A Hierarchical P2P Overlay for Hierarchical Mobile Ad hoc Networks (MANETs)

by

Walid Abdel Gelil

A thesis submitted to the Faculty of Graduate and Postodcotoral Affairs in partial fulfillment of the requirements for the degree of

Master of Applied Science (MASc.)

in

Electrical and Computer Engineering

Carleton University

To all the *Champions* of tolerance, fairness, justice, integrity, mercy, inclusion, openness, illumination and innovation.

To all the *Victims* and *Survivors* of hatred, fanaticism, tyranny, greed, corruption and discrimination.

We pray for you. We stand with you. *YOU ARE NOT ALONE* . . .

Acknowledgements

Thanks and praise to God, for forgiving my mistakes, for listening to my prayers, for getting me where I stand today, and for taking me to where his kindness, grace and wisdom will guide me.

Thanks to my Supervisor Prof. T. Kunz for giving me the chance, for his trust in my capabilities, for his support and guidance throughout the process. Also, Prof. B. Esfandiari who was involved in the project in its early stages, for stimulating the challenge, raising the bar, and providing valuable insights.

Thanks to my colleagues and former partners in the lab. and project, A. Cormier and the dear friend S. Echegini whose work, support and discussions since the early stages and later provided great mileage to complete my work.

I want also to acknowledge a great initiative from J. Rocamora, a fellow researcher, whose work and information she generously and openly provided were a true milestone to progress in my work.

For my "references" in the application process, Prof. Ashraf Elazouni, a friend who really encouraged me to take this step. Prof. Nayer Wanas, one day I was his mentor, we turned to be close friends, then I called him to be my reference. I am so proud of you . . . !

Finally, I want to acknowledge my dear family. They always provided support and affection (my wife), they were always the motivation and inspiration for many milestones in my life (my son Yahya), and for being such a joy and blessing (my son Youssef).

Thank you all . . .

Abstract

P2P applications deployment on MANETs is motivated by the popularity of these applications, coupled with the widespread use of mobile devices. P2P applications and MANETs have common features such as decentralization, and the absence of dedicated servers or infrastructure. The deployment often faces specific performance challenges resulting from topological overlay and underlay mismatch, limited bandwidth constraint and dynamic topology changes. Hierarchical MANETs are a special type of MANETs where some nodes have specific routing roles to allow inter- cluster communications. Such topologies (typical for tactical networks) render a successful P2P deployment more challenging. We developed a novel approach for P2P deployment in such networks by bringing topology-awareness into the overlay, and building a hierarchically-structured logical overlay on top of the hierarchical underlay. Simulation results demonstrated a significant performance advantage of our solution vs. a flat logical overlay using different configurations and mobility scenarios.

Contents

List of Figures **13**

List of Tables **15**

Nomenclature **17**

1 Introduction **1**

1.1 P2P Applications 1

1.2 Routing in MANET and MANET Configurations 3

1.3 Motivation and Challenges 6

1.4 Contribution 8

2 P2P Deployment on MANETs **11**

2.1 Challenges 11

2.2 P2P Deployment Classification 13

2.3 P2P Deployment Review 14

2.3.1 Legacy (layered) Approach 14

2.3.2 Cross Layer P2P Deployment Approach 15

2.3.3 Integration P2P Deployment Approach 19

2.4 Hierarchical Overlays 20

3 Proposed Hierarchical Overlay **25**

3.1 Scope 25

3.2 Proposal 26

3.2.1 Design Requirements 27

3.2.2 Overlay Design 28

3.3 Routing Messages in the Overlay 32

3.4 Cluster Head Joining/ RLM Assignment 35

3.5 Node Joining 38

3.6 Mobility Management 39

3.7 Put/Get Application Data Routing and Exchange 41

3.8 Proposal: Pros and Cons 43

4 Simulation **47**

4.1 Simulation Environment 47

4.2 Configurations 48

4.2.1 Nodes 48

4.2.2 Cluster Heads (CHs)/ Gateways 49

4.2.3 Overlay 51

4.2.4 Flat Overlay 54

4.3 Simulation Parameters Summary 56

4.3.1 Single-Hop Static Simulation Scenario 59

4.3.2 Multi-Hop Static Simulation Scenario 59

4.3.3 Mobile Simulation Scenario 60

4.4 Metrics 63

5 Simulation Results and Analysis **65**

5.1 Performance Evaluation: Single-Hop Scenarios 65

5.1.1 Success Ratio 67
5.1.2 Latency 67
5.1.3 Efficiency 70
5.1.4 Additional Metrics and Further Analysis 72
5.2 Performance Evaluation: Multi-Hop Scenarios 77
5.2.1 Multi-Hop Scenarios Performance Metrics 79
5.3 Mobility Scenarios 81

6 Conclusions **87**
6.1 Thesis Contributions 87
6.2 Future Work 88

Bibliography **91**

List of Figures

1.1 MANET Configurations 4

3.1 Physical and Corresponding Overlay Network 31

3.2 Inter-Cluster Message Route 33

3.3 Top Overlay 35

3.4 Beacon Broadcast 36

3.5 CH Joining Sequence 37

3.6 Node Joining Sequence 39

4.1 Node Structure 49

4.2 Cluster Head Structure 51

4.3 Simulation: Superpeer Hierarchical Overlay Network 52

4.4 Simulation: Hierarchical and Flat Overlays 57

4.5 Simulation: PUT/ GET Mechanism 62

5.1 Success Ratio Results with Different Test Intervals 66

5.2 Latency Results with Different Test Intervals 68

5.3 Traffic Overhead. Single-Hop, No Data 71

5.4 Number of collisions. Single-Hop, No Data 72

5.5 Number of Collisions with Data Exchange 74

5.6 Total Received Packets with Data Exchange 75

5.7 Multi-Hop: Received Packets/ Collisions 78
5.8 Multi-Hop: Success Ratio 80
5.9 Multi-Hop: Latency 82
5.10 Mobility: Success Ratio 84
5.11 Mobility: Latency 86

List of Tables

3.1 LID Format 30
3.2 LID Example 32
3.3 HID Format 33
3.4 Example: Remote DHT table 34

4.1 Simulation Environment Versions 48
4.2 Hierarchical Overlay Configurations 54
4.3 Simulation Parameters 58
4.4 Mobile Scenario Simulation Parameters 60

Nomenclature

Acronyms / Abbreviations

AODV Ad Hoc On-Demand Distance Vector

CCN Content Centric Networking

CGSR Cluster-head Gateway Switch Routing

DHT Distributed Hash Table

DSDV Destination-Sequenced Distance-Vector Routing Protocol

DSR Dynamic Source Routing Protocol

GSR Global State Routing

HA Home Agent

HCS Hierarchical Cluster-based forwarding Scheme for mobile social networks

HID Hierarchical Identifier

HNA Host and Network Association

HSR Hierarchical State Routing

LID Logical Identifier

MAC Medium Access Control

MANET Mobile Ad Hoc NETwork

MANIP MANET Information Plane

MANKOP MANET Knowledge Plane

MMOG Massively Multi-player Online Games

OLSR Optimized Link State Routing Protocol

P2P Peer-to-Peer

PDR Packet Delivery Ratio

RLM Random LandMarking

RPCs Remote Procedure Calls

RPGM Reference Point Group Mobility

RRGM Reference Region Group mobility

RSU Road Side Unit

rUID Resource Unique Identifier

RWP Random Waypoint

TCP Transmission control Protocol

TTL Time To Live

UDP User Datagram Protocol

UID Unique Identifier

$ULID$ Unique Logical Identifier

$VANET$ Vehicular Ad-Hoc Network

VRR Virtual Ring Routing

WSN Wireless Sensor Network

Chapter 1

Introduction

1.1 P2P Applications

P2P applications have gained increased popularity as they allow the exchange of information and resources among network users without the need for a dedicated infrastructure (servers). P2P-based computing allows for better utilization of the network resources by distributing the tasks among peers (nodes). This approach also increases the overall network resilience and fault tolerance by removing dependability on the servers' reliability and availability [1]. In P2P applications, nodes are connected with a logical overlay network and respond to queries or requests from other nodes to assign or request resources (data streaming, file sharing, ... *etc.*) through lookup services [1, 2]. The P2P overlay network may organize peers (nodes) in a random graph, in this case the network is classified as an "unstructured P2P network". These types of P2P application use random walks, flooding or ring Time-To-Live (TTL) to search or query for contents among peers [3]. A well-know implementation of unstructured networks is Gnutella [4]. In structured P2P networks, the topology is controlled to render search and queries between peers more efficient. Structured P2P networks mostly utilize Distributed Hash Tables (DHT) as a substrate [3] to implement the lookup services in the network. By using a DHT, a key-value pair (<*key, value*>) is

assigned to each node and resource in the overlay network [2, 5]. The organization of peers in the overlay network may differ depending on the protocol used. Chord [6] assigns each node a Logical ID (LID) by hashing the UID (Unique IDentifier) value of the node (its IP address, for example). Similarly, it calculates a key *k* for each content by hashing the content name, then it assigns a <*k, value*> pair to the node whose LID is equal to or the closest bigger than the value of the key. In Chord, nodes are arranged in a ring structure, and lookup queries are done by matching a key with a node LID. Another protocol, Pastry [7] is a hybrid protocol, the lookup for a key-value pair is performed either in a ring-like manner similar to Chord or a tree-like structure [8]. Other structured P2P protocols such as CAN [9] arrange nodes around a virtual multidimensional coordinate space, while Tapestry [10] proposes a tree-based P2P overlay. Whereas structured protocols organize nodes in the overlay network with different topological structures (ring, tree, grid, *etc.*), their operations, at a high level, can be expressed as store and retrieval operations on a DHT.

P2P applications were originally developed for stationary units and fixed networks, however, the widespread use of mobile devices resulted in the need to deploy P2P applications in wireless environments such as Mobile Ad Hoc Networks (MANETs) [2, 5]. MANETs are composed of autonomous mobile wireless units (nodes) that can use wireless communication technology to communicate with each other without the need for a specific infrastructure. Each participating node in a MANET acts as a host (sending and receiving data packets to and from other nodes) and also as a router (forwarding data packets from source to destination), hence providing low-cost, instant and flexible communication between groups of people that may not be located within a single hop (or direct transmission range) from each other [8, 11, 12]. MANETs have several application scenarios ranging from conference or campus communications involving a relatively small number of nodes, to emergency (disaster) situation or tactical units in a battle field, where hundreds or thousands of units may be involved [8, 11, 13].

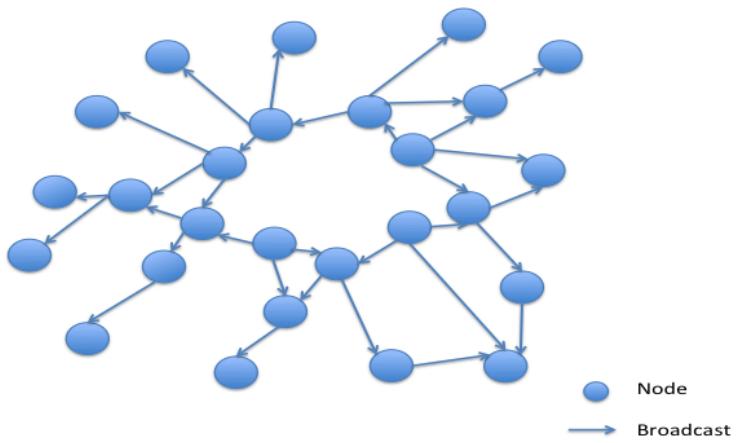
The deployment of a P2P overlay over MANETs is potentially complementary due to the presence of some common features between P2P applications and MANETs. Both P2P overlay networks and MANETs are self-organized, decentralized and are characterized by a dynamic and changing topology as nodes may join or leave (also called churn in the context of P2P applications) the overlay network. At the same time, nodes in MANETs may move away or join MANETs due to nodes' mobility [2, 14, 15]. In that sense, P2P addresses the issue of data (content) storage and retrieval in a MANET environment with its built-in routing and storage approach. Besides, P2P protocols with their sophisticated routing algorithms complement MANETs' routing capability especially in environments which are characterized by high mobility [11].

1.2 Routing in MANET and MANET Configurations

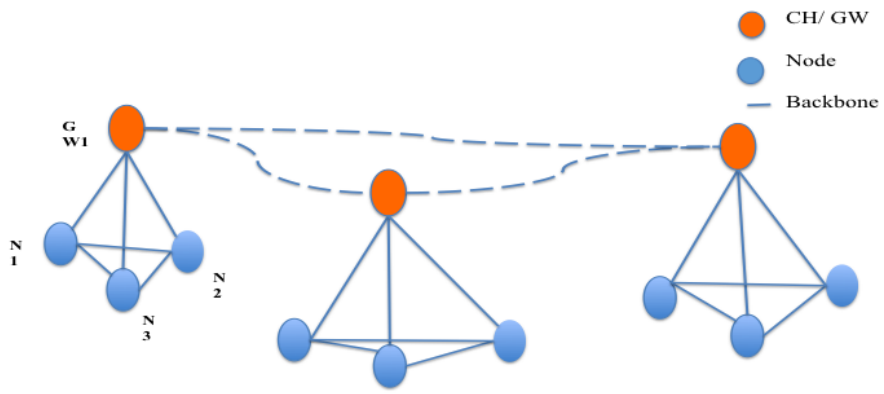
Routing protocols perform the critical task of of transferring packets between source and destination nodes in MANETs. The effectiveness of these protocols in delivering data and topology maintenance packets to the right destination in a timely manner, and the efficiency of the routing protocol in performing its critical task with minimum traffic overhead, reducing the size of the routing while maintaining the total network connectivity is key for the overall MANET performance [8].

Routing Protocols can be classified as Flat or Hierarchical based on the topology of the underlay (by underlay we mean the network topology at the network, MAC and physical layers), and the roles assigned to the nodes in the routing process accordingly. In flat routing protocols, no special tasks are assigned to specific nodes in the routing mechanism. Packets can be disseminated between nodes across the total network, nodes in the identified route will forward packets in a hop-by-hop manner from source to destination [8, 16]. Figure 1.1a shows an example of a MANET with flat underlay configuration and broadcasting messages in a flooding mode (no specific routing protocol).

Figure 1.1 MANETs with Different Underlay Configurations



(a) Flat MANET in a Flooding Broadcast Mode



(b) Hierarchical MANET with 2 Layers Underlay

In a hierarchical underlay topology, nodes may be grouped in local networks or clusters. In such a topology, nodes may be classified based on their roles and responsibilities in the routing process. Cluster Heads *(CHs)* are responsible for routing the information between different clusters, Gateways *(GWs)* provide the actual link and access to nodes in separate clusters, other nodes are simply part of a cluster or local network, performing regular routing tasks within the cluster boundaries. In some hierarchical configurations, CHs play the role of GWs and connect directly to other CHs through a backbone network [3, 8, 17, 18]. Figure 1.1b shows an example of a MANET with hierarchical underlay. In this Figure, CHs play the role of GWs and all are interconnected through a backbone network. All the nodes in the clusters are in single-hop distance from the CH/GW.

In order to be able to perform its task, a routing protocol needs to maintain updated routing information between nodes relative to the network dynamics. The network maintenance and routing updates should be done with minimum overhead to minimize bandwidth utilization and excessive processing. In a flat topology, using a reactive protocol (such as AODV [19], DSR [20], ...), the route discovery between source and destination is initiated and established on an as-needed basis by the source node [21]. This may impact the latency due to the route discovery process, yet the overhead for route discovery and control is minimized. As the networks grow in size and the message exchange rate increases, the overhead and bandwidth utilization due to routing increases, which degrades the network performance. Proactive protocols (such as OLSR [22], DSDV [23], ...) dynamically maintain up-to-date routing information to all other nodes in the network regardless of whether there are packets to be sent [18]. This mechanism significantly reduces latency, however the constant update and exchange of routing information through a flooding mechanism and the *potentially unnecessary* route discovery results in a significant traffic overhead, and consumes a non-trivial portion of the bandwidth. As the network size grows, this will result in increased

traffic for topology update, the performance of proactive protocols degrades, which translates into a poor network performance [8, 11, 24].

1.3 Motivation and Challenges

A flat network organization may be effective for small to medium size networks with moderate level of data exchange. However, as MANET applications may involve a large number of nodes and extensive data exchange, the issue of network scalability needs to be addressed. Historically, the same issue of scalability appeared as the Internet evolved from a limited number of nodes exchanging data packets through routers which used and updated paths to all destination addresses . As the Internet size grew exponentially year after year, involving large number of nodes interconnected in local networks, routing adapted a hierarchical model which split the routing protocol between Interior Gateway Protocols (IGP) used inside the local network (or more accurately, within an Autonomous System or AS, potentially a collection of local networks under a single administrative authority), and an External Gateway Protocol (EGP) used to link different local networks or Autonomous Systems in the Internet [25].

A hierarchical structure provides a solution to address network performance degradation due to scalability of MANETs, which is critical for applications involving a large number of nodes. The main concept which comes with the hierarchical design is the *localization* of the control and topology update messages exchange within the cluster boundary, hence significantly limiting the flooding of control messages and associated traffic across the network and reducing redundant traffic [8, 17]. In addition to addressing the scalability issue in MANETs, a hierarchical configuration may be a typical design for MANETs used in specific applications and environment such as tactical networks, or emergency and disaster management teams [13, 26]. An example of such tactical network is the case of a military operation where dismounted soldiers are grouped into different formations

(platoons, battalions, ...*etc.*). In addition, there are mobile tank units or trucks as part of the same deployment. Such units typically have advanced communication capabilities (more powerful transmitters, additional radio links, etc.). The communication within the same logical formation may take place among soldiers through relatively small bandwidth and limited range channels. In addition, soldiers may connect to the mobile units (tanks or trucks) which have a larger transmission range and are able to connect with each other and other units (airborne units, command and control/ intelligence, ...*etc.*) [27]. The resulting MANET structure is a hierarchical structure, where these logical formations form clusters, and these clusters are able to interconnect and establish inter-cluster communication through a more powerful channel. The ability to efficiently and successfully deploy a P2P application over such a hierarchical topology to allow members (nodes) in these networks to use P2P applications services (media files sharing, voice communication, ...*etc.*) is critical for the success of such operations.

The deployment of P2P applications on a hierarchical topology comes with many challenges, mainly as the nodes in physically separated clusters may be logically adjacent in the overlay. This will result in extended routes (number of hops between nodes), hence, extensive associated traffic generated by the maintenance and lookup messages between nodes in the overlay, the problem is known as the "route stretch" or "route mismatch" between logical and physical routes [14]. The extensive traffic during message exchanges, due to route mismatch, will eventually result in a considerable degradation of service in terms of resilience and ability of the overlay network to perform lookup tasks completely and correctly, also higher latency and slower response to queries and lookup requests [11, 14, 16].

Several P2P deployment solutions on MANETs, mostly using a cross layer approach, addressed the route stretch problem through the use of the link state and routing tables information in the underlay to build the logical relation between nodes in the overlay. The main concept was to maintain a neighboring logical relation in the overlay between physically

adjacent nodes [2, 8]. Whereas this approach was successful in reducing the impact of the route stretch, however the main limitation in these solutions was the assumption of the full visibility of the nodes in the entire network. This assumption could be valid for a flat underlay topology, however, for a hierarchical underlay structure, the node visibility is limited to only nodes located in the same cluster. In general, there was no specific solution to deploy P2P applications on Hierarchical MANETs which will not only mitigate the limitations in the existing solutions, yet further exploit and leverage any opportunity which comes with such hierarchical structure of the physical underlay network.

1.4 Contribution

In this work, we will present a novel approach for P2P deployment on hierarchical MANETs. The main design principle in our proposal is to maintain physically adjacent nodes in the underlay to be neighboring peers in the overlay, and to maintain an identical hierarchical topology and cluster formation for both the logical overlay and the physical underlay network structure. To the best of our knowledge, this is the first attempt to bring topology awareness for a hierarchical MANET to build a hierarchical logical overlay topology. Also, to the best of our knowledge, this is the only attempt we are aware of, that was successful in building a simulation model for a hierarchical logical overlay over hierarchical MANET, using hierarchical modules with complete layered design, and be able to generate concrete results with strong performance metrics achieved.

The rest of the document is organized as follows: In Chapter 2 we will share some related work on the approaches for P2P implementation, several implementation examples and protocols with some evaluation on the benefits, constraints, and the gap in these implementations that our solution aims to address. In Chapter 3, we will share the concept, and main design elements of our proposed solution for hierarchical P2P overlay deployment on hierarchical MANETs. In Chapter 4, we sill share the simulation setup, the implementation approach on

the simulation tool, the simulation scenarios and parameters. In Chapter 5, we will share the simulation results and analysis for all simulation scenarios. We will conclude in Chapter 6, then share some ideas and proposals for future work.

Chapter 2

P2P Deployment on MANETs

Despite the synergy resulting from the deployment of P2P over MANETs due to their common features and characteristics (distributed structure, decentralization, ... *etc.*), there are also many challenges for a successful P2P overlay deployment on MANETs. P2P protocols were originally designed for fixed networks and stationary nodes. MANETs have their own limitations relative to the capability of the nodes and the nature of the environment, beside the challenge of handling a highly dynamic environment and constant topology changes. A successful P2P deployment on MANETs implies certain modifications to existing protocols to adapt to MANETs. In the following we will summarize the main challenges and classify different deployment approaches and P2P overlay structures for MANETs.

2.1 Challenges

- *Limited Bandwidth:* MANET bandwidth resources are limited. Standard P2P algorithms are wasteful in bandwidth due to peer updates, multiple hop queries and logical topology maintenance. These protocols need to be adapted in order to minimize the overhead and to utilize less bandwidth and network resources [8, 11, 12, 16, 28]

- *Mobility:* The nodes in MANETs are mobile which lead to physical topology changes that may impact the overlay. A fast and efficient mechanism for updating the overlay with such physical topology changes needs to be included when P2P overlay protocols are deployed on MANETs [2, 3, 11, 29].

- *Route Stretch:* Route stretch is the ratio between the logical and the optimal physical path. The logical path in the overlay is translated into a physical path in the underlay physical network. Peers which are logically connected in the overlay may need several hops in the physical network to transmit a data packet in a message exchange. The same may apply for nodes which are physically neighbors but logically separated in the overlay network. All this translates into additional traffic in the network, hence optimizing route stretch is a critical challenge in P2P deployment [3, 11, 16, 29–31].

- *Processing Capacity:* Pending on the nature of the nodes in the network, some nodes may have limited processing power and memory capacity. This may be a limitation for deploying complex P2P overlay algorithms that may overload these nodes and result in excessive processing delays. P2P protocols need to adapt to the limited processing capacity of the nodes in a MANET environment [32].

- *Battery Life:* Mobile nodes are battery operated, batteries have limited power and discharge (operation) lifecycle. It is critical for a successful P2P overlay deployment to minimize the number of messages and the processing (execution) time for P2P algorithms to conserve battery power and extend battery lifecycle [11, 32].

- *Infrastructure-less Design:* Some P2P protocols are built assuming a fixed underlay network structure and make use of some static features of the underlay network accordingly. These features do not exist in the infrastructure-less topology of MANETs, these protocols need to be adapted accordingly [11].

2.2 P2P Deployment Classification

P2P deployment on MANETs may be classified into different categories based on the criteria used for the classification. In our work, we will refer to two different classification criteria that will be briefly described in the following:

1. *Deployment Approach:* A **legacy approach** for deployment is simply applying a given P2P protocol at the application layer on top of the network layer. Both layers are independent and operate in isolation. Such an approach proved to be inefficient and results in poor performance due to redundancy and excessive traffic overhead. A **cross layer approach** for deployment is violating the standard layered design by allowing exchange of information between layers. The exchange may take place in different ways, including direct exchange of specific data between two or more non-neighboring layers, through posting data into a management board accessible by all layers, or by implementing a middleware layer where data is exchanged, processed and accessed by different layers. The interchange of data between layers allows the underlay to share routing and topology status with the P2P overlay. This will help reducing the maintenance overhead, hence lead to a better performance. The third approach is the **integration approach** , which is considered another deviation from the standard layered design. In this approach, we integrate the P2P algorithms directly with the network layer for P2P deployment. Integration eliminates redundant signaling between the two layers at the same host which leads to an improved performance [11, 14, 33].

2. *Overlay Hierarchy:* P2P can be deployed as a **flat overlay** structure where all peers have the same role in routing. The flat overlay can be further classified as structured or unstructured as described previously. P2P can also be deployed in a **multi-tier hierarchical structure**. The purpose of the multi-level hierarchy is to limit the exchange of information among different domains (clusters) where heterogeneous or identical DHT

overlay networks are deployed inside each of these domains. The disjoint domains are connected at each upper tier level(s) by one DHT overlay network, named the *backbone* or the *interconnection overlay*. The backbone network role is to allow the exchange of information between domains (clusters) by routing queries and supporting data retrieval between peers located in different clusters [34, 35]. We will limit our review to two-tier structured P2P deployment, some studies have shown that a 2-3 tier architecture is optimal in terms of network performance (i.e. lookup efficiency, latency and maintenance traffic) [3].

2.3 P2P Deployment Review

In our review we classify the deployment according to the integration approach adapted for the deployment of the P2P overlay on MANET [11].

2.3.1 Legacy (layered) Approach

The legacy design (deploying an overlay P2P having no interaction with physical underlay network) proved to be inefficient and leads to poor performance [11]. In [36] Castro *et al.* evaluated the performance of a DHT protocol called Bamboo DHT [37] over a wireless multi-hop mesh network. Bamboo DHT is designed to handle a high level of churn in P2P networks through the combined use of active probing and recursive routing to accurately distinguish down nodes in the overlay network, then activates a congestion aware recovery scheme to reduce stress over the network. Bamboo uses a proactive management traffic approach to maintain the network structure. The simulation setup implied different levels of management traffic periods to vary the level of management traffic load generated by the overlay protocol. The Bamboo DHT overlay was applied over a static MANET to isolate the impact of mobility, and the AODV-UU protocol [38] was used for routing in NS-2. The

simulation results show an increased percentage of management overhead with larger network size. Besides, the results indicate a degradation in success ratio (related to percentage of lookups that are correctly delivered in the overlay) for larger networks , which was attributed by the authors to the increase in management overhead.

2.3.2 Cross Layer P2P Deployment Approach

Mei *et al.* proposed Convergence Chord [39], which targets to converge separate Chord rings in a MANET environment. The protocol uses a cross layer approach by leveraging the information from routing protocols to detect physically neighboring nodes that belong to alien Chord rings. The protocol implements a convergence process through using a Chord Neighbor Table (CNT) which stores information for detected neighboring nodes. Using this information yields a significant improvement in lookup efficiency and network resilience by restoring logical connectivity between disjoint Chord domains. The protocol is only limited to a flat overlay hierarchy, and aims to address the issue of reconstructing a single Chord ring that was split into disjoint domains due to nodes mobility. The traditional Chord was not able to handle such case, which will result in permanent Chord ring breakdown into smaller disjoint rings in such case. Besides, the protocol relies on the availability of routing information for all participating nodes at the network layer to detect physical vicinity and restore logical unity in the overlay, which does not apply to a hierarchical underlay network topology.

In [40] Seddiki *et al.* propose an adaptive P2P overlay over MANET. The objective of the adaptive overlay implementation is to build an unstructured overlay that likely matches the MANET physical underlay and constantly adapts to changes in its topology. Such matching, which considers the physical proximity between nodes and maps it into the logical overlay network, significantly improves lookup efficiency, reduces route stretch and traffic redundancy. Message paths in the physical layer are used to determine the physical neighbors

for each peer. Each node keeps a list of logical overlay neighbors with associated IP address, state, and physical distance for each entry in the list. To establish overlay connections, each peer targets the physically nearest peers as overlay neighbors and applies a neighbor replacement strategy to adapt the overlay to underlay topology changes. The mapping approach in building the overly network seems an attractive proposal, however, it has also some limitations. The proposal adapts an unstructured P2P overlay to match the underlay structure which will impose a constraint on the scalability of the protocol, given the fact that unstructured protocols use flooding techniques and random walks for queries and data dissemination. For large scale networks, adapting this approach will result in considerable performance degradation.

Chen *et al.* propose SPOON, a P2P content-based file sharing system [41]. They propose a hierarchical structure where nodes are clustered in communities. The nodes in the same community share a common interest, the interest is defined by the file categories stored and queried by each node. SPOON classifies nodes with frequent contact and common interests as a community to allow for interest-based file searches. Nodes in the same community also frequently meet in a social network context. Nodes have different role assignments as normal peers, coordinators or ambassadors. Nodes use information from the routing layer to track and trace stability and mobility of each node. Coordinators are stable nodes within a community that keep a file index for files stored within the community, and also assign and keep track of community ambassadors. Ambassadors are nodes with high mobility, responsible for routing queries to other communities if the query cannot be satisfied within the community. Nodes with different interests can be part of different communities. The highly mobile nodes are assigned by coordinators for inter-community routing. The Interest-oriented file searching Routing Algorithm (IRA) is the component responsible for intra-community and inter-community searching. If the home community cannot satisfy a request, the coordinator launches the inter-community search. The coordinator forwards the request to an ambassador

that will route the request to the foreign community that matches the request's interest. In this protocol, the use of routing data is limited to draw a node "profile" and used in the context of social networks where the level of dynamics is moderate. Also, the clustering is based on "interest" which may not apply to other types of P2P applications.

Macedo *et al.* propose a cross layer deployment platform MANKOP (MANET KnOwledge Plane) [42]. MANKOP is a horizontal knowledge plane that stores and provides information related and relevant to all protocol layers. Protocols in different layers, such as P2P application protocols, routing protocols, etc. can use the information stored in the knowledge plane to optimize and improve their performance. The plane was used in a case study to build transmission-power-control-aware protocols and proved to improve throughput significantly. The same middleware concept was further developed with MANIP [43], an information plane that provided a single distributed information service allowing information sharing. The information plane, in addition to storing information needed for network protocols, replaced the information repositories of context-aware systems and autonomic management solutions. Hence, MANIP was used to support the autonomic reconfiguration of a P2P network over MANETs achieving reduced response times and a better service (number of resolved queries) as per the simulation results.

Abid *et al.* propose 3D P2P overlay over MANETs [30]. The DHT logical overlay is implemented in 3D space to reflect the relationship between physically neighboring peers using an LID (Logical IDentifier) generation algorithm which maps the physical proximity in the 3D space. The LID of each node is a tuple {x|y|z}, each tuple is calculated using local information of logical neighbors of the node, the information is obtained by exchanging probing messages. The information includes the neighbor's LID, UID (Unique Identifier), the list of neighboring peers and their distance (in hops) from the neighbor. The LID generation scheme assigns weights to the links expressed in number of hops between a peer and its physical neighbors, the link information is obtained through the underlying agent routing

table (*e.g.* OLSR). The use of a 3D space provides a higher level of resilience to the network as the nodes are interconnected through alternative routes. In addition, mapping the physical proximity in the 3D space reduces the overhead resulting from the route mismatch problem. Whereas, such benefits are applicable for a flat underlay topology, using the same protocol for a hierarchical overlay may in some cases totally reverse the impact. Organizing the peers in a 3D logical space and providing alternative logical routes between peers located in separate clusters will yield excessive route stretch, generating massive redundant overhead traffic for an eventual communication failure. This is resulting from the fact that, while the protocol provides a descent presentation for physical proximity of nodes into the logical space between peers in a flat underlay topology, it provides a misleading logical relation between peers physically located at different clusters in a hierarchical underlay configuration. This solution still falls short of addressing a hierarchical environment, as routes in the logical space are only provided to the nodes which are "visible" in the underlay, which will be limited in this case to neighboring nodes in the local cluster.

A recent and successful P2P protocol deployment on MANETs is OneHopOverlay4MANET [2]. The implementation uses a cross-layer approach to leverage the synergy between underlay routing protocol and overlay. The protocol improves performance by minimizing the logical hops in the overlay to a single hop. This is achieved by using a structured overlay protocol similar to Epichord [44] which is able to implement lookups in a single hop, coupled with leveraging the routing information from the cross-layer channel to reduce overlay control traffic and maintain network consistency. The solution is successful in reducing traffic overhead and improving network performance. However, this solution is limited to a flat underlay network architecture where routing information for all nodes is available to the overlay application through a cross layer information board.

2.3.3 Integration P2P Deployment Approach

P2P protocols are integrated and implemented at the network layer using the integration approach. In this approach, routing decisions are based on both logical and physical neighbor information [8]. An example is Ekta [45] which integrates Pastry [7] with DSR [20, 46] into one structure to exploit the synergies and improve routing performance.

Zahn *et al.* introduced MADPastry [47], a protocol which integrates Pastry and AODV at the network layer to improve scalability and reduces the traffic overhead caused by the maintenance of the DHT routing structures. MADPastry utilizes both AODV and Pastry routing tables and data structures. It introduces the concept of a cluster of nodes sharing the same prefix, and landmark keys which equally divide the logical space of a DHT. A common ID prefix is assigned to the nodes which are likely to be physically close. A node whose value is the closest to a landmark key is considered a landmark node and periodically announces its presence and distance to neighboring nodes. The nearest nodes to the landmark node receiving the beacon join the cluster by assigning themselves the same prefix as the landmark node. To minimize the maintenance overhead, only entries to cluster heads are maintained in the overlay routing table. To improve the lookup and message routing efficiency, the nodes follow the Pastry routing procedure if the receiving node is the target overlay hop node. Otherwise, a node will consult its AODV routing table for the next physical hop towards the overlay hop target. MADPastry depends on overhearing traffic packets to update its overlay and routing table entries. For a low traffic network, updates may not be frequent and network information in the routing table may not be updated in a timely manner, which makes this protocol more suited for high data traffic networks.

2.4 Hierarchical Overlays

In this section, we will focus on some relevant work in implementing hierarchical networks and hierarchical routing protocols. The work presented in this section is the main inspiration for our proposed solution. In [48] Pei *et al.* introduced the HSR (Hierarchical State Routing) routing protocol. The protocol presented one of the earliest and most efficient approaches to address the issue of scalability and mobility for large networks by using a hierarchical logical overlay. Mobility is a major challenge to any hierarchical structure due to the need to timely update the routing information inline with topology changes caused by nodes' mobility as an entity or as a group. The protocol adapted the notion of logical subnets (in our literature we use *cluster* as the notion for a logical subnet formation), where a group of nodes are bound in a logical hierarchy and share some group mobility behavior (e.g. brigade in a battlefield, rescue teams in a disaster management situation, ... *etc*). HSR uses clustering election schemes to elect CHs and to build a hierarchical structure where cluster heads are elected at each level of the hierarchy to form a higher layer of a hierarchical topology. The main rational behind the use of a cluster formation is to localize the routing control message exchange and related traffic overhead within a limited number of nodes inside the cluster rather than propagating these messages across the whole network. Each node in the hierarchy is assigned an HID (Hierarchical Identifier) which defines the path to the node from the top hierarchy to the node in the bottom layer. Packets can be routed from source to destination using the HID of both source and destination nodes. In addition, nodes in the network are assigned logical addresses with a < *subnet, host* > format. Each subnet component identifies a particular group of nodes with a logical bind of common characteristics (*e.g.* a battalion in a tactical network, an emergency response team in disaster management, ... *etc.*). Each node registers its own HID address with a local Home Agent *HA* which is a member of the subnet and manages group membership. The local agents' HIDs are periodically propagated across all layers to all nodes in the network. A node aiming to send a packet to a destination node

will use the list of HIDs to identify the HID responsible for the subnet where the destination node belongs. A source node, aware of the logical address of its destination, will be able to resolve the address locally or through extracting the corresponding destination HA HID. The HA HID will provide the source node with the HID of the destination to start the direct communication exchange accordingly. The main advantage of the use of logical subnets is the separation between logical grouping and physical mobility. A node will be able to use the same logical ID to forward packets while the updated HID will reflect the actual path and route to destination nodes. The approach has the advantage of properly resolving the changes in HID due to mobility combined with reducing the overhead through the physical and logical cluster formation. HSR is basically a routing solution which uses clustering and hierarchical topology to reduce the overhead traffic, improve scalability and support group mobility scenarios. In this context, it is not considered as a P2P deployment solution, rather it provides some interesting features and design characteristics which can be used for successful P2P deployment.

In [49], Yu *et al.* proposed a similar approach to provide a scalable, topology-aware, responsive, and portable routing protocol to deploy NVEs (Networked Virtual Environment) used for MMOGs (Massively Multi-player Online Games) on MANETs. Yu *et al.* proposed a hierarchical overlay DHT-based routing solution to deploy P2P applications for MMOGs on a MANET platform. The solution uses a logical hierarchical structure in the overlay similar to HSR [48]. They adapted the use of RLM (Random Landmarking) [50] used in MADPastry [47] to create clusters with the same prefix ID (landmark key) in the overlay. To maintain topology awareness, nodes which are physically close are assigned the same landmark key as prefix. Each node in the overlay maintains a remote DHT table which maps landmark keys to existing clusters, hence unicast routing can be used to route packets to destination nodes across clusters. Each cluster has a cluster head, elected cluster heads in each layer form a higher overlay layer similar to HSR [48]. To route packets between nodes,

the HID is used as the destination address. Clusters are organized internally as VRR (Virtual Ring Table) used to route packets internally within the cluster. The combination of topology awareness through applying the RLM concept, the responsiveness to topology changes (given that MMOGs are real time applications) using HIDs which are periodically updated across all overlay layers, together with the clustering formation to localize control updates positions the solution as an attractive approach for P2P applications deployment. However, the solution suggests recursive layering and clustering in the overlay until a top layer of single CH and peer is reached, in addition to the periodic update and propagation of the remote DHT tables to all nodes, which will add a significant additional traffic overhead. This traffic overhead will have a negative impact on bandwidth utilization, latency and the ratio of successfully satisfied queries, which are all critical measures for online applications successful deployment. Besides, in this work, the concept is not tested to prove functionality with different test conditions, and there was no demonstration of solid (in fact any) performance metrics results with different test scenarios.

In summary, we reviewed different approaches and related work to deploy P2P applications on MANETs. Whereas these approaches have their own limitations on the implementation (can only be applied to a flat underlay configuration), or imply extensive traffic overhead that may impact the network performance, we applied several strong concepts used in these solutions to develop an efficient and successful solution for P2P deployment on hierarchical MANETs. In our proposed solution, we will leverage some fundamental concepts we see critical for a successful design:

- *Topology awareness* by mapping the underlay topology to the overlay and having the same cluster formation. Hence, we will be able to localize the DHT table maintenance messages and topology control messages within local cluster traffic.

- *Physical and Logical proximity* by maintaining physically adjacent nodes in the underlay as logical neighboring peers in the overlay to reduce the route stretch.

In the following, we will share our proposal for a successful P2P deployment on an hierarchical underlay topology.

Chapter 3

Proposed Hierarchical Overlay

In this chapter, we describe a proposed routing solution for hierarchical MANETs. We start by describing the problem nature and boundaries which our proposed solution is addressing (what our solution is about, what it is not ...). We review the main design principles that we followed in order to develop a successful solution. Then, we describe the details of the solution proposed (structure, topology, mechanism) and emphasize on the main advantages and limitations of our proposal.

3.1 Scope

The solution aims to provide an efficient P2P deployment on hierarchical MANETs. The network hierarchical structure assumes some basic characteristics for the nodes and the topology which can be summarized as follows:

1. All nodes in the network are interconnected in local clusters (subset of nodes). A node can be connected to one cluster at a time.

2. The cluster formation follows a hierarchical ranking approach. The nodes in the same cluster have one superior node in rank, the Cluster Head *CH*, which is assigned

different roles in the network. One of these roles is to allow the nodes in its own cluster to communicate with nodes pertaining to other clusters in the network.

3. Each cluster has a gateway(s) *GW* (the CH may also play the role of the GW). A gateway has a main role of routing the packets between clusters for inter-cluster communication.

4. All gateways are connected through a backbone network layer which allows the gateways in different clusters to route control and data packets.

The above description of the network characteristics may apply to tactical networks which are hierarchical by design [13, 51], or other configurations of MANETs such as emergency and disaster management applications [52].

Our solution aims to provide a successful P2P deployment on hierarchical MANETs by adapting an efficient routing mechanism for MANETs having similar configuration characteristics as described above. Clustering schemes [53] such as HCS [54] or cluster head selection mechanisms [53] (e.g. [55] which describes a mechanism for cluster head selection and rotation in WSNs (Wireless Sensor Networks), or [56] which describes another approach to select cluster heads in CCN (Content Centric Networking) to maximize the information dissemination across nodes) are not part of the scope of our solution.

3.2 Proposal

Our solution is based on the implementation of a hierarchical P2P overlay structure over a hierarchical/clustered MANET underlay architecture. There are different design requirements and features in order for a solution to be scalable, efficient and responsive. Our proposal is mainly inspired, with adaptations and modifications, by the previous work done to implement a DHT overlay on MANET in [47] described above. It also draws on the work done by Pei et al. [48] to propose the hierarchical routing protocol HSR (Hierarchical

State Routing), where the notion of a logical subnet (e.g. a brigade in a battlefield, etc.) is introduced to handle mobility. In HSR, each node is defined by a node ID (in this case the MAC address), and HID (Hierarchical ID). The HID is generated by concatenating the MAC addresses of the nodes forming the path to the node from the top hierarchy in the overlay to the node in its cluster. Hence, with any topology changes, the HID of each node can be locally and dynamically updated upon receiving the routing updates. We used the concept of Random Landmark Key (RLM) to create a topology aware P2P overlay [50]. Finally, we followed a similar approach to the DHT-based hierarchical overlay for P2P described in [49] with some modifications to adapt with the hierarchical structure of the underlay network, and to meet the scalability, efficiency and responsiveness requirements for a successful design. In the following, we will review a successful DHT structure design requirements, the proposed solution, and demonstrate how our proposal will meet these requirements.

3.2.1 Design Requirements

We may summarize the design requirements [57] for a successful structured DHT P2P overlay implementation in the following points:

- ***Requirement 1***: Ability to map the underlay structure in a way that neighboring logical neighbors in the overlay are also physically adjacent in the underlay [8, 57].
- ***Requirement 2***: The ability of the overlay routing protocol to handle different mobility scenarios and sustain system performance with frequent topology changes [49].
- ***Requirement 3***: Minimize the traffic overhead resulting from DHT routing structure maintenance [8, 40, 47, 49, 57].

3.2.2 Overlay Design

We propose a hierarchical DHT structured routing overlay at the application level. We bring topology awareness to the overlay through mapping the underlay structure. We benefit from the availability of topology information at the application layer to reduce overhead traffic and bandwidth consumption. Clusters are mapped to the overlay in a way to match the underlay physical network. In that sense, all cluster heads in the physical network are part of the overlay network. Cluster heads continue to play the same role in the overlay i.e. the ability to route message through the top layer in the overlay to other clusters forming the overlay hierarchical structure. In the remainder of this document, we will jointly and frequently use the term *Superpeer* [11, 34] to refer to CHs in the overlay.

The overlay logical space is equally segmented into separate logical spaces, each identified by landmark keys [50]. All Superpeer(s) at the overlay will be assigned a logical ID with a prefix corresponding to a specific landmark key which identifies the beginning of its logical space section. All nodes in the same cluster will share the landmark key assigned to the Superpeer. Besides, each node will be assigned a different ULID (Unique Logical IDentifier) which forms the last part of its LID (Logical IDentifier). This approach will ensure that neighboring nodes in the overlay, which occupy the same logical space identified by the same RLM, are physically adjacent in the underlay (Requirement 1).

Another advantage of mapping the same hierarchical structure of the underlay to the overlay is the ability to localize the overlay DHT substrate maintenance messages in the overlay to the same cluster (intra-cluster message exchange) rather than propagating these type of messages across the whole network (Requirement 3).

The number of layers in the overlay is limited to 2 layers. The hierarchy in this case is static, the upper (top) layer is the backbone layer which connects all Superpeers (all Superpeers will be part of the backbone top layer). This choice is rooted in the design principle to bring topology awareness to the overlay by mapping the underlay hierarchical structure

to the overlay. Both the overlay and underlay hierarchy will have an identical structure of cluster formation with Superpeers (CHs / GWs) connected through a top (backbone) layer (network). The rational behind our choice for a 2-tier overlay hierarchical static structure is the fact that such an overlay structure properly reflects the underlay topology. Also, with limited clustering and layering schemes in the overlay, we aim to achieve an efficient network structure in the overlay in terms of performance metrics (lookup hopcount, latency, size of overhead traffic due to maintenance ... etc.) [3].

CHs use the link state information of neighboring nodes in its cluster to map the same cluster formation in the logical space. Each cluster in the underlay uses a proactive routing protocol to update link status information for nodes within the cluster. The use of a proactive routing protocol is motivated by the fact that routing and topology information is updated regularly and available to every node which is part of the same cluster. Hence, the use of the link state information does not imply any additional latency to gather and update routing information as would be the case for reactive routing [49]. Besides, the use of a proactive routing protocol spares the need to use and exchange signaling messages from the overlay to manage and maintain a topology aware overlay. In case of using a reactive routing protocol, the overlay would signal messages to handle specific topology control and updates. This can be the case of a new node joining the overlay to decide on the cluster formation the new node needs to join [2]. The network in the underlay can be of a heterogeneous nature (which is an advantage of using a hierarchical topology). In the same way, CHs use link state information for neighboring CHs in the underlay backbone network to form the top layer logical network connecting Superpeers in the overlay. The nodes located within the same cluster in the overlay are organized as a virtual ring in a peer-to-peer distributed hash table structure. There is no restriction on the protocol to be implemented, any standard protocol such as Chord [6], Pastry [7], ... *etc.* may be utilized. As an example, VRR (Virtual Ring Routing) [58] is the protocol used in [49], VRR is a network routing protocol which provides both point-to-point

Table 3.1 LID Format

LID Format	
RLM Prefix	**ULID** (node)

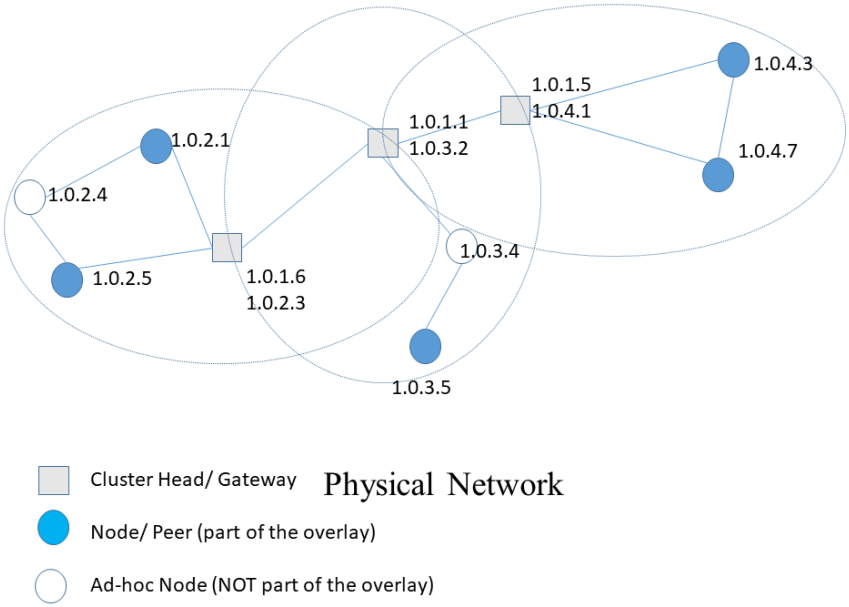
connectivity and DHT functionality by routing messages sent to a key to the node responsible for the key [58].

We differentiate between two Logical Space(s) *LS*. Each node in the overlay will have a ULID which can be generated by hashing a unique node identifier (IP, MAC address, etc). The ULID will prevent any LID duplication when a node joins different clusters in the overlay. The ULID will lie in logical space one (***LS1***). The ULID will be preceded by the RLM prefix which identifies the cluster containing the same node. The resulting node LID will lie within a new logical space ***LS2*** which is equally divided into sections, each identified by a RLM value.

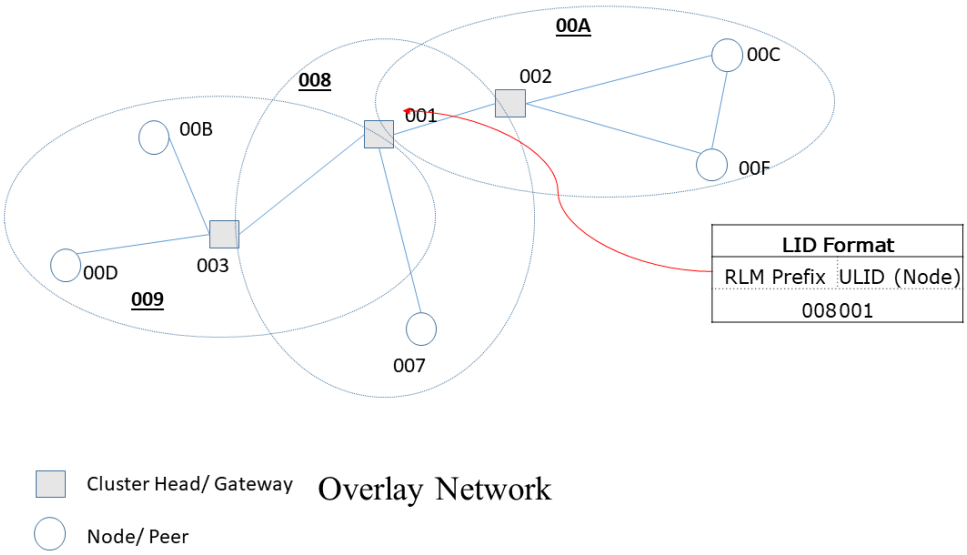
Example

We will consider the below physical network shown in Figure 3.1a. It shows the physical topology of the underlay with 3 clusters. In each cluster, the CH/GW has an IP interface address with the local cluster and the backbone network connecting all the CH(s). Figure 3.1a shows also other nodes in the underlay, each forming part of one of the local clusters.

The logical overlay formed over the hierarchical underlay is shown in Figure 3.1b. Each peer (we will use peers to identify nodes which joined the overlay) is assigned an LID. The LID is generated by the concatenation of the landmark key of the local cluster, and its ULID (can be the result of hashing the node IP address). In the example, the CH has an arbitrary landmark 008H value (for an hexadecimal address space), its unique ID is 001, hence it has the logical ID of 008001. Table 3.2 shows the LID of the CH for one of the clusters in the overlay, the same principle applies for other peers and CH(s) in the overlay.



(a) Physical Network Topology



(b) Overlay Topology showing sample HID

Figure 3.1 A Physical Network Sample and Corresponding Overlay Configuration

Table 3.2 LID Example

LID Format	
RLM Prefix	**ULID** (node)
008	001

3.3 Routing Messages in the Overlay

As indicated previously, the objective of the solution is to provide a routing mechanism which allows nodes physically located at different clusters in the underlay to communicate, share resources, and to exchange messages, which are the basic characteristics of P2P applications.

The intra-cluster communication in the overlay will be supported by the DHT protocol which manages the logical space assigned to the cluster. This may allow the use of heterogeneous protocols (i.e. Chord [6], PASTRY [7], CAN [59], ... *etc.*) in individual clusters. In that sense, the performance of the network for intra-cluster communication in the overlay will be reliant on the protocol efficiency and resilience metrics.

The inter-cluster message routing will follow a structured routing mechanism where the source node will forward the message to the Superpeer. The Superpeer will use the top overlay (backbone) to forward the message to the Superpeer responsible for the destination peer. The Superpeer in the destination cluster will forward the message internally to the destination node, using the local DHT overlay routing protocol. Figure 3.2 shows a potential route for a message where the source and destination nodes lie in different clusters.

At this stage, we will introduce the notion of another dimension in the logical space, the Hierarchical Identifier *HID*. The HID identifies the peer address in the logical space, which needs to include the same cluster Superpeer LID, which is part of the routing path to other peers located in remote clusters, then followed by its own peer LID. Table 3.3 shows the format of the HID in a 2-layer hierarchical structure.

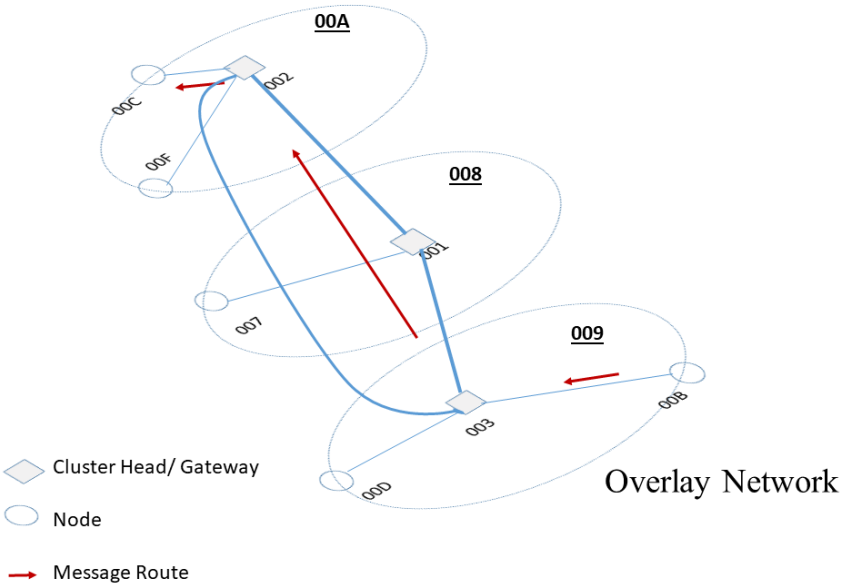


Figure 3.2 Example: Inter-Cluster Message Route

Table 3.3 HID Format

HID Format	
LID Superpeer	**LID** (node)

Table 3.4 Example: Remote DHT table

RLM	**HID** (node)
008	008001008007
009	009003009003
00A	00A00200A002

Each peer maintains a routing table (Remote DHT Table) consisting of the landmark keys of each cluster and the corresponding HIDs of an arbitrary node in that cluster (Table 3.3 shows the structure of a node HID, Table 3.4 shows an example of a remote DHT table). HIDs will be used as the destination addresses for overlay routed messages. A message routed to a destination peer which has a LID prefix (RLM) different from the source node will be routed to the local Superpeer, which will be selected as the next logical hop for any remote DHT table entry that points to destination nodes outside the local cluster. It is to be noted that for a 2-layer hierarchy, such as our solution, the Remote DHT can be kept at a simpler format with each entry showing the RLM and the corresponding CH LID. The use of the CH LID as the destination address for remote (inter-cluster) communication will be enough to route the messages from source and destination peers located in separate clusters. We have kept the remote HID table entries format (showing hierarchical path of nodes in successive overlay layers) to allow for the solution to expand in future work to accommodate more than a 2-layer hierarchy.

The top layer in the overlay is organized as a DHT structured layer, an example of a ring structure is given in Figure 3.3. The first part of the HID is the LID of the Superpeer responsible for the cluster where the destination node is located. The Superpeer at the source will use the routing protocol in the top overlay and the LID of the destination Superpeer to forward the message to the destination node Superpeer. The destination Superpeer will then use the LID of the destination node to route the message internally, within the cluster logical space, until the message reaches its destination.

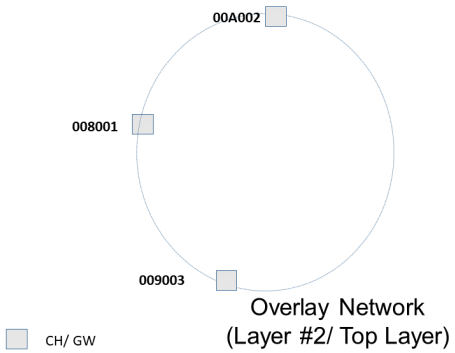


Figure 3.3 Example: Top Overlay Configuration with 3 Superpeers

3.4 Cluster Head Joining/ RLM Assignment

In this part, we review the mechanism for RLM assignment to a CH, bootstrapping the CH to the overlay hierarchy. Figure 3.5 summarizes the sequence of events.

a) The cluster head will propagate a JREQCH (Cluster Head Join REQuest). The JREQCH will be repeated with increasing backoff period. In case the CH does not get any response, within the assigned period, it will assume the logical space is empty and assigns itself an arbitrary RLM.

b) The closest node receiving the JREQCH will respond with a JRESPCH message. The message contains the node Remote DHT table (with all landmark keys already utilized).

c) The CH receives the JRESPCH message and, based on the contents of the DHT remote table received, it assigns itself a landmark key which is not already utilized.

d) The CH starts propagating its landmark key by sending a periodic beacon message. The beacon message will start by echoing the received remote DHT table updated with a new entry showing its own assigned landmark key and a blank field for related HID. A blank field in the DHT table will indicate the cluster is still in bootstrapping stage.

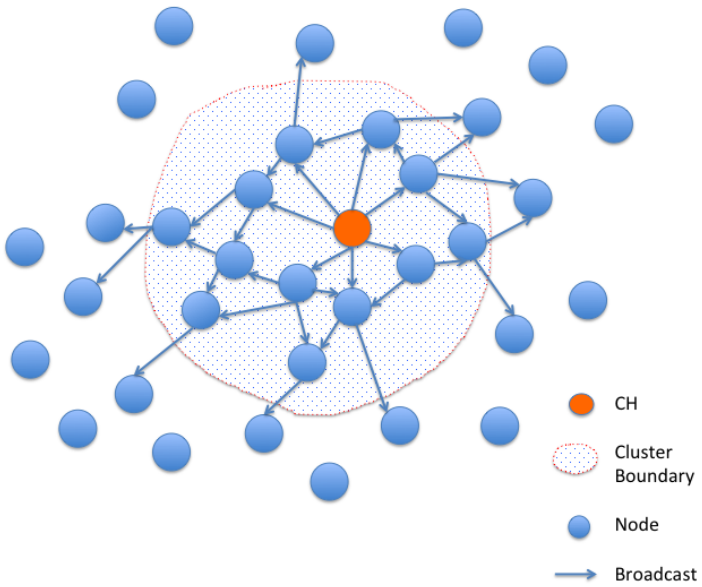


Figure 3.4 Beacon Broadcast in a Cluster Configuration

e) Each CH receiving the beacon message will examine the entries and the HID field in the message. If the HID field of the beacon message is blank, the cluster head will share the message with all cluster heads in the upper layer. Any other CH receiving the forwarded message with the same blank RLM entry but originated from a different CH will propagate an error message. The new entry is then blanked in all messages, and the gateway will respond to the new joining node with the blanked RLM field in the DHT table. In this case, the CH asking to join the overlay needs to assign itself a different RLM and restart the bootstrapping process. The intent of the process in this case is to prevent simultaneous duplicate RLM assignment if two or more clusters are joining at the same time.

f) Otherwise, the neighboring CH will respond back to the requesting CH with an updated remote DHT table adding the new RLM field.

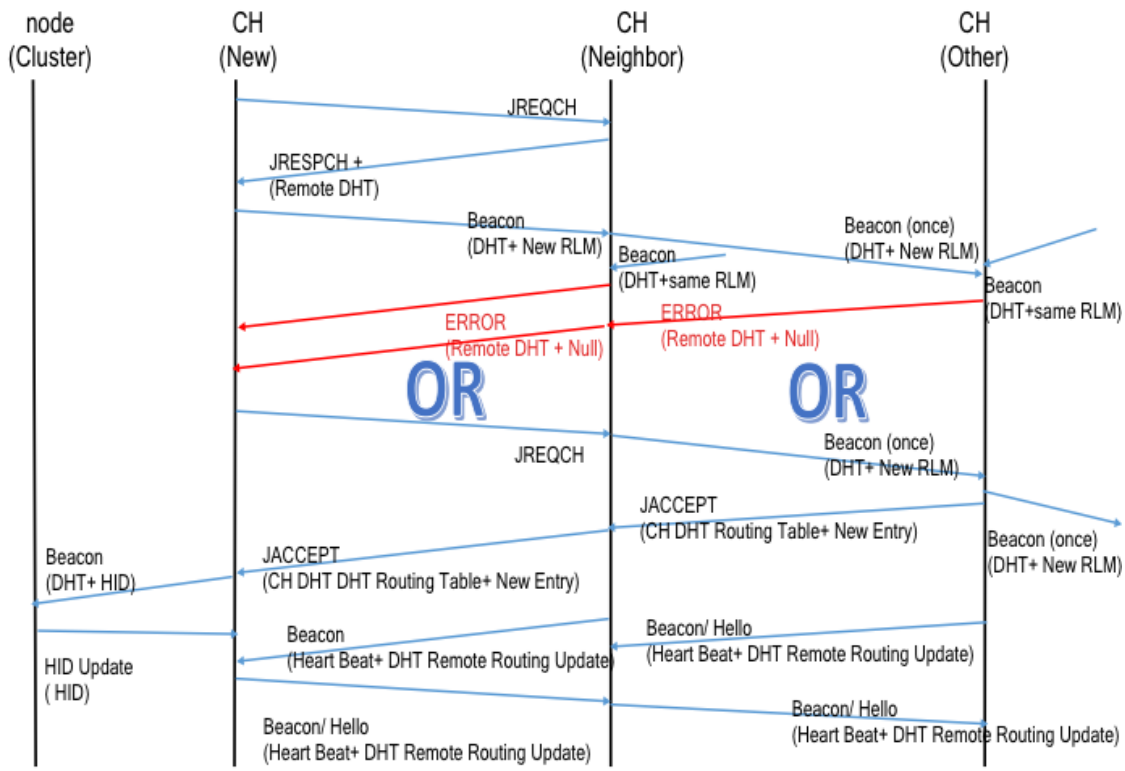


Figure 3.5 CH Joining, Sequence of Events.

g) The new CH, once it receives the response from other CH(s), will update both its remote DHT table, adding an arbitrary HID address to its own landmark and update the link status routing table with all CHs in the top layer.

h) The joining CH will then beacon the remote DHT with its newly assigned RLM and HID to all nodes in the cluster.

i) All nodes in the cluster, when they receive the beacon message from their local cluster CH (Figure 3.4) with the remote DHT table entry having a non-blank HID, will add the RLM prefix to build their own LID, then initiate a periodic HID update (HIDUPDATE) message with its new HID.

3.5 Node Joining

Next, we present the mechanism for a node to join the overlay structure, and to build its own HID in the hierarchical overlay. Figure 3.6 summarizes the sequence of events.

a) A node, being part of a hierarchical structure in the underlay will join the overlay by responding to the beacon message initiated by its own CH. The node will add the CH prefix to its LID and will periodically update its HID address by periodically propagating an HIDUPDATE message.

b) The HIDUPDATE message contains a *key* generated by hashing the node UID to generate a key lying in the LS2 space. The *value* is the current HID of the node reflecting its location and the cluster it is a part of. The <*key, value*> pair is periodically sent to the peer responsible for the key.

c) Any node will be able to extract the current node HID to start a communication by retrieving the <*key, value*> pair containing the HID from the designated node. The designated node holding the key will delete this entry if the HIDUPDATE message is not received for a predefined period.

d) A node which is not part of any existing cluster, and does not receive any beacon messages, will broadcast a JREQN (Node Join REQuest message). The node will send the message with increasing Backoff periods until it receives a response from any neighboring node in the overlay.

e) Any node responding with a JRESPN (Node Join Response) can do so simply by forwarding the beacon message from the CH, which updates the remote DHT table (same as JRESPCH).

f) The first response coming from the closest node will provide the information on the prefix which will be part of the node LID, hence, the node will assign itself a prefix of

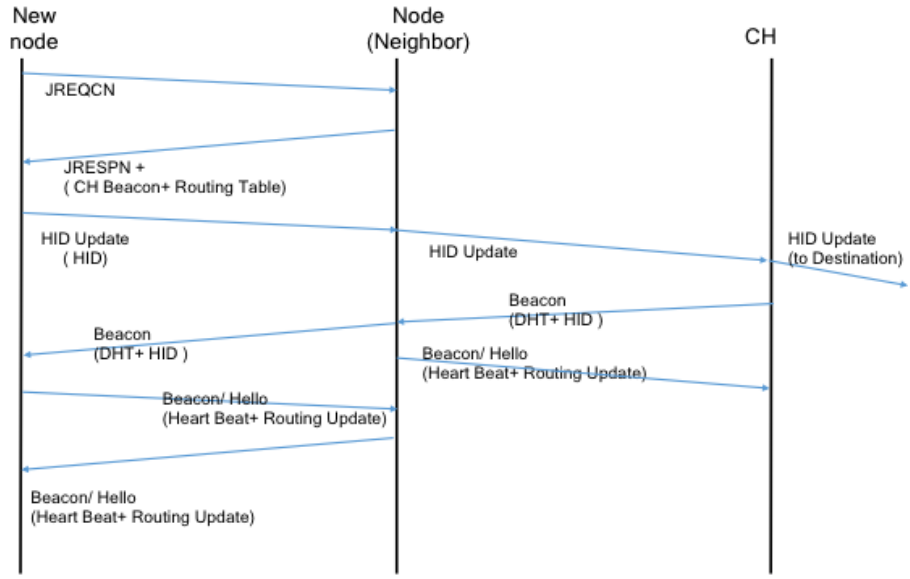


Figure 3.6 Node Joining, Sequence of Events.

the closest CH (RLM) and starts building its routing table with neighboring nodes in the same cluster.

g) Once the node is connected to the overlay DHT structure, neighboring nodes in the logical space will forward to the joining node the keys/values table the node is responsible for as per the mechanism used in the local cluster overlay DHT protocol.

3.6 Mobility Management

Any change in the physical location of a node, or cluster formation change (merging or splitting) is reflected in the HID. A node's HID may change due to mobility, the DHT structure allows the possibility to solve the logical address and maintain routing capability with dynamic topology changes (Requirement 2).

- As a general rule, all routing information are updated periodically with heart beat messages (HIDUPDATE for nodes within the overlay, Remote DHT update for CH). In

other words, the absence of any of these messages will trigger a timeout in the relevant tables and corresponding entries will be deleted. It is obvious in this case that the update frequency will have a significant impact on the network performance (control message overhead, resilience and successful data retrieval operations).

- Cluster Split: Nodes may leave a cluster, with a new additional cluster being formed. This implies that the split will result in two separate clusters, each with a separate CH. The new cluster and CH will follow the same process of cluster joining/bootstrapping, only one CH will keep the same RLM as the original cluster before the split.

- Cluster Merging: Nodes may also join into an existing cluster, in the limiting case resulting in the formation of a new cluster with only one CH and the same RLM prefix. One CH (for the cluster which will merge into the new cluster) will stop sending the heartbeat DHT remote table update. Once the heartbeat message stops, the nodes in the cluster will initiate the join process to join the new cluster and will update their LID and HID accordingly.

- Node Mobility: Adapting the periodic update/ timeout strategy translates into a “fail safe” mode. A node leaving a cluster will stop sending the HIDUPDATE message, however it will initiate a node "join" process with a new cluster.

- CH failure: A new cluster is formed by electing a new CH using the already adapted cluster selection scheme, and the joining process to the new CH will be initiated. If no cluster formation strategy is adapted, the scenario translates to a mass number of nodes initiating a new “Join” process to other clusters in the network as long as physical communication links can be established.

- Node Failure: Routing data will be updated after a timeout period in the absence of periodic updates. To improve network resilience in such case for <*key,value*>

retrieval, the redundancy mechanism applied by the overlay protocol is used to retrieve the <*key,value*> pairs from neighboring nodes.

3.7 Put/Get Application Data Routing and Exchange

The P2P application will use the DHT structure to store values, and to share resources in the hierarchical MANET. The following process *(algorithm 1)* will be used to store (PUT) <*key, value*> pairs in the hierarchical overlay logical space:

1. The resource Unique IDentifier *(rUID)* will be hashed to generate a key which lies within the LS2 space (as mentioned in Section 3.2.2). In this case the generated key will have the same format as the *LID* shown in Table 3.1.

2. The originating node will use the prefix of the <*key*> and compare it with the *RLM* entries in the remote DHT to decide on the numerically closest *RLM* entry in the table.

3. If the local *RLM* is the closest to the key value, the node uses the local DHT protocol to route the $< key, value >$ to the destination peer.

4. Otherwise, the node retrieves the *HID* from the remote DHT routing table entry which has the *RLM* numerically closest to the key value.

5. The node then forwards the PUT message with the HID_{Dest} to the CH which in return forwards the same message to the CH_{Dest} responsible for the HID_{Dest}.

6. The CH_{Dest} will then route the message to the destination peer responsible for the key. The <*key*> will be stored in the node responsible for the key value according to the DHT structure used in the overlay (i.e. for VRR and Chord the key/value pair is stored in the node ID with LID closest or higher than the *key* value).

A similar process (*algorithm 2*) will be followed when a node initiates a query to GET a value stored in the DHT overlay.

Algorithm 1 PUT Operation

function PUT $(rUID)$ **Input** : Resource Unique Identifier $rUID$
Output : $((Key, HID_{Dest})$
(key) $\leftarrow hash(rUID)$ [1.]
Compare key$_{Prefix}$ $with\ RLM_{DHTRemote}$ [2.]
if $key_{Prefix} \preceq RLM_{Local}$ **then**
  Route through Local DHT [3.]
else
  Identify Destination Cluster (HID_{Dest}) [4.]
  CH $\leftarrow$ Forward(<key, value>) [5.]
  CH$_{Dest}$ $\leftarrow$ Forward(<key, value>) [6.]
  Route through Local DHT
end if

Algorithm 2 GET Operation

function GET $< value >$ **Input** : Resource Unique Identifier $rUID$
Output : $(< value >$
(key) $\leftarrow hash(rUID)$ [1.]
Compare key$_{Prefix}$ $with\ RLM_{DHTRemote}$ [2.]
if $key_{Prefix} \preceq RLM_{Local}$ **then**
  Route Query message through Local DHT [3.]
  Node$_{Dest}$ $\leftarrow$ Query <*key*>Node$_{Orig}$ $\leftarrow$ Response <*key, value*>
else
  Identify Destination Cluster (HID_{Dest}) [4.]
  CH $\leftarrow$ ForwardQuery(<key>) [5.]
  CH$_{Dest}$ $\leftarrow$ ForwardQuery(<key>) [6.]
  Route Query message through Local DHT
  Node$_{Dest}$ $\leftarrow$ Query <*key*>
  CH$_{Dest}$ $\leftarrow$ Response <*key, value*> [7.]
  CH$_{Orig}$ $\leftarrow$ Response <*key, value*> [8.]
  Route Response message through Local DHT
  Node$_{Orig}$ $\leftarrow$ Response <*key, value*>
end if

1. The node which initiates the query will start by hashing the *rUID* to get the <*key*>.

2. The node looks into the remote DHT table to find the cluster which owns the *RLM* entry in the table which is numerically closest to the <*key*> prefix.

3. In case the node's own cluster has the numerically closest *RLM* prefix value, it forwards the query message within its DHT structure using the local DHT routing protocol. The node responsible for the key will respond with the value corresponding to the key entry in the DHT table.

4. Otherwise, the message is directly forwarded to its CH, with the *HID* destination corresponding to the closest numerical *RLM* value to the key.

5. The CH will then use the backbone top overlay network to forward the query message to the destination CH responsible for the node owning the <*key*>.

6. The CH of the destination node cluster will forward the message to the target destination using the local cluster DHT routing table.

7. The node responsible for the key will respond with the <*key, value*> and send it directly to its CH.

8. The CH will forward the response to the query message back to the originating CH, which in return will internally forward the response to the originating node using its local DHT routing table.

3.8 Proposal: Pros and Cons

The concept of bringing topology awareness to the overlay by mapping the underlay hierarchical structure to the overlay delivered certain performance improvements which can be summarized in the following:

1. *Localization of routing update and control messages:* [47, 60] The overlay protocol control messages are limited within the cluster and not propagated across the network. Hence, topology changes do not result in excessive traffic in the network which reduces overall traffic overhead, resulting in better latency and less bandwidth utilization.

2. *Route Mismatch Reduction:* [8, 47, 49, 61] By mapping the underlay hierarchy to the overlay, logical neighbors in the overlay are physical neighbors in the underlay. By this we mean, that nodes will be always neighboring in the same cluster both in the logical and the physical overlay and won't be scattered among different clusters logically. This will not always create the perfect route match, unless all nodes are one hop distant from each other, however, this will keep any route stretch local within the cluster boundary. Hence, message routing between peers will not result in excessive redundant traffic (if any) in the physical layer to reach the destination node, which will positively impact the latency and bandwidth utilization.

3. *Mobility Management/ Adaptability:* [13, 57, 8] The use of RLM and HID is bringing adaptive features to the overlay structure as any peer LID changes once connectivity is lost and a node joins a different cluster. The LID will change to a different prefix value (RLM), while the HID will be updated to reflect the new node location in the overlay, hence maintaining the resilience of the network.

On the other hand, the proposal may have some limitations (disadvantages) which we can summarize in the following points:

1. Hierarchical Architectures are heavily dependent on CH (gateways) to maintain connectivity between nodes in different clusters. The performance of our proposal, which does not cover the CH election scheme, will be heavily dependant on the effectiveness of CH election scheme to select the most stable nodes as CH. The selection of non-

stable CH will result in overall poor connectivity hence having low performance in network reliability and resilience. [34, 53, 62].

2. The proposal ensures logical neighbourhood in the overlay for nodes in the same cluster formation in the underlay by the use of a common prefix in the LID. However, adjacent nodes in the physical underlay (one hop apart) may not be immediate successors or predecessors to each other in the logical space. This slight mismatch of the internal cluster topology between physical underlay and logical overlay may result in some intra-cluster traffic overhead. A possible improvement is the implementation of a local DHT structure which minimizes such overhead traffic. OneHopOverlay4MANET [2] is an example of such protocol, it uses a cross-layer design approach to leverage the synergy between P2P applications and MANET by passing routing information between the underlay routing protocol and the DHT structured overlay. This information is used to achieve one logical hop lookup, hence reducing the traffic overhead generated by data message exchange in the overlay.

Chapter 4

Simulation

4.1 Simulation Environment

We used the OMNET++ simulator as the simulation platform. OMNET++ is a discrete event network simulator which provides tools and infrastructure to simulate networks. The main concept is the creation of architecture components, called modules, which represent in functions and features the real components of the system to be simulated. These modules can be assembled, then used with different parameters, to build diverse systems, configurations and test conditions [63–65].

We used the INET Framework [66] in order to support the implementation of protocols and models related to MANETs. INET is an open-source library for the OMNET++ environment which supports multiple types of networks, including mobile and adhoc wireless networks. We used the INET modules to implement the Internet model protocol stack (TCP, UDP, IPV4 [67], OLSR [22] and MAC IEEE 802.11 layer [68]). As INET is built over the OMNET++ infrastructure, it follows the same concept of modules communicating through messages. These modules may represent protocols, and agents that can be integrated with the structure of a node, a host, and a gateway or any other network element.

Table 4.1 Simulation Environment Versions

Environment	Version
OMNeT++	4.6
INET	2.3.0
Oversim	20121206
OS	Ubuntu 18.04.1 LTS

In order to implement the P2P overlay, we used OverSim [69]. OverSim is a flexible framework for overlay network simulation. OverSim is based on OMNET++ and includes both unstructured (e.g. GIA [70]), and structured P2P protocols (i.e. Chord [6], Koorde [71], Kademlia [72], ... *etc.*). In addition, to facilitate the implementation of any additional protocols, OverSim provides a number of common functions such as a generic lookup mechanism for structured P2P overlay networks, and an RPC interface to facilitate messages and calls exchange between peers in the overlay. Besides, OverSim provides additional statistical functions by using the *GlobalStatistics* module, which allows the tracing and analysis of data on an aggregate level across the all overlay modules and components [69, 73]. Table 4.1 shows the versions used in our simulation work.

4.2 Configurations

4.2.1 Nodes

The basic component for the nodes is the AdhocHost, which is an INET module representing a wireless host with sub-components such as battery, MANET routing and wireless interface. The AdhocHost module supports IPv4, UDP and TCP transport layer protocols. In order to construct the node model that properly represent our design, we made addtional modifications in the module structure, On top of the AdhocHost base, we have added the overlay modules using a similar configuration to the *SimpleOverlayHost* module in OverSim. The implementation of the node structure using this combination allowed an integral repre-

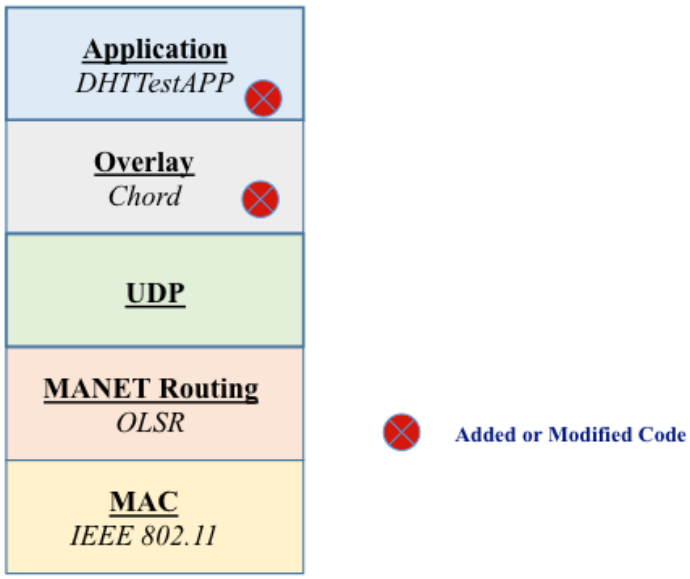


Figure 4.1 Simulation: Node Structure

sentation of a node in a MANET with capability to communicate in a wireless environment, in addition to the ability to join a logical overlay and deploy P2P applications in the top overlay tiers. Figure 4.1 shows the main building blocks of the node protocol stack layers. In the simulation, all nodes are part of clusters, all nodes are able to join the logical space and participate in the P2P DHT application.

4.2.2 Cluster Heads (CHs)/ Gateways

CHs have a different role in the network. CHs will be able to communicate with nodes in the same cluster in the underlay (network layer), as well as with other CHs in the network at the same level through a backbone network. For the latter purpose, an additional routing submodule was added at the network routing layer in addition to the original MANET routing module used in the regular node structure. Both routing modules use OLSR, however in order to connect with both local cluster and backbone network we use the additional HNA

(Host and Network Association) messaging functionality in OLSR specified by an auxiliary specification [22, 74, 75].

The HNA functionality allows a node in a MANET to have multiple interfaces with different wireless networks. The gateways use HNA messages to announce to other nodes in the local cluster their interface addresses and associated external networks through the wireless backbone network. These addresses will be used to forward any message destined to an address that is not available in the node local routing table. The gateways are equipped with 2 routing components (MANET Routing, and Backbone MANET Routing). Each of these routing components will be connected to a separate channel, one channel for the local cluster wireless network, and another channel for the backbone wireless network. Hence, using both interfaces, the gateway will be able to forward packets, data or control packets, internally (intra-cluster communication), or between different clusters through the backbone (inter-cluster) communication.

The CH also plays the role of the Superpeer in the overlay. In the logical space, the Superpeer or the CH is the node responsible for forwarding messages to other networks (subnetworks) in the hierarchical overlay structure. To implement this functionality in the simulation environment, we integrated the *SimpleMultiOverlayHost* modules in OverSim with Adhoc host modules in INET (including MANET and Backbone MANET Routing modules). The *SimpleMultiOverlayHost* module has the ability to accommodate more than one overlay structure per node. As our solution is a 2-layer hierarchical structure, described in Section 3.2.2, we have limited the number of overlays per Superpeer (CH) to 2 overlays. Accordingly, each CH will have 2 different logical IDs, one LID to connect with the local cluster (network) in the overlay logical space, the other LID is used to join the top overlay network, which connects different clusters in the overlay. Figure 4.2 shows the main building blocks for the CH structure used in the simulation.

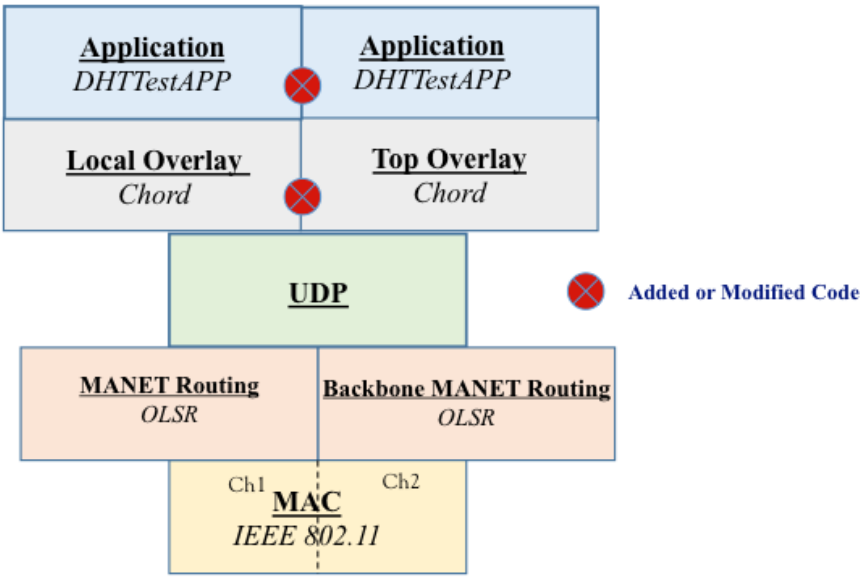


Figure 4.2 Simulation: CH Structure

4.2.3 Overlay

Hierarchical Overlay

As described in subsection 3.2.2, subsection 4.2.1 and subsection 4.2.2, the overlay has a hierarchical structure, with clusters forming suboverlays or subnetworks, and a top overlay layer connecting all superpeers. We used a similar overlay topology implemented by another researcher to study the performance of a hierarchical structured overlay under different churn profiles [76]. In the work, Rocamora *et al.* implemented a hierarchical overlay with a structured DHT protocol used in both subnetworks or clusters (mainly Chord), and a DHT structured backbone with only superpeers joining the top overlay (experiments were conducted using Chord and Kademlia in different simulation scenarios) and the performance of all scenarios was compared to the performance of a flat structured DHT overlay (Chord). Unlike our work, however, the underlay topology and structure was totally abstracted, the assumption for the underlay was a transparent flat network able to deliver and exchange data and control packets between nodes in the overlay. The simulation results demonstrated

a better performance of the hierarchical overlay structure vs. flat overlay using different churn profiles. Figure 4.3 shows a sample overlay configuration with 4 clusters and 10 nodes (including the CH) in each cluster. CHs are connected through a top (backbone) overlay network, in the same Figure, the backbone network is shown in thick black and yellow lines.

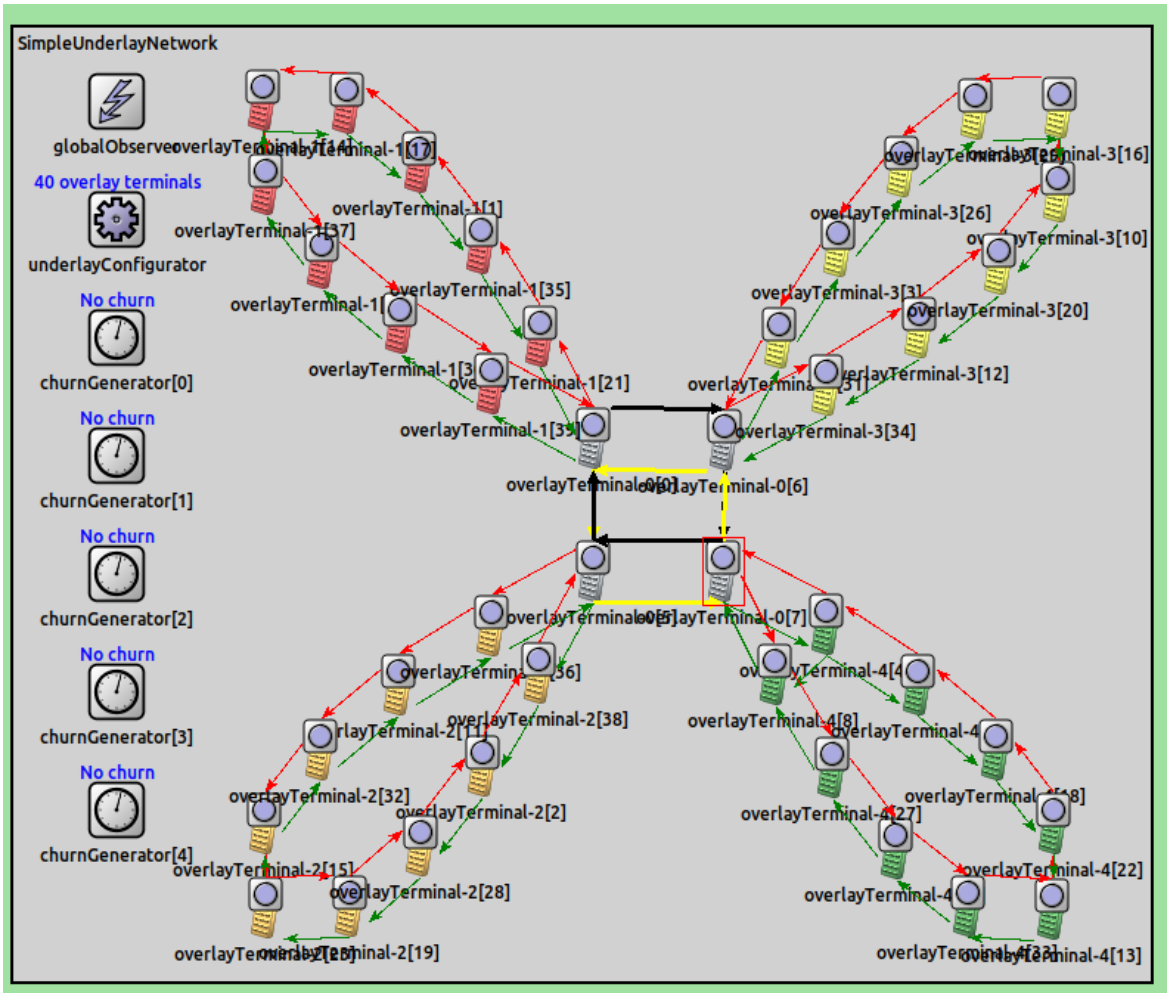


Figure 4.3 Simulation: A DHT Superpeer Network with 4ClustersX10Nodes.

We implemented several modifications and additions to the simulation environment in order to be able to construct a hierarchical underlay and build and identical hierarchical overlay on top of it. The bootstrap node selection mechanism used by Chord is modified to restrict the bootstrap process to one node (CH) as per the proposed design. By this modification, we avoided the formation of disjoint Chord rings in the overlay, and maintained

an integral single ring formation for each cluster. In addition, to allow for better visibility and differentiation between the overlay top layer and cluster formation, we modified the visual display and update functions to provide a different representation of the top layer overlay, while ensuring it timely and accurately represents the relation between all nodes in the overlay as dynamic changes take place in the network. We modified the BaseOverlay modules in the simulator, used by Chord, to allow for CHs to restore the *join* state and rejoin Chord for each overlay tier. We added our proposed protocol for global queries, messages structure, and global message exchange mechanisms (inter-cluster messaging) to the application layer (DHT) modules. Besides, we added special functions to allow for the message exchange mechanisms to identify local and global CHs as part of the global message forwarding scheme, beside handling the special cases for communication failure when executing the global queries.

For a hierarchical overlay configuration, all CHs will join the local logical space (subnetwork) and participate in the P2P application. Besides, all CHs (Superpeers) will join the top overlay network in the hierarchical structure to provide a logical link between clusters to allow inter-cluster communication and messages exchange. Using the DHTTestApp application, all peers participating in the application will generate PUTs/GETs for random keys in the logical overlay space, beside storing key values based on their assigned logical IDs and the DHT structure used in the simulation. DHTTestApp, the application used in the simulation, will use by default the local logical space (subnetwork) to store the <*key, value*> pair, i.e. the PUT operation will be done locally (intra-cluster process). We modified the application to allow nodes in different clusters to generate global GET requests and query for <*key,value*> pairs located in separate clusters in the overlay. In our simulation, we used different configurations with different cluster sizes (number of nodes per cluster) to study the impact of granularity. These configurations will be mapped to the overlay hierarchy as per our proposed solution. In these configurations, the CHs (Superpeers) are part of the clusters

(subnetworks) performing similar PUT/GET requests like similar peers in the cluster. In addition, the CH (Superpeer), as a member of the top layer, has the responsibility of routing messages to different clusters in the logical space, which is a similar functionality to the GW role in the underlay. Whereas Superpeers in the top overlay are using DHT Chord, the CHs/GWs in the underlay are using OLSR with auxiliary HNA interface capability to connect to the backbone, and to be able to propagate to all nodes in the cluster its interface address with the backbone network connecting it to other clusters. Table 4.2 shows different configurations used in the simulation runs.

Table 4.2 Simulation: Hierarchical Overlay Configurations

		Hierarchical Configuration				
	unit	3X13 *	4X10	5X8	8X5	10X4
Top Overlay Size	CH	3	4	5	8	10
Cluster Size	peer	12	9	7	4	3

* In order to keep the total number of nodes using the P2P application equal, one cluster has 13+1 peers to match the total number of peers (40 peers) in other configurations.

4.2.4 Flat Overlay

In order to test the efficiency and success of our solution, we compared the performance of the hierarchical overlay proposal with a flat overlay. In other words, using the same test parameters, we deployed a flat overlay over a hierarchical underlay and tracked the same performance metrics. We used equivalent module structures for both CHs and nodes as in the hierarchical overlay, to implement the flat overlay configurations. The objective was to maintain the same functionality for the CHs in routing, and for the nodes to generate and respond to logical queries. The hierarchical underlay is able to perform inter-cluster communication and packet exchange between nodes in different clusters using the (OLSR + HNA auxiliary functionality) protocol at the network layer in both the backbone network and inside the clusters.

To achieve full routing capability in the underlay between nodes located in separate clusters, we made additional modifications to the structure and modules used in our proposal. We deployed a routing solution designed for an hierarchical underlay network, which is not a standard OMNeT++ or INET module. This routing capability was essential to allow the nodes in different clusters to join one logical flat overlay, and to connect to each other arbitrarily, in order to form that logical overlay on top of interconnected clusters in the underlay. For this purpose, we used the routing solution described in [75]. The solution is basically using a DHT routing table which provide CHs in the backbone with routing information to nodes in different clusters. Hence, CHs in the underlay will be able to locate and route packets originated from nodes in the local cluster, to destination nodes located in remote clusters using the backbone network. This routing solution was deployed in the tier2 of the application layer, the same tier2 in the application layer is used in our hierarchical overlay implementation by the P2P application (in our test case the DHTTestApp) to generate random PUT/GET requests. The CH module in the flat overlay used in [75] to provide this hierarchical routing capability in the underlay, has a limitation which does not allow for multiple-tier overlay implementation. In other words, only the DHT application used for routing will be deployed in tier2 of the application layer in the overlay. Hence, CHs will not be able to join the P2P application in the overlay, and will not be able to generate their own PUT/GET requests. In order to maintain an equivalent topology between hierarchical and flat overlay in terms of total number of nodes participating and joining the P2P application in the overlay, we added an additional node for each cluster in the flat overlay configuration. This node will replace the CH in the P2P application overlay, hence maintaining the same logical network size (i.e. number of peers) and same level of data messages and traffic (PUT/GET) generated in the overlay. Accordingly, the CHs will only play the role of a GW, routing packets in the backbone between different clusters in the flat overlay/ hierarchical underlay configuration. CHs will not participate in the P2P application logical space, their role in this

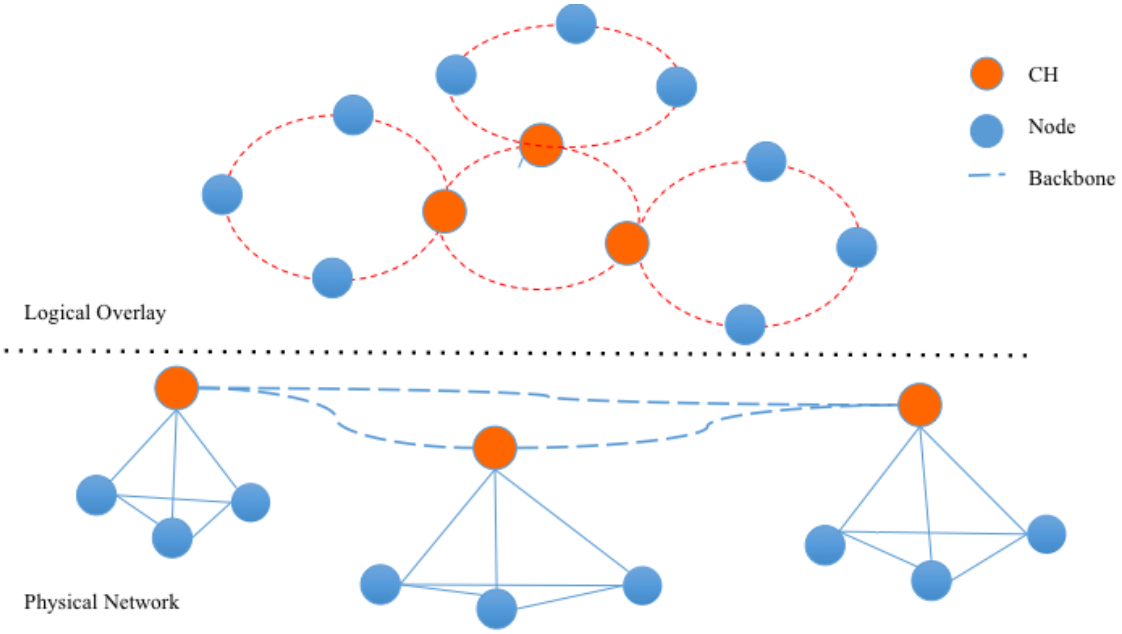
part will be compensated by the additional node in the cluster which will join the overlay to maintain the same logical space size as the equivalent hierarchical topology. Figure 4.4 shows a sample of a 3X3 hierarchical network. Figure 4.4a shows the hierarchical overlay mapping the underlay topology. Figure 4.5b shows the flat overlay implementation by adding an additional node in the cluster, this node will join the overlay to maintain the overall logical network size (# of participating nodes), which is 12 peers in that Figure, the same in both configurations (flat and hierarchical).

4.3 Simulation Parameters Summary

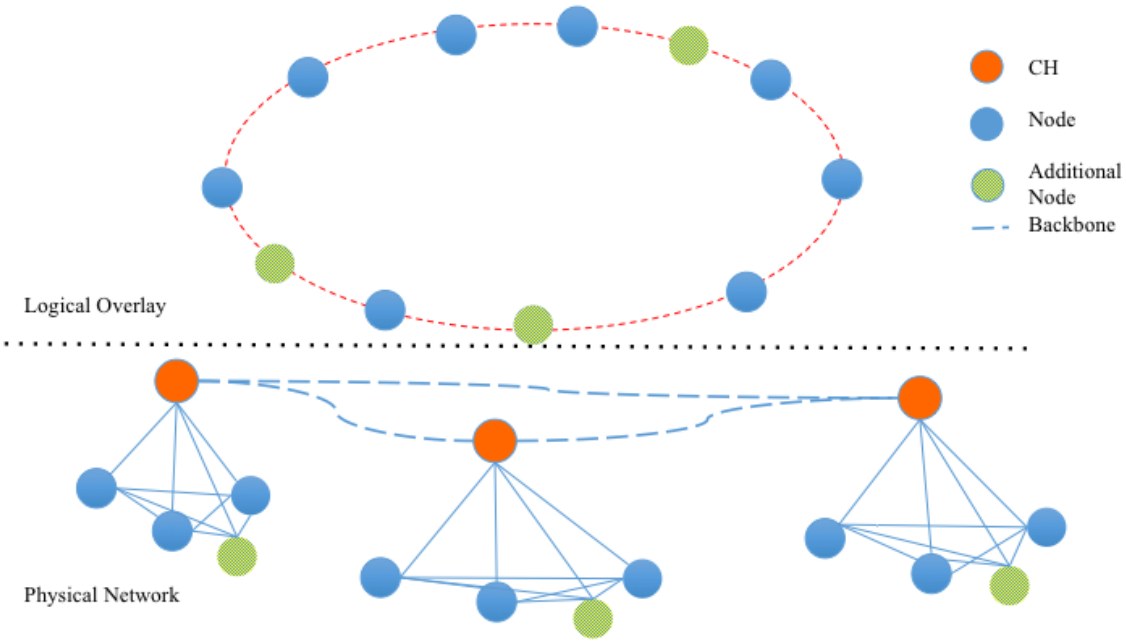
In this section, we share a brief summary of the main common simulation components and parameters used in the simulation. We will elaborate on the reason and justifications behind specific choices in our simulation. Table 4.3 summarizes the parameters used in our simulation for both the hierarchical and the flat overlay.

UDP [77], a connection-less communication protocol, is the transport layer protocol used by the P2P MANET application as it provides better performance in bandwidth utilization (smaller datagram overheads), and latency by dropping i.o. processing delayed packets (however, this may impact resilience and hit rate). This is the main reason UDP is utilized by real time services such as video streaming, live conferences, ... *etc.* . OLSR [22], a proactive routing protocol, is used as it provides a better response time by eliminating the need for route discoveries (however, this will come at the expense of extra overhead for TC messages and regular topology updates). In subsection 4.2.2, we discussed in more details the OLSR features used.

IEEE 802.11 [68] is the MAC protocol used in our simulation. IEEE 802.11 is a contention-based protocol where nodes compete for channel access, no other node is allowed to access the communication media once it is occupied by one of the nodes [78]. IEEE 802.11 is one of the most commonly used MAC protocols in MANET where the type of



(a) Simulation: Hierarchical Overlay vs. Physical Network



(b) Simulation: Flat Overlay with Additional Node per cluster

Figure 4.4 Simulation: Hierarchical Underlay Network and the Corresponding Hierarchical/ Flat Overlays.

Table 4.3 Simulation Parameters

Parameter	Unit	Hierarchical Overlay	Flat Overlay
Transport Layer Protocol	N/A	UDP	UDP
Internet Protocol	N/A	IPv4	IPv4
Routing Protocol	N/A	OLSR	OLSR
MAC layer	N/A	IEEE 802.11	IEEE 802.11
Area	mxm	4000x4000	4000x4000
Node Transmission Range [1]	m	1400	1400
Node Transmission Range [2]	m	500-600	500-600
GW Transmission Range [3]	m	3000-4000	3000-4000
Total Simulation Time	sec	600	820
Transition Time	sec	150	370 [4]
Measurement Time	sec	450	450
DHTTestApp Interval	sec	30,15,10	30,15,10
Simulation Runs	#	10	10

[1] Single-hop scenario. The specific range used in a configuration is maintained for both hierarchical and flat simulations runs.
[2] Multi-hop scenario. Transmission range is reduced to allow for multi-hop connection between nodes and CHs.
[3] The transmission range for GWs ensure one-hop distance between CHs in the backbone network.
[3] The flat configuration requires more time to stabilize.

communication is intermittent or takes place in bursts [78]. The contention mechanism allows better utilization of bandwidth, and access to communication media is granted on an immediate basis when the channels are free, and when there is no conflict between nodes trying to initiate a transmission of data packets in the same time. However, conflicts may occur with instantaneous attempts to access the communication media by several nodes. This will result in collisions, and re-transmission attempts will take place following random contention periods, which implies some delays in the communication process.

We have selected 3 different values for the PUT/GET requests generation intervals (30, 15 and 10 sec) respectively. The values are selected to generate different patterns of data traffic intensity. We did not perform a failure test by generating intensive data traffic to reach a full network contention condition resulting in consistent communication failures. The data

in such case won't be relevant in studying the network performance under average data traffic conditions with different network configurations.

We ran the simulation for 10 times per configuration, each with a different seed value, then used the results to generate the mean values and calculate the 95% confidence intervals. The measurement time to track all metrics was set to 450 sec, however the total run time is different for the hierarchical and flat overlays. The flat overlay took a longer time to stabilize as logical links in the flat overlay are established for a single larger overlay network size (40 nodes).

4.3.1 Single-Hop Static Simulation Scenario

In different scenarios, the transmission ranges are selected in a way to vary the configuration (number of hops) between nodes in the same cluster or the backbone. In Table 4.3 the transmission range is selected to ensure one-hop communication between the nodes and CHs, and between the CHs joining the backbone. The rational behind maintaining this condition is to eliminate any impact of multi-hop communication (which cannot be normalized between all nodes) on the latency results. These values for the transmission range may be larger than typical ranges used for IEEE802.11 type of networks where the transmission range lies between 250 m-500 m.

4.3.2 Multi-Hop Static Simulation Scenario

In the multi-hop simulation scenario, we aim to achieve a multi-hop connection between the CH and some/all nodes in the local cluster. In order to be able to materialize such a topology in the simulation environment, we reduced the transmission range of both the nodes in the local cluster and the CH in the same cluster and using the same channel for intra-cluster communication.

Table 4.3 shows the transmission ranges used for the multi-hop scenario. It is to be noted that the transmission range in the backbone is kept at one hop transmission range between CHs. The reason is to reduce the variance in latency and response time, and minimize the number of variables that may affect the performance metrics due to varied hop counts between CHs in the backbone.

4.3.3 Mobile Simulation Scenario

In the mobile scenario, we assessed the network performance with dynamic topology changes resulting from continuous movement of participating nodes. Typically, for emergency and disaster handling teams, or for tactical units, the nodes move as a group and keep their unit formation while in a dynamic movement situation [79, 80].

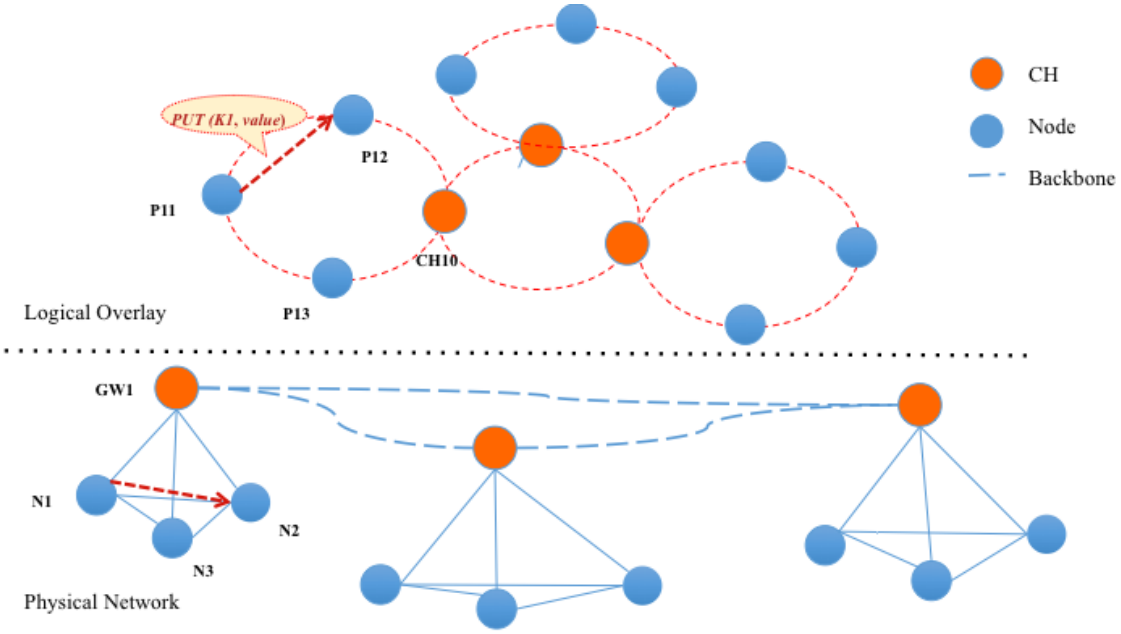
Table 4.4 Mobile Scenario Simulation Parameters

Parameter	Unit	Mobile configuration
Area	mxm	4000x4000
Nodes per Cluster [1]	#	9
Number of Gateways [1]	#	4
Node Transmission Range [2]	m	700m
GW Transmission Range [3]	m	1500
Max. Speed [4]	m/sec	20
Min. Speed [4]	m/sec	10
Total Simulation Time	sec	600
Transition Time	sec	150 [5]
Measurement Time	sec	450
DHTTestApp Interval	sec	30
Simulation Runs	#	10

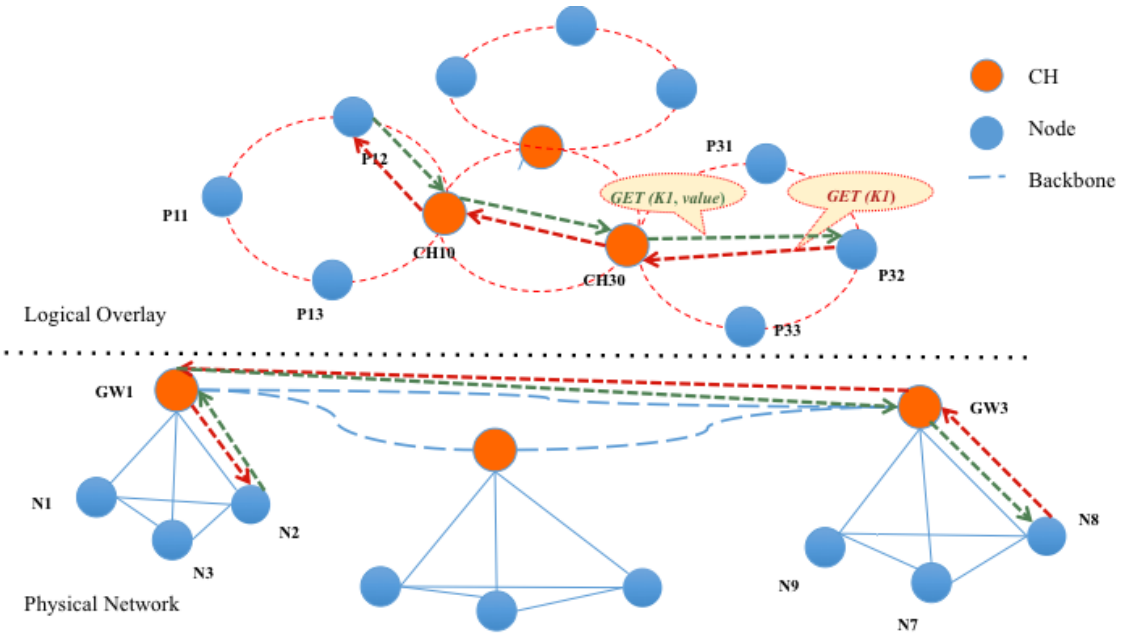
[1] We only used a 4X10 nodes configuration in the mobility scenario.
[2] The transmission range will allow the nodes within the cluster to be one-hop or multiple-hop count from the CH.
[3] The transmission range will allow CHs to be at a single-hop or multiple-hop distance from each other in the backbone pending on the mobility scenario.
[4] This is equivalent to a motorized vehicle speed. Applies for all nodes (CHs and nodes).
[5] We did not change the transition time to maintain the same baseline test conditions for both static and dynamic scenarios

In our simulation, we used the RPGM (Reference Point Group Mobility) [81] as it better represents the mobility characteristics of hierarchical MANETs. In this model, which can be classified under the "models of spacial dependency" category, nodes move in formations which are logically correlated [82]. A main feature of the RPGM model is the fact that it allows a random motion of a group of nodes while keeping the group formation coherence, as well as providing a degree of motion freedom for individual mobile nodes within the group. The group movements follow the path of a logical center (the reference point) for each group in the network. The motion of the group logical center completely defines the movement of its corresponding group node members, including their speed and direction. At the same time, mobile nodes in the group can randomly move about their group specific predefined reference point(s) [80, 81].

Table 4.4 shows the different parameters used in the mobility scenarios. All scenarios adapted the same hierarchical configuration (4X10 nodes), while they vary in terms of mobility patterns and also in the configuration of the backbone network. In order to maintain the connectivity in the backbone, we have added 0 (null), 2, 3 and 4 static nodes to the backbone for 4 mobility scenarios tested respectively. The role of the static nodes added is to maintain the link between dynamic CHs moving in the space, hence, avoid a communication failure condition resulting from a CH being disconnected from the backbone network. Such failure will simply result in clear deterioration in the network performance which is not related to the proposed solution functionality, but rather to the mobility scenario dynamics generating this failure condition. By adding the static nodes, we ensured the backbone connectivity is maintained between CH during the simulation, with varying number of hop counts between CHs as the topology in the backbone dynamically changes due to mobility.



(a) Simulation: PUT Request (Local)



(b) Simulation: GET Request (Global)

Figure 4.5 Simulation: Local PUT/ Global GET for the Same <*key, value*> in a Hierarchical Overlay Network

4.4 Metrics

In the context of our simulation, we need to differentiate between *Local* and *Global* data exchange mechanisms (e.g. PUT/GET local and global requests in a DHT overlay). A *local* data exchange takes place between 2 peers located in the same cluster in a hierarchical overlay. A *global* query or data exchange takes place between 2 peers located in different clusters in the hierarchical overlay. Figure 4.5 illustrates a *local* PUT mechanism and a *global* GET mechanism for the same <*key, value*> pair. It is also to be noted that the hierarchical overlay illustrated in the same figure is based on our proposal, by mapping the underlay structure to the overlay, a similar number of logical and physical hops in the overlay and underlay exists for the same data exchange.

We used 3 common MANET performance measures in order to assess the performance of our proposal:

1. **Success Ratio:** Network resilience will be measured by the ***Success Ratio*** [2, 36, 41, 83, 84]. In our simulation, the success ratio is the percentage of the number of successful GETs vs. overall GET requests initiated by the DHTTestApp application across all participating nodes in the network. We will consider only the GET requests, the PUT requests in the application are bound to the default inbound cluster nodes (local). We will track both the local GET requests (i.e. the key is located inside the same cluster as the node generating the GET request), and the the global GET requests (i.e. the key is stored in a peer located in a different cluster from the node generating the request). Figure 4.5 illustrates the PUT/GET mechanism in the hierarchical overlay. The success ratio can be calculated using the following equation:

$$Success\ Ratio = \frac{\#\ Successful\ GET\ Requests}{\#\ Total\ GET\ Requests}\ \% \tag{4.1}$$

2. **Latency:** Latency is an important measure of routing performance, and a critical measure for QoS. Latency will be measured by the time elapsed to complete a successful GET request, which is the time elapsed from the time the node initiates a GET request, until the time it receives a correct GET response [2, 36, 41, 83, 84]. It is to be noted that, in the single-hop simulation scenario, we maintained a one physical hop distance between nodes and CHs in the same cluster to normalize the impact of multi-hop routing delays across the network. In the other scenarios, we tracked the impact of multi-hop distance between nodes and mobility on the latency metric.

3. **Efficiency:** In our simulation, Efficiency will be measured by the level of traffic overhead generated in both the backbone and local cluster networks. The overhead traffic is defined as all traffic not related to the data exchange. In this case, the overhead traffic includes the packets generated by the routing protocols for maintenance (such as DHT entry updates, stabilization messages for Chord, multi-hop and mobility scenarios OLSR TC messages, ... *etc.*) [2, 36, 41, 83, 84]. In our simulation, the overhead traffic will be measured in number of packets traveling, both in the backbone and locally within the clusters, in the absence of any data exchange (PUTs/GETs generated by DHTTestApp). The number of packets will be tracked for the same simulation period in both flat and hierarchical topologies.

Chapter 5

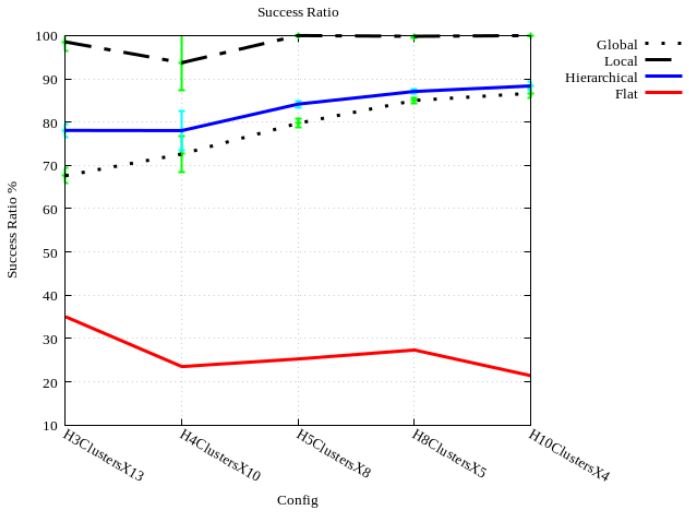
Simulation Results and Analysis

In the following, we will present the simulation results for all the scenarios we tested (Single-Hop, Multi-Hop, Mobility) and share the relevant analysis and justifications.

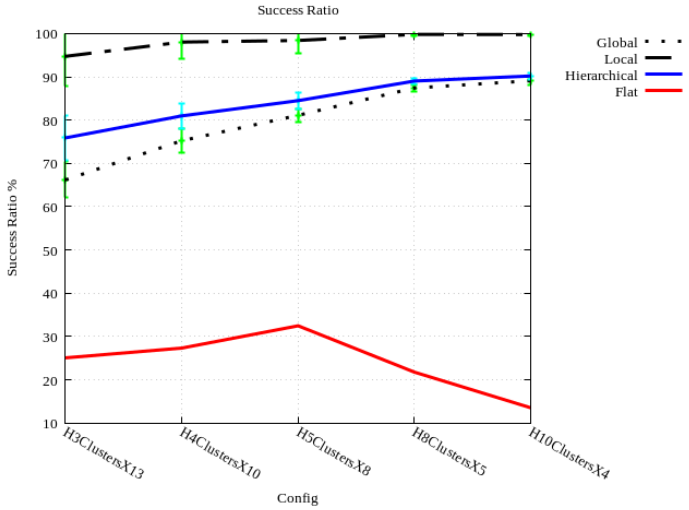
5.1 Performance Evaluation: Single-Hop Scenarios

In this section we will describe in details the single-hop configuration results. By maintaining this connectivity configuration in these scenarios, we eliminate the effect of the routing protocol's efficiency and response on the performance metrics with packets being exchanged between 2 directly connected nodes. As described in section 4.4, we measured the performance of a hierarchical overlay (our proposal) vs. a flat overlay deployed on the same underlay configuration. We used the metrics described in that section, and also the simulation parameters described in subsection 4.3 . In this context, the test results for these scenarios are better suited to represent (as baseline) the performance of our solution vs. the alternative configuration of a flat overlay deployed over the hierarchical underlay structure.

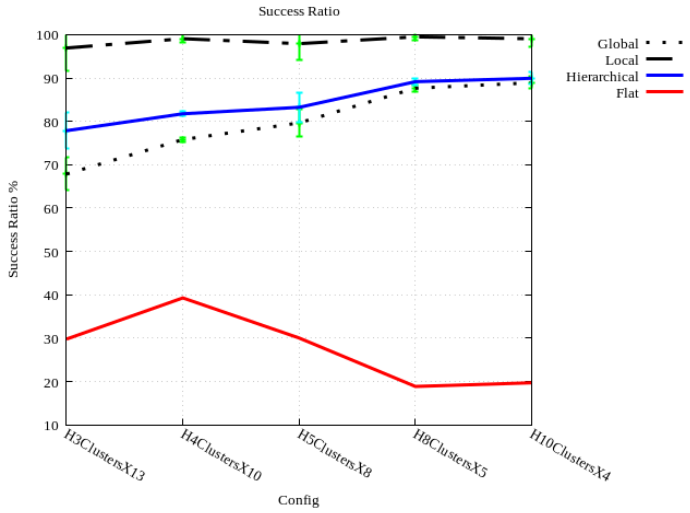
Figure 5.1 Success Ratio Results with Different Test Intervals



(a) Success Ratio: DHTTestApp interval = 30 sec



(b) Success Ratio: DHTTestApp interval = 15 sec



(c) Success Ratio: DHTTestApp interval = 10 sec

5.1.1 Success Ratio

We used the *DHTTEstApp* application in OverSim to generate the PUT/GET requests with different test intervals (30, 15 and 10 sec respectively) as listed in Table 4.3. The simulation results clearly indicate that the hierarchical overlay has better performance in terms of success ratio. Besides, it shows a clear positive trend in success ratio as we move to a finer granularity (higher number of clusters). On the other hand, we see a consistently poor performance for a flat overlay topology, which shows a better suitability of a hierarchical overlay implementation on a hierarchical underlay vs. a different overlay design (flat design in our simulation case).

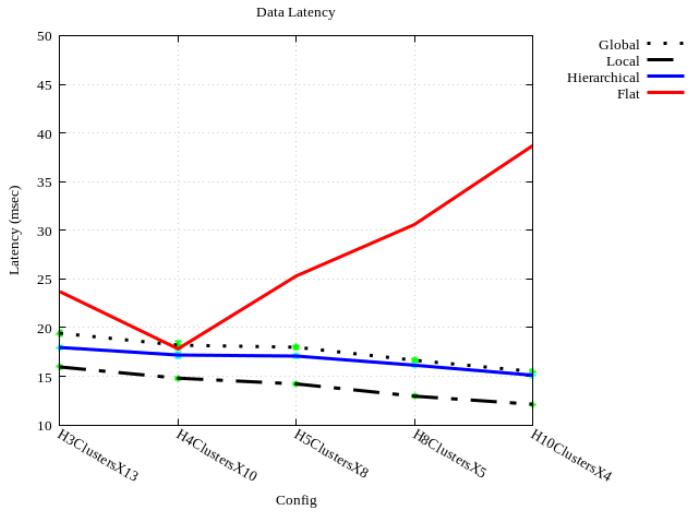
The dashed and dotted lines in the graph represent the local and global success ratio results respectively (these apply only for the hierarchical overlay solution). It is clear from the graphs that the local GET request success ratio results improve significantly as the cluster size shrinks in number of nodes. On the other side, the global GET requests follow the same trend of improvement as we move to a fine granularity (higher number of clusters with smaller cluster sizes). This can be partially attributed to the fact that parts of the global GETs are local in nature (performed locally at the source and destination clusters), which are executed with better success ratio as the size of a cluster is reduced.

The Error Bars are calculated based on a 95% confidence level. The error intervals in the graph are quite minor, which reflect a good level of consistency and stability in the hierarchical overlay performance.

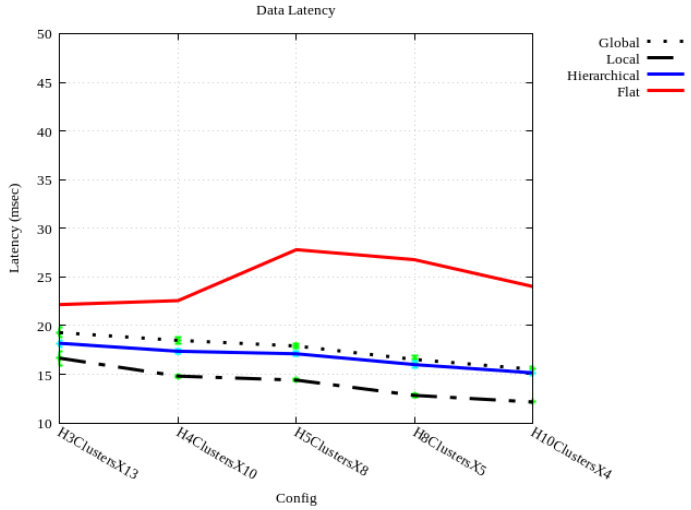
5.1.2 Latency

Latency results for the hierarchical overlay clearly indicate better response time for a hierarchical overlay vs. the flat overlay as shown in Figure 5.2. The results also show a stable performance and a clear improvement trend in latency results as the level of granularity increases. This can be attributed to the fact that the local GET delay significantly improved

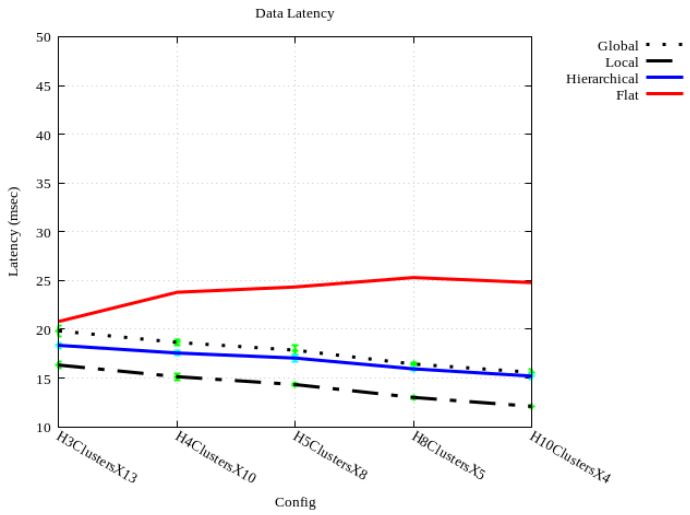
Figure 5.2 Latency Results with Different Test Intervals



(a) Latency: DHTTestApp interval = 30 sec



(b) Latency: DHTTestApp interval = 15 sec



(c) Latency: DHTTestApp interval = 10 sec

(shown in dotted lines in Figure 5.2) when the size of the cluster is reduced, hence the level of local traffic inside the cluster decreases accordingly. In return, the global GET request latency trend (shown as a dashed line in Figure 5.2) improves as well, the global queries involve both intra-cluster and inter-cluster communication. Besides, the inter-cluster communication always translate to a constant overhead in terms of number of hops. This is due to the fact that in the single-hop configuration, all CHs are maintained at a single hop distance from each other, which was previously described in subsection 4.3.1.

The concept of mapping the underlay hierarchy to the logical overlay allows to leverage the enhancement in performance resulting from a better connectivity, and shorter paths between nodes in the underlay. This main benefit of our proposal is clearly demonstrated in both success ratio and latency performance results. On the other hand, the flat overlay topology is showing higher latency results vs. the hierarchical overlay simply due to the fact that logical paths translate to a number of hops in the physical underlay, which are usually a stretch *vs.* the logical path. This *randomness*, which in general characterizes the performance of the flat overlay in the simulation, can be attributed to two main factors:

1. The arbitrary choice of the LIDs for the nodes joining the flat overlay. This arbitrary choice of the peer LIDs results in the changeability of the flat overlay logical configuration in different scenarios. The arbitrary choice of the LIDs in the overlay is deliberately maintained as the IP address is assigned to each node in the simulation configuration file. By keeping the LID randomly generated for the nodes in the overlay, we are able to generate different overlay topologies and different ring formations (successor/ predecessor relations) for the overlay. Assuming the case we use a hash function to hash the node IP address in the simulation to generate the node LID, it will result in having both the IP address and corresponding LID as constants in all configurations or scenarios, hence the results will be representing a specific condition and logical relationship between nodes, rather than representing a generic solution.

This change in the logical overlay topology results in different number of logical hops involved in communication between 2 peers. This variation is reflected by the physical path between the nodes, as the logical path varies when nodes are assigned different LIDs in different simulation runs [8, 11].

2. The arbitrary choice for generated keys, and PUT/GET requests by the application used in the simulation *DHTTestApp*. This will result in different peers with different LIDs being engaged in the process of storing (PUT) or retrieving (GET) <*key, value*> pairs. This will have a similar impact on the number of hops involved in communication as described above.

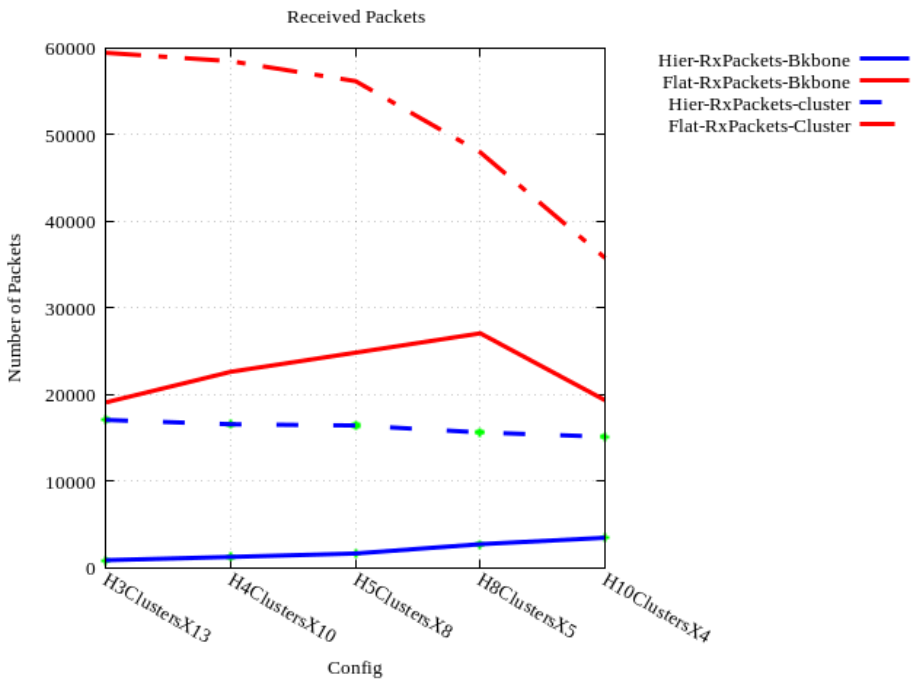
This randomness in the physical path materialization, which results in an arbitrary number of hops between the nodes exchanging data queries, is a reason for the lack of stability (high variability), or the absence of a consistent trend in performance results for the flat overlay. Different configurations will have different LID assigned to peers, hence, different communication patterns will be generated, and different routes will be used to exchange messages. This will translate to a large variability in latency and response time, also in the traffic generated due to the data exchange (as shown in Figure 5.1- 5.2).

5.1.3 Efficiency

As described in section 4.4, we measured the efficiency of our design by tracking the traffic (in number of packets) exchanged within both the backbone and local clusters in the absence of any data exchange. We used the same total measurement period (450 sec) and the same simulation times (600, 820 sec) for the hierarchical and flat overlays respectively.

Figure 5.3 shows the level of traffic (in packets) for both a hierarchical overlay and flat overlay network. The Figure shows the level of traffic in the backbone network indicating lower level of traffic in the backbone for the hierarchical overlay network. This can be attributed to the impact of "localization" where the DHT maintenance traffic is constrained

Figure 5.3 Traffic Overhead (with Zero Data Packets)

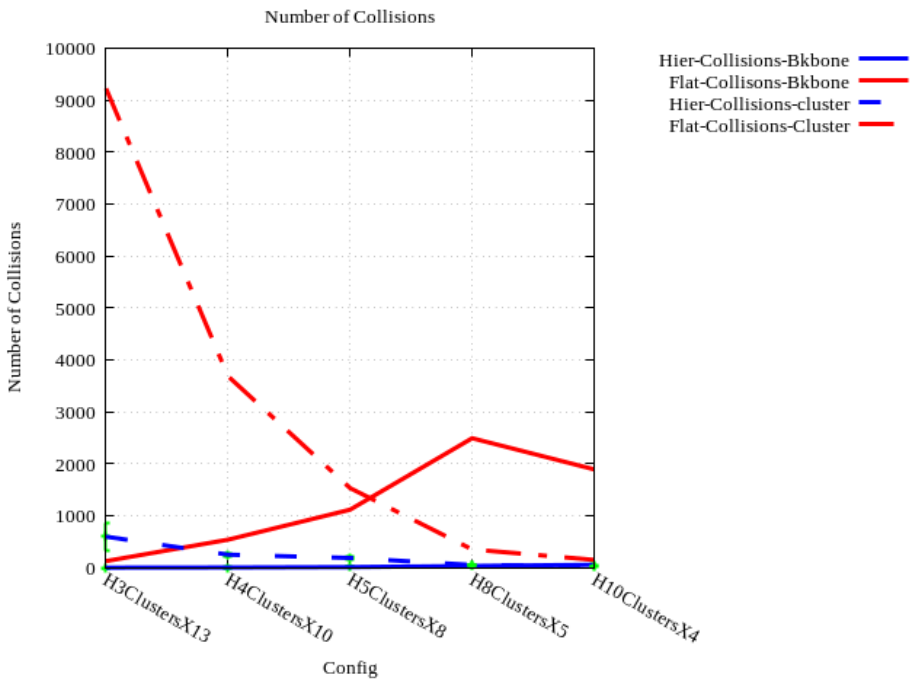


within the local cluster network and does not propagate (through route stretch) using the backbone network to other outside nodes. This also explains the reason for having a slightly higher local cluster traffic level in the hierarchical overlay vs. the flat overlay. Most of the control messages in the cluster will be routed internally inside the cluster with the hierarchical overlay vs. the possibility of propagating the control packets through the backbone network in a flat overlay structure.

The same Figure 5.3 also shows a direct relationship between the overhead traffic in the local clusters and clusters size. As the cluster size shrinks (fewer peers in the cluster) with fine granularity, the traffic overhead in the clusters is reduced, whereas the traffic in the backbone tends to slightly increase as more CHs join the backbone.

It is worth to be mentioned that efficiency is a critical metric in improving the overall MANET performance, given the fact that less overhead traffic will result in lower bandwidth utilization. This will allow higher capacity for data exchange, lower congestion probability

Figure 5.4 Number of Collisions with Zero Data Exchange



and fewer collisions, which will result in higher resilience (lookup success rate) and lower latency [2, 8, 36].

5.1.4 Additional Metrics and Further Analysis

Further to the above evaluation of performance metrics, we tracked an additional parameter as an in-process measure for the network performance. We have tracked the number of collisions in the network for both backbone and cluster sub-networks. In general, we can consider collisions as an indication of a failure in the network [85, 86]. Collisions can be an indication of the level of congestion (efficiency), communication failure (success ratio), and packet transmission retrials (latency). In this analysis, we did not evaluate our results vs. a benchmark, yet we compared the same metrics for both hierarchical and flat overlay configurations. Also, we tried to identify the link and analogy with other performance

metrics, and use this analogy to support (or contradict) our interpretation of the simulation performance metrics results.

We consider the status where there is no data exchange between nodes (no PUT/GET requests and queries are generated) as the baseline status, packets will only be exchanged for control, topology updates and DHT maintenance. Figure 5.4 shows the number of collisions during the simulation time for both hierarchical and flat network. The main observations can be summarized in the following:

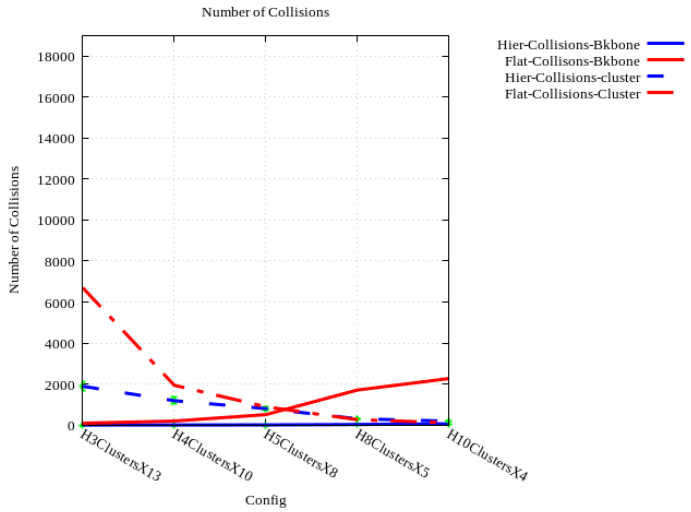
- Overall, the number of collisions in the hierarchical overlay is considerably lower than the number of collisions in the flat overlay. Besides, the backbone network for the hierarchical overlay demonstrates a steady low level of collisions across all configurations.

- The number of collisions in the flat network backbone proportionally increases with the level of granularity level (number of clusters). Due to the increase in the number of clusters, the probability of having neighboring peers in the logical overlay which are located in separate clusters in the physical underlay increases. This will result in more control traffic migrating to the backbone network (Figure 5.3), hence the probability of more collisions to occur.

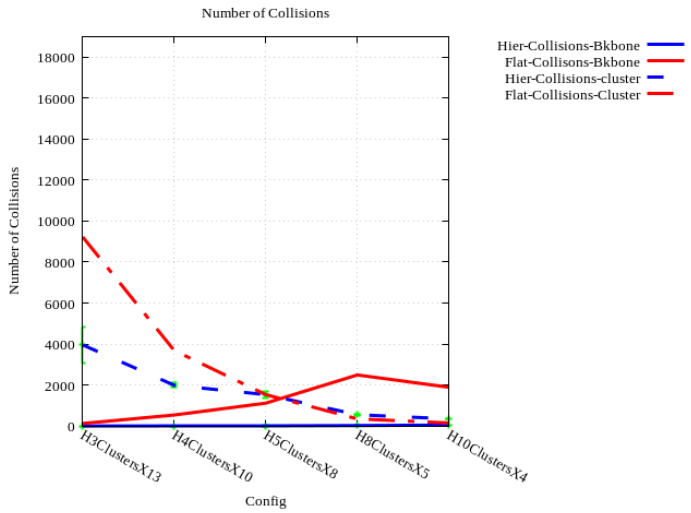
- The number of collisions is partially a function of traffic level (beside the communication channel bandwidth and other parameters). This explains the falling trend in the number of collisions in the local clusters as the number of nodes per cluster declines, hence the intra-cluster traffic is reduced as shown in Figure 5.3.

We monitored the same metric (number of collisions) in a single-hop configuration, this time with the activation of the P2P application *DHTTestApp*, and the generation of PUT/GET requests at different intervals (30, 15 and 10 sec respectively). Figure 5.5 shows a similar trend of an increasing number of collisions in the backbone with granularity, comparable

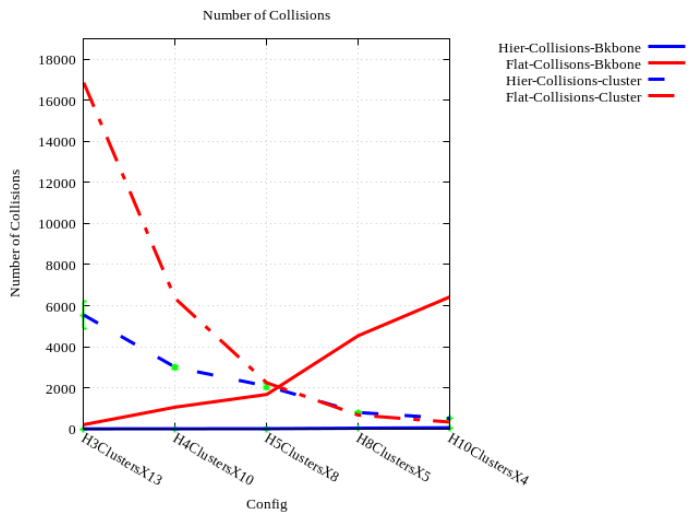
Figure 5.5 Number of Collisions with Data Exchange



(a) Number of Collisions with DHTTestApp Interval= 30 sec

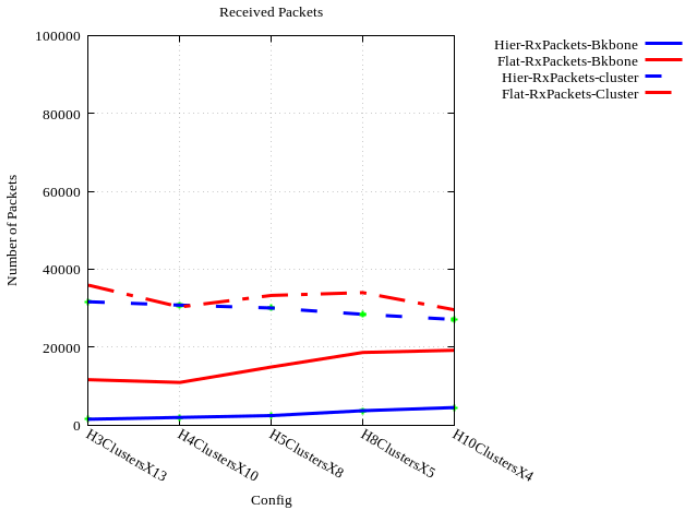


(b) Number of Collisions with DHTTestApp Interval= 15 sec

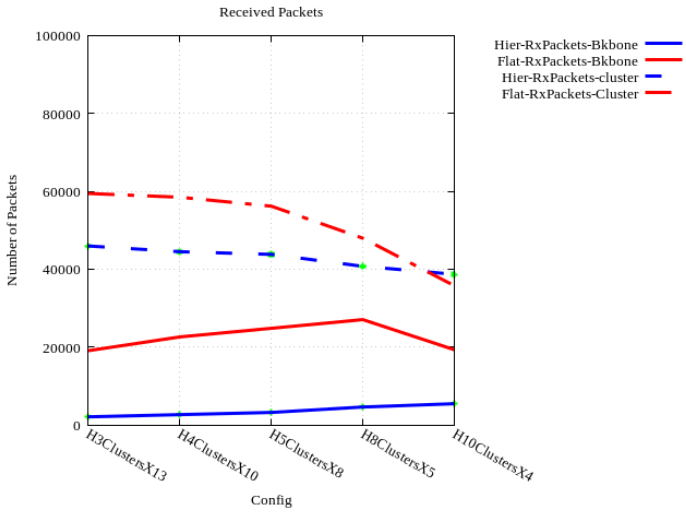


(c) Number of Collisions with DHTTestApp Interval= 10 sec

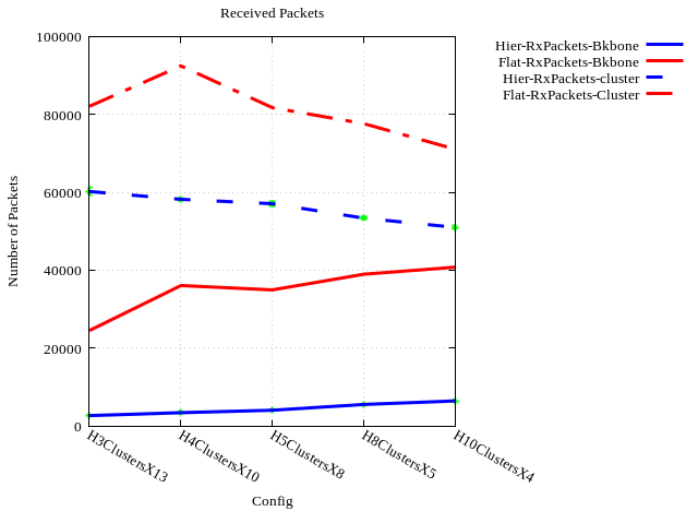
Figure 5.6 Received Packets with Data Exchange



(a) Total Received Packets with DHTTestApp Interval= 30 sec



(b) Total Received Packets with DHTTestApp Interval= 15 sec



(c) Total Received Packets with DHTTestApp Interval= 10 sec

to Figure 5.4, which can be attributed to an increasing portion of traffic migrating to the backbone as the local data traffic probability (GET requests) is reduced with higher number of peers located in external clusters. Also, we can see the link between the increased frequency of queries (higher PUT/GET exchange) and the growing number of collisions in both the backbone network and the local clusters vs. the baseline (i.e. the no-data exchange status). The same applies with the negative trend in number of collisions in local clusters which tends to be Zero as the cluster size becomes small, reducing to almost zero the probability of collision and communication failure within the cluster. The latter trend is a good explanation of the improvement of success ratio and latency as the level of granularity increases (as described in subsection 5.1.1, and 5.1.2).

Figure 5.6 shows the level and distribution of traffic (in packets) in the clusters and the backbone at different test intervals (30, 15 and 10 sec respectively). From the same Figure, we may excerpt the following observations:

- In general, the level of traffic in a flat overlay is higher than a hierarchical overlay for the same number of nodes involved in the data exchange. This significantly reflects the bigger traffic overhead for the flat overlay network vs. the hierarchical overlay structure for the same rate of data exchange.

- As the level of data traffic increases with the test intervals periods getting shorter, an increase of overall data traffic levels is observed (for both local and backbone networks), which should be the normal impact. However, while we see a moderate increase in the traffic level for the hierarchical overlay commensurate to the increase in the frequency of data exchange mechanisms (PUT/ GET requests), the rate of the traffic increase is much higher for the flat overlay. This can be attributed to increased number of packets taking stretched routes to reach their destinations across several hops.

- The traffic level in the hierarchical structure backbone is almost the same in all test intervals. The impact of the data exchange frequency increase is quite moderate in this case given the fact that the actual data volume is not really massive. Besides, it is only that portion of global GET request packets which travel through the backbone. On the other side, the traffic gets significantly higher in the flat overlay configuration backbone network. In the absence of the localization concept, and disproportionate level of traffic traveling through the backbone, the rate of increase in the backbone network traffic (almost 100% increase for 35-50% test period reduction) which suggests a high potential for traffic congestion in the backbone to become a bottleneck. This bottleneck condition will be the reason for communication failures, and performance deterioration for the whole network.

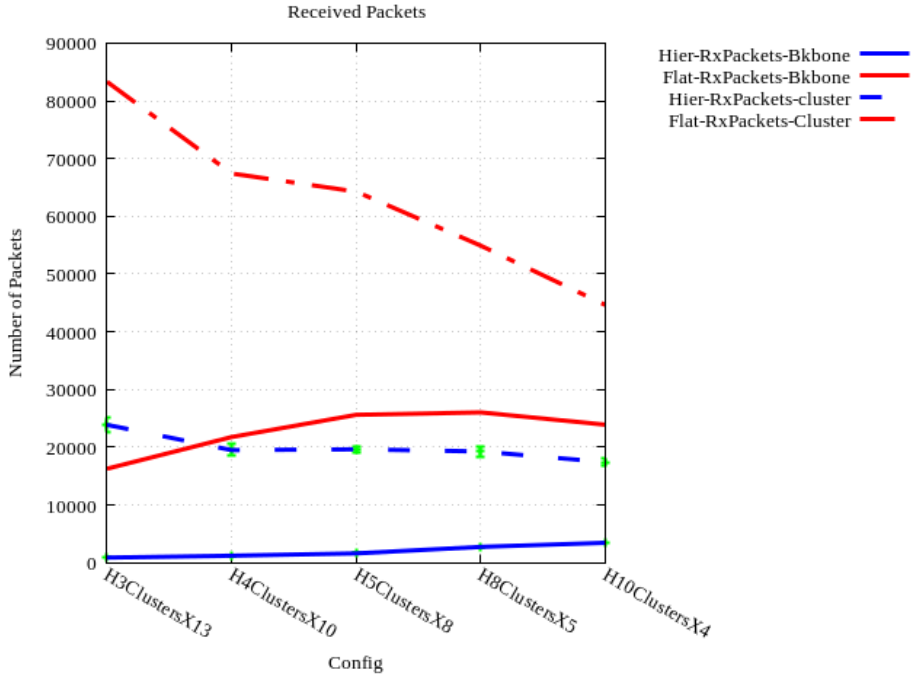
5.2 Performance Evaluation: Multi-Hop Scenarios

As previously described in subsection 4.3.2, the main characteristic of the multi-hop scenarios is the change in the underlay physical topology to allow for some nodes in the underlay to be at multiple-hop distance from the CH(s). Each node is allocated an area in the test space, as the simulation run seed number varies, the nodes location may vary (each within its allocated area) to be at single or multiple hops from the CH(s). On the other side, all CHs joining the backbone remain at one hop distance from each other.

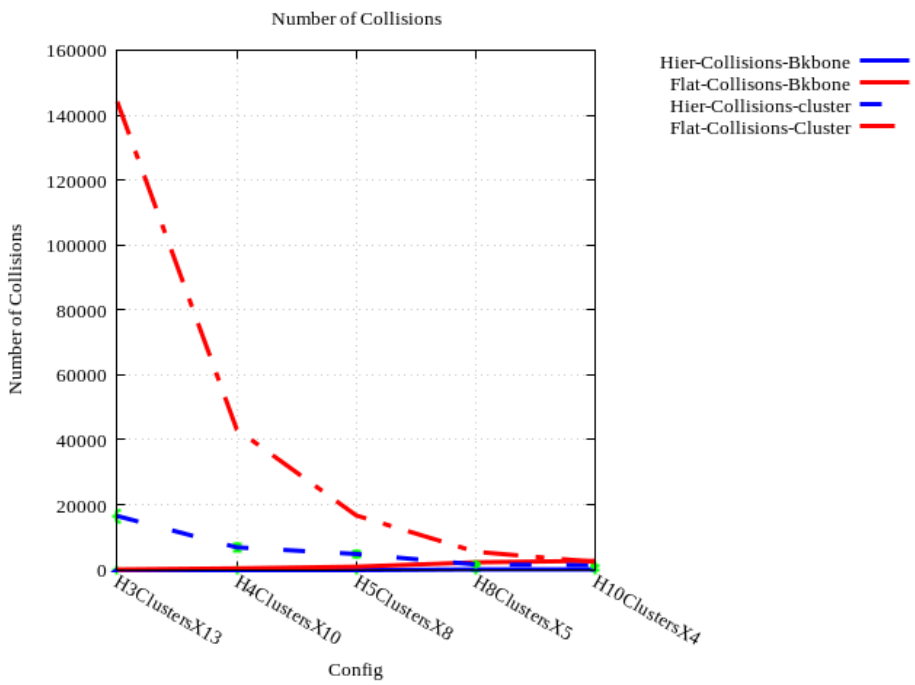
The change in the physical topology to a multi-hop communication between nodes in the same cluster implied the following:

a) The use of TC messages in OLSR for topology update in the local clusters which adds a significant amount of traffic to the local cluster networks.

Figure 5.7 Multi-Hop Configuration - No Data Exchange



(a) Multi-Hop: Total Received Packets with No Data Exchange



(b) Multi-Hop: Total Number of Collisions with No Data Exchange

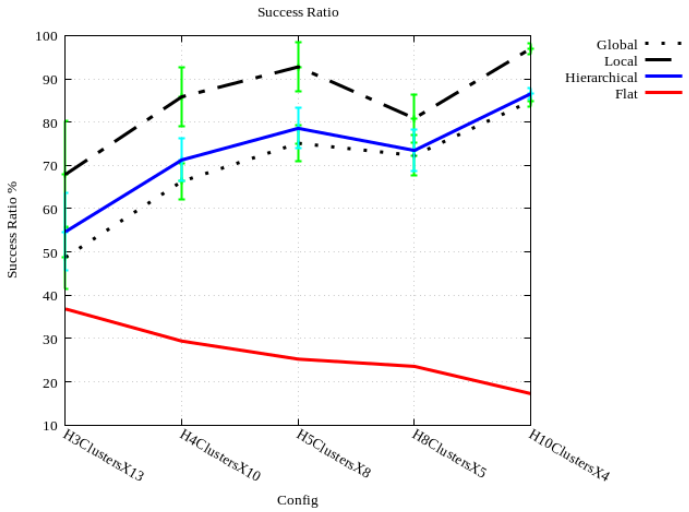
b) The logical adjacent peers in the same overlay subnetwork are subject now to even more stretched physical routes in the underlay. This will add additional traffic to the local cluster networks for both the hierarchical and flat overlay topologies.

Figure 5.7a shows a considerable variation (upside) in the traffic pattern for control and topology update messages vs. the level of traffic shown in Figure 5.3 . In the flat overlay topology, the coarse granularity configurations (small number of clusters and high density of nodes per cluster) have a higher number of nodes located at multiple hops from the CH, hence, they are subject to an increased traffic intensity. As we move to fine granularity, fewer nodes are located at multiple hop distances, approaching almost the same traffic level we observed for the single hop topologies. The change is more obvious for the hierarchical overlay topology, where the traffic level almost doubled from the baseline (single-hop configuration). This increase explains the significant growth in the number of collisions with the multi-hop configuration as shown in Figure 5.7b . The higher number of collisions in the network is one indication of potential deterioration in the network performance metrics.

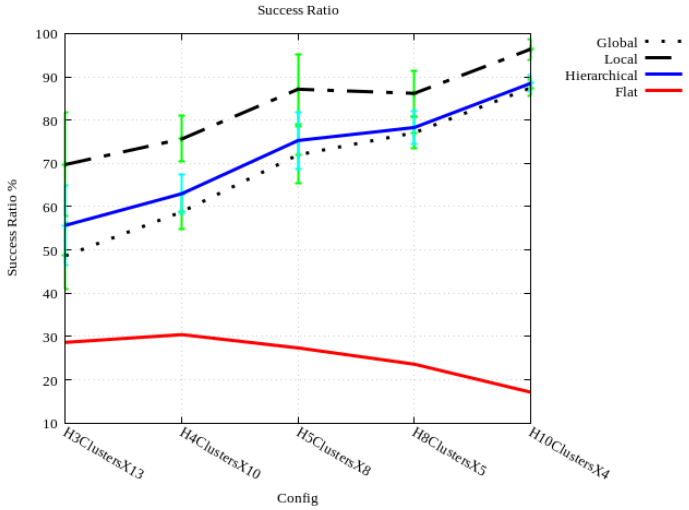
5.2.1 Multi-Hop Scenarios Performance Metrics

In Figure 5.8, there is a drop in the success ratio results specially for the hierarchical overlay with coarse granularity configurations. This can be attributed to the fact that, as the number of nodes in the cluster increases, we move to a flat-like overlay topology. By this analogy we mean a configuration with a high number of nodes interconnected within a network carrying intensive traffic. The traffic is generated by traffic control messages, DHT lookup and update messages stretched over consecutive hops and transfer of packets in the physical route between nodes. As we move to fine granularity, message exchange becomes more localized in a small number of nodes, traffic migrates to the backbone network for global queries, the performance of local lookups improves, followed by an overall performance improvement in global, and overall success ratios. In Figure 5.8a , there is a slight dip in

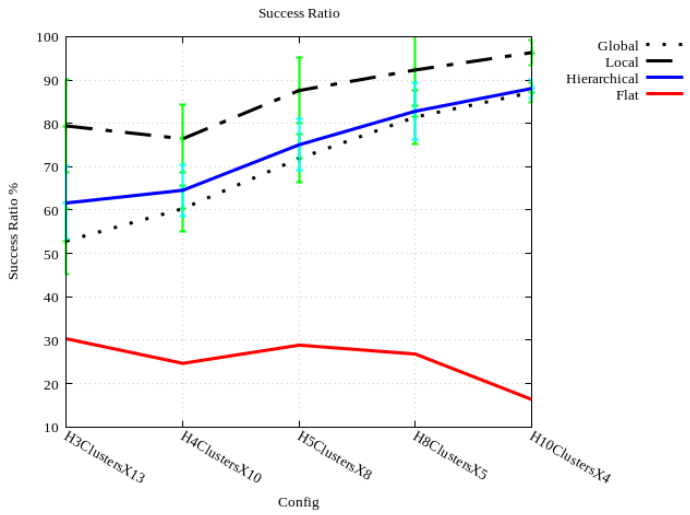
Figure 5.8 Multi-Hop: Success Ratio



(a) Multi-Hop Configuration Success Ratio (Interval= 30sec)



(b) Multi-Hop Configuration Success Ratio (Interval= 15sec)



(c) Multi-Hop Configuration Success Ratio (Interval= 10sec)

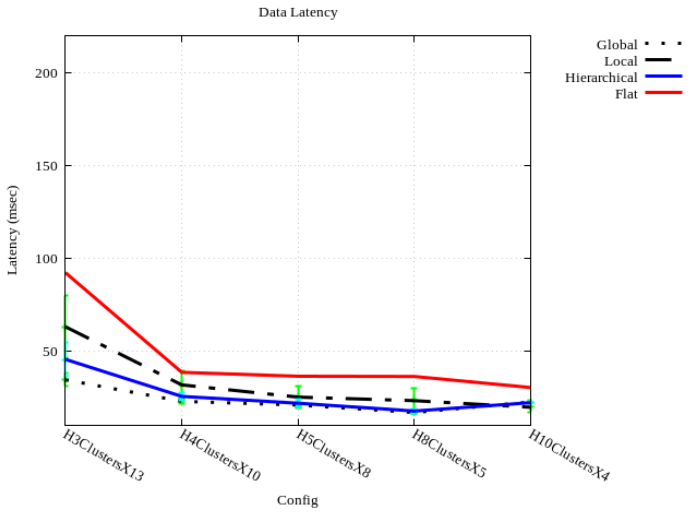
success ratio results for the 8ClustersX5Nodes configuration, which can be partially attributed to the fact that for a small number of nodes in the cluster, the ratio of single-hop to multi-hop nodes in the same cluster significantly varies with the probability of nodes being at single or multiple-hops away. Also, as the number of queries in general is the lowest using this test interval (30 sec), a limited number of unsuccessful queries may have a higher impact on the overall result. Using smaller test intervals, generating a higher number of queries, we can see a smooth trend in the success ratio results (Figure 5.8b , 5.8c). It is also to be mentioned, as the level of granularity increases, and the portion of global queries growing accordingly, the global and overall success ratio trends slightly converge and tend to merge at the highest granularity level.

Figure 5.9 shows a similar trend, yet in this case, the latency for successful queries is decreasing as the level of granularity increases. The same reasons mentioned above can be used to justify the latency trend. However, it should be mentioned the impact of muti-hop communication vs. single-hop is more obvious in the latency results for both the flat and hierarchical coarse granularity configurations. This emphasizes the importance of reducing the number of hops between nodes, shortening the routes and leveraging the physical vicinity between nodes in exchanging messages and communication packages.

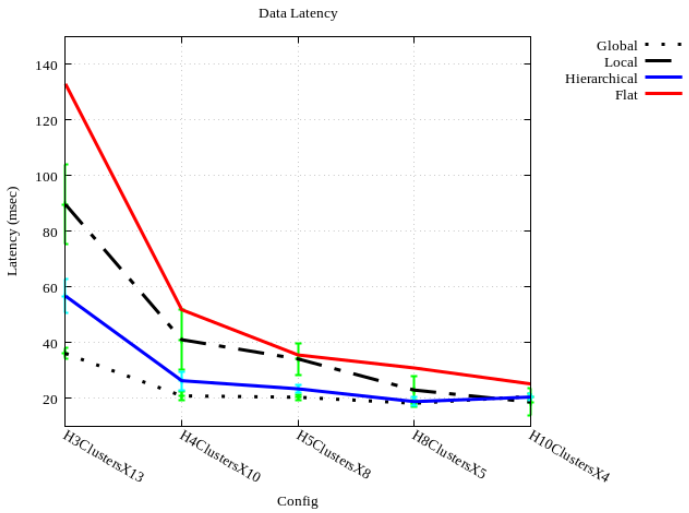
5.3 Mobility Scenarios

As previously mentioned in subsection 4.3.3, we used the RPGM [80, 82] model in our mobility scenarios. The reason was to maintain the same structural formation of the network (same cluster formation) as nodes move within the simulation area. This mobility model is a typical characteristic for tactical and disaster management networks [82]. The mobility scenarios tested can be classified as a high mobility condition, with the nodal speed ranging from 10 m/sec to 20 m/sec. This speed range is similar to the case of VANET (Vehicular

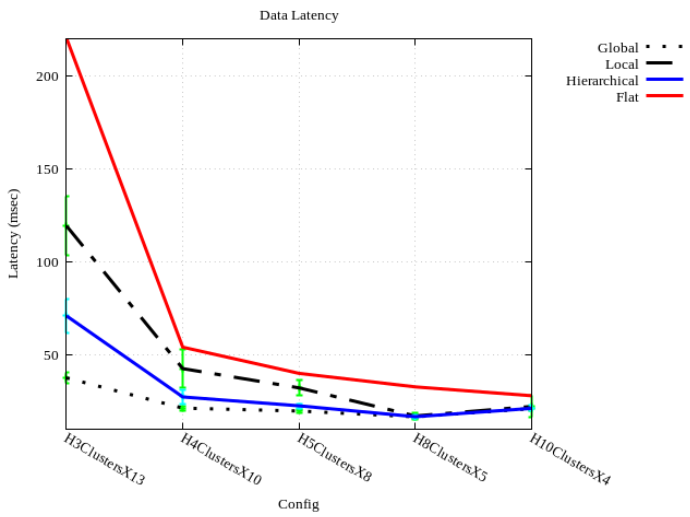
Figure 5.9 Multi-Hop: Latency



(a) Multi-Hop Configuration Latency (Interval= 30 sec)



(b) Multi-Hop Configuration Latency (Interval= 15sec)



(c) Multi-Hop Configuration Latency (Interval= 10sec)

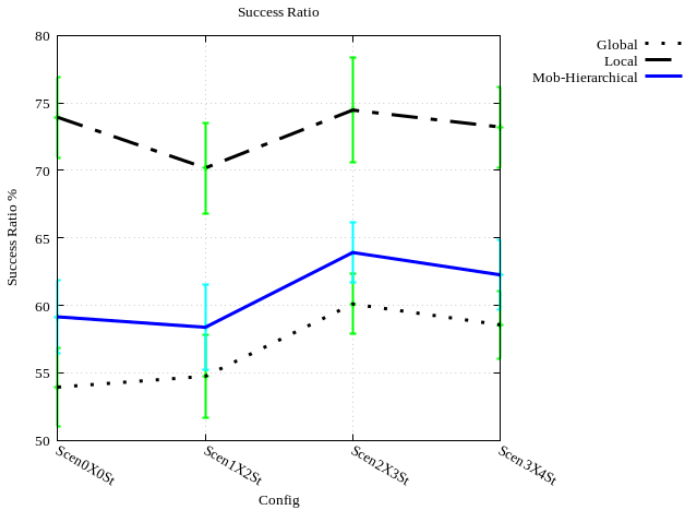
Adhoc Network) nodes moving in an urban zone. Also, there is no pause time set for the nodes, i.e. the nodes are constantly moving throughout the simulation period.

Figure 5.10 shows the success ratio at different test intervals. Despite the fact that the success ratio in the mobility scenario is far less than for Single-hop and Multi-hop scenarios, the hierarchical solution in a mobile scenario still performs significantly better than a flat overlay solution in static conditions. Yet we can also see the impact of mobility on the performance. The overall success ratio drops to a range of $\tilde{6}0\%$, similarly, the global success ratio drops to a range of $\tilde{5}0\%$. To put these results in the right context of our simulation, we need to note the following:

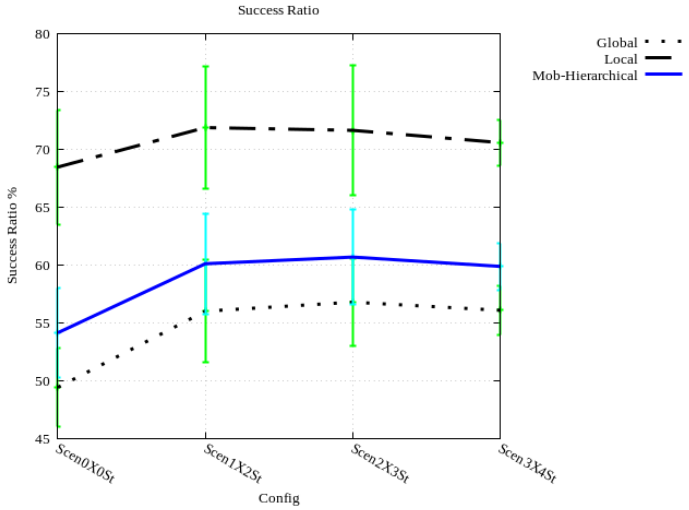
- The global queries are performed in the underlay through routing protocols, namely OLSR for both intra-cluster and backbone routing, hence the performance of the solution is highly depending on the performance of the routing protocols used. Several studies indicated a deterioration in OLSR performance in mobile environments and analyzed the reasons. In [87] Plesse *et al.* analyzed the performance of OLSR on a platform used to simulate military scenarios. The study indicated the presence of periods that range from 2-4sec where the route is lost. This is the period where the old route is not valid and the new route is not yet updated through the TC messages. The same deterioration in performance was indicated in another study to analyze the performance of AODV [19] and OLSR [22] protocols under realistic mobility patterns [88]. In this study, Haerri *et al.* indicated that for OLSR the Packet Delivery Ratio (PDR) in a mobile scenario was in the range of $\tilde{4}0\%$.

- The same applies in the overlay, where Chord is used for both the top layer and intra-cluster DHT lookup and routing. Chord was originally implemented for a wired network environment, which is static by nature. Chord deployment and performance in a wireless mobile environment faces several challenges and results in significant performance degradation. In [62], Liu *et al.* proposed some enhancements to improve

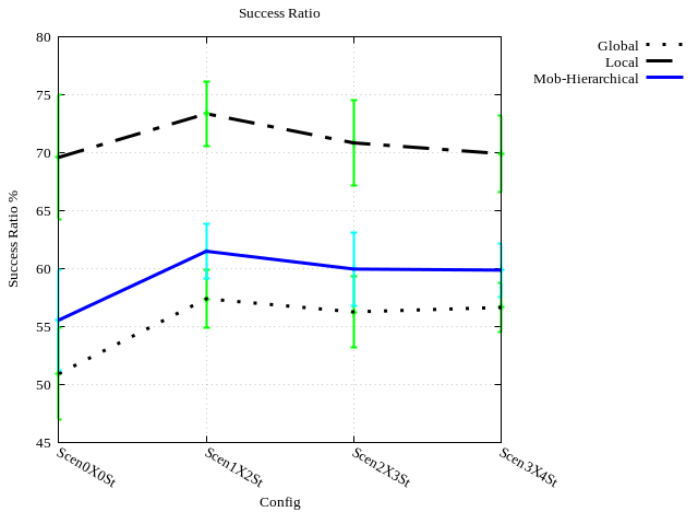
Figure 5.10 Mobility: Success Ratio with Different Intervals



(a) Mobility Scenarios Success Ratio (Interval= 30 sec)



(b) Mobility Scenarios Success Ratio (Interval= 15 sec)



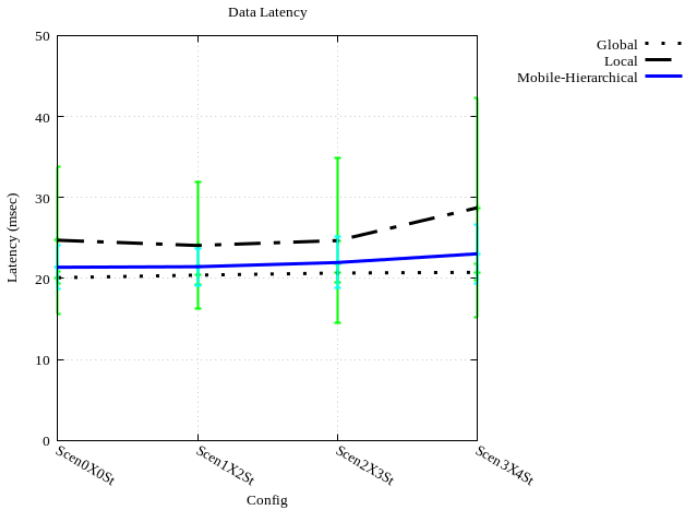
(c) Mobility Scenarios Success Ratio (Interval= 10 sec)

Chord performance in mobile (specifically vehicular mobility) environments. Chord demonstrated poor performance in a mobile environment (simulation was done with a similar configuration in terms of number of nodes in the overlay), correct query responses dropped to $\tilde{5}0\%$ with pedestrian mobility, then to $\tilde{1}0\%$ with vehicular mobility. The enhancements to improve performance included overlay table broadcasting i.o. ping keep alive messages, aggressive table updates, greedy forwarding, and passive bootstrapping. In our simulation, enhancing Chord was not part of the scope of our simulation, the mobility simulation performance results were impacted by the deterioration in Chord performance accordingly.

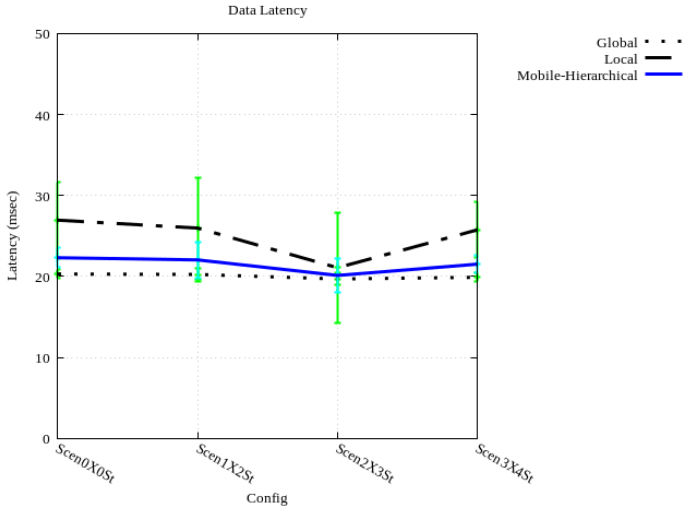
Figure 5.11 shows a consistent performance in response (latency) for the mobile scenarios. This can be attributed to the fact that number of hops is maintained limited and consistent as the configuration is kept constant in all scenarios, this is the reason there is no significant change in latency results vs. multi-hop scenarios. The impact of introducing additional stationary nodes in the backbone to maintain connectivity, which can be compared to the role of RSUs (Road Side Units) in VANETs, did not significantly impact the number of hops in communication exchange. In our scenarios, the mobility is confined within a limited space and which did not imply extensive topology changes and updates in the backbone.

Overall, despite the deterioration in the network performance with the mobile scenarios, the overall network performance metrics under aggressive mobility conditions still outperforms the performance of a flat overly in a static scenario. This deterioration is mainly attributed to the performance of some of the building blocks (DHT lookup and routing protocols) of our simulation. Our scope of work did not include fine tuning some of the parameters used in the simulation protocols to achieve improvements in the network performance. We maintained the same parameters for all scenarios (static single-hop, multi-hop, and mobile scenarios) to have a fair basis for comparison, and to isolate the impact of an arbitrary change in the simulation scenarios from other external factors.

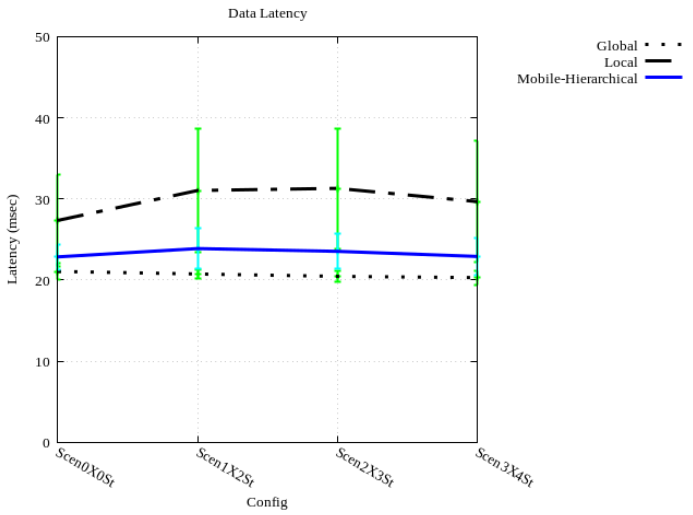
Figure 5.11 Mobility: Latency with Different Test Intervals



(a) Mobility Scenarios Latency (Interval= 30 sec)



(b) Mobility Scenarios Latency (Interval= 15 sec)



(c) Mobility Scenarios Latency (Interval= 10 sec)

Chapter 6

Conclusions

6.1 Thesis Contributions

The deployment of P2P applications on MANETs faces many challenges due to the nature of MANETs, and the requirements for a successful P2P deployment. MANETs, which are distributed by nature, have limitations on bandwidth, limited battery lifetime, and are characterized by dynamic topology changes due to the mobility of the nodes. Such limitations, constraints and specific nature of MANET yield to a significant performance degradation mainly in terms of resilience and responsiveness. A successful P2P deployment on MANET implies specific design requirements to limit the overhead traffic due to maintenance, topology control and update messages, reduces the mismatch between logical and physical message routing, while being able to adapt to frequent topology changes due to mobility in a timely and efficient manner. To deliver such requirements, many design approaches were used, including the use of topology aware adaptive protocols to generate and maintain a logical overlay topology with neighboring nodes having physical proximity in the underlay. We reviewed in this work different deployment approaches and shared the benefits and limitations for each.

A special type of MANETs exists, the Hierarchical MANET, where the underlay is grouped into subnets (or clusters) of nodes, and these clusters of nodes are interconnected by a backbone network to allow for inter-cluster communication. Whereas the deployment of P2P overlay on a MANET with a simple flat underlay topology is challenging by nature, the successful deployment of P2P applications on a hierarchical underlay topology brings a new dimension to the challenge. The hierarchical underlay topology adds an additional constraint by limiting the visibility and extent of the routing tables owned by the nodes in the underlay to the boundary of the cluster they belong to. In this case, the concept of bringing topology awareness to the overlay, in light of this limited visibility of the routing tables, becomes a new challenge.

In our proposal to deploy P2P applications on hierarchical MANETs, we provided a novel approach to bring the topology awareness to the logical overlay, and maintaining logical vicinity in the overlay between adjacent nodes in the underlay. The solution is based on mapping the underlay hierarchical topology to a hierarchical logical overlay, and having the hierarchy of the nodes in the underlay (CH/GW and nodes) maintained for the same nodes in the overlay (Superpeer and peers). Using this approach, we were able to match physical and logical proximity between nodes, and minimize the route stretch between logical and physical routes, hence, reduce the overhead traffic and improve the network efficiency. The solution was tested with different static and mobility scenarios using OMNeT++. All test results demonstrated that the hierarchical overlay significantly outperformed a flat logical overlay deployed on the same underlay topology in terms of overall queries success ratio and response time.

6.2 Future Work

In our work, the static single-hop scenarios demonstrated best performance results, mainly as the route stretch for a single-hop configuration is minimal. It also minimizes the negative

impact due to the routing protocols traffic overhead and efficiency on PDR and latency. As we moved to multi-hop and mobile scenarios, the impact of these factors resulted in a certain performance degradation. We would strongly recommend the implementation of more efficient DHT routing protocols in the overlay. OneHopOverlay4MANET [2], which we already described in subsection 2.3.2, is a strong candidate for implementation, as it aims for one-hop distance between peers in the overlay. This should imply significant improvement vs. Chord, specially with MANETs of coarse granularity (large cluster size) when deployed in the local clusters overlay, or with MANETs having a high number of CHs participating in the backbone. Besides, in order to improve routing efficiency and responsiveness in the underlay, we may recommend the use of other routing protocols which may be better suited for the hierarchical underlay structure. Although there is no obvious answer on what would be the best suited routing protocol (in terms of overhead generated and latency), yet experimenting with different configurations using different routing protocols whether proactive (GSR [89], ...*etc.*), hierarchical (CGSR [17], CEDAR [8], ...*etc.*), or reactive (AODV [19], ...*etc.*) may lead to interesting findings and conclusions on the best fit for specific configurations, application types, or the combination of both.

As mentioned in subsection 4.2.4, there was a limitation on the CH module design used in the flat overlay implementation. This design limitation inhibited the deployment of another P2P application in the overlay module of the flat network CHs. In turn, this implied the use of an additional node to replace the CH in the flat overlay network to maintain the same total number of peers participating in the overlay P2P data exchange as the case for the hierarchical overlay network. We followed a conservative strategy in our research, hence we maintained the integrity of the modules used in the hierarchical routing solution which was adapted to build the flat overlay network. However, we may recommend the use of a MultiOverlay module in the CHs, to allow the deployment of another P2P application to the

CH and experiment the full participation of the CHs in the data exchange between peers in the overlay.

Besides, our solution, like similar hierarchical structures, may endure severe performance degradation with CH failures and full cluster(s) sudden separation from the network. Whereas this is a common disadvantage for hierarchical configurations, it would be a significant improvement in reliability if a redundancy scheme is implemented at the cluster level. Redundancy is already a common solution to restore data in case of node failures or churn in several DHT routing protocols (*e.g.* Chord [6]). Devising a suitable scheme to backup data at the cluster level is an area to explore. However, this should be carefully analyzed and evaluated in terms of additional overhead, complexity and processing delays that may result from such scheme.

Bibliography

[1] Xianghan Zheng and Vladimir Oleshchuk. A Survey on Peer-to-Peer SIP Based Communication Systems. *Peer-to-peer networking and applications*, 3(4):257–264, 2010.

[2] Mohammad Al Mojamed and Mario Kolberg. Design and Evaluation of a Peer-to-Peer MANET Crosslayer Approach: OneHopOverlay4MANET. *Peer-to-Peer Networking and Applications*, 10(1):138–155, 2017.

[3] Zhonghong Ou. *Structured Peer-to-Peer Networks: Hierarchical Architecture and Performance Evaluation*. PhD thesis, University of Oulu, Faculty of Technology, Department of Electrical and Information Engineering, 2010.

[4] Matei Ripeanu. Peer-to-Peer Architecture Case Study: Gnutella Network. In *Peer-to-Peer Computing, 2001. Proceedings. First International Conference on*, pages 99–100. IEEE, 2001.

[5] Mehdi Asadi, Abolfazl Toroghi Haghighat, and Bager Zarei. A Scalable Lookup Service for P2P File Sharing in MANET. In *Proceedings of the 2007 international conference on Wireless communications and mobile computing*, pages 511–516. ACM, 2007.

[6] Ion Stoica, Robert Morris, David Liben-Nowell, David R Karger, M Frans Kaashoek, Frank Dabek, and Hari Balakrishnan. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. *IEEE/ACM Transactions on Networking (TON)*, 11(1):17–32, 2003.

[7] Antony Rowstron and Peter Druschel. Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In *IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing*, pages 329–350. Springer, 2001.

[8] Shahbaz Akhtar Abid, Mazliza Othman, and Nadir Shah. A Survey on DHT-based Routing for Large Scale Mobile Ad hoc Networks. *ACM Computing Surveys (CSUR)*, 47(2):20, 2015.

[9] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. *A Scalable Content-Addressable Network*, volume 31. ACM, 2001.

[10] Ben Y Zhao, Ling Huang, Jeremy Stribling, Sean C Rhea, Anthony D Joseph, and John D Kubiatowicz. Tapestry: A Resilient Global-scale Overlay for Service Deployment. *IEEE Journal on selected areas in communications*, 22(1):41–53, 2004.

[11] Mohammad Al Mojamed and Mario Kolberg. Structured Peer-to-Peer Overlay Deployment on MANET: A Survey. *Computer Networks*, 96:29–47, 2016.

[12] Marco Conti and Silvia Giordano. Mobile Ad Hoc Networking: Milestones, Challenges, and New Research Directions. *IEEE Communications Magazine*, 52(1):85–96, 2014.

[13] Izegbuwa Okundaye, Thomas Kunz, and Semra Gulder. Inter-domain routing for tactical mobile ad-hoc networks. In *Vehicular Technology Conference (VTC Fall), 2014 IEEE 80th*, pages 1–6. IEEE, 2014.

[14] Irfan Al-Anbagi, Melike Erol-Kantarci, and Hussein T Mouftah. A Survey on Cross-Layer Quality-of-Service Approaches in WSNs for Delay and Reliability-Aware Applications. *IEEE Communications Surveys & Tutorials*, 18(1):525–552, 2016.

[15] Mario Bisignano, Giuseppe Di Modica, Orazio Tomarchio, and Lorenzo Vita. P2P over MANET: a Comparison of Cross-layer Approaches. In *Database and Expert Systems Applications, 2007. DEXA'07. 18th International Workshop on*, pages 814–818. IEEE, 2007.

[16] Marcel C Castro, Andreas J Kassler, Carla-Fabiana Chiasserini, Claudio Casetti, and Ibrahim Korpeoglu. Peer-to-Peer Overlay in Mobile Ad-hoc Networks. In *Handbook of Peer-to-Peer networking*, pages 1045–1080. Springer, 2010.

[17] Mehran Abolhasan, Tadeusz Wysocki, and Eryk Dutkiewicz. A Review of Routing Protocols for Mobile Ad hoc Networks. *Ad hoc networks*, 2(1):1–22, 2004.

[18] R Shenbagapriya and N Kumar. A Survey on Proactive Routing Protocols in MANETs. In *Science Engineering and Management Research (ICSEMR), 2014 International Conference on*, pages 1–7. IEEE, 2014.

[19] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. IETF, RFC 3561, July 2003. [Online] Available: https://www.rfc-editor.org/info/rfc3561.

[20] D. Johnson, Y. Hu, and D. Maltz. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. IETF, RFC 4728, February 2007. [Online] Available: https://www.rfc-editor.org/info/rfc4728.

[21] Daxesh N Patel, Sejal B Patel, Hemangi R Kothadiya, Pinakin D Jethwa, and Rutvij H Jhaveri. A Survey of Reactive Routing Protocols in MANET. In *Information Communication and Embedded Systems (ICICES), 2014 International Conference on*, pages 1–6. IEEE, 2014.

[22] Ed. Clausen, T. and Ed. P. Jacquet. Optimized Link State Routing Protocol (OLSR). IETF, RFC 3626, October 2003. [Online] Available: https://www.rfc-editor.org/info/rfc3626.

[23] Guoyou He. Destination-Sequenced Distance Vector (DSDV) Protocol. *Networking Laboratory, Helsinki University of Technology*, pages 1–9, 2002.

[24] Abdalkarim Awad, Reinhard German, and Falko Dressler. Exploiting Virtual Coordinates for Improved Routing Performance in Sensor Networks. *IEEE Transactions on Mobile Computing*, 10(9):1214–1226, 2011.

[25] Romualdo Pastor-Satorras and Alessandro Vespignani. *Evolution and Structure of the Internet: A Statistical Physics Approach*. Cambridge University Press, 2007.

[26] Jun Liu, Xiaoyan Hong, Jiejun Kong, Qunwei Zheng, Ning Hu, and Phillip G Bradford. A Hierarchical Anonymous Routing Scheme for Mobile Ad-hoc Networks. In *Military Communications Conference, 2006. MILCOM 2006. IEEE*, pages 1–7. IEEE, 2006.

[27] Jack L Burbank, Philip F Chimento, Brian K Haberman, and William T Kasch. Key Challenges of Military Tactical Networking and the Elusive Promise of MANET Technology. *IEEE Communications Magazine*, 44(11):39–45, 2006.

[28] Lei Chen and Wendi B Heinzelman. A Survey of Routing Protocols that Support QoS in Mobile Ad Hoc Networks. *IEEE Network*, 21(6), 2007.

[29] Shahbaz Akhtar Abid, Mazliza Othman, Nadir Shah, Mazhar Ali, and AR Khan. 3D-RP: A DHT-based Routing Protocol for MANETs. *The Computer Journal*, 58(2):258–279, 2014.

[30] Shahbaz Akhtar Abid, Mazliza Othman, and Nadir Shah. 3D P2P Overlay over MANETs. *Computer Networks*, 64:89–111, 2014.

[31] Thomas Kunz, Babak Esfandiari, and Frank Ockenfeld. Efficient Routing in Mobile Ad-Hoc Social Networks. In *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 216–222. IEEE, 2017.

[32] Waleed Alsalih, Selim Akl, and H Hassancin. Energy-Aware Task Scheduling: Towards Enabling Mobile Computing over MANETs. In *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*, pages 8–pp. IEEE, 2005.

[33] Boangoat Jarupan and Eylem Ekici. A Survey of Cross-layer Design for VANETs. *Ad Hoc Networks*, 9(5):966–983, 2011.

[34] Luis Garces-Erice, Ernst W Biersack, Keith W Ross, Pascal A Felber, and Guillaume Urvoy-Keller. Hierarchical Peer-to-Peer Systems. *Parallel Processing Letters*, 13(04):643–657, 2003.

[35] Isaias Martinez-Yelmo, Roberto Gonzalez, and Carmen Guerrero. Benefits of an Implementation of H-P2PSIP. 2010.

[36] Marcel C Castro, Eva Villanueva, Iraide Ruiz, Susana Sargento, and Andreas J Kassler. Performance Evaluation of Structured P2P over Wireless Multi-Hop Networks. In *Sensor Technologies and Applications, 2008. SENSORCOMM'08. Second International Conference on*, pages 796–801. IEEE, 2008.

[37] Sean Rhea, Dennis Geels, Timothy Roscoe, John Kubiatowicz, et al. Handling Churn in a DHT. In *Proceedings of the USENIX Annual Technical Conference*, volume 6, pages 127–140. Boston, MA, USA, 2004.

[38] Ian D Chakeres and Elizabeth M Belding-Royer. AODV Routing Protocol Implementation Design. In *Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Conference on*, pages 698–703. IEEE, 2004.

[39] Jing-qing MEI, JI Hong, and LI Ting. Cross-layer Optimized Chord Protocol for Separated Ring Convergence in MANET. *The Journal of China Universities of Posts and Telecommunications*, 16(4):84–90, 2009.

[40] Manel Seddiki and Mahfoud Benchaïba. An Adaptive P2P Overlay for MANETs. In *Proceedings of the 2015 International Conference on Distributed Computing and Networking*, page 24. ACM, 2015.

[41] Kang Chen, Haiying Shen, and Haibo Zhang. Leveraging Social Networks for P2P Content-based File Sharing in Disconnected MANETs. *IEEE Transactions on Mobile Computing*, 13(2):235–249, 2014.

[42] Daniel F Macedo, Aldri L dos Santos, Guy Pujolle, and José Marcos S Nogueira. MANKOP: A Knowledge Plane for Wireless Ad hoc Networks. In *Network Operations and Management Symposium, 2008. NOMS 2008. IEEE*, pages 706–709. IEEE, 2008.

[43] Daniel F Macedo, Aldri L Dos Santos, José Marcos S Nogueira, and Guy Pujolle. A Distributed Information Repository for Autonomic Context-Aware MANETs. *IEEE Transactions on Network and Service Management*, 6(1):45–55, 2009.

[44] Ben Leong, Barbara Liskov, and Erik D Demaine. Epichord: Parallelizing the Chord Lookup Algorithm with Reactive Routing State Management. *Computer Communications*, 29(9):1243–1259, 2006.

[45] Himabindu Pucha, Saumitra M Das, and Y Charlie Hu. Ekta: An Efficient DHT Substrate for Distributed Applications in Mobile Ad Hoc Networks. In *Mobile Computing Systems and Applications, 2004. WMCSA 2004. Sixth IEEE Workshop on*, pages 163–173. IEEE, 2004.

[46] David B Johnson, David A Maltz, Josh Broch, et al. DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks. *Ad hoc networking*, 5:139–172, 2001.

[47] Thomas Zahn and Jochen Schiller. MADPastry: A DHT Substrate for Practicably Sized MANETs. In *Proc. of ASWN*, 2005.

[48] Guangyu Pei, Mario Gerla, Xiaoyan Hong, and C-C Chiang. Wireless Hierarchical Routing Protocol with Group Mobility (WHIRL). In *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, volume 3, pages 1538–1542. IEEE, 1999.

[49] Anthony Yu and Son T Vuong. A DHT-based Hierarchical Overlay for Peer-to-Peer MMOGs over MANETs. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*, pages 1475–1480. IEEE, 2011.

[50] Rolf Winter, Thomas Zahn, and Jochen Schiller. Random Landmarking in Mobile, Topology-Aware Peer-to-Peer Networks. In *Distributed Computing Systems, 2004. FTDCS 2004. Proceedings. 10th IEEE International Workshop on Future Trends of*, pages 319–324. IEEE, 2004.

[51] Jerzy Dołowski and Marek Amanowicz. An Application of Chord Structure in Tactical Ad hoc Network. In *Communications and Information Systems Conference (MCC), 2012 Military*, pages 1–5. IEEE, 2012.

[52] Yongchun Tian and Wei Guo. A Mobility Management Scheme for Hierarchical Multi-hop Mobile Wireless Networks. In *Parallel and Distributed Computing, Applications and Technologies, 2003. PDCAT'2003. Proceedings of the Fourth International Conference on*, pages 400–403. IEEE, 2003.

[53] Oleksii Zghurskyi and Sergey Bunin. A Survey of Clustering Protocols for MANET with Weighted Metric for Cluster Head Selection. In *Problems of Infocommunications Science and Technology, 2014 First International Scientific-Practical Conference*, pages 54–56. IEEE, 2014.

[54] Sun-Kyum Kim, Ji-Hyeun Yoon, Junyeop Lee, and Sung-Bong Yang. HCS: Hierarchical Cluster-based Forwarding Scheme for Mobile Social Networks. *Wireless Networks*, 21(5):1699–1711, 2015.

[55] Muhammad Arshad, Mohamad Y Aalsalem, and Farhan A Siddiqui. Energy Efficient Cluster Head Selection in Mobile Wireless Sensor Networks. *Journal of Engineering Science and Technology*, 9(6):728–746, 2014.

[56] Muluneh M Tulu, Ronghui Hou, Chuanyin Li, and Melkamu D Amentie. Cluster Head Selection Method for Content-Centric Mobile Social Network in 5G. *IET Communications*, 12(4):402–408, 2017.

[57] Nadir Shah, Depei Qian, and Rui Wang. MANET Adaptive Structured P2P Overlay. *Peer-to-Peer Networking and Applications*, 5(2):143–160, 2012.

[58] Matthew Caesar, Miguel Castro, Edmund B Nightingale, Greg O'Shea, and Antony Rowstron. Virtual Ring Routing: Network Routing Inspired by DHTs. In *ACM SIGCOMM Computer Communication Review*, volume 36, pages 351–362. ACM, 2006.

[59] Gang Peng, Shanping Li, Hairong Jin, and Tianchi Ma. M-CAN: A Lookup Protocol for Mobile Peer-to-Peer Environment. In *Parallel Architectures, Algorithms and Networks, 2004. Proceedings. 7th International Symposium on*, pages 544–549. IEEE, 2004.

[60] Sangheon Pack, Kunwoo Park, Taekyoung Kwon, and Yanghee Choi. SAMP: Scalable Application-Layer Mobility Protocol. *IEEE Communications Magazine*, 44(6):86–92, 2006.

[61] Raphaël Kummer, Peter Kropf, and Pascal Felber. Distributed Lookup in Structured Peer-to-Peer Ad-hoc Networks. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 1541–1554. Springer, 2006.

[62] Che-Liang Liu, Chih-Yu Wang, and Hung-Yu Wei. Mobile chord: Enhancing P2P Application Performance over Vehicular Ad hoc Network. In *2008 IEEE Globecom Workshops*, pages 1–8. IEEE, 2008.

[63] András Varga. OMNeT++ Simulation Manual, January 2019. [Online] Available: https://doc.omnetpp.org/.

[64] András Varga. Using the OMNeT++ Discrete Event Simulation System in Education. *IEEE Transactions on Education*, 42(4):11–pp, 1999.

[65] András Varga and Rudolf Hornig. An Overview of the OMNeT++ Simulation Environment. In *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, page 60. ICST (Institute for Computer Sciences, Social-Informatics, 2008.

[66] INET Framework User's Guide, January 2019. [Online] Available: https://inet.omnetpp.org/.

[67] Jon Postel et al. RFC 791: Internet Protocol, 1981. [Online] Available: https://tools.ietf.org/html/rfc791.

[68] P. Calhoun, M. Montemurro, and D. Stanley. Control and Provisioning of Wireless Access Points (CAPWAP) Protocol Binding for IEEE 802.11. IETF, RFC 5416, March 2009. [Online] Available: https://tools.ietf.org/html/rfc5416.

[69] OverSim The Overlay Simulation Framework, January 2019. [Online] Available: https://inet.omnetpp.org/.

[70] Yatin Chawathe, Sylvia Ratnasamy, Lee Breslau, Nick Lanham, and Scott Shenker. Making Gnutella-like P2P Systems Scalable. In *Proceedings of the 2003 conference on*

Applications, technologies, architectures, and protocols for computer communications, pages 407–418. ACM, 2003.

[71] M Frans Kaashoek and David R Karger. Koorde: A Simple Degree-Optimal Distributed Hash Table. In *International Workshop on Peer-to-Peer Systems*, pages 98–107. Springer, 2003.

[72] Petar Maymounkov and David Mazieres. Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. In *International Workshop on Peer-to-Peer Systems*, pages 53–65. Springer, 2002.

[73] Ingmar Baumgart, Bernhard Heep, and Stephan Krause. OverSim: A Flexible Overlay Network Simulation Framework. In *IEEE global internet symposium*, volume 2007, pages 79–84, 2007.

[74] Ngozi Echegini, Thomas Kunz, and Babak Esfandiari. Evaluating Chord Over a Hierarchical MANET. In *Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2017 8th IEEE Annual*, pages 608–617. IEEE, 2017.

[75] Ngozi Echegini. A DHT-Based Routing Solution for Hierarchical MANETs. Master's thesis, Carleton University, 2018.

[76] Josyl Mariela B Rocamora and Jhoanna Rhodette I Pedrasa. Evaluation of Hierarchical DHTs to Mitigate Churn Effects in Mobile Networks. *Computer Communications*, 85:41–57, 2016.

[77] R. Braden. RFC 1122: Requirements for Internet Hosts – Communication Layers, 1989. [Online] Available: https://tools.ietf.org/html/rfc1122.

[78] Saira Andleeb Gillani, Peer Azmat Shah, Amir Qayyum, and Halabi B Hasbullah. MAC Layer Challenges and Proposed Protocols for Vehicular Ad-hoc Networks. In *Vehicular ad-hoc networks for smart cities*, pages 3–13. Springer, 2015.

[79] Soon-Young Oh, Davide Lau, and Mario Gerla. Content Centric Networking in Tactical and Emergency MANETs. *Wireless days*, 10, 2010.

[80] Tracy Camp, Jeff Boleng, and Vanessa Davies. A Survey of Mobility Models for Ad hoc Network Research. *Wireless communications and mobile computing*, 2(5):483–502, 2002.

[81] Xiaoyan Hong, Mario Gerla, Guangyu Pei, and Ching-Chuan Chiang. A Group Mobility Model for Ad Hoc Wireless Networks. In *Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, volume 53, page 60. Citeseer, 1999.

[82] Fan Bai and Ahmed Helmy. A Survey of Mobility Models in Wireless Adhoc Networks. *Wireless Adhoc Networks. University of Southern California, USA*, 206:147, 2004.

[83] Farida Chowdhury and Mario Kolberg. Performance Evaluation of EpiChord under High Churn. In *Proceedings of the 8th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*, pages 29–36. ACM, 2013.

[84] Joseph Macker. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. 1999.

[85] Leonardo B Oliveira, Isabela G Siqueira, Daniel F Macedo, Antonio Alfredo Ferreira Loureiro, Hao Chi Wong, and Jose M Nogueira. Evaluation of Peer-to-Peer Network Content Discovery Techniques Over Mobile Ad hoc Networks. In *Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*, pages 51–56. IEEE, 2005.

[86] Kumar Viswanath, Katia Obraczka, and Gene Tsudik. Exploring Mesh and Tree-based Multicast Routing Protocols for MANETs. *IEEE Transactions on mobile computing*, 5(1):28–42, 2006.

[87] Thierry Plesse, Cedric Adjih, Pascale Minet, Anis Laouiti, Adokoé Plakoo, Marc Badel, Paul Muhlethaler, Philippe Jacquet, and Jérôme Lecomte. OLSR Performance Measurement in a Military Mobile Ad hoc Network. *Ad Hoc Networks*, 3(5):575–588, 2005.

[88] Jerome Haerri, Fethi Filali, and Christian Bonnet. Performance Comparison of AODV and OLSR in VANETs Urban Environments under Realistic Mobility Patterns. In *Proceedings of the 5th IFIP mediterranean ad-hoc networking workshop*, number i, pages 14–17, 2006.

[89] Tsu-Wei Chen and Mario Gerla. Global State Routing: A New Routing Scheme for Ad-hoc Wireless Networks. In *ICC '98. 1998 IEEE International Conference on Communications. Conference Record. Affiliated with SUPERCOMM'98 (Cat. No.98CH36220)*, volume 1, pages 171–175 vol.1, June 1998.