHOPOVERLAY4MANET

In this chapter, further evaluation of the proposed OneHopOverlay4MANET system is carried out. Specifically, a performance evaluation is conducted for OneHopOverlay4MANET against two P2P systems (E-SP2P and MA-SP2P) which were proposed intentionally as a P2P overlay for mobile ad hoc networks. The proposed system shows an improvement over both systems as shown by the presented results.

6.1 INTRODUCTION

The proposed system, OneHopOverlay4MANET, was evaluated in the previous two chapters. Firstly, it was compared to two P2P protocols (Chord and EpiChord) which were deployed as they are over MANET. Both systems were just layered as they are, following the IP stack layering system. These systems, EpiChord and Chord, were chosen because OneHopOverlay4MANET builds a similar structure to their structure. However, OneHopOverlay4MANET adopts a different mechanism and optimizes the similarities between P2P overlay and MANET to maintain overlay consistency and scale down the typical traffic that results from deploying P2P over MANET.

Secondly, OneHopOverlay4MANET was cross-layered with different underlay routing types. The reason for carrying out such an evaluation was to investigate the suitability of the proposed system and how it behaves when deployed over different MANET technologies. It was cross-layered with OLSR and BATMAN as proactive underlays. Moreover, it was also cross-layered with DSR, AODV and

DYMO as reactive underlays. As the previous chapter concludes, OneHopOverlay4MANET achieves its best performance when combined with OLSR as the underlay routing protocol. However, the proposed system has not been evaluated against some P2P technologies that were developed specifically for mobile ad hoc networks. There are many P2P systems that were proposed for mobile ad hoc networks. Most of these systems have modified existing P2P technologies to suit mobile ad hoc networks or propose a similar structure to current internet-based P2P structures. In order to compare the proposed system to P2P MANET systems, OneHopOverlay4MANET was evaluated and compared to two recent systems, E-SP2P and MA-SP2P, that were proposed for MANET. Both systems are not only structured P2P for MANET but also adopt the same underlay as OneHopOverlay4MANET system. They adopt OLSR as the underlay routing protocol.

6.2 E-SP2P AND MA-SP2P

Efficient Structured P2P Overlay over MANET (E-SP2P) [82] hashes files to m-bit identifier and builds an identifier space ranging from 0 to 2^m -1. Each participant peer maintains a disjoint portion of the ID space. The portion that a peer is responsible for ranges from Lower End (LE) to Upper End (UE). Any hashed value of shared file that happens to be in this range will be under the responsibility of this node. A root peer is used in this system to organize the relationship between neighboring peers. It is used as a reference point to nominate one of each neighboring pair in the logical overlay to be in charge of maintaining the relationship. A peer which is closer to the root peer will be the one that sends the probe messages to its neighboring peer and hence maintains the relationship. The ID space is distributed between peers such that each peer P has to be connected to a peer P1, providing that P's lower end value is greater than P1's upper end value. Regarding logical overlay routing tables, each peer has to maintain a routing table that contains information about the root peer and neighbor peers.

When a peer joins the network, it sends an information request message to all of its surrounding peers. Each of the surrounding peers should reply with information about its directly connected peers, root peer and the portion of the ID space that happens to be under its responsibility. The joining peer will then construct a graph that consists of itself, its direct neighboring peers and 2 logical hops away neighbors. The graph will be used to execute a Minimum Spanning Tree (MST). The MST will be used to decide the joining peer's directly connected peers. The directly connected peer that has the largest portion of the ID space will be then chosen by the joining peer to share its ID space. The ID space will be split into two halves, and the lower half of the portion is allocated to the peer that happens to be furthest away from the root peer.

E-SP2P was extended by the MANET adaptive structured P2P overlay (MA-SP2P) [80]. MA-SP2P is similar to its predecessor, where a peer builds a graph that consists of itself, its directly connected peers and two logical hops away neighboring peers in the logical overlay. The graph is then used to construct MST where the peer itself is the source. However, MA-SP2P eliminates the use of the root peer mechanism which was used by E-SP2P as a reference to assign one of each pair of peers to take the responsibility for maintaining the relationship. In addition, MA-SP2P is different from E-SP2P in the limit of the portion of the ID space that a peer can be responsible for. In MA-SP2P, a peer can be in charge of the disjoint portion of the ID space which can be non-contiguous. However, a portion of the ID space at peer P (which can be non-contiguous) has to be consecutive to each P's directly connected neighbor peers. The upper end of the ID space portion at peer P points at one of P's neighbors that has a greater portion of the ID space. On the other hand, the lower end at P points to one of P's neighbors that stores the ID space portion with a lower value. Probe messages are exchanged among peers in the network, and when a peer receives updated information, it builds the graph of itself, its direct peer neighbors and 2 hops away logical neighbors. It then constructs the MST.

A joining peer broadcasts a join request message to find peers that are close in the physical topology. In response, a join reply should be sent by any neighboring peer that receives the request. A join reply consists of the initiator's directly connected neighbor peers, along with their distance in physical hops from the initiator of the join reply. These replies will be then used by the joining peer to construct a weighted undirected graph that consists of the joining peer itself, its directly connected neighbor peers and 2 hops away logical peers. The MST is consequently built from this graph. In order to take over some part of the logical ID space, the joining peer sends a connection-request message to its directly connected neighbor peers. Whenever a peer receives this message, it constructs its new MST and splits one of its continuous portion of the ID space into 2 halves and assigns one of them to the joining peer when any of the following happens:

- If a peer finds out from the MST that it can only reach one of its current directly connected neighbor peers through the joining peer. It splits the corresponding ID space with the joining peer.
- If the joining peer is a neighbor according to the new MST. In this case, it splits the largest continuous portion of the ID space and assigns one half to the joining peer.

Both systems try to push the overlay to match the physical underlay to avoid the criss-crossing problem. They employ a multi-hop overlay system where a peer maintains a minimum spanning tree of its directly neighbor peers and 2 logical hops away neighbor peers. The former approach has a key weakness due to its use of the root peer. All peers in the network will contact the root peer, which results in collisions and losing packets. Optimizing a root peer is not an ideal solution since the root peer may become a bottleneck when the size of peers gets larger. Moreover, the accuracy of the logical routing would be affected by the way that the ID space portions are exchanged. Due to mobility, peers will have to exchange ID space portions and build new relationships with new neighbors. This however, would result in less accurate logical routing.

In a similar way, MA-SP2P would be affected by the nature of mobility in MANET. As a consequence of mobility, a peer may find itself far away from its previous directly connected neighbor peers. This would require the peer to start the minimum spanning tree procedure and build new relationships with the new neighbors. In addition, a peer may have different portions of the ID space that are not contiguous. This is a result of the splitting process of the ID space portions whenever a peer builds a new relationship. Such splitting of the ID spaces would affect the efficiency of the system to route file lookup. This can be noticed on the performance evaluation of MA-SP2P when the peers speed is increased.

6.3 PERFORMANCE EVALUATION

6.3.1 *Simulation Set up*

The performance of OneHopOverlay4MANET is compared to MA-SP2P and E-SP2P. Table 6.1 provides an overview of the key simulation parameters. These parameters were chosen because they match the ones used in the E-SP2P and MA-SP2P papers and thus the results are comparable. Through simulation it is shown that OneHopOverlay4MANET outperforms MA-SP2P and ESP2P based on data lookup latency, network load and lookup failure. The authors of MA-SP2P and E-SP2P have previously shown that their proposal outperforms Modified Chord [89] and P2P-WANT [90]. For clarity in the graphs, these results are not represented here. Thus the proposed system also outperforms Modified Chord and P2P-WANT.

The used parameters are listed in Table 6.1. All the simulated scenarios were repeated ten times and the result is the average of the repetitions. Random churn for P2P overlay is used, where nodes randomly join and leave the network providing that the stated peer ratio is maintained. The used mobility model is the Random Way Point model, which is commonly used for simulating ad hoc networks and was used by MA-SP2P and E-SP2P. For each of the simulated scenarios, the network is given about 60 seconds to stabilize. After the 60 seconds, the measurements of the metrics begin. Lookups for shared files are randomly initiated for about 100 random files in the network. Metrics were monitored with different scenarios of the network. The ratio of peers in the network is changed from 10% up to 50% of the network size. Networks with 10%, 20% 30%, 40% and 50% as the peer rate to 100 mobile nodes are simulated. Different node mobility speeds were considered. The speed was varied from 0.4 m\s to 1.6 m\s.

Table 6.1: Simulation Parameteres

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

The following performance metrics are evaluated from the conducted simulations:

- Average file discovery delay: The average amount of time required in order to solve a lookup query. It starts from sending a key lookup until the time when a peer receives the answer of that lookup. This reflects the ability of the system to retrieve a shared key on the network.
- Network traffic load: The total number of packets transmitted at the routing layer in the network.
- Fail rate: The ratio of the number of unanswered file lookup queries for files that exist in the network to the total initiated file lookup queries in the network.

6.3.2 *Result*

6.3.2.1 *File Discovery Delay*

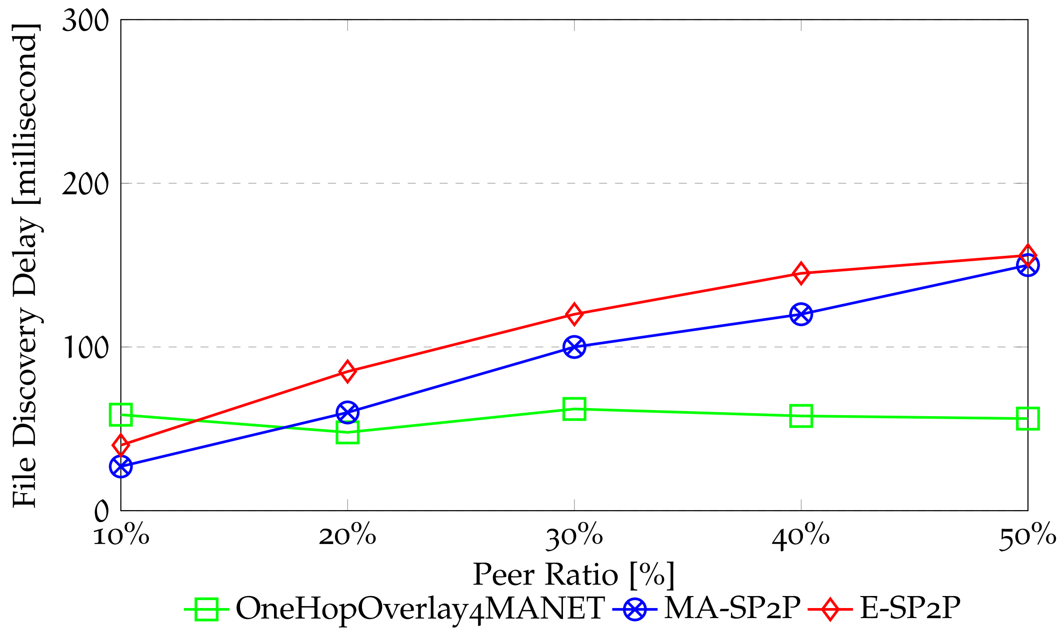


Figure 6.1: File discovery delay at 0.4 m\s node speed

Figure 6.1, 6.2, 6.3 and 6.4 show the file discovery delay for all three systems with different peer ratios for 0.4 and 0.8 as well as 1.2 and 1.6 m\s speeds. Looking at each individual graph, it can be seen that the latency for a lookup in OneHopOverlay4MANET stays about constant at 70 ms as the number of peers in the network increases. This is in contrast to the MA-SP2P and E-SP2P systems where the latency for the 50% peer ratio network increases markedly to about 3 or 4 times the latency experienced at 10% peer ratio.

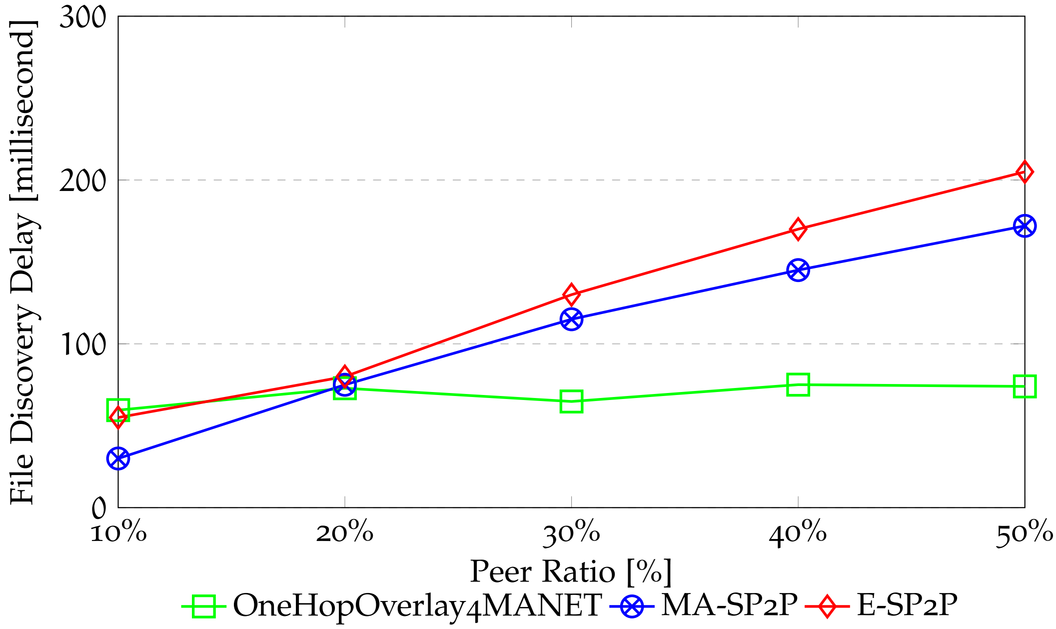


Figure 6.2: File discovery delay at 0.8 m\s node speed

For most of the simulated scenarios across different node speeds and peer ratios, OneHopOverlay4MANET manages to achieve an equal or less latency than the other two systems. The performance of E-SP2P and MA-SP2P is also affected by increased node velocity, albeit to a lesser extent than the peer ratio. As the figures show, the file discovery delay with 50 % peer ratio network at 0.4 m\s node mobility of about 150 ms increases to about 250 ms for 50 % peer ratio at 1.6 m\s node mobility. An obvious reason for the increased discovery delay with the higher speed scenario is the resulting change in the MANET underlay topology which affects the overlay routing efficiency. Hence, peers may move away from their recently connected neighbors. However, this would require building new relationships with the new neighbors, which involves reconstructing a minimum spanning tree.

OneHopOverlay4MANET achieves better results in terms 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 lookup 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 may go through a number of overlay hops with a queried peer forwarding the lookup to other peers in the overlay until the lookup is resolved. In both systems, each peer maintains only information about up to 2 logical hop away neighbors. As a consequence, a key lookup will not be resolved from the first logical hop unless it resides in one of the 2 hop away neighbors. Clearly, this would lead to a higher discovery delay when resolving key lookups.

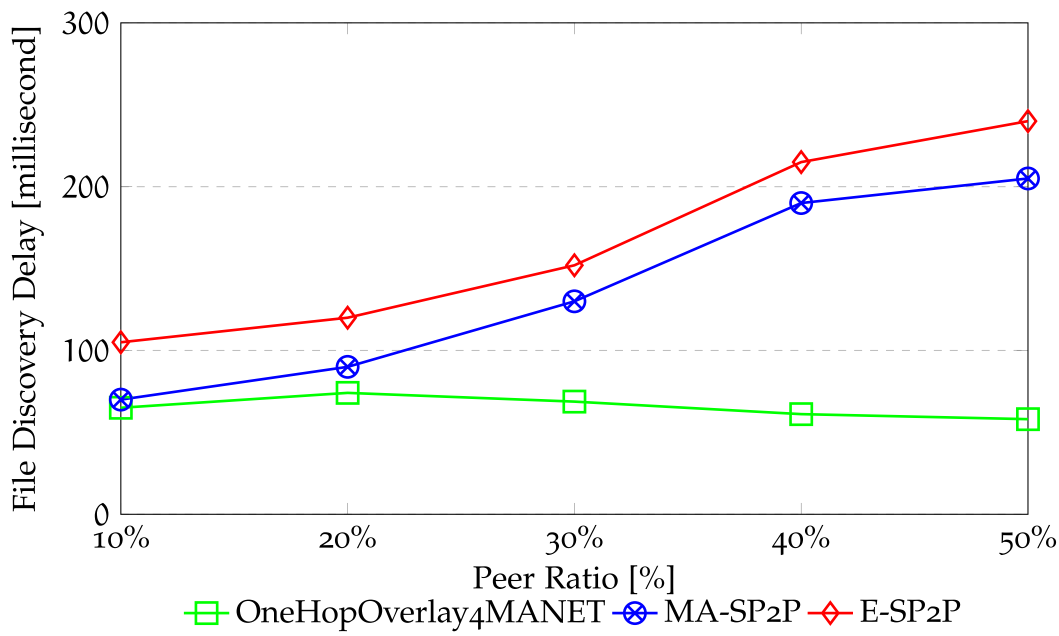


Figure 6.3: File discovery delay at 1.2 m\s node speed

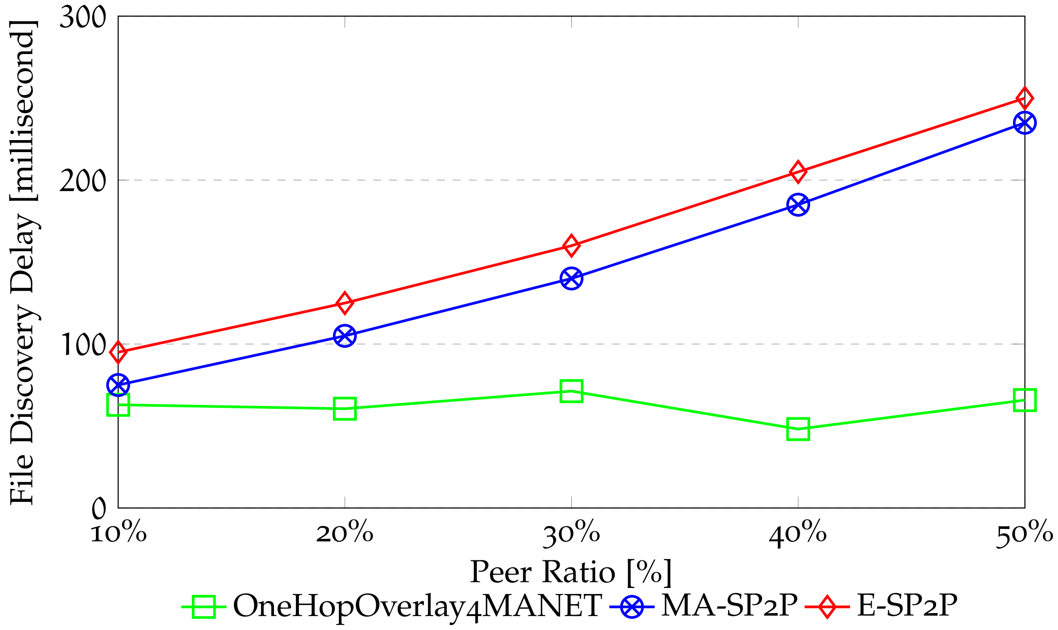


Figure 6.4: File discovery delay at 1.6 m\s node speed

6.3.2.2 *Network Load*

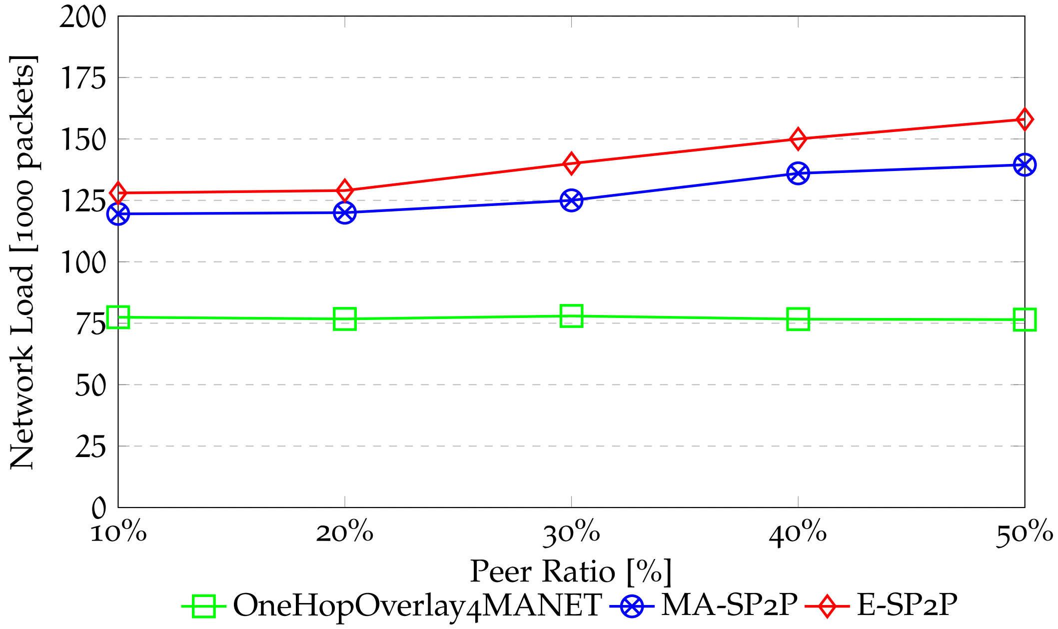


Figure 6.5: Network load at 0.4 m\s node speed

Figures 6.5, 6.6, 6.7 and 6.8 show the generated traffic in the network. As a consequence of optimizing the underlay (OLSR) routing information through cross-layering, OneHopOverlay4MANET generates the lowest amount of traffic in all the simulated scenarios over increasing overlay sizes. Adding new peers to the OneHopOverlay4MANET network by increasing the overlay peer ratio will not affect the overall traffic since underlay routing information is fully optimized and overlay management traffic released by OneHopOverlay4MANET is kept to a minimum. For example as figure 6.8 shows, OneHopOverlay4MANET loads stay almost steady when increasing the overlay size (peer ratio) from 10% up to 50% of the network size. The total released traffic is under 100 k for all the different overlay sizes. However, as a result of running intensive overlay management, MA-SP2P and E-SP2P have a load increase from 150 k with 10% ratio to 190 k with 50% and from 175 k with 10% ratio to 225 k with 50% ratio respectively (figure 6.8). This is an increase by approximately 40 k packets for MA-SP2P and about 50 k packets for E-SP2P.

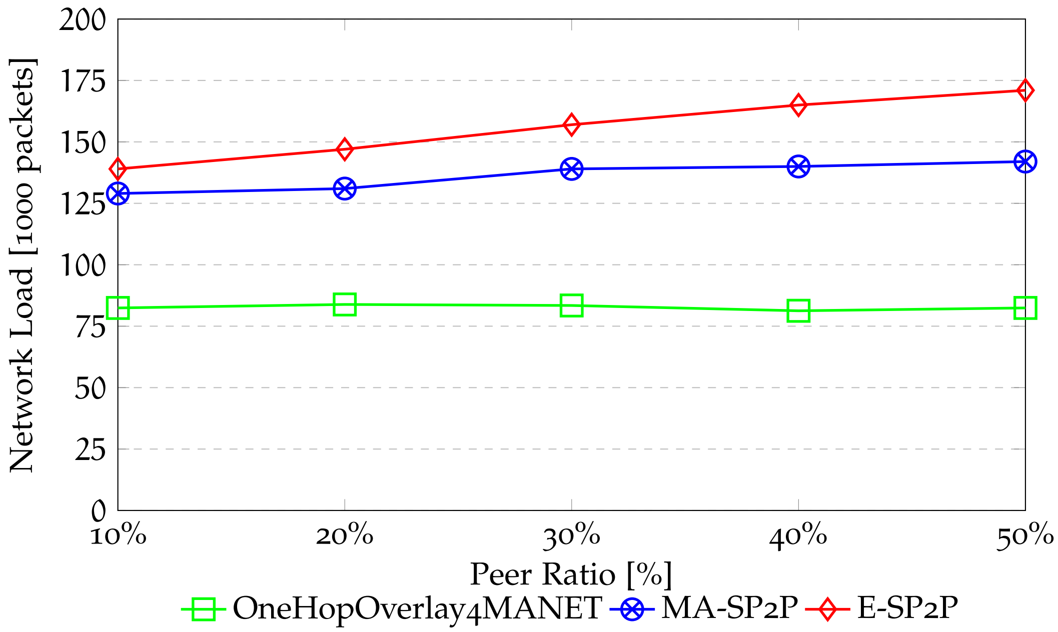


Figure 6.6: Network load at 0.8 m\s node speed

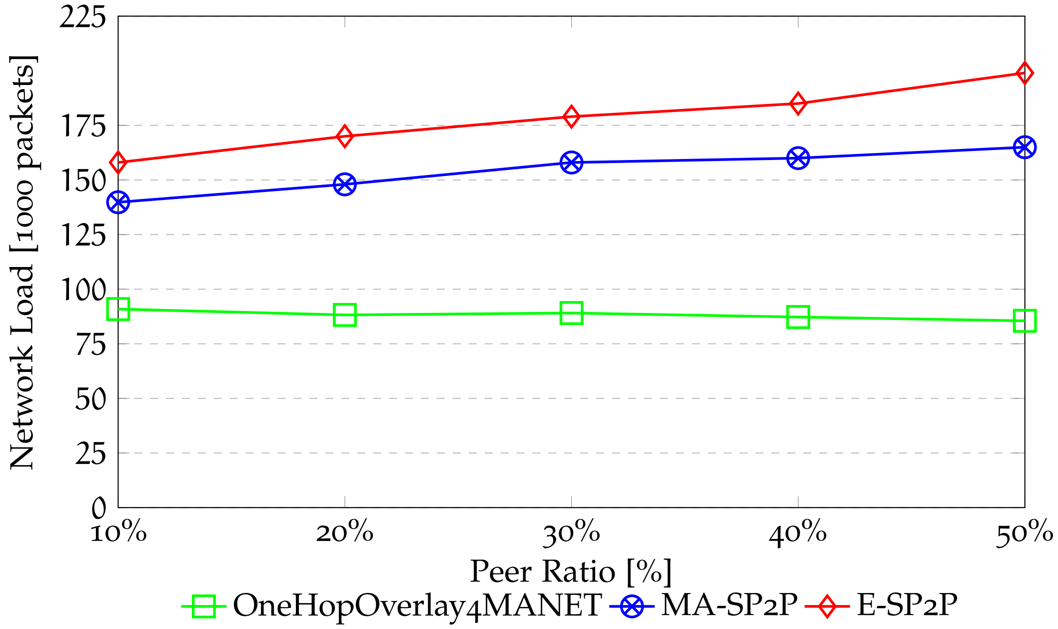


Figure 6.7: Network load at 1.2 m\s node speed

Across different node velocities, OneHopOverlay4MANET traffic varies only slightly, with about 75 k at 0.4 m\s (figure 6.5) up to about 90 k at 1.6 m\s (figure 6.8) being required with 50% peer ratio. This is expected, as nodes move away from each other causing link failure. However, the additional traffic is mainly caused by the underlay routing agent since OneHopOverlay4MANET does not operate separate overlay management. This is a result of exploiting the synergies between the underlay and the overlay. OneHopOverlay4MANET uses the underlay routing tables to populate the overlay routing tables without incurring additional traffic in the network, and consequently cuts the typical P2P management traffic.

As the figures show, MA-SP2P and E-SP2P loads increase with larger overlay sizes. A reason for this is that when increasing the peer ratio, new traffic is generated by the new peers, and more traffic is required to maintain the consistency of the logical overlay. Moreover, when the node speed increases, all three systems incur more traffic in the network as a result of node movement and link failure. Nonetheless, the generated extra traffic required by OneHopOverlay4MANET is much lower than the additional traffic required by MA-SP2P and E-SP2P. MA-

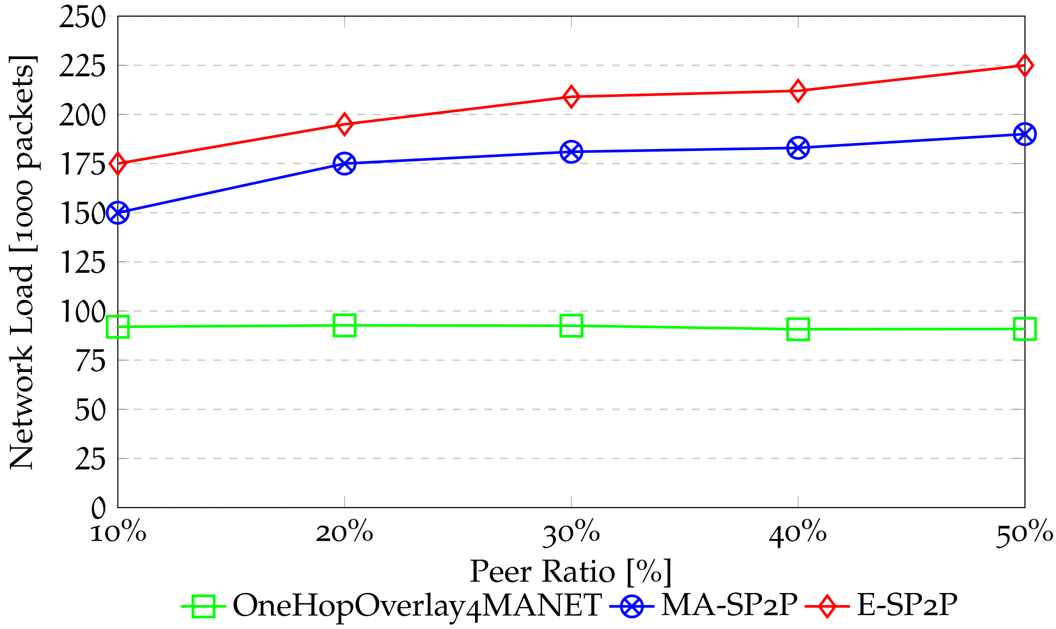


Figure 6.8: Network load at 1.6 m\s node speed

SP2P and E-SP2P produce more than 140 k at 50% ratio and 0.4 m\s node speed (figure 6.5), and about 200 k at 50% node ratio and 1.6 m\s (figure 6.8). This is about double what OneHopOverlay4MANET needs (It generates less than 100 k at 1.6 m\s and 50% peer ratio).

Clearly, OneHopOverlay4MANET can handle the logical routing updates which are required due to increased node mobility much better than MA-SP2P and E-SP2P systems. The reason for this is that with MA-SP2P and E-SP2P, peers may move away from their connected logical neighbor peers. as a consequence, peers are required to operate the minimum spanning tree procedure and create new relationships with new neighbor peers. Hence, more traffic will be introduced.

6.3.2.3 *Fail Rate*

Figures 6.9, 6.10, 6.11 and 6.12 depict the ratio of failed lookups for existing keys to the total initiated lookups. They indicate that MA-SP2P and E-SP2P performance decreases when the peer ratio increases. In other words, the fail rate of both systems goes up as a result of having more peers in the network. A likely reason for this is that as the peer ratio increases, more traffic is created to build the relationships between neighbors and maintain the overlay consistency. This results in an increased number of collisions and lost packets. Hence, the performance of the overlay routing is compromised.

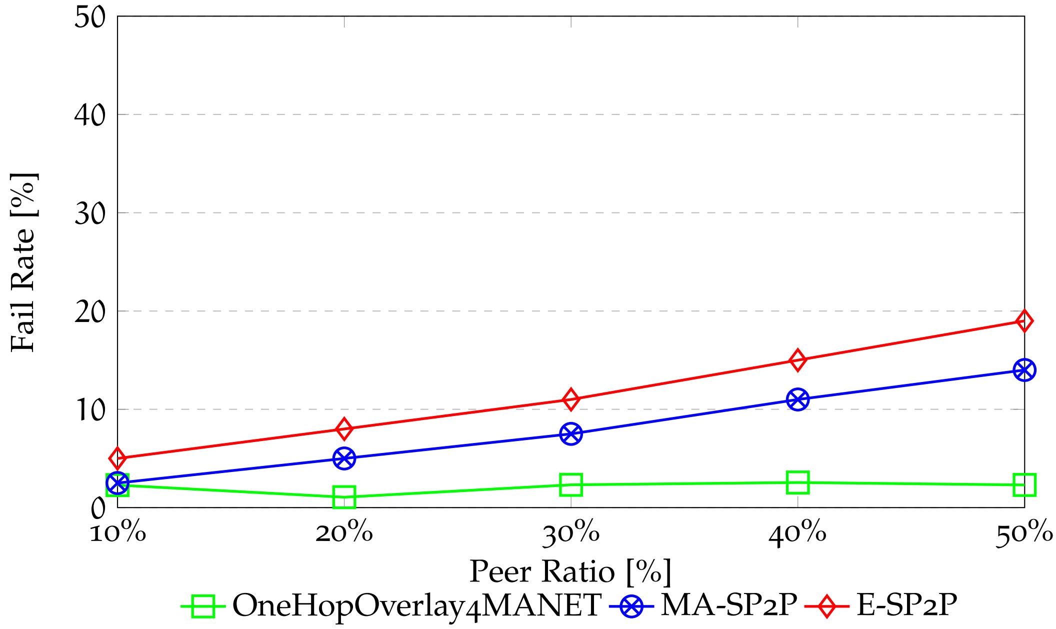


Figure 6.9: Fail rate at 0.4 m\s node speed

Moreover, the performance of both systems is influenced by the movement speed of the mobile nodes. It can be seen that, when the node speed is increased to 1.6 m\s and half the existing nodes participate in the overlay, the success rate drops to under 80% , which is more than a 20% fail rate as figure 6.12 shows. One possible reason for this drop is the frequent topology changes which causes the peers to incur more maintenance traffic to keep the entries in the overlay routing tables up-to-date. These require peers to rebuild the minimum spanning trees and build new relationships with new neighbors. This however, results in more traffic being released to maintain the consistency of the overlay. Consequently, packets collision and loss may accrue resulting in inefficient overlay routing.

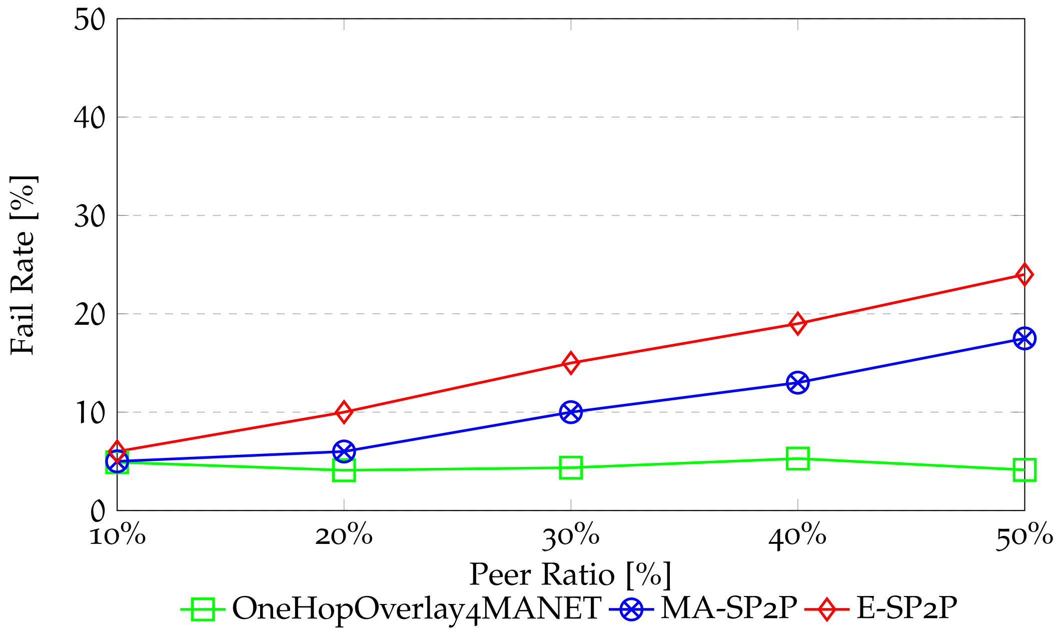


Figure 6.10: Fail rate at 0.8 m\s node speed

Furthermore, OneHopOverlay4MANET outperforms both systems, especially with a higher peer ratio. The main reason for this is that the increase in the number of peers will not cause a noticeable increase in the traffic, since each peer in OneHopOverlay4MANET uses the underlay routing information to populate the overlay. Even with higher node speeds, OneHopOverlay4MANET manages to maintain accurate routing tables and keep the fail rate at less than 10%, which means resolving more than 90% of the lookup queries. This is because OneHopOverlay4MANET does not introduce further management traffic with the higher moving speed. It only optimizes the underlay routing information and hence it can maintain good performance.

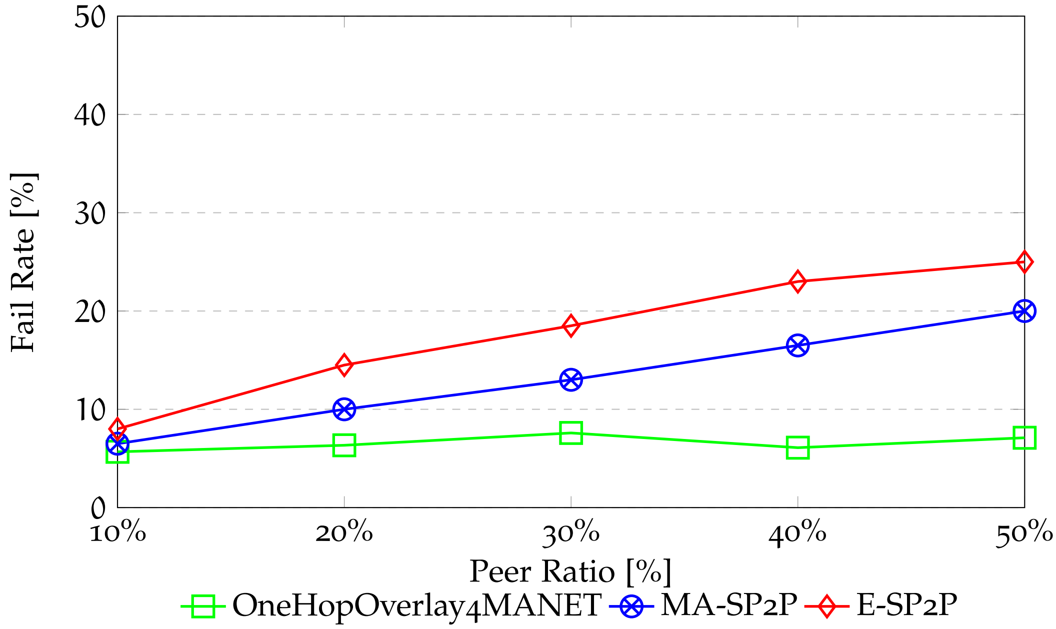


Figure 6.11: Fail rate at 1.2 m\s node speed

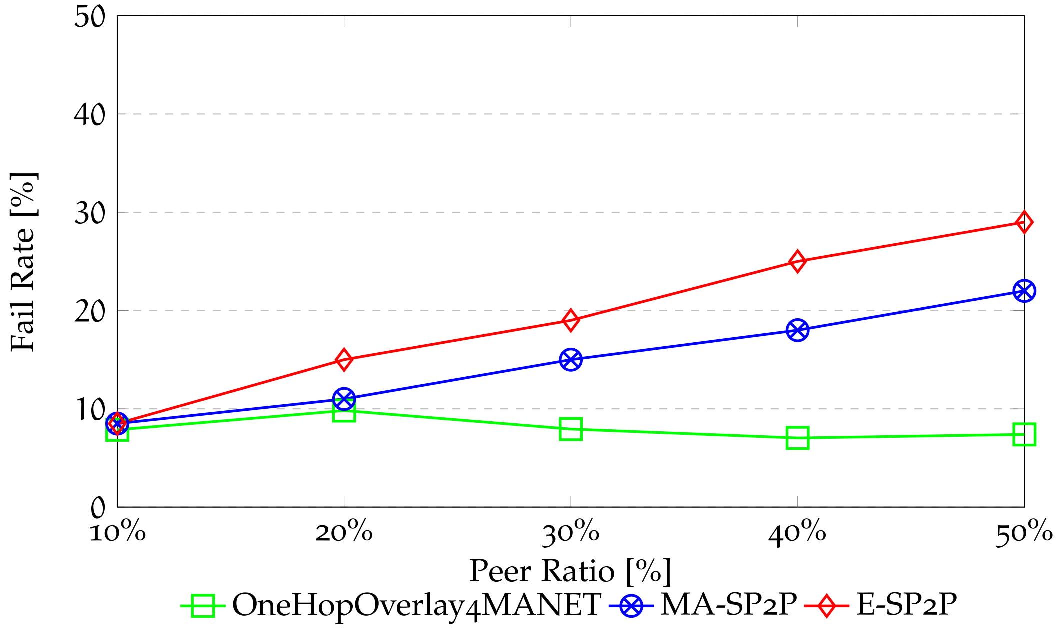


Figure 6.12: Fail rate at 1.6 m\s node speed

6.4 SUMMARY

This chapter focused on the comparison of the proposed system, OneHopOverlay4MANET, to existing P2P over mobile ad hoc network solutions. The performance of the proposed system was compared to the performance of two recently developed systems that proposed structured P2P overlays for mobile ad hoc network (E-SP2P and MA-SP2P). All three systems (OneHopOverlay4MANET, E-SP2P and MA-SP2P) adopted the same underlay routing protocol, OLSR. The results of the conducted experiments showed that OneHopOverlay4MANET outperformed E-SP2P and MA-SP2P in all the monitored metrics. It managed to retrieve a shared file in the minimum time compared to the required time by E-SP2P and MA-SP2P. In addition, due to optimizing the underlay routing information through cross-layering, OneHopOverlay4MANET produced the least amount of traffic when increasing the number of peers in the overlay. Meanwhile, it achieved better results in terms of successfully retrieving shared objects from the overlay.

7

CONCLUSION AND FUTURE WORK

This chapter concludes the thesis and discuses potential future work. It provides a summary of the undertaken study, reviews the research aims and objectives, and discusses how they were addressed. Limitations of the presented work and possible future research direction are then discussed.

7.1 THESIS SUMMARY

This research has investigated the applicability of deploying an efficient P2P system over mobile ad hoc networks. The thesis began by specifying the thesis statement, aims and objectives of the thesis. The research contributions were also stated in the first chapter.

P2P and MANET networks were introduced in the second chapter to provide the background of the research domains. This included introductions to both networks and descriptions of a number of protocols from each of the networks which are of interests for the research presented in this thesis including Chord, Pastry, Kademlia, EpiChord, OLSR, BATMAN, DSR, AODV and DYMO.

Chapter three focused on P2P over mobile ad hoc networks. It discussed the challenges of deploying P2P over MANET, and then introduced the possible approaches of the deployment namely, straight deployment, integration and cross-layering. A wide range of relevant related works were then surveyed. It was observed that there are different purposes for using P2P in the MANET domain. Some of the existing systems have used P2P techniques to enhance the performance of traditional MANET routing protocols. On the other hand, other systems have deployed P2P over MANET to serve the purpose it was originally designed for, Data storage and retrieval, as with P2P over the traditional internet.

Existing solutions for P2P over MANET for data storage and retrieval were further classified and explained by chapter three. They were classified according to the underlay used. These were proactive underlay based systems, reactive underlay based systems, GPS-based systems, and finally, underlay independent systems. In addition, all the surveyed systems were also discussed in terms of adopted P2P systems, the use of clustering and registration techniques, consideration of locality or physical proximity, which deployment approach was used, and finally, how the performance of the proposed systems was evaluated.

As a result of this review it was concluded that there is no adoption of one-hop P2P systems into MANET. All the adopted structures were multi-hop P2P structures or modified structures similar to multi-hop P2P systems to suit MANET. An obvious reason for this is that one-hop overlays are well known for their intensive maintenance traffic, which can lead to overwhelming constraint network such as MANET.

Chapter four started by stating the biggest challenge when deploying a P2P system over MANET. Each single logical hop in the overlay may result in a path in the MANET underlay topology. As a consequence, each logical hop in the overlay is translated into multiple physical hops. Peers which are neighbors in the overlay are likely to be separated by many hops in the MANET topology. When multi-hop P2P systems are used in such settings, each overlay (logical) hop represents multiple hops in the underlay. Progressing through the overlay path to the final destination may result in contacting some MANET nodes more than once and may well mean passing MANET nodes which are very close to the last destination of the logical route. Thus, multi-hop P2P overlays are not well suited to MANETs. One-hop overlay is much better suited as they avoid the aforementioned criss crossing issue.

In chapter four, the aforementioned issue was experienced and evaluated by deploying multi-hop overlays and one-hop overlay over MANET. For first time, this thesis presented the first (to the best knowledge of the author) investigations of the performance of one-hop overlay in MANET by deploying the EpiChord system. EpiChord can achieve one-hop performance under intensive lookup usages. OMNeT++ was used as the simulation environment, and the OverSim and INET frameworks were used to carry out the simulations. The straight layering approach was adopted to investigate the performance of EpiChord, Chord, Pastry and Kademlia. All the overlays were layered as they are over OLSR as the MANET underlay routing protocol.

As noted before, EpiChord, when achieving one-hop performance, manages to achieve better success rate compared to other simulated P2P systems. The result proved that one-hop overlay is more suited to MANET and is a promising solution to the criss crossing issue. However, such a system needs to be optimized to better suit MANET and produces better performance.

The proposed system presented in this thesis, OneHopOverlay4MANET, is then introduced in chapter four. OneHopOverlay4MANET employs a logical structure similar to Chord and EpiChord. It assigns a permanent logical identifier for mobile nodes by hashing their IP addresses. The logical identifiers are then ordered in a logical circular address space. Data items are hashed and stored on the peer that most closely follows the item's key in the logical space, which is known as the successor. OneHopOverlay4MANET optimizes the similarities between the overlay and the underlay to build an efficient P2P system for MANET. OneHopOverlay4MANET is designed to construct an overlay system for MANET that can resolve lookup queries for shared items in a short time, incurring minimum traffic in the network.

OneHopOverlay4MANET maintains a Successor list and a Predecessor list to guarantee consistency in the network. Each OneHopOverlay4MANET peer maintains routing information to four succeeding peers and four preceding peers. This is to guarantee DHT functionality when OneHopOverlay4MANET does not resolve a key lookup in the first step. OneHopOverlay4MANET also maintains a cache to store as much information as possible about existing peers in the network.

OneHopOverlay4MANET opts for using cross-layering to fully optimize underlay routing information. The cross-layer channel is constructed between the application layer, where the overlay operates, and the network layer, where MANET routing operates. OneHopOverlay4MANET uses the underlay routing information as the main source to build the logical overlay. MANET routing information is passed up to the overlay using the constructed channel. Consequently, using underlay routing information reduces the overlay maintenance traffic.

In the employed cross-layering mechanism, an underlay routing protocol is modified to notify the overlay of any changes in its routing tables through the cross-layer channel. Whenever the routing agent learns new routes, it sends the IP addresses of all the nodes, to which it has a route, up to the overlay.

Upon receiving notifications from the underlay, a OneHopOverlay4MANET peer checks first whether the logical ID of each received entry falls between itself and the last entry in its Successor list. If it does fall in this range, the ID will be placed in the Successor list. Secondly, the same procedure will be applied to the Predecessor list to check whether the peer's logical ID falls in the range of the Predecessor list. Otherwise, the received entry will be stored in the cache. This, however, would insure that each peer in the overlay maintains a Successor list and a Predecessor list to guarantee DHT performance. In addition to the use of the underlay routing information, OneHopOverlay4MANET uses lookup responses to update the overlay routing tables. Each lookup response contains some overlay routing information from the originator of the lookup response.

For joining the overlay, OneHopOverlay4MANET considers two methods; joining the network through optimizing underlay routing notification or joining the network through the use of join messages. In the former method, underlay information can be sufficient to instantiate the overlay, providing that the used MANET protocol is a proactive routing protocol. The reason for this is that a proactive protocol builds and populates its routing table without any need from nodes to push them to start. Consequently, the overlay gets fed and instantiated from the underlay update through cross layering. However, this is not the case when combining OneHopOverlay4MANET with a reactive underlay that works only on demand. If no join messages were used, this would result in a reactive overlay over a reactive underlay. Thus, the underlay routing protocols would not start. For the latter method, OneHopOverlay4MANET uses joining messages to allow a peer to join the overlay. In addition, such join messages would generate demand on the reactive underlay which would result in more routing entries in the underlay routing table. Hence, this would be reflected in the overlay when it gets notified of such entries. Chapter five covers the procedure that a peer goes through in order to join the overlay. It illustrates the joining process using examples of both a proactive routing protocol and a reactive routing protocol.

Implementation of the proposed system was developed in the OverSim framework along with the INET framework for the OMNeT++ simulation environment. The performance of the proposed system was compared to the performance of EpiChord and Chord when they were layered as they are over MANET. Chapter four shows the first series of OneHopOverlay4MANET's evaluation. OneHopOverlay4MANET manages to achieve better performance across different simulated scenarios. The results show that a better success rate for solving key lookups was achieved alongside with emitting the least amount of traffic in the network compared to EpiChord and Chord networks. In terms of required logical hops to retrieve a shared item, OneHopOverlay4MANET manages to solve key lookups in almost one logical hop.

OneHopOverlay4MANET suitability over different MANET technologies was also investigated. The proposed strategy of optimizing underlay routing information through cross-layering to build one hop overlay for MANET was adopted in this thesis with several underlay routing protocols. In chapter five, OneHop-Overlay4MANET was deployed over two proactive MANET routing protocols, namely OLSR and BATMAN, and over three reactive MANET routing protocols, namely DSR, AODV and DYMO. The best performance was achieved when the proposed system was cross-layered with OLSR. Regarding the reactive underlays, the DYMO based OneHopOverlay4MANET produced a competitive performance when the overlay demand was not intensive. Moreover, the AODV based system produced a reasonable performance when the network size was small, and its best performance was with more static networks. However, all the reactive underlays tended to overload the network when there was an intensive demand from the overlay or with cases when the node velocity was high and network size was large.

In chapter six, further evaluation of OneHopOverlay4MANET was carried out. The performance of the proposed system was compared to two P2P systems, MA-SP2P and E-SP2P which were specifically proposed for MANET. OneHop-Overlay4MANET showed an improvement in the performance over both systems. As a result of building a one hop overlay, OneHopOverlay4MANET requires less delay to solve key lookups. In addition, it also achieved a better success rate. In scenarios where the overlay size increased, OneHopOverlay4MANET exhibited a linear performance trend compared to the exponential trend achieved by both MA-SP2P and E-SP2P.

7.2 Meeting the objectives

The aims and objectives of this research were stated in the first chapter of this thesis. The main focus was to design a one hop overlay system for mobile ad hoc networks to address the criss crossing issue. In order to satisfy the main aim, several objectives were developed and satisfied during this study. The following list reviews the objectives of this research and how they were addressed.

- **To investigate the performance of multi-hop and one hop overlay over MANET**
  Chapter four experimented with deploying multi-hop overlays and one-hop overlay as they are over MANET. It evaluated the performance of EpiChord over MANET for the first time to the best knowledge of the author. It concludes that one-hop overlay is a promising solution for P2P networks over MANET. However, one-hop systems are well known for their high amount of maintenance traffic. Thus, such a deployment requires more optimization to suit MANET better.

- **To design a one hop overlay system for MANET that optimizes the synergies between MANET and P2P networks**
  Chapter four presented the proposed system in this thesis to build a one-hop overlay for MANET optimizing all the possible synergies between these technologies. It presented OneHopOverlay4MANET to solve lookups in a short time while incurring less traffic in the network and providing highly accurate logical routing.

- **To investigate the use of Cross-Layering**
  The proposed system has built a cross-layer channel between MANET underlay routing protocols and the overlay to allow routing information exchange between Network Layers. The underlay routing information is then used to build the overlay routing tables.

- **To reduce overlay maintenance traffic**
  The designed system, OneHopOverlay4MANET, reduced typical overlay maintenance traffic through using underlay routing information. The adopted MANET routing protocol's routing table contents are passed up to the overlay to provide a view of the existing nodes in the network. So, OneHopOverlay4MANET limits the need for the typical quantity of management traffic employed by overlay systems.

- **To investigate the suitability of the proposed system over different MANET routing protocols**
  OneHopOverlay4MANET was combined with proactive underlays (OLSR and BATMAN) and with reactive underlays (AODV, DYMO and DSR). The cross-layer channel was built with all the aforementioned routing protocols. These were presented in chapter five.
- **To evaluate the performance of the proposed system**
  OneHopOverlay4MANET was comprehensively evaluated in different MANET scenarios, considering different velocities and network sizes. The performance of OneHopOverlay4MANET over OLSR was also compared with two recent P2P over MANET systems (MA-SP2P and E-SP2P) that use OLSR as the underlay routing protocol. These evaluations were presented in chapters four, five and six.

7.3 FUTURE WORK

Several limitations exist in the presented research. Such limitations represent possible future work directions to extend the proposed system. The following are some potential future work directions:

- OneHopOverlay4MANET optimizes the underlay routing information to build the overlay. The optimization of underlay information throughout this research was limited to the reuse of underlay routing information when building the overlay. More optimization of the synergy could be further investigated. In addition, Notifications from the underlay routing protocols are currently sent to the overlay whenever changes happen in the routing table of the adopted MANET protocol. This however, is not optimal when considering energy consumption. This area of the proposed system requires more investigation to identify the optimal frequency for emitting underlay notifications to the overlay. This would make the system more energy efficient, however, further investigation is required to consider energy efficiency.

- Due to node movement and restricted transmission range for mobile nodes, the network may face the partitioning issue. The partitioning issue happens when the network topology gets split and results in a number of disconnected parts of the logical overlay. This would mean that a peer in a separate part of the overlay would not be able to reach services in another part. This area requires more investigation to enable the system to detect and merge separated parts of the network when they get closer to each other. One possible strategy to overcome partitioning issue is to use instant recovery procedure when a neighbouring peer is disconnected. This would help to detect network partitioning before it occurs. Another approach that can be used is to use underlay routing information through cross layering to detect any new nodes in the network.

- The proposed system was evaluated using software simulation tools. When carrying out such evaluation, the most popular mobility model, Random Way Point mobility model, was used to simulate movements of mobile nodes. The performance of the system can be further investigated considering different mobility models. Moreover, churn rate is another factor that may require further investigation. The work in this thesis considered two types of churn. In the first, node mobility was considered to generate churn where mobile nodes may move to a one direction and end up out of range of other nodes. Random churn was the second type of churn considered, where some random nodes get killed during the simulation period. The effect of real churn has not been thoroughly considered. This could be further investigated to find the capability of the system when more nodes are frequently killed during the simulations.

- The work in this research was limited in terms of the use of a data item replication strategy. As potential future research direction, data replication can be investigated. In data replication, data items are replicated a number of times inside the network. This would increase reliability, availability and accessibility of shared items within the developed system, especially in a network like MANET where mobile nodes may disappear at will.

- The performance of the proposed system was evaluated for increasing network sizes up to 140 nodes networks. However, the performance of the system should be further evaluated with larger network sizes to investigate the scalability of the proposed system, OneHopOverlay4MANET.

- The proposed system was evaluated with P2P systems that can achieve one hop performance when deployed over MANET, namely EpiChord. However, further evaluation could be done to investigate the performance against other one hop P2P systems such as OneHop when deployed over MANET.

- The proposed system has been evaluated using a simulation tool. Implementing the proposed system in a real environment is another possible direction for future work. Obviously such implementation and evaluation is costly due to the required number of wireless devices, people and programming efforts to carry out such evaluation. However, this would reflect the performance of the system in the real world.

BIBLIOGRAPHY

[1] Q. H. Vu, M. Lupu, and B. C. Ooi, *Peer-to-peer computing: Principles and applications*. Berlin, Heidelberg : Springer-Verlag, 2010.

[2] J. F. Buford and H. Yu, "Peer-to-Peer Networking and Applications: Synopsis and Research Directions," in *Handbook of Peer-to-Peer Networking*. Springer US, 2010, pp. 3–45.

[3] J. F. Buford, H. Yu, and E. K. Lua, *P2P Networking and Applications*. Burlington, USA: Morgan Kaufmann, Elsevier, 2009.

[4] Q. Vu, M. Lupu, and B. Ooi, "Architecture of Peer-to-Peer Systems," in *Peer-to-Peer Computing*. Berlin, Heidelberg: Springer, 2010, pp. 11–37.

[5] "OpenNap: Open Source Napster Server," 2001. [Online]. Available: Http://opennap.sourceforge.net/

[6] T. Klingberg and R. Manfredi, "Gnutella 0.6," 2002. [Online]. Available: http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html

[7] M. Duigou, "JXTA Protocol," 2002. [Online]. Available: http://tools.ietf.org/pdf/draft-duigou-jxta-protocols-00

[8] I. Clarke, O. Sandberg, B. Wiley, and T. Hong, "Freenet: A distributed anonymous information storage and retrieval system," in *Designing Privacy Enhancing Technologies*. Berlin Heidelberg: Springer, 2001, pp. 46–66.

[9] "The Annotated Gnutella Protocol Specification v0.4." [Online]. Available: http://rfc-gnutella.sourceforge.net/developer/stable/

[10] D. Korzun and A. Gurtov, *Structured Peer-to-Peer Systems : Fundamentals of Hierarchical Organization, Routing, Scaling, and Security*. New York: Springer Science + Business Media, 2013.

[11] K. Dhara, Y. Guo, M. Kolberg, and X. Wu, "Overview of Structured Peer-to-Peer Overlay Algorithms," in *Handbook of Peer-to-Peer Networking*. US: Springer, 2010, pp. 223–256.

[12] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, New York, USA, 2001, pp. 161–172.

[13] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord : A Scalable Peer-to-peer Lookup Service for Internet," in *Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, New York, USA, 2001, pp. 149–160.

[14] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Middleware*. Berlin Heidelberg: Springer, 2001, vol. 2218, pp. 329–350.

[15] H. Shen, C. Z. Xu, and G. Chen, "Cycloid: A constant-degree and lookup-efficient P2P overlay network," *Performance Evaluation*, vol. 63, no. 3, pp. 195–216, 2006.

[16] A. Gupta, B. Liskov, and R. Rodrigues, "One Hop Lookups for Peer-to-Peer Overlays," in *Hot Topics in Operating Systems (HotOS)*, Lihue, HI, 2003.

[17] A. Gupta, B. Liskov, and R. Rodrgues, "Efficient Routing for Peer-to-Peer Overlays." in *the 1st Symposium on Networked Systems Design and Implementation (NSDI)*, California, 2004.

[18] B. Leong, B. Liskov, and E. D. Demaine, "EpiChord: Parallelizing the Chord lookup algorithm with reactive routing state management," *Computer Communications*, vol. 29, no. 9, pp. 1243–1259, May 2006.

[19] L. R. Monnerat and C. L. Amorim, "D1HT : A Distributed One Hop Hash Table," in *International Parallel and Distributed Processing Symposium (IPDPS)*, Rhodes Island, 2006.

[20] J. Li, J. Stribling, R. Morris, and M. Kaashoek, "Bandwidth-efficient management of DHT routing tables," in *The 2nd USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Boston, Massachusetts, USA, 2005, pp. 99–114.

[21] A. Brown, M. Kolberg, and J. Buford, "Chameleon: An adaptable 2-tier variable hop overlay," in *6th IEEE Consumer Communications and Networking Conference (CCNC)*, Las Vegas, NV, 2009, pp. 1–6.

[22] C. G. Plaxton, R. Rajaraman, and A. W. Richa, "Accessing nearby copies of replicated objects in a distributed environment," *Theory of Computing Systems*, vol. 32, no. 3, pp. 241–280, 1999.

[23] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," in *Peer-to-Peer Systems*. Berlin Heidelberg: Springer, 2002, vol. 2429 LNCS, pp. 53–65.

[24] F. Chowdhury and M. Kolberg, "Performance evaluation of structured Peer-to-Peer Overlays for Use on Mobile Networks," in *6th International conference on Developments in eSystems Engineering (DESE)*. Abu Dhabi: IEEE, 2013, pp. 57–62.

[25] J. Furness, F. Chowdhury, and M. Kolberg, "An Evaluation of EpiChord in OverSim," in *the Fifth International Conference on Networks & Communications (NetCom)*, ser. Lecture Notes in Electrical Engineering, vol. 284. Chennai, India: Springer International Publishing, 2014, pp. 3–19.

[26] F. Chowdhury and M. Kolberg, "Performance Evaluation of EpiChord under High Churn," in *the 8th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks (PM2HW2N)*, Barcelona, Spain, 2013, pp. 29–36.

[27] R. S. King, "Building a Subversive Grassroots Network: How Commotion Wireless plans to enable digital communication in the face of an Internet

shutdown," *IEEE Spectrum*, 2011. [Online]. Available: http://spectrum.ieee.org/telecom/internet/building-a-subversive-grassroots-network

[28] X. Hong, K. Xu, and M. Gerla, "Scalable routing protocols for mobile ad hoc networks," *IEEE Network*, vol. 16, no. 4, pp. 11–21, 2002.

[29] A. Boukerche, B. Turgut, N. Aydin, M. Z. Ahmad, L. Bölöni, and D. Turgut, "Routing protocols in ad hoc networks: A survey," *Computer Networks*, vol. 55, no. 13, pp. 3032–3080, 2011.

[30] E. Alotaibi and B. Mukherjee, "A survey on routing algorithms for wireless Ad-Hoc and mesh networks," *Computer Networks*, vol. 56, no. 2, pp. 940–965, 2012.

[31] R. Rajaraman, "Topology control and routing in ad hoc networks: A Survey," *ACM SIGACT News*, vol. 33, no. 2, pp. 60–73, 2002.

[32] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," *RFC 3626, doi:10.17487/RFC3626*, 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3626.txt

[33] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector (DSDV) for Mobile Computers," in *the ACM conference on Communications architectures, protocols and applications (SIGCOMM)*, London, UK, 1994, pp. 234–244.

[34] J. Garcia-Luna-Aceves and M. Spohn, "Source-tree routing in wireless networks," in *International Conference on Network Protocols*, Toronto, Ontario, Canada, 1999, pp. 273–282.

[35] S. Murthy and J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks," *Mobile Networks and Applications-Special issue: routing in mobile communications networks*, vol. 1, no. 2, pp. 183–197, 1996.

[36] A. Munaretto and M. Fonseca, "Routing and quality of service support for mobile ad hoc networks," *Computer Networks*, vol. 51, no. 11, pp. 3142–3156, 2007.

[37] C. Liu and J. Kaiser, "A Survey of Mobile Ad Hoc network Routing Protocols," Tech. Rep., 2003. [Online]. Available: http://www.minema.di.fc.ul.pt/reports/report_routing-protocols-survey-final.pdf

[38] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," *RFC:3561, doi:10.17487/RFC3561.*, 2003. [Online]. Available: https://www.ietf.org/rfc/rfc3561.txt

[39] D. Johnson, Y. Hu, and D. Maltz, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4," *RFC: 4728, doi:10.17487/RFC4728*, 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4728.txt

[40] D. Johnson, D. Maltz, and J. Broch, "DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks," in *Ad hoc networking*. Boston, MA, USA: Addison-Wesley Longman Publishing Co, 2001, pp. 139–172.

[41] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks," *Ad Hoc Networks*, vol. 2, no. 1, pp. 1–22, 2004.

[42] Z. Haas, "A new routing protocol for the reconfigurable wireless networks," in *IEEE 6th International Conference on Universal Personal Communications (ICUPC)*, vol. 2, San Diego, CA, 1997, pp. 562–566.

[43] M. R. Pearlman and Z. J. Haas, "Determining the Optimal Configuration for the Zone Routing Protocol," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1395–1414, 1999.

[44] N. Nikaein, C. Bonnet, and N. Nikaein, "Harp - Hybrid Ad Hoc Routing Protocol," in *International symposium on telecommunications (IST)*, TEHRAN, IRAN, 2001, pp. 56–67.

[45] M. Gerla, X. Hong, and G. Pei, "Landmark routing for large ad hoc wireless networks," in *IEEE Global Telecommunications Conference (Globecom)*, vol. 3, San Francisco, CA, 2000, pp. 1702–1706.

[46] N. Nikaein, H. Labiod, and C. Bonnet, "DDR-distributed dynamic routing algorithm for mobile ad hoc networks," in *First Annual Workshop on Mobile and Ad Hoc Networking and Computing (MobiHOC)*, Boston, MA, 2000, pp. 19–27.

[47] M. K. Marina and S. R. Das, "Ad hoc on-demand multipath distance vector routing," *Wireless Communications and Mobile Computing*, vol. 6, no. 7, pp. 969–988, 2006.

[48] R. Leung, J. Liu, E. Poon, A.-l. C. Chan, and B. Li, "MP-DSR : A QoS-aware Multi-path Dynamic Source Routing Protocol for Wireless Ad-Hoc Networks," in *26th Annual IEEE Conference on Local Computer Networks (LCN)*, Tampa, FL, 2001, pp. 132–141.

[49] J. Xie, R. R. Talpade, A. Mcauley, and M. Liu, "AMRoute : Ad Hoc Multicast Routing Protocol," *Mobile networks and Applications*, vol. 7, no. 6, pp. 429–439, 2002.

[50] J. Garcia-Luna-Aceves and E. Madruga, "A multicast routing protocol for ad-hoc networks," in *Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*. New York, NY: IEEE, 1999, pp. 784–792.

[51] S.-j. Lee, W. Su, and M. Gerla, "On-Demand Multicast Routing Protocol in Multihop Wireless Mobile Networks," *Mobile Networks and Applications*, vol. 7, no. 6, pp. 441–453, 2002.

[52] J. Biswas, M. Barai, and S. Nandy, "Efficient hybrid multicast routing protocol for ad-hoc wireless networks," in *29th Annual IEEE International Conference on Local Computer Networks (LCN)*. Tampa, Florida, U.S.A.: IEEE, 2004, pp. 180–187.

[53] M. Witt and V. Turau, "BGR: blind geographic routing for sensor networks," in *Third International Workshop on Intelligent Solutions in Embedded Systems*. Hamburg, Germany: IEEE, 2005, pp. 51–61.

[54] J. Boleng and T. Camp, "Adaptive location aided mobile ad hoc network routing," in *IEEE International Performance, Computing, and Communications Conference (IPCCC)*. Phoenix, Arizona: IEEE, 2004, pp. 423–432.

[55] Y.-B. Ko and N. Vaidya, "Geocasting in mobile ad hoc networks: location-based multicast algorithms," in *IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*. New Orleans, LA: IEEE, 1999, pp. 101–110.

[56] W. Liao, Y. Tesng, K. Lo, and J. Sh, "GeoGRID: A Geocasting Protocol for Mobile Ad Hoc Networks Based on GRID," *Internet Technology*, vol. 1, no. 2, pp. 23–32, 2000.

[57] R. Roy, "Mobile Ad Hoc Networks," in *Handbook of mobile ad hoc networks for mobility models*. Boston, MA: Springer US, 2011, pp. 3–22.

[58] A. Hinds, M. Ngulube, S. Zhu, and H. Al-Aqrabi, "A Review of Routing Protocols for Mobile Ad-Hoc NETworks (MANET)," *International Journal of Information and Education Technology*, vol. 3, no. 1, pp. 1–5, 2013.

[59] A. Neumann, C. Aichele, M. Lindner, and S. Wunderlich, "Better approach to mobile ad-hoc networking (BATMAN)," *IETF draft*, 2008. [Online]. Available: https://tools.ietf.org/html/draft-wunderlich-openmesh-manet-routing-00

[60] "Better Approach To Mobile Ad hoc Networking (B.A.T.M.A.N.), Date accessed: 2016-02-02." [Online]. Available: https://www.open-mesh.org/projects/open-mesh/wiki

[61] C. Perkins, S. Ratliff, and J. Dowdell, "Dynamic MANET On-demand Routing Protocol (DYMO)," *Draft-ietf-manet-dymo*, 2005. [Online]. Available: https://tools.ietf.org/html/draft-ietf-manet-dymo-00

[62] M. Zhao, Y. Li, and W. Wang, "Modeling and Analytical Study of Link Properties in Multihop Wireless Networks," *IEEE Transactions on Communications*, vol. 60, no. 2, pp. 445–455, 2012.

[63] "The network Simulator, NS." [Online]. Available: Http://www.isi.edu/nsnam/ns/

[64] "riverbed (OPNET)." [Online]. Available: Http://www.riverbed.com/products/steelcentral/opnet.html?redirect=opnet

[65] "OMNET++ network simulator." [Online]. Available: http://omnetpp.org/

[66] "QualNet." [Online]. Available: Http://web.scalable-networks.com/content/qualnet

[67] M. Hawa, "An OPNET simulation model for peer-to-peer networks," *Simulation: Transactions of the Society for Modeling and Simulation International*, 2013.

[68] "The Overlay Framework (OverSim)." [Online]. Available: http://www.oversim.org/

[69] "INET Framework for OMNeT++." [Online]. Available: Https://inet.omnetpp.org/

[70] R. Sargent, "Verification and validation of simulation models," *Journal of Simulation*, vol. 7, no. 1, pp. 12–24, 2013.

[71] G. Jayakumar and G. Gopinath, "Performance Comparison of Two On-demand Routing Protocols for Ad-hoc Networks based on Random Way Point Mobility Model," *American Journal of Applied Sciences*, vol. 5, no. 6, pp. 659–664, Jun. 2008.

[72] S. Sharma, "P-OLSR: Position-based optimized link state routing for mobile ad hoc networks," in *IEEE 34th Conference on Local Computer Networks*. Zurich, Switzerland: IEEE, 2009, pp. 237–240.

[73] J. Furness, M. Kolberg, and M. Fayed, "An Evaluation of Chord and Pastry Models in OverSim," in *European Modelling Symposium*. Manchester: IEEE, 2013, pp. 509–513.

[74] M. C. Castro, A. J. Kassler, C.-F. Chiasserini, C. Casetti, and I. Korpeoglu, "Peer-to-Peer Overlay in Mobile Ad-hoc Networks," in *Handbook of Peer-to-Peer Networking*. US: Springer, 2010, pp. 1045–1080.

[75] N. Qadri and A. Liotta, "Peer-to-Peer over Mobile Ad-Hoc Networks," in *Next Generation Mobile Networks and Ubiquitous Computing*. New York: Information Science Reference, 2011, pp. 105–121.

[76] S. Lee, L. Quan, K. Lee, T. Cho, and J. Jang, "A Peer-to-Peer Search Scheme over Mobile Ad hoc Networks," in *International Scientific Practical Conference (ISPC)*, 2004.

[77] 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 (PerCom)*. Hawaii: IEEE, 2005, pp. 60–64.

[78] 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)*. Lake District, UK: IEEE, 2004, pp. 163–173.

[79] R. Hwang and C. Hoh, "Cross-layer design of P2P file sharing over mobile ad hoc networks," *Telecommunication Systems*, vol. 42, no. 1-2, pp. 47–61, 2009.

[80] N. Shah, D. Qian, and R. Wang, "MANET adaptive structured P2P overlay," *Peer-to-Peer Networking and Applications*, vol. 5, no. 2, pp. 143–160, 2012.

[81] S. Abid, M. Othman, and N. Shah, "3D P2P overlay over MANETs," *Computer Networks*, vol. 64, pp. 89–111, 2014.

[82] N. Shah and D. Qian, "An efficient structured P2P overlay over MANET," in *the Ninth ACM International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE)*. Indiana, USA: ACM Press, 2010, pp. 57–64.

[83] A. Yu and S. T. Vuong, "A DHT-based hierarchical overlay for Peer-to-Peer MMOGs over MANETs," in *7th International Wireless Communications and Mobile Computing Conference (IWCMC)*. Istanbul: IEEE, 2011, pp. 1475–1480.

[84] 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 (MobiShare)*. California, USA: ACM Press, 2006, pp. 48–53.

[85] G. P. Millar, E. A. Panaousis, and C. Politis, "ROBUST: Reliable overlay based utilisation of services and topology for emergency MANETs," in *Future Network & Mobile Summit*. Florence: IEEE, 2010, pp. 1–8.

[86] G. P. Millar, E. Panaousis, and C. Politis, "Distributed Hash Tables for Peer-to-Peer Mobile Ad-hoc Networks with Security Extensions," *Journal of Networks*, vol. 7, no. 2, pp. 288–299, 2012.

[87] M. Conti, G. Maselli, G. Turi, and S. Giordano, "Cross-layering in mobile ad hoc network design," *Computer, IEEE Computer Society*, vol. 37, no. 2, pp. 48–51, 2004.

[88] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz, "Handling Churn in a DHT," in *USENIX Annual Technical Conference*, Boston, MA, 2004, pp. 1–14.

[89] 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," *Computer Communications*, vol. 32, no. 13-14, pp. 1445–1459, 2009.

[90] H. Sözer, M. Tekkalmaz, and I. Korpeoglu, "A peer-to-peer file search and download protocol for wireless ad-hoc networks," *Computer Communications*, vol. 32, no. 1, pp. 41–50, 2009.

[91] J.-C. Liang, J.-C. Chen, and T. Zhang, "An adaptive low-overhead resource discovery protocol for mobile ad-hoc networks," *Wireless Networks*, vol. 17, no. 2, pp. 437–452, 2010.

[92] M. C. Castro, E. Villanueva, I. Ruiz, S. Sargento, and A. J. Kassler, "Performance Evaluation of Structured P2P over Wireless Multi-hop Networks," in *Second International Conference on Sensor Technologies and Applications (SENSORCOMM)*. Cap Esterel: IEEE, 2008, pp. 796–801.

[93] M. Jing-qing, J. Hong, and L. Ting, "Cross-layer optimized Chord protocol for separated ring convergence in MANET," *The Journal of China Universities of Posts and Telecommunications*, vol. 16, no. 4, pp. 84–90, 2009.

[94] M. Li, W.-C. Lee, and A. Sivasubramaniam, "Efficient peer-to-peer information sharing over mobile ad hoc networks," in *the Second Workshop on Emerging Applications for Wireless and Mobile Access (MobEAII)*, 2004.

[95] S. Burresi, C. Canali, M. E. Renda, and P. Santi, "M ESH C HORD : A Location-Aware , Cross-Layer Specialization of Chord for Wireless Mesh Networks," in *Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom)*. Hong Kong: IEEE, 2008, pp. 206–212.

[96] R. Kummer, P. Kropf, and P. Felber, "Distributed lookup in structured peer-to-peer ad-hoc networks," in *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE*. Springer Berlin Heidelberg, 2006, pp. 1541–1554.

[97] S. G. Fantar and H. Youssef, "Locality-aware Chord over Mobile Ad Hoc Networks," in *Global Information Infrastructure Symposium (GIIS)*. Hammemet: IEEE, 2009, pp. 1–6.

[98] G. Peng, S. Li, H. Jin, and T. Ma, "M-CAN: a lookup protocol for mobile peer-to-peer environment," in *7th International Symposium on Parallel Architectures, Algorithms and Networks*. Hong Kong: IEEE, 2004, pp. 544–549.

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

[100] M. Thaalbi, A. Meddahi, T. Bejaoui, and N. Tabbane, "An enhanced chord-based P2P lookup protocol for mobile Ad hoc networks," in *Wireless Days (WD)*. Niagara Falls, ON: IEEE, 2011, pp. 1–5.

[101] M. Thaalbi, N. Tabbane, T. Bejaoui, and A. Meddahi, "Enhanced Backtracking Chord protocol for mobile Ad hoc networks," in *2nd International Conference on Communications and Information Technology (ICCIT)*. Hammamet: IEEE, 2012, pp. 191–195.

[102] C. Cramer and T. Fuhrmann, "Proximity Neighbor Selection for a DHT in Wireless Multi-Hop Networks," in *Fifth IEEE International Conference on Peer-to-Peer Computing*. Konstanz, Germany: IEEE, 2005, pp. 3–10.

[103] M. Shin and W. A. Arbaugh, "Efficient peer-to-peer lookup in multi-hop wireless networks," *Transactions on Internet and Information Systems*, vol. 3, no. 1, pp. 5–25, 2009.

[104] M. Caesar, M. Castro, E. Nightingale, G. O'Shea, and A. Rowstron, "Virtual ring routing: network routing inspired by DHTs," in *Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*. Pisa, Italy: ACM, 2006, pp. 351–362.

[105] T. Fuhrmann, P. Di, K. Kutzner, and C. Cramer, "Pushing chord into the underlay: Scalable routing for hybrid manets, Universität Karlsruhe (TH), Fakultät für Informatik," Tech. Rep., 2006.

[106] Y. Hu, S. Das, and H. Pucha, "Exploiting the Synergy between Peer-to-Peer and Mobile Ad Hoc Networks." in *Ninth Workshop on Hot Topics in Operating Systems (HotOS)*. Lihue, Hawaii: USENIX, 2003, pp. 37–42.

[107] T. Zahn and J. Schiller, "MADPastry: A DHT substrate for practicably sized MANETs," in *5th Workshop on Applications and Services in Wireless Networks (ASWN)*, 2005.

[108] Q. Meng and H. Ji, "MA-Chord: A New Approach for Mobile Ad Hoc Network with DHT Based Unicast Scheme," in *International Conference on Wireless Communications, Networking and Mobile Computing (WiCom)*. Shanghai: IEEE, 2007, pp. 1533–1536.

[109] A. C. Viana, M. D. de Amorim, and S. Fdida, "An Underlay Strategy for Indirect Routing," *Wireless Networks*, vol. 10, no. 6, pp. 747–758, 2004.

[110] J. Eriksson, M. Faloutsos, and S. V. Krishnamurthy, "DART: Dynamic Address RouTing for Scalable Ad Hoc and Mesh Networks," *IEEE/ACM Transactions on Networking*, vol. 15, no. 1, pp. 119–132, 2007.

[111] M. Caleffi, G. Ferraiuolo, and L. Paura, "Augmented tree-based routing protocol for scalable ad hoc networks," in *IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems (MASS)*. Pisa, Italy: IEEE, 2007, pp. 1–6.

[112] M. Caleffi and L. Paura, "P2P over MANET: Indirect tree-based routing," in *IEEE International Conference on Pervasive Computing and Communications (PerCom)*. Galveston, TX: IEEE, 2009, pp. 1–5.

[113] B. Zhao, Y. Wen, and H. Zhao, "KDSR: An Efficient DHT-Based Routing Protocol for Mobile Ad Hoc Networks," in *Ninth International Conference on Hybrid Intelligent Systems (HIS)*. Shenyang: IEEE, 2009, pp. 245–249.

[114] J. I. Alvarez-Hamelin, A. Viana, and M. D. Amorim, "DHT-based Funcionalities Using Hypercubes," in *IFIP 19th World Computer Congress, TC-6, IFIP Interactive Conference on Ad-Hoc Networking*. Springer US, 2006, vol. 212, pp. 157–176.

[115] J. J. Garcia-Luna-Aceves and D. Sampath, "Scalable integrated routing using prefix labels and distributed hash tables for MANETs," in *IEEE 6th*

International Conference on Mobile Adhoc and Sensor Systems (MASS). Macau: IEEE, 2009, pp. 188–198.

[116] S. a. Abid, M. Othman, and N. Shah, "Exploiting 3D structure for scalable routing in MANETs," *IEEE Communications Letters*, vol. 17, no. 11, pp. 2056–2059, 2013.

[117] S. Khan, A. Gani, and M. Sreekandath, "The routing performance of logarithmic-hop structured P2P overlay," in *IEEE Conference on Open Systems (ICOS)*. Langkawi: IEEE, 2011, pp. 202–207.

[118] "Specifications for the SECURE HASH STANDARD," 2002. [Online]. Available: http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf

[119] B. Fu, Y. Xiao, H. Deng, and H. Zeng, "A survey of cross-layer designs in wireless networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 110–126, 2014.