Scalable Distributed Hash Table Based Routing with Distributed Partition Detection and Merging in Mobile Ad Hoc Network

Ph.D Thesis

By

Saleem Zahid
85-FBAS/PHDCS/F12

Supervisor

Dr. Nadir Shah
COMSATS University (CUI-Wah).

Co-Supervisor

Dr. Syed Husnain A.N
DCS & SE, FBAS, IIUI.

Department of Computer Science & Software Engineering
Faculty of Basic & Applied Sciences
International Islamic University, Islamabad
2019

A dissertation submitted to the
Department of Computer Science & Software Engineering,
International Islamic University, Islamabad
as a partial fulfillment of the requirements
for the award of the degree of
Doctor of Philosophy in Computer Science.

Department of Computer Science & Software Engineering, International Islamic University, Islamabad

Date: July 6, 2019

Final Approval

It is certified that we have examined the thesis report submitted by Mr. Saleem Zahid, Registration No. 85-FBAS/PhD(CS)/F12, and it is our judgment that this thesis is of sufficient standard to warrant its acceptance by the International Islamic University, Islamabad for the Doctor of Philosophy in Computer Science.

Committee:

Declaration

I hereby declare that this thesis, neither as a whole nor as a part thereof has been copied out from any source. It is further declared that no portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Saleem Zahid

Dedication

Dedicated to My family, especially to my parents and brothers, my wife and my daughter *Hoorain*.

Saleem Zahid

Acknowledgments

I am very grateful to *ALLAH* the *ALMIGHTY* for without His grace and blessing this study would not have been possible.

Foremost, I would like to express my sincere gratitude to my supervisor *Dr. Nadir Shah* for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.

I would like to extend immeasurable appreciation and deepest gratitude for the help and support of my Co-Supervisor *Dr. Husnain Abbas Naqvi*, who gave all his knowledge, guidance and support to boost my confidence and learning. His mentoring and encouragement have been specially valuable, and his early insights launched the greater part of this dissertation.

It is my pleasure to thank Dr. Nadir Sha for introducing and motivating me to work in the field of large scale routing in MANETs. I am extremely thankful to Dr. Shahbaz akhtar abid for his sincere and constant support, guidance and contributions throughout my research.

I am thankful to Dr. Ayyaz Hussain (Chairman DCSE) and Dr.Anwar Ghani for their valuable suggestions and sincere support during my study.

I am extremely grateful to Higher Education Commission (HEC) of Pakistan for the financial support throughout my Master and PhD degree.This work is completed under the 5000-Indigenous PhD fellowship program.

Last but not least I wish to avail this opportunity, to express a sense of gratitude and love to my beloved parents, brothers for their moral support, strength, help and for everything.

I would also like to thank my wife who has supported me patiently and firmly during completion of this task.

Abstract

Distributed Hash Table (DHT)-based routing in Mobile Ad Hoc Networks (MANETs) completely eliminates flooding at the control plane as well as at the data plane, thus makes the network scalable. In DHT-based MANETs, a logical structured network is built over the ad hoc physical topology in a fully distributed manner and routing is performed using logical identifiers (i.e. transient addresses) rather using IP addresses.

This thesis investigates the DHT-based routing paradigm in MANETs and points out that existing state-of-the-art DHT routing protocols assume ideal network environments and, ignore the adversarial environment offered by MANETs. Limited radio range, mobility, lack of infrastructure and decentralized nature introduce frequent and unpredictable changes to network topology in MANETs (i.e. connectivity/dis-connectivity, node(s)/link(s) failure, network partition, frequent merging). The network dynamics severely damage the logical structured network (i.e. the logical space (LS) distributed among the nodes) and completely halt communication. Specifically, existing work fails to address issues such as node(s)/link(s) failure and its impact (i.e. anchor node failure and lookups), network partitioning, lost LS recovery and reusing (i.e. disrupted LS) and merging considerations. Curtailing the information loss due to the network dynamics is imperative for the successful communication in DHT networks. Similarly, the key factor that defines overall routing performance in DHT networks is the successful resolution of lookup requests with minimum possible delay. However, we found that existing DHT protocols suffer from longer delay and, fail to ensure the successful resolution of the lookup requests. Therefore, effective distributed solutions under the scalability constraints are needed to tolerate the faults in the logical network and to provide end-to-end connectivity in such an adversarial environment. It is worthy to mention that the targeted problems are completely unexplored and had never been addressed by the research community. For the first time, we are exploring the problems and providing solutions under the constraints.

The first part of our work explores the impact of network dynamics on the intrinsics of DHT routing (i.e. lookup requests and successful resolution with minimum delay). A novel address publication mechanism, also called Anchor_Request, is proposed. The mechanisms exploit k-hop topological information to detect critical regions in the network and replicate the index information (stored on the anchor node) across those regions. The considered pre-failure measures (i.e critical regions detection and replication) are found to have good side effects on the resolution of lookup requests and delay, despite the failures. Simulation results endorse the significant gains in terms of lookup delay and success ratio.

The second part targets the problems of distributed partition detection, unavailability of the anchor node due to partition, lost logical space (LS) recovery and reusing, and merging in DHT networks. We are using the philosophies of detection, replication and recovery to solve the identified problems. The proposed solutions ensure access to the index information despite the network partition. Similarly, prior to the network partition event, LS recovery and reusing is performed, this contributes an evenly distributed and connected logical network. The LS recovery process maintains evenly distributed LSs in each instance of the network after partition. Also, this sets grounds for smooth merging of the disjoint instances. Simulation results confirm the effectiveness of the proposed solutions.

Contents

List of Figures

List of Symbols/Abbreviations

Symbol	**Detail**
MANET	Mobile Ad Hoc Network
DHT	Distributed Hash Table
LIS/LS	logical identifiers space/Logical Space
LSP	Logical space portion
SII	Store Index Information (SII)-packet
SMI	Store Mapping Information (SMI)-packet
MREQ	Mapping Request Packet.
MREP	Mapping Reply packet
LID	Logical Identifier
UID	User Identifier (IP/MAC address)
AN	Anchor Node
N(p) or ${M_p}^+$	Open neighborhood of node p.
N[p] or M_p	Closed neighborhood of node p
S	Minimmum LID in chord
E	Maximmum LID in chord
min	Minimmum
d(x,y)	Logical distance between node x and node y in the logical network.

List of Publications

Published:

S. Zahid, S.A.Abid, S.Hussnain A.N, Nadir Sha and W. Ahmad, ”Distributed Partition Detection with Dynamic Replication Management for DHT-based Routing Protocols in MANETs”, IEEE-Access (2018), volume 6, pp. 18731 - 18746, 2018.**(Impact Factor: 4.098).**

Under Review:

S. Zahid, S.A.Abid, Nadir Shah and S. Hussnain A.N, ”Fault Tolerant DHT-based Routing in MANETs”, Computer Networks,Elsevier.

Chapter 1

Introduction

This chapter provides an overview of the DHT-based routing paradigm in MANETs and points out the research gap. Based on the analysis of the identified problems, we give motivation to our work. Furthermore, major contributions of the thesis are listed along with the proposed frame work to solve the identified problems. Thesis organization is given at the end of this chapter.

1.1 About this Work

The thesis studies problems in distributed hash table (DHT) based routing paradigm for mobile ad hoc networks (MANETs). The thesis is based on the recently published journal paper [3]. In particular the thesis targets problems such as critical node(s)/link(s) identification, dynamic replica deployment, distributed partition detection, logical space recovery and reusing, and merging considerations in DHT-based MANETs.

1.2 Introduction

MANETs are distributed systems composed of mobile nodes with limited radio transmission ranges. Nodes in the network move freely and dynamically self-organize themselves into temporary topologies without any fixed infrastructure. Nodes in MANETs (PDAs, laptops etc) can communicate directly with other nodes sharing identical media (radio, infrared etc) within their radio transmission range. For nodes outside this range, hop-to-hop communications are used. End-to-End communications among distant nodes are possible only through multi-hop forwarding. In this way, each node in a MANET operates as a host (send/receive data) and a relay/router (by acting as relays for neighbor). MANETs have a vast application domain; applications range from military to education, disaster management, civil services, etc. However, the decentralized nature, mobility (i.e. dynamic topology), limited transmission range and lack of infrastructure make the deployment of such networks challenging . There are a number of design issues in the deployment of such networks, namely scalability, fault tolerance etc. The network dynamics elicit problems such as node(s)/link(s) failure, network partition, merging etc., and completely halt network communication. Therefore, designing a scalable routing protocol that can perform efficiently in the adversarial environment offered by MANETs is a challenging task.

Generally, in MANETs, routing protocols use *IP* addresses for nodes identification as well as for routing and consider node identity equal to routing address (i.e., static addressing). These identifiers are independent of the relative location of nodes in the network and do not provide any information to guide the routing process. Therefore, such protocols rely on flooding or network wide dissemination of routing information and degrade performance as network scales. On the contrary, the identity and location of nodes should be considered separately because nodes are mobile and the network topology continuously changes. Therefore, a node should have a transient routing address that reflects its relative position with respect to its neighboring nodes. This emerges the concept of dynamic or transient addressing, where a node changes its address according to its location. In this manner, the transient addresses build a logical structured network that guides the routing process. This completely eliminates flooding at the control plane as well as at the data plane and makes the system scalable. Distributed Hash Table (DHT) provides a scalable way to decouple node location from its identity and facilitates general mapping between them. This allows DHT to scale with extremely large numbers of nodes and to handle continual node arrivals and departures.

Existing work targets the scalability problem using the concept of Distributed Hash Tables (DHTs). DHT-based routing protocols completely and effectively eliminate flooding at the control plane as well as at the data plane in a fully distributed manner, thus make the system scalable. It is well understood that flooding based routing protocols cannot scale well with growing network size [1][2][4][5], whereas, flooding-free routing protocols (i.e. DHT-based protocols) scale well with the increasing network size [2] [4] [5]. However, existing DHT-based routing protocols assume ideal network environment and this limits the applicability of such protocols in the adversarial environment offered by MANETs [3, 4, 6].

1.2.1 DHT Based Routing Paradigm in MANETs

DHT routing paradigm is comparatively a new emerging research domain in MANETs. DHT completely eliminates flooding at the control as well at the data plane. This makes DHT-

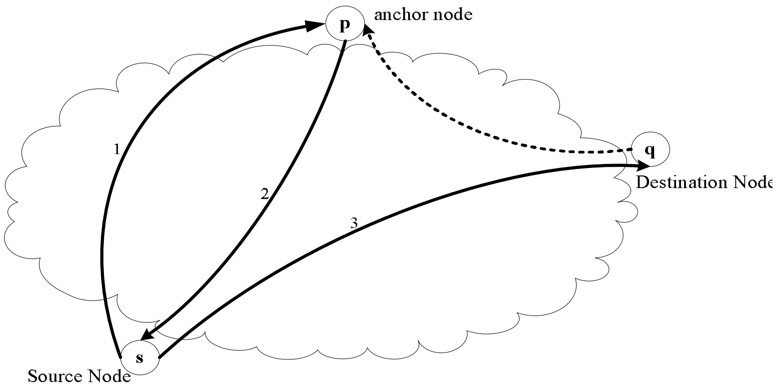


Figure 1.1: Address publication/Anchor Request, lookup and Routing in DHT networks

based routing protocols scalable.

In DHT-based MANETs, a logical structured network is built over the ad hoc physical topology in a fully distributed manner. Specifically, a node is assigned a logical identifier (LID), in addition to its permanent identifier (i.e. MAC/IP address, also known as user id (UID)), based on the LIDs of its physical neighbors. The LID is drawn from a predefined logical identifier space (LS) (the LS forms a logical structured network following a tree, ring, chord etc. structure). A node has to maintain a disjoint portion of the whole LS referred as logical space portion (LSP), the LSP acts as the LID of the node. Moreover, each node keeps track of its logical neighbor nodes that have LIDs close to its LID by following a ring, or chord, or multidimensional structure. Thus, a logical structured network is built over the ad hoc physical topology. Routing is performed in the logical structured network using the transient addresses (i.e. LIDs). The logical structure guides the routing process and eliminates flooding. Furthermore, evenly distributed LS is always needed for successful routing.

Address publication/Anchor Request: In network bootstrapping, nodes compute LIDs and store their index information, also called mapping information (i.e. (LID, UID) pair), by sending store index information (SII) packets, also called store mapping information (SMI) in the network, at certain nodes in the network referred as anchor nodes (AN). A node determines its anchor node by applying a hash function over its UID and generates a hashed value, say *h(v)*. The hashed value *h(v)* is drawn from the same LS that is used for LIDs assignments. A node *p* would act as the AN for node *q* if either *h(v)* falls in the LSP of node *p* or node *p*'s LID is closest to *h(v)* depends on the protocol specification. The SII packet is unicasted in the logical network to node *q* and node *q* stores the index information for node *p* (i.e. node *q* starts acting as a designated anchor node for node *p*). The address publication process is shown in Figure 1.1, dotted line shows the address publication process.

Lookup phase/Address Resolution: In DHT based MANETs, routing data packets is performed in two phases; i) Lookup Request/Address Resolution and, ii) actual data forwarding. To send a data packet, a source node *s* retrieves the destination node *q*s LID from *q*s AN i.e. node *p*, (flow 1 and flow 2 in Figure1.1 shows the lookup process). For this purpose, node *s* applies a hash function over *q*s UID that gives the LID of node *q*s AN, i.e., node *p*. Based on the computed LID, node *s* forwards a mapping request message (MREQ) towards node *p* in order to get the index information/mapping information of node *q*. Upon receiving the index information of node *q* in mapping reply message (MRPY), node *s* then sends the data packet towards node *q* based on the node *q* LID, flow 3 in Figure1.1 shows actual data packet forwarding. Greedy forwarding is used in the logical structured network i.e. the neighbor with closest LID to the intended destination node is selected as a next hop. The address publication (i.e. Anchor_Requests) and lookup (i.e. Address resolution) processes are the main factors those define overall network performance. Therefore, sophisticated techniques are needed to keep the system in function, especially in MANETs.

Table:1 Terminologies/Definitions related to DHT-based Routing in MANETs

Anchor Node (AN)	A node that stores mapping information of others nodes in the network. Any node in the network can act as anchor node. A source node retrieves the mapping information before starting a communication session with the destination node.
Logical Identifier (LID)	A unique ID (i.e. drawn from the logical space) that identifies a node in the logical network. The LID of a node shows its relative position in the logical network. LIDs are transient addresses and change with nodes positions.
Mapping information or Index information	A pair of LID and UID i.e. (LID, UID) pair, is called mapping or index information. Mapping information are maintained at anchor nodes and are needed to be retrieved for starting communication session between a source and destination pair.
Anchor Request Or Address Publication	After computing LID, a newly joining node publishes/stores its index information on the anchor node by sending Store Mapping Information (SMI) or Store Index Information (SII) packets in the network.
Lookup Request Or Address Resolution	A source node retrieves destination's mapping information stored on the corresponding anchor node to obtain destination node LID. The source node sends Mapping Requests (MREQ) packets and receives mapping reply (MRPY) message from the corresponding anchor node. DNS query resolution is an analogy of the address resolution process.
Logical Space (LS)	An address space (range) from which nodes obtain their LIDs. The address space follows certain structure (i.e. tree, cord etc) and arranges nodes in that specific structure. For example, in VCP[10] the address space ranges from 0 to 1, where nodes get LID between this range and form a virtual cord.
Logical Space Portion (LSP)	Each node in the network maintains a disjoint portion of LS and called as LS portion of that node. The concept of LSP is similar to that of LIDs.
Logical Network(LN)	The interconnection of nodes in the network based on the LIDs. In other words, the relationship among the nodes in the logical space.
Universal Identifiers (UID)	It is a unique and permanent identifier of a node. UID uniquely identifies a node and remains the same throughout the network life contrast to LIDs. The UID of a node can be its IP or MAC address.

Table 1 provides an overview over the basic definitions and terminologies used in DHT-based MANETs.

1.3 Research Motivation and Main Contributions

Historically Mobile Ad-hoc Networking (MANET) paradigm was proposed for tactical networks. But, the advancement in computing and wireless communication brought a revolutionary change in mobile communication and information technology domain [7, 8]. This enabled the development of smart mobile computing devices (such as PDAs, Cell phones, Laptops, etc.) equipped with short range wireless capabilities, i.e., Wi-Fi, Blue-tooth, which leads us to the deployment of commercial MANETs outside the military domain. MANETs have a vast application domain; applications range from military to education, disaster management, civil services etc. Flooding free routing protocols (i.e. DHT-based routing protocols) scale well and make possible the deployment of MANETs in such applications.

However, existing state-of-the-art DHT routing protocols for MANETs assume ideal network environments. But, the adversarial environment offered by MANET limits the scope of these protocols. Limited radio range, mobility, and lack of infrastructure introduce frequent and unpredictable changes to the network topology, i.e., connectivity/ dis-connectivity (can cause node(s)/link(s) failures, network partition, frequent merging etc.). Therefore, an end-to-end path that is believed optimal at a given time might not be available just after a moment. The case is more serious in DHT networks, where a logical structured network is built over the ad hoc physical topology. For example, nodes failure in the physical network would cause the failure of anchor nodes in the logical network. The failure of anchor node(s) may completely halt the communication between a source and destination pair, despite the fact that the source and the destination nodes are connected (since index information, stored on the anchor node(s), are always needed before starting communication). Similarly, physical network partition severely damages the logical structured network (i.e. logical space (LS)) and, also causes the failure of anchor nodes. Specifically, in network partitioning the logical network also gets partition and a portion of LS is lost. This results into disrupted LS(s) in the disjoint instances of the network. In DHT networks, complete and evenly distributed LS is always needed for smooth networking operations. Therefore, it is imperative to recover and reuse the lost LS during network partition. This would result evenly distributed LS(s) in the disjoint partitions. LS recovery and reusing need timely detection of the partition event. Therefore, partition detection is another important sub-problem in this specific domain. Moreover, network partition also causes the failure of anchor nodes, this problem arises when both source and destination are in one part of the network but corresponding anchor node remains in another part after network partition. Similarly, mobility causes frequent merging of the disjoint instances of the network. In DHT network merging needs to merge the logical structures i.e. LS(s). Existing merging algorithms [6] assume complete and evenly distributed LSs in the partitions going to merge. However, as mentioned earlier, in network partition a portion of LS is lost, this results unevenly distributed and disrupted LS(s) and prohibits the merging to be initiated. Therefore, LS recovery and reusing is needed for a smooth merging process. Furthermore, the key factor that defines overall routing performance in DHT networks is the successful resolution of lookup requests with minimum possible delay. However, in the literature review, we found that existing DHT protocols suffer from longer delays and fail to ensure the successful resolution of the lookup requests. Moreover, the adversarial environment (i.e. AN failure, network partition etc.) exacerbates the delay and may fail the lookup resolution process. Therefore, curtailing the information loss due to the network dynamics (i.e. node(s)/link(s) failure, network partition etc) is imperative for the successful communication in DHT networks. To solve the identified problems, distributed solutions under the scalability constraints are needed. It is worthy to mention that the targeted problems are completely unexplored in DHT-based routing paradigm. For the first time, we investigate the problems and define the scope in the context of DHT based MANETs, and provide fully distributed solutions.

Main objectives of this study are summarized below.

- To elaborate several core issues raised by the node(s)/link(s) failure, anchor node(s) failure and network partitioning in DHT paradigm such as unavailability of the index information, disrupted LS, LS loss and recovery, longer lookup delays etc.
- To design and develop distributed solutions based on the localized neighbors information to address the identified problems, which best suits, the scalability constraints of DHT-based paradigm. More specifically, following contributions cover the solution domain.
  - A distributed solution is provided that measures network dynamics using k-hop connectivity information and identifies critical node(s)/link(s) in the network. The identified critical node(s)/ link(s) plays a vital role in the replication of index information.
  - A novel address publication (also called, Anchor_Request) mechanism is proposed. In the address publication process the index information are replicated across the critical regions in the network in a fully distributed and dynamic manner. The mechanism makes sure access to the index information despite of anchor node failure or network partition.
  - A distributed partition detection algorithm is given which plays a vital role in taking pre-partition measures such as lost LS recovery and reusing after network partition. The LS recovery and reusing mechanisms set grounds for smooth net-

work merging.

- Impact of the dynamic replication management scheme on lookup (also called, address resolution) is studied and found significant gains.

To solve the targeted problems, the generalized overall structure of our work is given as under.

The Frame Work

1. Nodes collect *k*-hop topological information using the existing periodic *hello* messages. The *hello* messages are used for LIDs computation in DHT protocols.
2. Compute critical node(s)/link(s) using *k*-hop topological information **(Algorithm 4.1)**
3. Replicate the index information /mapping information across the critical regions in the address publication process (also called Anchor Request). **(Algorithm 3.1)**
4. Detect the partition event using *k*-hop topological information. **(Algorithm 4.2)**
5. In case of network partition, recover the lost logical space (LS) and reuse in the disjoint partitions. **(Algorithm 4.2)**

The proposed solutions are fully distributed and rely on the local information available to nodes without network wide dissemination of control information. The solutions exploit existing *hello* messages, used for LID computation in DHT protocols, thus our approaches do not incur any extra transmission overhead.

1.4 Structure of the Thesis

Rest of the thesis is organized as follows. A comprehensive review of the related work is provided in Chapter 2. At the end of Chapter 2, the research gap is identified and the problem statement is given.

Chapter 3 targets the problems of node(s)/link(s) failure, network partition, lookup resolution and longer lookup delay. Specifically, the impact of node(s)/link(s) failure and network partition on DHT-based routing in MANETs is explored. It is advocated that node/link failure and network partition elicit the problem of anchor node(s) failure. The anchor node failure exacerbates lookup delay and causes the failure of address resolution process. To solve the problems, a novel address publication (i.e. Anchor_Request) mechanism is proposed. The proposed algorithm 3.1 replicates index information in the address publication process. The algorithm confirms access to the index information despite anchor node failure and network partition. The proposed mechanism significantly reduces longer lookup delay and ensures access to the index information stored on ANs. Performance of the proposed solution is evaluated using simulation results comparisons with existing approaches. Performance evaluation using different parameters confirms effectiveness of the proposed solutions.

Chapter 4 deals with the problems of critical node(s)/link(s) identification, distributed partition detection, LS recovery and reusing. Algorithm 4.1 computes critical node(s)/link(s) using *k*-hop topological information. Critical node/link plays vital role in replication decisions(used in Chapter 3) and partition detection. Algorithm 4.2 detects partition event and recovers/reuses the lost LS in the disjoint parts of the partitioned network. The LS recovery/reusing ensures uniformly distributed LS(s) in the network instances and sets ground for smooth network merging. Simulation based results comparisons endorse the significant gains.

Chapter 5 deals with performance evaluation and simulations details. The problems targeted in Chapter 3 and in Chapter 4 are interrelated, therefore, Chapter 5 presents the combined results analysis of Chapter 3 and Chapter 4.

Finally, Chapter 6 collects some concluding remarks and provides future directions.

Chapter 2

Literature Review

2.1 About the Chapter

This chapter provides a comprehensive review on the existing state-of-the-art DHT routing protocols in MANETs. In particular, the classification, working principles and limitations of existing protocols are given. For DHT routing protocols, to operate in the adversarial environment offered by MANETs, some crucial requirements are identified. The problem statement is listed at the end of this chapter.

2.2 Routing in MANETs

The basic aim of a routing protocol is to compute best route from source to destination in MANET. But, due to nodes limited transmission ranges, establishing end-to-end connectivity can only be achieved by exploiting multi-hop communication. However, mobility and lack of infrastructure introduce frequent and unpredictable changes to network topology. An end-to-end path that is believed optimal at a given time might not be available just after a moment, as shown in Figure2.1. These specific attributes of MANET make routing an important and challenging task.

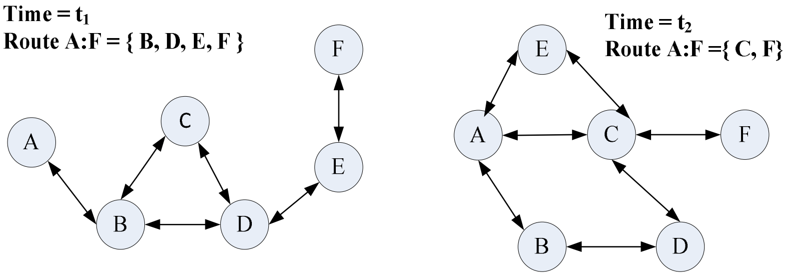


Figure 2.1: Network Dynamics/ Dynamic Routes

Several routing protocols have been proposed for MANETs. These routing protocols can be classified into different categories depending upon different criteria [7]. Each of them has its own applications and advantages.

By the manner in which they react to network topology, the routing protocols can be classified into reactive and proactive routing protocols. Reactive routing protocols find routes on demand (i.e. the routing protocol finds routes to destination nodes when needed). AODV [9] and DSR[10] are the examples of reactive protocols. Reactive protocols have larger delivery delay and are inefficient for huge traffic. In proactive routing, the routing information are maintained and periodically updated for the whole network. Proactive protocols maintain routes to all nodes in the network whether needed or not, e.g. OLSR[11]. Proactive routing can have unnecessary routes discoveries and produces extra traffic overhead.

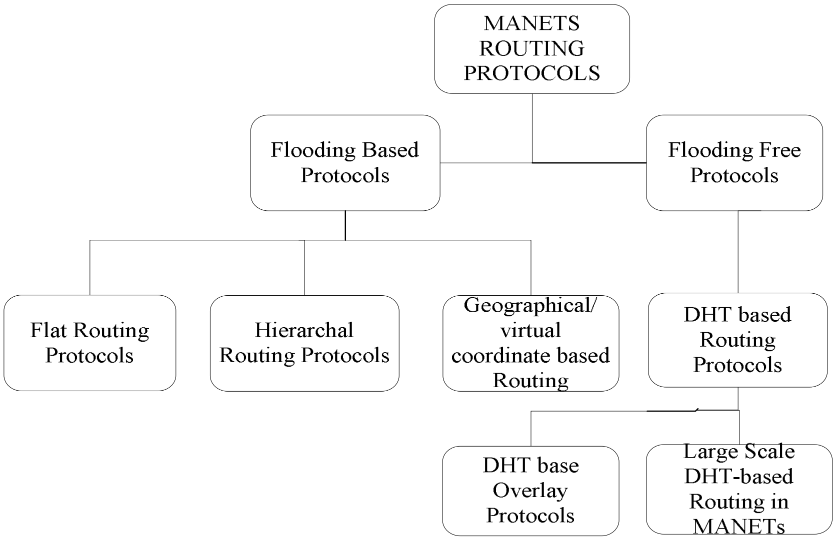


Figure 2.2: MANETs Routing Protocols

Based on the network organization and role of the nodes, the routing protocols can be classified into cluster based (i.e. hierarchal routing) and flat routing protocols. In cluster-based routing protocols, nodes in the network are grouped in clusters. Certain nodes are elected as cluster heads and gateway nodes in each group. The gateway nodes are used for inter-cluster communication. In this manner, only selected nodes take part in the route discovery process, thus reduces transmission overhead. Cluster-based routing protocols scale well with growing network size. The examples of cluster-based routing protocol are [12–16]. The flat routing protocols assume a flat network topology (i.e. flat structure) and nodes in the network play similar role. Flat routing uses flooding in the route discovery process and degrades performance with growing network size (e.g. AODV[9] etc.). Based on the use of location information, routing protocols in MANET can be classified into geographic and non-geographic routing. Geographic routing utilizes nodes's location information in the routing process [17–23]. The use of location information guides the routing process and confines the route searching space into a minimum estimated range. In this manner, the routing overhead is reduced to some extent. Geographic routing protocols scale well as compare to non-geographic routing, but obtaining and maintaining nodes physical location information is expansive and need line-of-sight. These factors limit applicability of the geographic routing protocols.

Based on flooding mechanism, we divide the existing protocols into two categories, flooding-based and flooding free-routing as shown in Figure 2.2. Flooding based routing protocols use flooding mechanism in route discovery phase. The examples of these routing protocols are DSR[10], AODV[9], OLSR[11], CBRP[12], etc. Flooding-free protocol, also called DHT-based routing protocol, aims to achieve scalability by eliminating flooding. DHT-based routing protocols can be divided in to two categories, DHT-based overlay protocols for Peer-to-Peer networks and DHT-based routing protocols in MANETs.

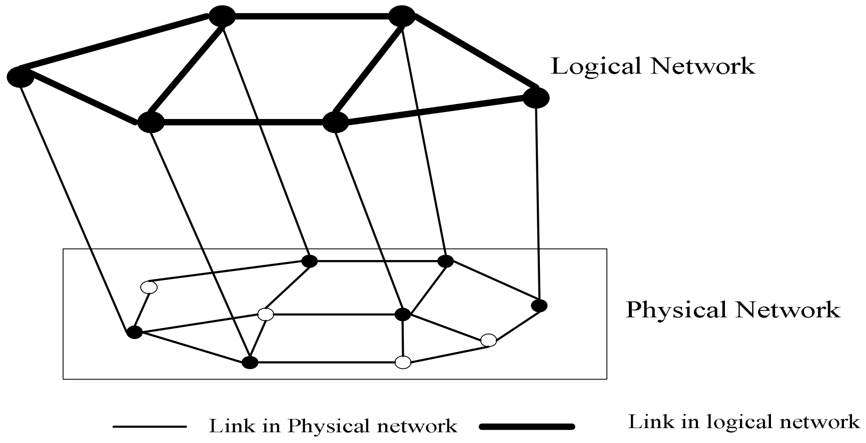


Figure 2.3: Logical Network vs Physical Network in P2P Networks

Initially DHT was proposed for structured Peer-to-Peer networks. In per-to-peer networks, each network node plays a role of both client and server simultaneously. Peers in the network self-organize themselves in a distributed manner without any centralized support and form an overlay network over the underline physical network. In other words, Peer-to-peer networks form an overlay layer on top of application layer, so that a direct link between two peers in an overlay network at the application layer may be equal to multi-hop path in the physical network [24, 25], as depicted in Figure 2.3. Fundamental DHT-base overlay protocols in Peer-to-Peer (P2P) networks found in the literature are CAN[26], Chord [27], Pastry [28], Kademlia [29]. There exist several common features between MANETs and P2P networks such as self-organization, decentralized architecture and transient topology. We can exploit the existing synergy to develop large scale routing in MANET [30]. This allows DHT to scale with extremely large numbers of nodes and to handle continual node arrivals, departures, and failures.

The research community exploits the synergy between peer-to-peer networks and MANETs. There exist a significant amount of research work in the literature that implement DHT at the network layer in MANETs. DHT-based routing protocols in MANETs eliminate flooding at the control plane as well as the data plane and make the system scalable. Some well known DHT-based routing protocols in MANETs are VCP[31][32], Tribe[33], PROSE[34], DART[1], VIRO[35], 3D-RP[2], M-DART[36], Motion-Mix[37] [38] [39] [6] [40] [41] [42] [43]. A comprehensive review (working, features, shortcomings) over the DHT-based routing protocols is given in section 2.3.

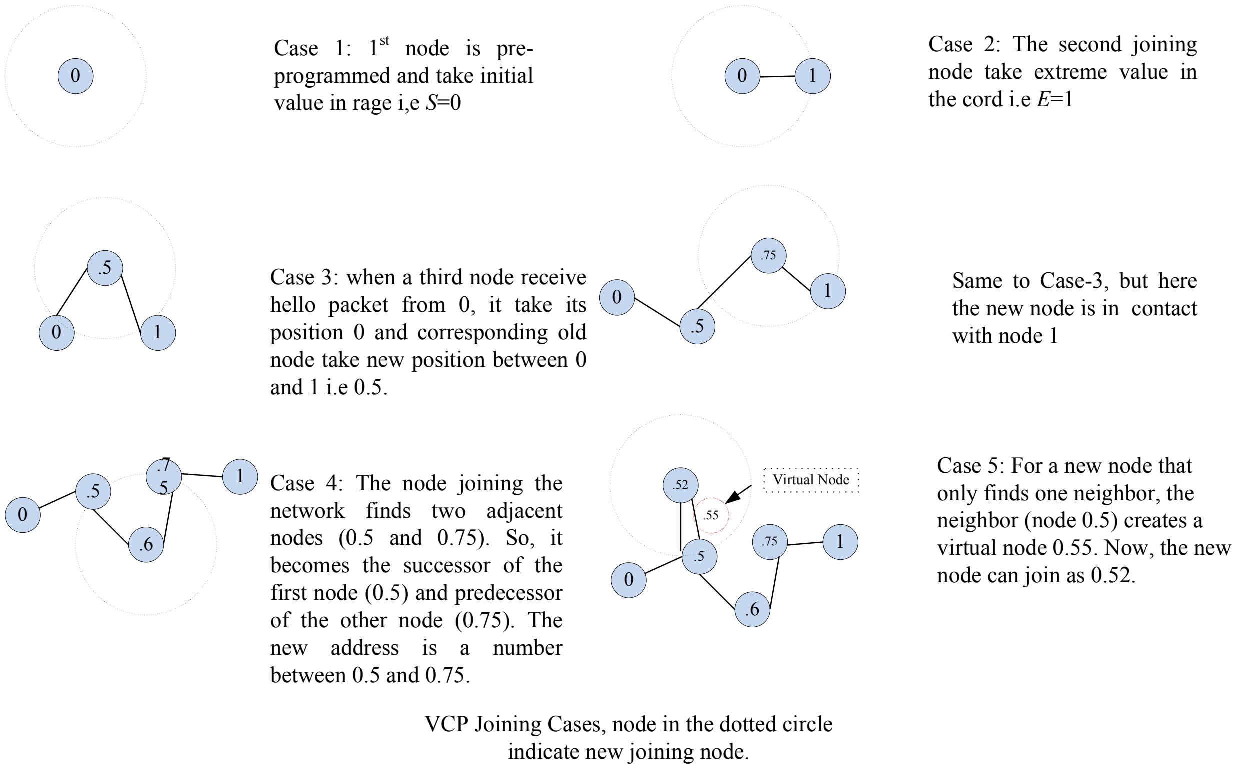
Since scalability is a more important concern for any routing mechanism to be used in MANET. A routing protocol is said to be scalable if the operations carried out by the protocol are in local scope and affects the network locally. It is well understood that flooding based routing protocols cannot scale well with growing network size [1][2][4][5], whereas, flooding-free routing protocols (DHT-based protocols) scale well with the increase of network size [2] [4] [5].
The subsequent section provides detail description of existing state-of-the-art DHT routing protocols in MANETs. Scope, limitations along with some identified open issues are given at the end.

2.3 DHT Routing Protocols in MANET

A detail description of the existing state-of-the-art DHT-based routing protocols is listed in the subsequent paragraphs.

Virtual Cord Protocol (VCP)[31][32] embeds the ad hoc physical topology of MANET into a logical structured network i.e. virtual cord. In other words, in network bootstrapping nodes organize themselves into a logical cord structure. Each node is given a logical identifier(LID) in addition to its UID (i.e IP/MAC) from a pre-defined logical space (LS) that ranges [*S*, *E*] (i.e. *S* and *E* represent starting LID and ending LID in the cord). The LID of a node represents its relative position in the cord. A joining node computes its LID based on the LIDs of its directly connected physical neighbors using various decision choices (a detail description is given in Figure 2.4. For this purpose, VCP uses periodic *hello* messages after every time interval to discover the directly connected neighbors and their position in the cord. In this manner, a joining node gets and maintains knowledge about the physical neighbors in the physical network and logical neighbors in the cord. In start, the first node is preprogrammed with the smallest value *S*. Joining the network a node is assigned a logical LID between *S* and *E*. Similarly, each node pro-actively maintains its logical neighboring nodes in the chord along with its physical neighbors. This makes the routing table of size *O(m)* at each node, where *m* refers to the logical (predecessor/successor nodes) and the one-hop physical neighbors of a node. VCP employs greedy forwarding that uses both the logical and physical neighbors of a node to ensure the successful delivery of the data packet. VCP assumes ideal network environments and does not consider network dynamics. A node/link failure, partition etc would cause a disrupted and unevenly distributed LS i.e. cord. This limits the use of VCP in MANETs.

Routing: VCP uses greedy forwarding on the virtual relative positions in the cord using locally available information (i.e. successor/predecessor as well as nodes physical neighbors), where at each routing step the data packet gets closer to destination. Let *N* be the set of all nodes in the network, where *P* represents relative positions in the chord (where $S \leq P \leq E$). If a source node *S*, with position *P* in the chord, wants to forward a data packet *D* to a destination position D_p, the source node sends the packet to the neighboring node with closest position to D_p. In other words, Node *S* selects node $N_i \in N$ as next hop such that $\forall N_j \in N, i \neq j$ and $|Pi\text{-}Pd| < |Pj - Pd|$). VCP uses logical distance in the chord. This process is repeated until no further progress is possible. Greedy forwarding ensures data delivery



Case 1: 1st node is pre-programmed and take initial value in rage i,e *S*=0

Case 2: The second joining node take extreme value in the cord i.e *E*=1

Case 3: when a third node receive hello packet from 0, it take its position 0 and corresponding old node take new position between 0 and 1 i.e 0.5.

Same to Case-3, but here the new node is in contact with node 1

Case 4: The node joining the network finds two adjacent nodes (0.5 and 0.75). So, it becomes the successor of the first node (0.5) and predecessor of the other node (0.75). The new address is a number between 0.5 and 0.75.

Case 5: For a new node that only finds one neighbor, the neighbor (node 0.5) creates a virtual node 0.55. Now, the new node can join as 0.52.

VCP Joining Cases, node in the dotted circle indicate new joining node.

Figure 2.4: VCP Joining Cases

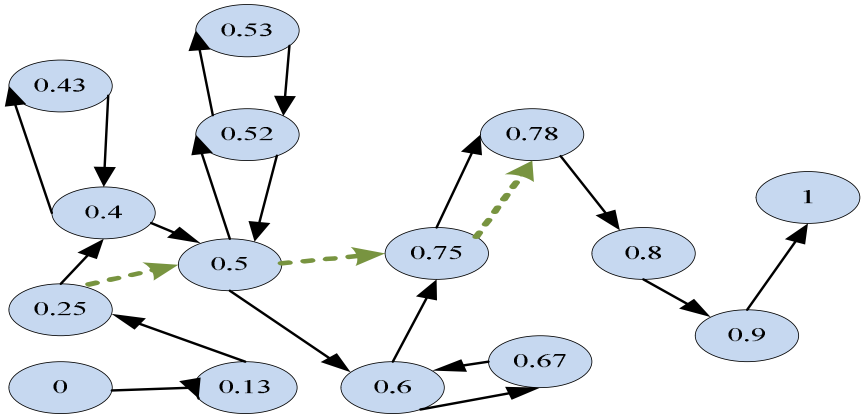


Figure 2.5: Routing in VCP

without sticking in dead ends, because cord has no dead ends (In case of no node failure). This process is depicted in Figure 2.5. In Figure 2.5, a network of 15 nodes is depicted by joining the cord (i.e. nodes LIDs range from *S=0* to *E=1*), here a source node 0.25 transmits a data packet to destination node 0.78. In first step, sender performs greedy forwarding by passing packet to 0.5 which is the closest available physical node, in the logical chord, to the destination i.e. |0.78-0.5|= 0.28, |0.78-0.4|= 0.38 and |0.78-0.13| =0.65. Similarly 0.5 forward it to 0.75 and then to 0.78 (destination).

Failure Handling in VCP: VCP exploits two types of control packets (*NP-I* i.e no path interval packet and *NPB* i.e. No path back packet) for failure handling. These control packets are used to find alternate routes in case of path break or next hop failure. The failure handling mechanism targets two cases i.e. In first case, if the failing node is the destination node then the packet is dropped. Otherwise, an *NP-I* packet is created by a neighbor of the failing node. The *NP-I* packet is stored and forwarded by all neighbors. If a node receives a *NP-I* packet twice then the node sends a *NPB* packet towards the sender of *NP-I* packet. The reception of *NPB* packet, nodes resume greedy fording strategy and forward the data packets

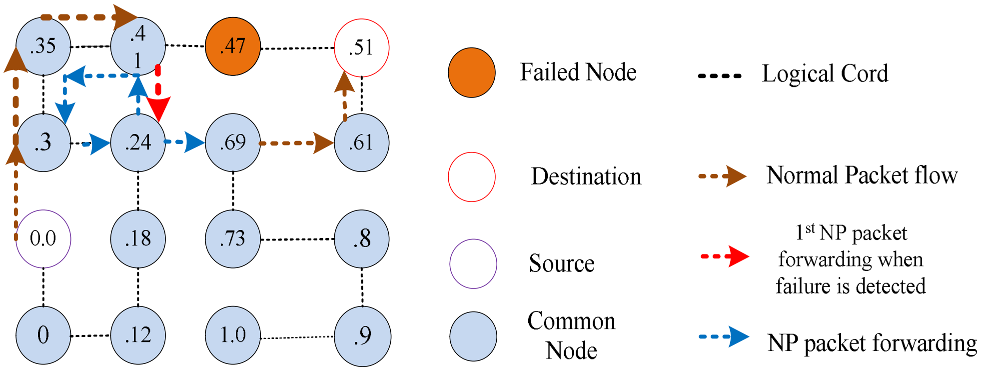


Figure 2.6: Failure Handling in VCP

to intended destination.This scenario is shown in Figure 2.6. Suppose node(0.0) has a packet for node(0.51). According to the greedy routing of VCP, the data packet will be forwarded by nodes 0.30 and 0.35 until it reaches node 0.41 as shown in the Figure 2.6. At this point, a dead end is detected because the previously existing node 0.47 has failed. Thus, node 0.41 will create a no path interval (*NP-I*) and send an *NP* packet back to node 0.35. Similarly, the *NP* packet is forwarded until it reaches node 0.24. The node(0.24) sends *NP* packet to node(0.41) again, however, 0.41 detects a loop and sends an *NPB* packet to node 0.24. In turn, node 0.24 tries to find another path by sending an *NP* packet to node 0.69. Finally, this node can resume greedy forwarding toward destination node 0.51. The whole process of failure handling in VCP is shown in Figure 2.6.

A routing protocol called TRIBE is proposed in [33] for indirect routing in MANETs. Tribe embeds the ad hoc physical network topology into a virtual space in range [0, 2^{m-1}]. The embedding preserves nodes physical proximity in the logical network, so that nodes closer in the physical topology also close in the virtual space. In virtual space construction, each node in the network is associated to a control region (i.e. a certain range) in the given virtual space. The starting minimum range value in the associated control region becomes virtual id (i.e. LID) of the node. Nodes obtain LIDs during the joining process from the neighboring nodes. In TRIBE, the assigned LIDs or control regions are used for nodes identification as well as for routing. Consider a newly joining node *i*, let the control region of node *i* is $R_i=[R_i^{\alpha}, R_i^{\beta}]$, where $[R_i^{\alpha},R_i^{\beta}]$ a subset of $[0,2^{m-1}]$. The starting range value is considered as LID of node *i* i.e. $E_i= R_i^{\alpha}$, where α and β denote initial and final value of the given range respectively. Suppose, the newly joining node *i* contacts *k* number of nodes in its neighborhood i.e. $A_i=\{a_1, a_2, a_3, ..., a_k\}$. In TRIBE, nodes exchange LID and corresponding control region information in a periodic *hello* message. Therefore, node *i* collects neighbors information list i.e. $L_i = \{ [Ea_1,Ra_1], [Ea_2,Ra_2], ... [Ea_k,Ra_k] \}$. Node *i* selects the node with largest control region(i.e. $max(Ra_k)$) as a parent node denoted as P_i. Parent node P_i share its control region with the newly joining node *i* as follows.

$$L_i = \{ [E_{a_1}, R_{a_1}], [E_{a_2}, R_{a_2}], ... [E_{a_k}, R_{a_k}] \}$$

From this list a node having largest control region i.e $max(R_{a_i})$ becomes nodes *i* parent node P_i , this parent node share its control region with the new joining node *i* in the following manner.

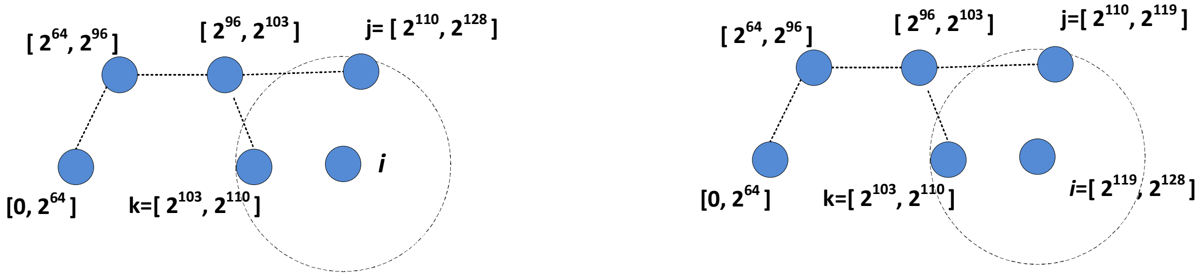
Node *i* control region gained from P_i:

$$R_i = \{ R_i^{\alpha} = (R_{P_i}^{\alpha} + R_{P_i}^{\beta})/2, R_i^{\beta} = R_{P_i}^{\beta} \}$$

New control region of P_i:

$$R_{P_i} = \{ R_{P_i}^{\alpha}= R_{P_i}^{\alpha}, R_{P_i}^{\beta} = (R_{P_i}^{\alpha} + R_{P_i}^{\beta})/2-1 \}$$

This process is depicted in Figure 2.7.



Before Joining of node *i*

After Joining of node *i*

Figure 2.7: Joining case in TRIBE

All nodes in the neighborhood are updated about the new node control region and parent node region changes. After obtaining virtual id (vid i.e. LID) during the joining procedure, each node stores its mapping information i.e.((IP, vid) pair) at a certain node in the network called home node (i.e. anchor node). This is done by simply hashing the vid and store at the closest node i.e Home agent, to the hashed value.

Routing: During the joining process the newly joining node obtains its vid, control region as well as the vids and control regions of its neighbors, as discussed earlier. This neighborhood information builds the routing table of the joining node. Similarly all nodes in the network pose this information, where simple greedy forwarding is used based on the nodes critical region closeness to the destination relative vid(i.e. LID). Furthermore, the routing process is similar as in other DHT-based protocols. For example, let a source *s* wants to route a packet to a destination *d*. For this purpose, node *s* will have to contact home node(anchor node) of *d* for retrieving mapping information. When node *s* finds vid(i.e. LID) of *d* from the corresponding home node i.e. anchor node , then a key based greedy forwarding mechanism is performed. Furthermore, handling with node mobility requires two tasks to be accomplished i.e the control region of the moving node should be taken by the remaining nodes especially by parent node and the leaving node is required to be assigned a new region and vid based on its neighborhood. There can be two situations, first when the leaving nodes control region and its child control region forms a contiguous region with its parent node. This case is handled in a straight forward manner without causing any changes to the neighboring nodes. The rejoining of the leaving node in such scenario is called smooth assignment/joining and is shown in Figure 2.8.

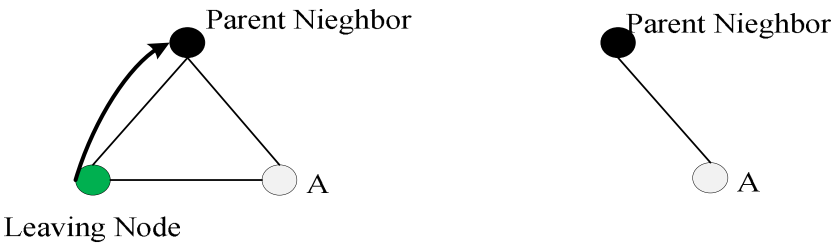


Figure 2.8: Soft Joining in TRIBE

In smooth assignment/joining, the leaving node simply hand overs its control region, location database to its parent node and the parent node directly establishes kinship relationship with the siblings of the leaved node without any further processing. On the other hand, if the leaving node's childes do not form contiguous region with the leaving nodes parent. This situation is critical and additional measures are needed to be taken for uniform region partitioning and merging. When the moving node leaves it sends its control region, neighborhood list (vids, control regions), associated location information and its child (associated neighbors) to its parent. Based on the collected information, the parent node divides the handed over region among nodes in the neighborhood. In this manner, a uniformly distributed control region throughout the network is achieved and nodes those have lost their kinship relationship find appropriate relationship. For example, when a leaving node *n* hand overs its control region R_n , its neighbor list L_n and associated data base to its parent P_n. The parent

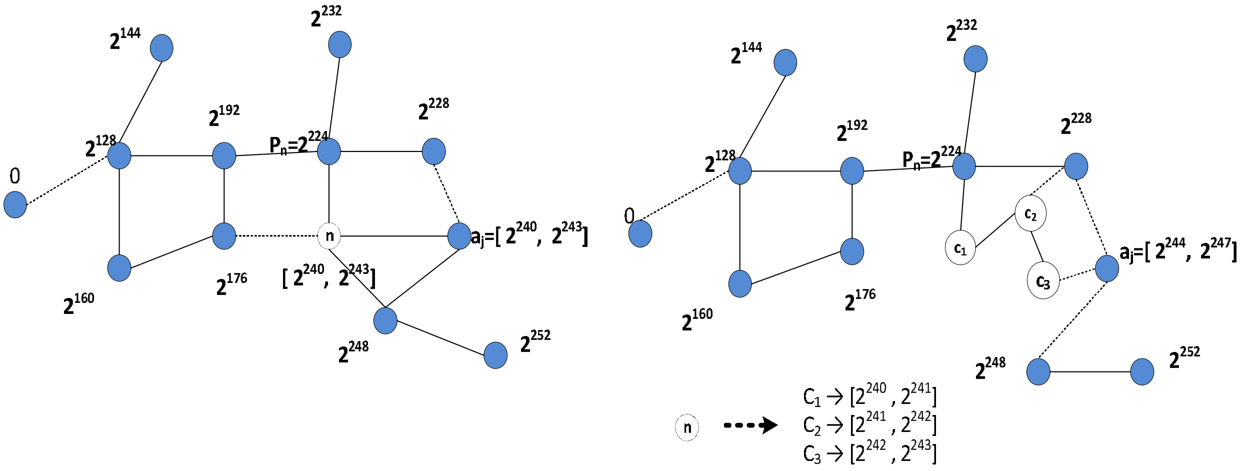


Figure 2.9: Hard Joining in TRIBE

sends (floods) a DISCOVER message to nodes in the received neighbor list of *n* i.e. L_n, (to find a node $a_j \in L_n$, having contiguous region to the leaving nodes control region R_n). In response, P_n receives a PATH messages containing the neighbor list of all the neighbors of *n*, where, P_n selects the node having list of neighbors closest to the leaving node neighbors. The selected Path-message (every path message) also contains vids of all nodes traversed and corresponding number of hops to a_j. These vids (nodes) along the traversed path to a_j forms a virtual path. Thus the handed over control region of the leaving node is divided by the nodes in the virtual path in an increasing order from parent P_n to a_j. Similarly, each node in the virtual path gaining critical region also take responsibility for managing location information database associated with its own region as well as its assigned new region. The whole process is depicted in Figure 2.9. In Figure 2.9, the moving node with vid *n*= 2^{240} and control region $\{2^{240}, 2^{243}\}$, the path message contains Path(a_j)= $\{2^{224},2^{228},2^{244}\}$, so node P_n (2^{224}) divides the n’s region into three parts (${R_n}^{i/3}$) and gives it to these nodes. Similarly, the location data base is divided equally.

TRIBE performs routing in the virtual space constructed using the computed LIDs. However, TRIBE assumes ideal network environments and ignore the crucial aspects of network dynamic such as node/link failures, anchor node failure, partition etc. Similarly, node(s) joining/leaving needs re-labeling a large portion of the network. Therefore, TRIBE cannot be used for highly mobile and dense environments. This also affects network scalability. These issues limit the scope and applicability of TRIBE in MANETs.

In [34, 44], a DHT-based routing protocol, called PROSE, arrange nodes in the network into a logical tree structure. In other words, the identifiers of nodes form a rooted tree. A DHT like routing mechanism is employed on the logical tree structure. Specifically, PROSE performs routing using prefix labeling and distributed hashing. In network bootstrapping, nodes are assigned prefix labels i.e. identifiers. The labels denotes nodes locations relative to an elected root node in a breadth-first manner. The label assignment takes place in such a way that there exists at least one route between any two nodes. Nodes compute identifiers (i.e. labels) and publish mapping information to be stored on the anchor nodes. A source node retrieves the mapping information from the corresponding anchor nodes prior to route packets. A prefix label Λ for a node is a word in Σ* (a set of all strings over Σ, where Σ alphabet containing finite number of symbols such that $|\Sigma| \geq 2$ (i.e 0,1,2). Similarly, every node labels its links to each of its neighbor with letter w from Σ, for a link *i*, $w_i = \{ w_i \in \Sigma \mid w_i \neq w_{i+1}$ for all $i \leq$ degree of the node - 1 $\}$, so that a unique letter is assigned to each link of node *i*) such that $\Lambda = \Lambda_{parent}.\mathbf{I}$, here Λ_{parent} is the prefix label given by the parent and **l** is a unique suffix over *k* (degree of the parent node) forms prefix label for a node. This prefix label defines a predecessor relation in the resulted rooted tree. PROSE employs greedy routing mechanism based on maximum prefix match. Each node along the path selects a neighbor with longest prefix match. Additionally, the routing process not only rely on the prefix tree but in case if there exist a two-hop neighbor that is lexicographically closer to the intended destination, then the packet is forwarded through that node instead of forwarding through prefix-tree parent. Furthermore, if there exist two nodes(i.e. source and destination) with equal prefix length and then there might be a common node in the ancestor tree through which they are reachable to each other. Similarly, if labels length of the source and destination are deferent, then they are reachable/accessible through the root node. However, PROSE is inflexible to network dynamics. The mobile nature of MANETs causes a large portion of the network to be reconfigured/relabeled. In case of a node leaving, if the leaving node is leaf of the prefix tree then relabeling is simple and dealt as the node joining (i.e the node gets its label from the immediate predecessor at the joining point). In case if the leaving node is an intermediate node in the prefix tree then the sub-prefix-tree is required to be reconstructed from the immediate ancestor of the leaving node. In case of anchor node leaving/failure, for example let node x is defined as anchor node for node y in the network. Node y updates its anchor node (i.e. node x) proactively about its current label and when the anchor node(i.e. node x) leaves/fails, node y selects another node with maximum matching prefix as a secondary anchor node. In such arrangements all the in-transit lookup requests remains unsuccessful.

The work in [45] targets the scalability problem in MANETs by using a combination of DHT and Optimized Link State Routing(OLSR). The proposed protocol(DHT-OLSR) enhances the proactive OLSR by adding the functionality of DHT along with dynamic clustering. The protocol improves scalability through partially eliminating flooding and confines control traffic within a limited scope i.e cluster size. In other words, intra-cluster communication is based on flooding mechanism similar to OLSR, where inter-cluster communication is achieved in a DHT-based uni-casting manner. However, the protocols lacks to address network dynamics and failures. There exist no explicit mechanisms to handle network partition, nodes failures and the impacts on routing. Similarly, network partition and merging, as common in MANETs, have not been addressed.

KDSR [46] integrates Kademlia[29] and DSR[10] at the network layer for indirect as well as direct routing in MANETs. The scheme considers no logical embedding of the physical topology of the network (i.e no joining algorithm or logical space construction). Simply all nodes hash their IP and take an 160-bit logical id for the purpose of indirect routing. Similarly, each node in the network maintains a kademlia like binary tree routing table, where *k* buckets exist. At each bucket nodes (node ids) between 2^i and 2^{i+1} $1 \leq i \leq 160$ distance are maintained and updated in a timely fashion, XOR is used as a distance function. So, each node in the network can find others node from this routing table using the exclusive OR distance function. The *k*-buckets are updated using Least Recently Discovered (LRD) replacement strategy. Each entry in the *k*-buckets include four entries i.e Node id, IP, distance and a vector to source routes. Routing: Routing tables are built in a manner similar to Kademlia, but packet forwarding in the network is based on source routing. For example, if a node *x* has to route data packets to a destination *d*, node *x* consults its routing table at appropriate bucket(i.e. corresponding to the XOR distance), if a source route exist in the bucket then data is forwarded. In case when the corresponding bucket is empty then a flooding based route discovery mechanism is initiated similar to DSR. In case of failure the node searches for alternate paths in the bucket or forward the packet to one of its neighbor based on the XOR distance, if no path exist in both cases so the packet is dropped. In KDSR, the routing tables are built and updated using flooding. This results into a huge transmission overhead. Similarly, construction of the routing table (i.e building kademlia binary tree) takes unpredictable time. Failure detection and handling may also need flooding. Moreover, KDSR is based on source routing which causes huge packet size and needs large buffer size needed at every node in the network. These requirements limit the applicability of KDSR in the adversarial environment of MANETs. Similarly, KDSR is not scalable.

Virtual Ring Routing(VRR)[47] is a proactive DHT-based routing for MANETs. VRR embeds nodes in the network into a virtual ring in certain order. In other words, the LIDs of nodes form a virtual ring structure. Nodes in the network maintain logical neighbors information along with physical neighbors. Each node in the network marks its physical links and maintains the links value above a certain threshold value. Nodes in the network maintains the lists of physical neighbors (i.e. *pset*) and logical neighors(i.e. *vset*). Routing is performed on the LIDs in the virtual ring. Nodes use greedy forwarding mechanism based on the logical identifiers. VRR addresses network partition and merging issues. Nodes in the network detect failures/merging and initiate the recovery mechanisms (i.e. merging). However, the mechanisms employed partially addresses the partition problems. There exist no explicit mechanisms to recover the lost information due to network dynamics. For merging the rings, each disjoint ring maintains representative nodes and if nodes across the network find nodes with different representative nodes then the merging process is initiated. However, nodes failures and network dynamics recovery mechanisms incur extra transmission overhead. Therefore, VRR is not suitable for dynamic environments.

Dynamic Address Routing(DART)[1] embeds the ad hoc physical toplogy of MANET into a virtual binary tree. The embedding preserves physical proximity in the logical network i.e. virtual binary tree. In network bootstrapping, the joining nodes form a virtual binary tree and obtain an L-bit logical identifier (LID). The LID of a node represents the node position in the tree of height, $h=L+1$. The logical network construction(i.e. tree) preserves physical proximity and maintains a key invariant property:

Key-invariant:

Adjacent nodes in the logical network (i.e.tree) share a common prefix and represent a connected sub-graph in the physical network topology. In other words, non-empty adjacent (i.e. with common prefix) sub-trees must be connected in the physical network topology. For any two logical adjacent trees, say $T_{(1)}$ and $T_{(2)}$, there must be nodes , say *i, j*:

$$\{ \exists\ i \text{ and } j \text{ where } i \in T_{(1)} \wedge j \in T_{(2)} \mid (i,j) \in E \}$$

In DART, a joining node computes LID based on the LIDs of its neighboring nodes. If the joining node finds more than one neighboring nodes, then the joining node takes LID from the neighbor with empty highest-level insertion point in the routing table. In other words, the new node takes its address from one of the neighbor with empty sibling tree. DART maintains an address size (i.e. L bit) array. Each node keeps routing information in the array in a manner that at position i the level-i sibling of the node is stored. In this way, each node stores all its corresponding possible sub-tree short addresses. In this manner every node stores all its corresponding possible siblings (sub-trees). A node can find a required destination node by simply finding corresponding sub-tree containing the destination node. This is found by finding the most significant bit that is deferent between the destination node LID and current node LID. Furthermore, nodes use periodic routing updates to notify their neighbors of the current state of their routing table Routing. Let consider a node with 3-bit LID 100 wants to route a data packet to a node having LID 000 (suppose the sender got this by contacting the anchor node of 000). Discussed earlier, the sender 100 should have the routing entries about all its siblings i.e 0xx,11x and 101 as shown in Figure 2.10. The sender node first need to find the most significant bit differs between its own LID and destination LID i.e. the second bit. The sender node looks its routing table for 0xx sibling and the packet is forwarded to

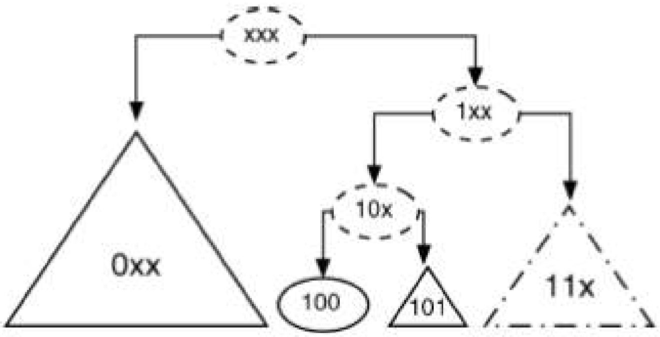


Figure 2.10: Routing in DART(curtesy from [1])

that sub-tree. The process is repeated until it reaches to the destination. DART scales well with the increasing network size. However, DART assumes ideal network environments and ignore the adversarial environment offered by MANET. DART only considers the failure of anchor node and provides solution for secondary anchor node selection. But, tree-based structures are vulnerable to single point of failure and instigate congestion problems.

M-DART[36] is an enhanced version of DART. M-DART sole purpose was to explore and pro-actively maintain multiple paths between a source and destination pair. M-DART pro-actively and blindly maintains multiple paths between a source and destination pair. Therefore, in case of a route failure alternate paths, if available, can be exploited. This is an inherent feature of the multi-path routing i.e. redundancy. M-DART completely fails in scenarios where a single path exists between a source and destination.There exist no explicit mechanisms for guaranteed alternate routes. M-DART does not claim fault tolerance capability and does not provide any explicit solution to the problem. Although, M-DART is evaluated under several scenarios and showed that M-DART performed well. After all, blind assumptions would not solve the problem in its spirit, especially, in a decentralized, dynamic and fully self organized DHT network(i.e. MANET). Again the impact of network dynamics

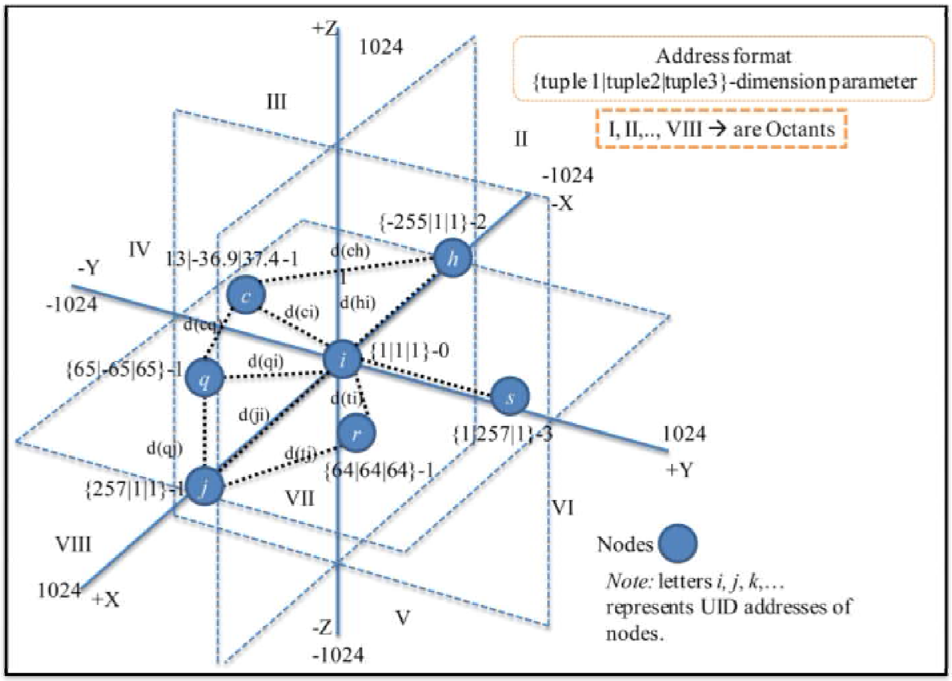


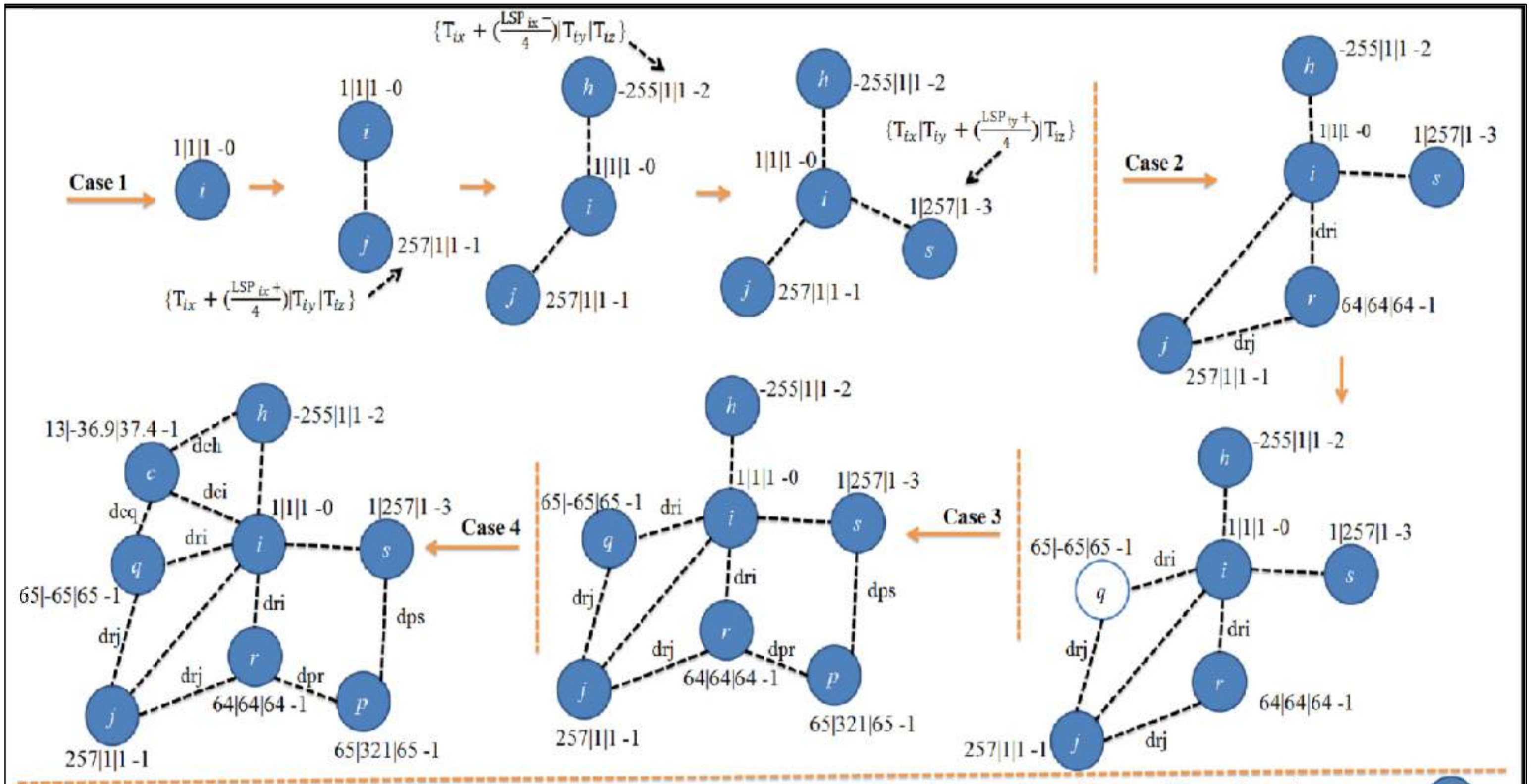
Figure 2.11: A view of 3D-LIS of node *i* in 3D-RP.

on the logical network(i.e. tree) is not considered.

3D routing protocol (3D-RP)[2, 48] is primarily designed to address the mismatch problem. Each node in 3D-RP envisions its neighboring nodes in a three-dimensional coordinate system while keeping itself at the origin, which is referred as local 3D logical identifier structure (3D-LIS) that divides the space into three planes having six dimensions and eight octants. The core idea of using 3D-LIS is to logically reflect the physical inter/intra-neighbor relationship of nodes. Each node in 3D-LIS computes its LID that reflects its relationship with its neighboring nodes. The computed LIDs consist of three parts (i.e. ordered tuple) $\{x|y|z\}$, where *x, y, z* are M-bit identifiers drawn from a pre-define 3D-LS. Each dimension (i.e., *x,* *y*, and *z*) ranges from 1 to ± 2^M in 3D-LS. Nodes in the network exploits *hello* messages to construct and maintain the the logical space structure(i.e. 3D-LS). Similarly, each node group the neighboring nodes into different dimensions using a dimension parameter (*dim*). 3D-RP computes physical distances among the nodes using receiving signal strength (RSS) metric. Weights are assigned to each link based on the calculated distances. Furthermore, the weights are used in the interpolation function to compute relative positions of nodes with respect to the neighboring nodes. LIDs assignment in 3D-RP is based on different decision choices. For example, a single starting node *i* in the network reserves the initial positions in each tuple and sets its dim value to zero. LID of the starting node *i* sets as $\{1|1|1\}$-0. Similarly, if a new joining node *j* finds only a single neighboring node *i*, node *j* computes using equation (1) in the first dimension of node *i* and sets its LID to $\{257|1|1\}$-1, this scenario is shown in Case 1 of Figure 2.12. In a similar fashion, the neighboring nodes *h* and *s* of node *i* compute LIDs but in different dimensions. The different dimensions show that node *h* and node *s* have no physical connectivity in the physical network. Similarly, nodes *h*, *j*, and *s* compute LIDs in different directions. In this manner, nodes *q* and *r* compute LIDs according to the Case 2 using the interpolation equation, as shown in Figure 2.12. According to Case 3, as shown in Figure 2.12, node *p* computes LID based on the relative positions of node *r* and node *s*. Furthermore, the newly joining node *c* computes LID using the inter-neighbor relationship with nodes *q*, *h*, *i* and *q*. The local view of node *i* in the 3D-LS along with the neighboring nodes is shown in Figure 2.11, the dashed lines show physical links in the physical network topology. Figure 2.11 shows mapping between the physical neighbors of node *i* and 1-hop neighbors in the logical network. It is clear from Figure 2.12 and Figure 2.11 that physically connected nodes in the physical network topology are also closer to each other in the logical network (i.e. 3D-LS). This feature of 3D-RP minimizes the effects of mis-match problem and, avoids longer routes in the routing process along with a reasonable reduction in delay and overhead. *Forwarding:* In the forwarding decisions, nodes exploit 1-hop logical neighbors (i.e. L$_{nbr}$). To send a packet to a destination with LID $\{x|y|z\}$-dim, nodes along the path select next hop from the 1-hop logical neighbors with similar dim value and least sum of difference (LSD) to the destination node. If there exist no such neighbor, then the packet is forwarded to the base node (i.e. the node involved in LID computation). However, 3D-RP is purely designed to avoid the mismatch problem and, it completely ignores network dynamics and its impact on the logical structures built (i.e. 3D LS.)

Mismatch Problem: In DHT-based routing, a logical network is build over the ad hoc physical topology of MANET. In logical network construction nodes compute LIDs and build a logical structured network (i.e. following chord, tree, 3D structures). The logical network construction and its structure play an important role in the overall network performance. The logical network should preserve physical proximity of nodes. Nodes closer in the physical topology should also be closed in the logical network. If the logical neighbors of a node are not adjacent in the physical network topology, this results into mis-match or ill-match between logical and physical network. The mis-match problem results into longer routes, delay and additional overhead. There exist several mechanisms to address this issue, for example, VCP considers logical as well as physical neighbors in packets forwarding. This reduces the impact of mis-match problem to some extent. In 3D-RP, the problem is addressed using 3D structures, it is advocated that 3D structure has the capability to precisely interpret the intra-neighbor relationship of nodes in the physical network in the logical network. 3D-RP preserves physical proximity in the logical network and avoid the mis-match problem.

Motion-Mix[37] exploits past mobility patterns of nodes in the network and builds a partial overlay network. In the overlay network only 1-hop neighbors are maintained despite maintaining a global overlay network. In this manner, Motion-Mix successfully reduces the cost of joining algorithm. However, the logical structure used is not resilient against high mobility. Mobility causes frequent LIDs reassignments. Since LIDs assignment considers past



Case 1: In case of more distinct neighbors, the remaining 3 dimensions out of the six uses the following formula to assign LID to nodes.

$\{T_{ix}|T_{iy} + (\frac{LSP_{iy}^{-}}{4})|T_{iz}\}$ $\{T_{ix}|T_{iy}|T_{iz} + (\frac{LSP_{iz}^{+}}{4})\}$
$\{T_{ix}|T_{iy}|T_{iz} + (\frac{LSP_{iz}^{-}}{4})\}$

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

$$\text{LID} = \{ \sum_{k=1}^{L_{nbr}} \frac{W_{mk}}{\sum_{j=1}^{L_{nbr}} W_{mj}} * T_{kx} \left| \sum_{k=1}^{L_{nbr}} \frac{W_{mk}}{\sum_{j=1}^{L_{nbr}} W_{mj}} * T_{ky} \right| \sum_{k=1}^{L_{nbr}} \frac{W_{mk}}{\sum_{j-1}^{L} W_{mj}} * T_{kz} \} \ldots. \text{Eq(2)}$$

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

$$\text{LID} = \{ \sum_{k=1}^{L_{nbr}} T_{kx} \left| \sum_{k=1}^{L_{nbr}} T_{ky} \right| \sum_{k=1}^{L_{nbr}} T_{kz} \} \ldots \text{Eq(3)}$$

Node ●

Note: letters *i*, *j*, *k*,... represents UID of a node

Figure 2.12: Joining cases in 3D-RP. Dashed lines show physical links.(curtesy from [2])

mobility patterns, therefore, an increase in nodes moving speed would increase the frequency of LIDs assignments and nodes should have to collect/exchange relatively more information. This would cause more extra overhead. Similarly, the protocol assumes ideal network environments and does not provide any fault tolerance capabilities to handle network dynamics.

Mesh-DHT [38] embeds the ad hoc physical topology into a 2-dimensional (2D) logical structured network. In the embedding the physical proximity of nodes is preserved in the logical structured network. In the embedding process closer nodes in the physical network attract each other and obtain closer coordinates in the logical network. Similarly, distant nodes in the physical topology produce the effects of repulsion and are assigned coordinates accordingly. Moreover, nodes exchange 1-hop neighbors coordinates and iteratively update the coordinates to achieve a proximity aware embedding. In this manner, Mesh-DHT minimizes the effects of ill-match problem [2][4]. However, the protocol assumes ideal network environments and does not consider network dynamics.

A DHT based routing and lookup mechanism called VIRO[35] employed on top of a topology aware virtual id space. Physical topology of the network is embedded into a structured virtual space like kademlia virtual tree[49] (depth=L). The embedding preserves physical proximity of the network in the virtual space. In such embedding the leaves of the tree represent actual nodes and all the intermediate nodes in the tree are logical nodes represent viro switches(i.e. logical nodes providing access to physical nodes through some association in the virtual tree structure). The LID of a node is an L-bit binary string spanning from root to leaf of the tree. The virtual space construction and LIDs assignment proceed in a bottom-up fashion starting from the leaf node. First, the lower-level bits are determined and then higher level bits are assigned recursively. VIRO constructs routing tables in a round-by-round and bottom-up fashion i.e for round k=1 to L. Each round results into the corresponding level entry. At each round *k*, each node discovers level-*k* gateway nodes to reach nodes in the corresponding sub-tree. After computing LID, a joining node publish its index information or mapping information i.e. (LID, UID) pair and store on the anchor nodes. The index information stored on the AN are needed to be retrieved by the source node to route packets towards the intended destination. In other words, VIRO follows the conventions of DHT paradigm i.e. address pulication or Anchor Request and lookup requests. VIRO employs greedy forwarding strategy once the routing tables are built using the logical identifiers. Let node *x* sends a data packet to destination node *y*. Node *x* forwards the packet to *y* directly if *y* is one of its neighbor, otherwise it computes logical distance *k= (x,y)= L- lcp(vid(x), vid(y))* (*(x,y)* represents logical distance between nodes *x* and *y*, where *lcp(vid(x), vid(y))* means longest common prefix between node *x* and node *y*) and forwards the packet to next node as present in the level-*k* routing entry. The same process is repeated by all nodes along the path. If the corresponding level-*k* entry is empty, the packet is dropped.

Merging and its impact on mismatch-problem in DHT-based networks has been studied in[6]. It is advocated that logical structure i.e LS plays vital role in avoiding mismatch-problem and multi-dimensions structures(i.e. 3D structures) are more resilient for smooth network merging as compare to tree, ring, cord etc. A leader election algorithm is proposed, where a node in the network with smallest LID is elected as a leader. Nodes in the network maintain and exchange leader information. If nodes across the network find different leaders, a merging detection event is triggered. In fact, the merging detection event in the physical network initiates the merging process in the logical network. In the merging process, nodes in the smaller network change their LIDs and join the larger network. The merging process ensures a uniformly distributed logical space structure. Moreover, the work targets the mismatch problems to avoid longer routes. It is shown that 3D structures are more suitable to deal with merging and to avoid longer routes. However, the work considers only post-partitioning measures while ignoring pre-partitioning considerations such as partition detection, node/link failure, LS alignment and recovery, index information recovery and their impact on merging. In other words, the work targets merging problem in the context of mismatch problem. Furthermore, the work still considers ideal network environment and ignore the adversarial environment of MANETs.

Similarly, some other well known DHT protocols in MANETs are [39] [40] [41] [43] [42].

All these protocols assume ideal network environments and failed to address the practical aspects of network dynamics in MANETs. Table 2.2 summarizes the features and limitations of the existing DHT routing protocols in MANETs. It is clear from the Table 2.2 that existing state of the art DHT routing protocols mainly target the scalability issue in MANETs and eliminate flooding at control plan as well at the data plan. In the perspective of fault tolerance, Table 2.2 shows that non of the existing protocols deal with network dynamics (i.e. node failure, AN failure, LS lost, network partition etc). However, some existing protocols partially address the fault tolerance problem, as shown in the table. The summarized features and limitations listed in Table 2.2 endorse the importance of network dynamics(node failure, AN failure, LS lost, network partition etc). Therefore sophisticated mechanisms are needed to address the issues identified in Table 2.2.

In MANETs, mobility and limited transmission range can cause frequent network partitioning besides other effects (i.e loss of LS, anchor node failure/leaving). Network partitioning is the process of breaking down a connected network topology into two or more than two disjoint parts such that nodes in one partition can not access nodes in another partition [50]. There exist several protocols in the literature [50], [51], [52], [53], [54] to deal with the partitioning problem in MANETs. However, theses approaches target MANETs in general and cannot be used for DHT-based MANETs, because in DHT-based networks, partitioning makes sever damages and need additional effort to cope with [3] (see Chapter 3 for details).

Furthermore, distributed approaches based on the localized knowledge of node scale well and are needed to address network dynamics (i.e. node/link failure, AN node failure, network partition etc.) in DHT-based MANETs.

Table2.2 Summarized Features of DHT-based Routing Protocols in MANET

*P*rotocol / Metric	VCP	TRIBE	PROSE	VRR	KDSR	DART	M-DART	VIRO	DHT-OLSR	DHT-Merging	3D-RP	Mesh-DHT	Motion-Mix
Routing Philosophy (Reactive, Proactive)	**Proactive**	**Proactive**	**Proactive**	**Proactive**	**Reactive**	**Proactive**	**Proactive**	**Proactive**	**Proactive**	**Proactive**	**Proactive**	**Proactive**	**Proactive**
Scalable	**Yes**	**Yes**	**Partially**	**Partially**	**No**	**Yes**	**Yes**	**Yes**	**No**	**Yes**	**Yes**	**Partially**	**Yes**
Fault Tolerance	**Partially**	**Partially**	**Partially**	**Partially**	**No**	**No**	**Partially**	**No**	**No**	**No**	**No**	**No**	**No**
Control Overhead	**Medium**	**Medium**	**High**	**High**	**High**	**Medium**	**Medium**	**Medium**	**High**	**Medium**	**Medium**	**Medium**	**Medium**
Logical Structure (LS)	**Cord**	**Tree**	**Tree-like**	**Ring**	**Tree-like**	**Tree**	**Tree**	**Tree**	**cluster**	**Tree, 3D structure**	**3d structure**	**2D**	**linear**
DHT-based	**yes**	**yes**	**yes**	**DHT-like**	**DHT-like**	**yes**	**yes**	**yes**	**DHT-like**	**yes**	**yes**	**yes**	**yes**
Partition Handling	**No**	**implicitly**	**No**	**yes**	**No**	**Partially**	**No**	**No**	**No**	**No**	**No**	**No**	**No**
Anchor Node failure Handling	**No**	**No**	**No**	**Partially**	**No**	**partially**	**Partially**	**Partially**	**No**	**No**	**Partially**	**No**	**No**
LS Recovery	**No**	**No**	**No**	**No**	**No**	**No**	**No**	**No**	**No**	**No**	**No**	**No**	**No**

2.4 Problem Statement

A detail description over the classification, functionality, features and limitations of DHT routing protocols in MANETs is given in Section 2.3. In this section, we discuss the requirements, challenges and open issues that are critical to be addressed by DHT routing protocols to operate in the adversarial environment of MANET. Based on the identified research gap, we are providing the problem statement.

In DHT-based networks a logical structured network is built over the underline ad hoc physical topology of MANETs. In other words, a structured logical space (LS) (i.e. the LS follows a tree, cord or multi dimensional structures etc.) is distributed among the nodes in the network bootstrapping. Each node in the network maintains a logical space portion (LSP) of the LS and contributes in maintaining an evenly distributed LS throughout the network. The LSP of a node acts as its LID (i.e. transient address). Routing is performed in the logical space constructed using LIDs. Therefore, an evenly distributed and connected LS is always needed for routing. Furthermore, certain nodes in the network store index information (i.e. LID and UID pair) on the behalf of others nodes and act as anchor nodes. Each node stores its index information on the anchor nodes in the joining process (i.e. this process is called Anchor_Request or address publication). A source node retrieves the index information stored on the anchor node to obtain destination's LID before starting communication (this process is known as lookup or address resolution). Therefore, the Anchor_Request and lookup processes are the intrinsic parts of DHT-base routing in MANET. However, MANETs are fully self organized, mobile and infrastructure-less networks. Nodes, with limited transmission ranges, use wireless broadcasting mechanism in a multi-hop fashion to achieve end-to-end connectivity. Limited radio range, mobility, faulty nodes and lack of infrastructure introduce frequent and unpredictable changes to network topology (i.e. connectivity/ dis-connectivity, node(s)/link(s) failure, partition, merging). The network dynamics severely damage the logical structured network built over the ad hoc physical topology of MANET and completely halt communication. We explore the impact of network dynamics on DHT networks and identify following problems.

Problem-I: We found that nodes/links failure raise the issue of anchor nodes failure in DHT networks. The anchor nodes hold index information and play vital role in the routing process. The anchor node failure causes the failure of address resolution process (i.e. lookup). This completely halt communication between the source and destination. Therefore, mechanisms are needed to recover the lost index information stored on the anchor nodes. Chapter 3 elaborates the impact of node/link failure on DHT networks and provides distributed solutions.

Problem-II: We study the impact of network partition on DHT networks. We found that network partition elicits three major problems in DHT networks 1) unavailability of the anchor nodes and, 2) Loss of LS, 3) and need partition detection event. All these problems are elaborated in Chapter 4.

1. Unavailability of the anchor nodes occurs when a DHT-based network is partitioned into two disconnected partitions such that both the source and the destination nodes are in the same network partition, but the corresponding AN remains in a disjoint partition. In this case, despite that the source and the destination nodes are reachable, but unable to communicate due the unavailability of destination node's AN.

2. Network partition severely damages the logical space (LS) structure built over the ad hoc topology of MANET and results into an unevenly distributed and disrupted LS(s) in the network instances after partition, such that even physically connected nodes in the disjoint part would not be able to communicate with each other. The lost LS due to partition can be reused in the disjoint partitions. The LS recovery ensures evenly distributed LS(s). Similarly, the lost LS recovery, also, sets grounds for smooth

merging. For detail please see Chapter 4.

3. The LS recovery and reusing need timely detection of the partition event. Therefore, partition detection is another correlated problem.

Problem-III: The successful resolution of lookup requests with minimum delay is always needed in DHT networks. However, we found that existing DHT protocols suffer from longer lookup delay and network dynamics exacerbate it further. Therefore, reducing lookup delay is another ultimate goal of this work. A complete detail is provided in Chapter 3.

Existing DHT-based routing protocols lack these capabilities and can not be used in the adversarial environment of MANETs. We are using the philosophies of detection, replication and recovery in a distributed manner to solve the identified problems.

Chapter 3

Dynamic Replication and its impact on DHT Routing in MANETs

3.1 About the Chapter

In this chapter we explore the problems raised by network dynamics in DHT-based MANETs and provide fully distributed solutions for the identified problems. In particular, we consider node(s)/link(s) failure and its impact on routing, lost information recovery in case of anchor node failure, access to the index information in case of network partition, longer lookup delay and the address publication process. We are using the philosophies of detection and replication to overcome the failures etc. Specifically, a novel Anchor_Request(also known as Address_Publication) is proposed in DHT-based Routing paradigm for MANETs. The proposed solution advantages are two fold, fist it ensures recovery of the lost index information due to anchor node failure and network partition, secondly it brings significant gains in terms of lookup delay and lookup success ratio. Furthermore, the chapter is based on the published results in the reference [3].
Note: The terms index information and mapping information are used interchangeably and convey similar meaning. Similarly, the control packets Store Index Information (SII) and Store Mapping Information (SMI) are similar.

3.2 DHT-based Routing in MANETs

In DHT-based MANETs, routing completes in two phases i.e. Anchor Requests phase/Address Publication and Lookup request phase/Address Resolution (for detail please see Section 1.2.1 in Chapter 1). These two phases define overall network performance of the DHT routing protocols, especially in MANETs. However, limited transmission range, mobility, self organizing nature and lack of infrastructure in MANETs introduce issues such as node/link failure, partitioning and frequent merging. The issues raised by network dynamics have a severe impact on DHT routing in MANETs.

In DHT-based networks a logical structured network is built over the underline ad hoc physical topology. In other words, the ad hoc physical topology is embedded into a logical structured network following tree, cord or multidimensional structures. The relation between physical network and logical network is tight, in other words, a change in the physical toplogy has a direct impact on the logical network. Therefore, changes in the physical network should be immediately incorporated into the logical network. For optimal network performance, the logical network construction must have to accommodate the changes in the physical networks from time to time. As discussed earlier, MANETs are fully self organized, mobile, distributed and infrastructure-less networks. Nodes, with limited transmission ranges, in the network use wireless broadcasting mechanism in a multi-hop fashion to achieve end-to-end connectivity. Limited radio range, mobility, faulty nodes and lack of infrastructure introduce frequent and unpredictable changes to network topology i.e. connectivity/dis-connectivity. Similarly, node(s)/ link(s) failures are common in MANETS due to mobility, limited transmission range, resource constraints, selfish behavior etc. In such adversarial environment, building and maintaining a logical structured network (i.e. DHT network) is a crucial and challenging task. In DHT networks, a connected and evenly distributed logical address space (LS) (i.e. logical structured network) is always needed for smooth network operations. The network dynamics severely damage the logical network structure and result an unevenly distributed and disrupted LS. A detail explanation of the problems raised by network dynamics is provided in Section 3.3. Therefore, mechanisms are needed to tolerate the faults and to ensure the successful completion of the routing process phases (i.e. Anchor_Request and lookup phases) in DHT-based MANETs

We use detection and replication philosophies to minimize the impact of network dynamics (i.e. failure, partition etc.). The employed distributed mechanisms ensure successful resolutions of lookup requests and significantly reduce lookup delay.

3.3 The Problems raised by Network Dynamics

In DHT networks, certain nodes in the network store index information and act as anchor nodes. The failure of anchor node may completely halt communication between a source and destination pair, despite the fact that source and destination are connected (since index information, stored on the anchor node, are always needed before starting communication). It is found that anchor node fails in two cases: i) due to network partition. ii) due to node/link failure or movement.

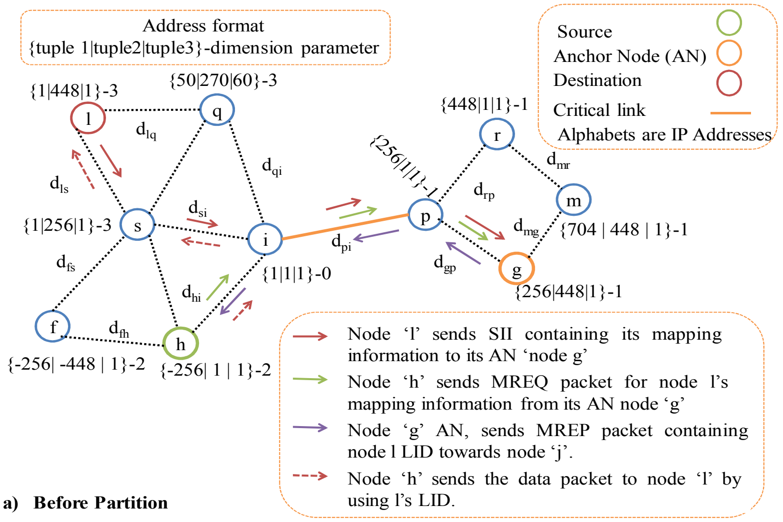
Similarly, successful resolution of the lookup requests with minimum possible delay in such adversarial environment is another challenging and correlated problem. A detail description of the identified problems is given in the subsequent sections.

3.3.1 Unavailability of an Anchor Node in both partitioned and unpartitioned Network

First we need to identify whether the leaving node is causing partition or just leaves without creating partition, therefore , the sub problem of partition detection, is also needed to be addressed (please see Chapter 4 for partition detection).

In DHT-based MANETs, newly joining nodes compute LIDs in addition to its permanent identifier (i.e., MAC/IP address, also known as user id (UID)) and store index information((LID, UID) pair) , also known as mapping information, on certain nodes in the network called anchor nodes (AN). The AN holds the index information (or mapping information) of various other nodes in the network and a source requires the mapping information to communicate with a destination node. The process of storing index information on ANs is known as Anchor_Request or Address_Publication. Similarly, the process of retrieving mapping information from the anchor node is called lookup or address resolution. The resolution of lookup queries is mandatory to initiate the routing process. However, unavailability of the destination node's AN completely disrupts the communication between the source and the destination node. Since, the availability of the destination node's AN is imperative for the uninterrupted communication in DHT-based routing protocols.

Inaccessibility of AN occurs when a DHT-based network is partitioned into two disconnected partitions such that both the source and the destination nodes are in the same network partition, but the corresponding AN remains in a disjoint partition. In this case, despite that the source and the destination nodes are reachable, but unable to communicate due to inaccessibility of the destination node's AN. Figure 3.1(a) illustrates the situation. Nodes LIDs in Figure 3.1 are computed using [2, 48](for complete detail over LID assignment please see Chapter 2 and Figure 2.12). After joining the network, node *l* publishes its mapping information (i.e. (LID, UID) pair) by sending Store Index Information(SII) packet and stores on the AN i.e. node *g*. Node *g* being logically closest to node *l* starts acting as AN for node *l*. To communicate with node *l*, the source node *h* retrieves the index information of *l* from *g* using Mapping Request(MREQ) packets. After receiving the index information, node *h* directly starts its communication with node *l*. For instance, if the link $i \leftrightarrow p$ breaks and the network partitioning occurs as shown in Figure 3.1 (b). The source node *h* and the destination node *l* are in the same partition, but the corresponding AN (i.e. node *g*) remains in the disjoint partition. In such a case, node *h* is unable to retrieve the mapping information of the node *l* from its AN (i.e. node *g*), which would disrupt the communication between node *h* and node *l*. Moreover, a new lookup request for the node *l*'s mapping information would not be resolved until *l* selects a new AN and updates its mapping information there. This would



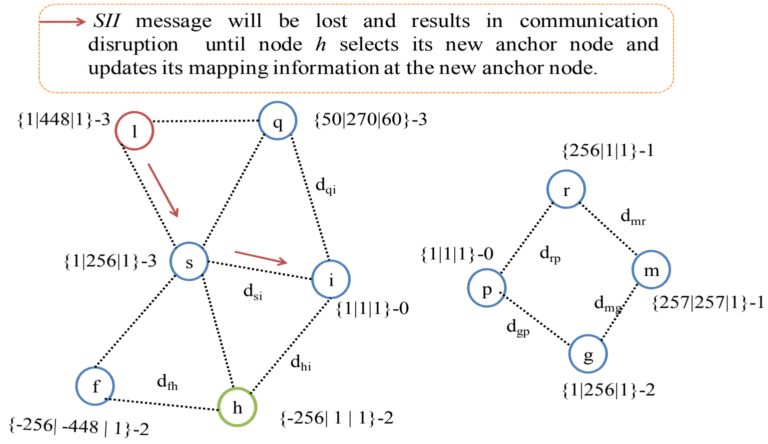
Node ‘l’ sends SII containing its mapping information to its AN ‘node g’

Node ‘h’ sends MREQ packet for node l’s mapping information from its AN node ‘g’

Node ‘g’ AN, sends MREP packet containing node l LID towards node ‘j’.

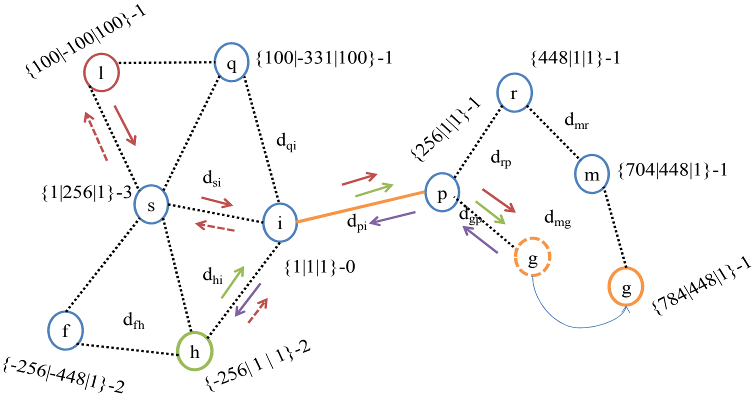
Node ‘h’ sends the data packet to node ‘l’ by using l’s LID.

a) Before Partition



SII message will be lost and results in communication disruption until node *h* selects its new anchor node and updates its mapping information at the new anchor node.

b) After Partition



c) Anchor node moves and its LID has been changed

Figure 3.1: Network Partitioning in a DHT-based Routing Protocol(i.e. 3D-RP[2])

cause the longer lookup delays and information loss.

Similarly, when an existing AN fails/moves without creating network partitioning, it is crucial to ensure the availability/recovery of the mapping information stored at that AN. In Figure 3.1 (c), the anchor node *g* moves away from node *p* and obtains a new LID. In this case, node *l* would select a new AN in order to store its mapping information. Meanwhile, the in-transit lookup queries might get lost and would result in communication disruption, information loss and longer lookup delays.

3.3.2 Lookup and Longer Delay

The successful resolution of the lookup request (i.e. retrieving index information stored on anchor nodes) is imperative for routing in DHT MANETs. Successful resolution of the lookup queries need guaranteed access to the index information (i.e. anchor nodes). However, network dynamics (i.e. node/link moves, fails, partition etc) make it uncertain to resolve the lookup queries. Therefore, mechanisms are needed for the successful resolutions of lookup requests despite the failures.

Furthermore, network dynamics exacerbates the lookup delay problem. Therefore, minimizing lookup delay in the current setup is another challenging and correlated problem.

The Anchor_Request and Lookup_Request define overall network performance in DHT networks. However, as discussed earlier, the adversarial environment offered by MANET makes completion of the processes in questions. Our proposed solutions solve the problems in fully distributed manners using the philosophies of detection and replication.

3.4 Proposed Solutions

We are using the philosophies of detection and replication in a fully decentralized manner to address the problems. For this purpose, we measure network dynamics (i.e. network connectivity/dis-connectivity) and identify/declare critical regions (i.e. critical nodes/links) in the network. It is found that the identified critical regions are more prone to failure. For critical node(s)/link(s) identification, we are exploiting nodes relative connectivity (i.e. 2-hop neighbor information) and local network variation (i.e. node degree). Nodes in the network exchange topological information and dynamically react against the changes take place in the network. The proposed mechanisms replicate index information across the critical regions along the path to anchor node in the address publication process. This makes sure access to the index information despite the failures and reduces lookup delay.

For example a node *u* is said to be critical with respect to node *v*, if the neighbors of node *u* excluding node *u*, remains unreachable to node *v* if node *u* moves/fails and vice versa. Link between critical nodes is treated as critical. To compute critical node(s)/link(s), we are using *k*-hop topological information. Complete detail of critical node/link is provided in Chapter 4 along with algorithmic detail.

3.4.1 Anchor_Request and Dynamic Replica Deployment

We are exploiting the existing *hello* messages, used for LID computation, in DHT networks for dynamic replica deployment. Each node in the network periodically exchanges *hello* messages with its 1-hop neighboring nodes after a pre-defined *hello interval*. The *hello* message of a node *x* contains the list of *x*'s 1-hop neighboring nodes in addition to its LID and UID. By exchanging the *hello* messages, each node maintains the connectivity information of up to 2-hops physical neighboring nodes. Based on the exchanged 2-hop topological information each node computes its status as *critical* or *non-critical*. Algorithm 4.1 in Chapter 4 computes critical nodes/links in a decentralized manner. A critical node or the nodes across the critical link sets their status as *critical*. Otherwise, the nodes status remains as *non-critical*. Each node announces its status (i.e. *critical/non-critical*) to its 1-hop neighboring nodes through the *hello* messages.

The proposed replication strategy targets critical regions (i.e. critical nodes/links in the network). Critical regions are more prone to failure (i.e. AN failure, partition etc), therefore, a trivial way is to replicate the index information across the critical regions, so that, index information would remain reachable despites actual anchor node fails or moves/partition etc. This makes sure guaranteed access to index information in the adversarial environment. In the address publication process (i.e. Anchor_Request), a newly joining node publishes its index information(i.e. (LID, UID) pair) to store on certain nodes in the network i.e. anchor nodes. we replicate the index information across the critical regions(critical nodes/links) along the path to the intended anchor node. Particularly, consider a newly joining node *j*, after computing LID node *j* publish and stores its index information(LID, UID pair) on anchor node (AN). Node *j* determines its AN by applying a hash function over its UID, say *v*, generating a hashed value, say *h(v)*, and stores index information on a node (i.e. A.N) with LID closest to *h(v)*. For address publication, node *j* sends SII packets to be routed towards the intended AN. we replicate index information across all the critical links/nodes along the path to AN. This makes sure access to the index information in case of AN failure, partition etc. In the lookup process, nodes in the network would be able to access node *j*s index information from the replicas despite the actual AN. Algorithm 3.1 performs the replications decisions using *k*-hop topological information.

Formally, let *U* be the set of nodes in the network and $M_k \subseteq U$ are 1-hop neighboring nodes of *k* excluding node *k* (i.e. open neighborhood) , where $M_k^+ = M_k \cup \{k\}$, M_k^+ refers to the closed neighborhood of *k* (i.e., including node *k*). In network bootstrapping, each node computes LID and publish index information by sending SII packets in the network. When the SII packet is received at a node *k* from *s* along the path to the intended anchor node AN, node *k* performs the replication and forwarding decisions based on the outcome of function *f*.

$$f(M_k^+, AN) = \begin{cases} \text{k}, & \text{if} \quad d(k, AN) = min_{j \neq s \in M_k^+}(d(j, AN)), \quad else \\ \exists i \neq s \mid i \in M_k \wedge d(i, AN) = min_{j \neq s \in M_k}(d(j, AN)) \end{cases}$$

The function f at k checks the logically closest neighboring node to the intended AN by computing the least sum of difference of their LIDs for storing the index information. If *k* finds itself as the logically closest among its 1-hop neighboring nodes, then it stores the mapping information in SII and stops forwarding the SII further. Thus, the node *k* becomes the designated AN for SII. If there exists another node $i \neq s \in M_k$ such that the logical distance between *i* and the intended AN is least among the 1-hop neighboring nodes of *k*, then *k* would forward the SMI towards *i*. The distance function computes the logical distance and depends on the underline logical structure. Before forwarding SMI, the replication of index information is achieved by the following decision choices based on the status (i.e., critical/non-critical) of the node *k* and its neighborhood:

1. If *k* is *non-critical* and function *f* returns *k*, then node *k* stores the index information in SII message and acts as the designated AN. If function *f* returns *i* then *k* forwards SII towards *i* without making the replica. In this case, the status of node *s* and node *i* do not effect the handling of SII.

2. If *k* is *critical* with respect to $p \in M_k$, then the index information in SII is stored at the

nodes k and p across the critical link $k \leftrightarrow p$.

Algorithm 3.1 illustrates the proposed address publication process/Anchor_Request. The mechanism ensures availability of the index information in disjoint partitions. This would avoid the communication disruption within partitions, but also reduces the lookup delay to retrieve the index information even in case the critical link fails. Moreover, the proposed algorithm significantly reduces the lookup delay even network partitioning does not occur because the nodes would easily be able to retrieve the mapping information from the deployed replicas besides the actual AN.

Algorithm 3.1: Anchor_Request

Required: When *SII or SMI* message is received at node k from node s along the path to intended AN, where $M_k \subseteq U$ an
$M_k^+ = M_k \cup \{k\}$ denote 1-hop neighbors of node k excluding k and including k respectively and U is the set of all nodes in the network. Function $f(M^+_k, AN)$ makes forwarding decisions.

```
On the reception of SII at a node k from node $s \in M_k^+$
if ( $f(M^+_k, AN)$ returns k )              then
    if ( STATUS(k) == non-critical )            then
        $k \leftarrow SII$                  // Store index information and node k acts as designated AN.
    end-if
    if ( STATUS(k) == critical w.r.t $p \in M_k$ )     then
        $k \leftarrow SII$                  // p can be either s or any other node
        $p \leftarrow SII$                  // replicate SII across the critical link $k \leftrightarrow p$.
    end-if
end-if
if ( $f(M^+_k, AN)$ returns $i \in M_k$ )          then
    if ( STATUS(k)== non-critical )             then
        Forward SII to node i w.o.t making any replica.
    end-if
    if ( STATUS(k)== critical w.r.t node i )      then
        $k \leftarrow SII$                  // store index information across the critical ($k \leftrightarrow i$).
        $i \leftarrow SII$
    end-if
end-if
```

3.4.2 Impact on lookup and lookup delay

It is observed that dynamic replication significantly reduces lookup delays and improves lookup success ratio. The replication strategy targets critical regions (i.e. critical nodes/links in the network). Critical regions are more prone to failure (i.e. AN failure, partition etc), therefore, a trivial way is to replicate the index information across the critical regions, so that, index information would remain reachable despites actual anchor node fails or move/partition etc. This makes sure guaranteed access to index information in the adversarial environment and significantly improves lookup success ratio. Similarly, nodes can find index information from the deployed replicas despites of the actual AN, therefore, significantly reduces lookup delays.

3.5 Illustrative Example

In Figure 3.1, the address publication process of node *l* is shown. After computing LID node *l* publish its index information by sending SII packets. As shown in the figure, the SII is forwarded to node *g* (i.e. designated anchor node for node *l*) by following the path $\{s, i, p, g\}$. The proposed forwarding function replicates index information (i.e. SII) across all the critical links along the path to node *g*. In the given scenario of Figure 3.1, node *i* and *p* are critical nodes across the critical link $i \leftrightarrow p$. Therefore, node *i* keeps copy of SII before forwarding. In case of link $i \leftrightarrow p$ failure, a source node *h* can find index information from the replica stored on node *i*, despite that actual anchor node i.e. node *g*. In this manner, the proposed mechanisms ensure access to the index information despite anchor node failure or partition.

3.6 Chapter Summary

The problems of anchor node(i.e. anchor nodes store index information needed for routing in DHT networks) failure and unavailability of index information due to node/link failure or network partition are solved in a fully distributed manner. Similarly, longer lookup delay in the routing process is another correlated and challenging problem. The proposed novel Anchor_Request replicates index information across the critical regions in the network. For this purpose, the proposed algorithm computes critical node(s)/link(s) using *k*-hop topological information. In the address publication process (i.e. Anchor_Request), a newly joining node dynamically replicates its index information based on the identified critical node/link positions. Moreover, the replication mechanism significantly reduces the lookup delay and increases lookup success rate. Performance evaluation based on the simulation results comparisons confirms the effectiveness of the proposed solutions (please see Chapter 5 for simulation and results analysis).

Chapter 4

Distributed Partition Detection and Merging

4.1 About the Chapter

In this chapter, we explore the problems raised by network partition in DHT-based MANETs. Specifically, the problems of detecting critical nodes/links, partition detection event, logical space (LS) recovery and reusing are addressed. Similarly, smooth merging needs evenly distributed LSs, therefore, prior to network partition, LS recovery is needed for smooth merging. we are exploiting the philosophies of detection and recovery using *k*-hop topological information to address the mentioned problems. This chapter is based on the published results in [3].

4.2 Network Partition and merging in DHT-based MANETs

Network partitioning is the process of breaking-down a connected network topology into two or more than two disconnected parts such that nodes in one partition can not access nodes in another partition [50]. Partition causes multiple instances of the same network to coexist. Similarly, merging is the process of combining disjoint parts of the network into a single connected network.

In DHT-based MANETs, a logical structured network (i.e. following cord, tree, 3D structures etc.) is built over the underlined ad hoc physical topology of MANET. In network bootstrapping, nodes compute LIDs and form logical structured network. In other words, a structured logical space (LS) is distributed among the nodes in the network. In this manner, a logical space portion (LSP) is maintained by each node and is used as a logical identity (i.e. logical identifier(LID)). In this way, every node in the network contributes in building and maintaining a uniformly distributed LS. In DHT networks, routing is performed on the logical structures built, using nodes LIDs. For optimal performance, a complete, evenly distributed and connected LS is always needed. However, the decentralized nature, limited transmission range, mobility etc. in MANETs introduce frequent network partition and merging.

The partition of physical network topology causes logical network (LS) partition [4][55]. This divides the LS into multiple instances such that each instance holds a portion of the LS. This results into an unevenly distributed and disrupted LS(s) in the disjoints parts of the DHT network after partition. However, successful routing needs complete and uniformly distributed LS (i.e. logical structures such as cord, tree, 3D etc). Therefore, nodes even physically connected within the disjoint parts remain unable to communicate with each other (i.e. due to disrupted and uneven LS(s)). We believe that when a network gets partitioned into disconnected disjoint parts, the LIDs (i.e. lost LS) of one partition can be reused in another partition. This would result into evenly distributed LS(s) in the disjoint partitions and would enable communication in the disjoint partitions.

Furthermore, Network partition also interrupts access to the index information stored on the anchor nodes. A complete detail on anchor node failure due to network partition along with proposed solution is given in Chapter 3.

Similarly, network dynamics cause frequent merging. Simple merging of the physical networks topologies does not solve the problem and results into physically connected but logically disconnected network [55]. In DHT networks, merging needs extra effort to merge the logical structures (i.e. LSs). But smooth merging needs/assumes uniformly distributed LS(s) in the partitions to be merged. But, the loss of LS due to network partition prohibits smooth merging. We believe that recovery of the LS (lost due to partition) is imperative for smooth merging.

Furthermore, it is found that LS recovery needs timely detection of the partition event. Therefore, detection of the partition event is another important sub-problem in this specific domain. We are using the philosophies of detection and recovery in a fully distributed manner to solve the identified problems.

A complete description of the identified problems with diagrammatic representation is given in Section 4.4.

4.2.1 The Problems

It is concluded from the above discussion that network partition in DHT-based MANETs elicits two major problems: *i*) unavailability of anchor node (a complete discussion with proposed solution is given in Chapter 3). *ii*) Loss of LID space (LS). It is found that LS loss further elicits three sub-problems.

- LS loss in network partition causes uneven and disrupted LSs in the created network instances, so that, physically connected nodes in the disjoint instances of the partitioned network might not able to communicate with each other, because successful routing needs uniformly distributed LS.
- Recovery of the lost LS due to logical network partition needs timely detection of the partition event in the physical network. Therefore, partition detection is another sub-problem to be addressed for LS recovery.
- Smooth merging needs evenly distributed instances of the same type (i.e. complete LS(s) of similar type). But due to LS lost in partitioning, evenly distributed instances might not exist. This prohibits smooth merging process to be accomplished by the merging algorithms. Therefore, the loss of LS during partition has a direct impact on merging as well. It is worthy to mention that existing merging techniques [6][55] in DHT networks hold this assumption (i.e. assume that partitions going to merge are

uniformly distributed). But this assumption do not hold due to the loss of LS during partitioning. This limits the use of these protocols in practice.

4.3 Proposed Solutions

We consider pre-partition measures such as critical node/link detection, replication and recovery to solve the identified problems raised by network partition in DHT-based MANETs. The proposed mechanisms exploit *k*-hop topological information (i.e. network connectivity/ dis-connectivity information) in a fully distributed manner. Based on the *k*-hop information we identify and declare certain nodes/links as *k*-hop critical nodes/links. The concept of *k*-hop critical nodes/links helps us in identifying critical regions (i.e. prone to failure) in the network and in the detection of partition event. The timely detection of the partitioning event in the physical topology before actual network partitioning allows the induction of pre-partitioning measures such as replication of index information (see chapter 3 for detail on replication) and LS recovery & reusing in the logical network. We recover the lost LS and reuse it in the disjoint partitions, and this results into uniformly distributed LS(s) in the disjoint partitions and, also enables communication in the disjoint partitions. Similarly, the LS recovery process sets ground for smooth merging of the disjoint instances of the network. Algorithms 4.1 and 4.2 solve the identified problems in a fully distributed manner. Please see the subsequent sections for further detail.

4.3.1 Critical Node(s)/Link(s) Detection

We are using the concept of critical node(s)/link(s) to handle network partition at the network layer in DHT-based MANETs. To improve the performance of DHT-based routings, it is imperative to identify/detect the critical links/nodes that instigate partitioning in the physical network. This would lessen the information loss and communication disruption in case a network gets partitioned. Critical link refers to a link whose failure/disconnection would lead to network partitioning whereas critical node refers to a node whose failure would result in network partitioning. Figure 4.1 elaborates both the scenarios. A distributed way of identifying a critical link or a critical node is through its common k-hop neighbors. A link $s \leftrightarrow b$ is said to be k-hop critical as shown in Figure 4.1 if and only if node s and node b has disjoint neighbor set assuming that the link between node s and node b does not exist. For k=1, the $s \leftrightarrow b$ link is said to be 1-hop critical as 1-hop neighbor set of nodes s and b are disjoint. For k=2, the $s \leftrightarrow b$ link is said to be 2-hop critical as there exists no common neighbor between 2-hop neighbors of nodes s and b. The link $s \leftrightarrow b$ in Figure 4.1 is a globally critical link because even if the value of k is increased, there would not be any common neighbor between node s and b. Similarly, node e in Figure 4.1 is globally critical node as its neighbors can be divided into two sets that are disjoint. In general, if nodes across a link are critical

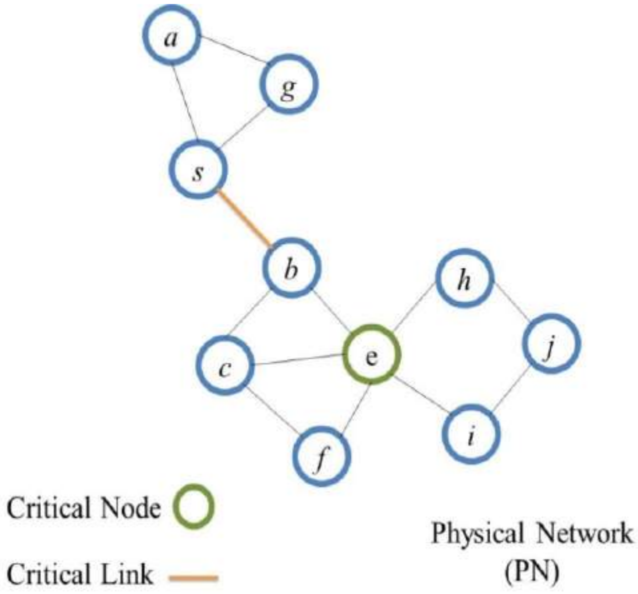


Figure 4.1: x-hop critical link and critical node scenario

with respect to each other then the link is treated critical as well.

Algorithm 4.1: Critical_Node/Link_Detection

Required: Each node in the network exchanges the list of its *k*-hop neighbors in the periodic *hello* message.

```
Each node u periodically broadcasts its closed neighborhood $N_k[u]$ and waits to listen messages from other nodes.
for each node u in the network do
    for each node $v \in adj[u]$ do
        If ( ( $N_k[u] \cap (N_k[v] \setminus u,v) = \{\}$ )
            STATUS (u) ← critical
            STATUS (v) ← critical
            link(u, v) ← critical
        else
            STATUS (u) ← non-critical
            STATUS (v) ← non-critical
        end-if
    end-for
end-for
```

/* For any two adjacent neighbors (u, v) across the link (u,v), where $N_k[u]$ and $N_k[v]$ denotes k-hop closed neighborhood of the adjacent nodes u and v respectively. */

Formally, *k*-hop neighbors of a node can be expressed as neighborhood. For a network $G= (V, E)$, the neighborhood of a vertex $v \in V$ is the induced sub-graph of G consisting of all *k*-hop vertices adjacent to *v* and all edges connecting such vertices. Similarly, the open neighborhood, $N_k(v)$, of the vertex *v* consists the set of *k*-hop vertices adjacent to *v* excluding *v*, i.e. $N_k(v)= \{w \in N : vw \in E\}$ and the *k*-hop closed neighborhood of *v* is $N_k[v]= N_k(v) \cup \{v\}$. Nodes in the network exchange the closed neighborhood in the periodic *hello* messages. For any two adjacent nodes $u \in V$ and $v \in V$ across the link(u,v), if there exist common nodes by comparing $N_k[u]$ and $N_k[v]$ excluding *u* and *v* (i.e. $(N_k[u] \cap N_k[v] \setminus u,v) \neq \{\}$), then node *u* and node *v* declare their status as *non-critical* otherwise *critical*. Link between two critical nodes is treated as *critical*. In this manner, Algorithm 4.1 computes critical node(s)/link(s) in a fully distributed manner.

We consider relative network connectivity (i.e. 2-hop neighbor information) and local network variation (i.e. nodes degree) to detect/identify the critical link(s)/node(s) and to replicate the index information based on the location of critical node(s)/link(s) (please see Section 3.4.1 in Chapter 3). For this purpose, nodes are needed to exchange the list of directly connected neighbors (i.e. 1-hop neighbors). This provides a network view up to 2-hop neighbors. There exist several approaches [56][57] in the literature to compute critical nodes/links. We use this approach for critical node(s)/link(s) identification due to two reasons:

- This approach is distributed and scalable in nature and complies with DHT-based routing protocols.
- This approach relies on existing *hello* messages that are exchanged among nodes for the LID computation in DHT-based protocols, thus our approach does not incur extra message overhead.

4.3.2 Distributed Partition Detection and LS Recovery

The critical node(s)/link(s) are prone to failure and instigate network partitioning in the physical topology. Therefore, we are using the concept of critical nodes/links to handle network partition at the network layer in DHT-based MANETs. The employed mechanism detects the partition event in physical network and adaptively makes arrangements (i.e. LS recovery and reusing) in the logical network. This avoids communication disruptions in the logical networks despite the network partitioning. The LS recovery and reusing ensure network availability (i.e. evenly distributed LS(s)) in the disjoint parts after network partition.

Algorithm 4.2 detects the partition event in the physical network and recovers/reuses the lost LS in the logical network. The algorithm detects critical links across the network using *k*-hop connectivity/dis-connectivity information in a distributed manner. For this purpose, nodes in the network exchange (*k-1*)-hop neighbors with directly connected nodes in the periodic *hello*, besides other information i.e. LID, UID. In our current setup (for *k*=2), nodes are needed to exchange 1-hop neighbor list. In such a way, every node maintains 2-hop topological information. we express the k-hop neighbors of a node as its neighborhood. As discussed earlier, for a network $G=(V, E)$, the neighborhood of a vertex $v \in V$ is the induced sub-graph of G consisting of all k-hop vertices adjacent to v and all edges connecting such vertices. Similarly, the open neighborhood, $N_k(v)$, of the vertex v consists the set of k-hop vertices adjacent to v excluding v, i.e. $N_k(v)= \{w \in N : vw \in E\}$ and the k-hop closed neighborhood of v is $N_k[v]= N_k(v) \cup \{v\}$. Every node exchange neighbors list i.e. closed neighborhood, in the periodic *hello* messages. Therefore, node u and node v periodically exchange $N_k[u]$ and $N_k[v]$ respectively. For any two adjacent nodes $u \in V$ and $v \in V$ across the link $u \leftrightarrow v \in E$, if there exist no common node(s) by comparing $N_k[u]$ and $N_k[v]$ (i.e. $(N_k[u] \cap N_k[v] \setminus u,v) = \{\}$ or $(N_k(u) \cap (N_k(v)) = \{\}$), then node u and node v declare their status as *critical*. Link between critical nodes is treated as *critical*. If the critical node u across the critical link $u \leftrightarrow v$ do not hear from node v for a certain time interval, we call it *Partition_Timer*, then it means that the network has partitioned. *Partition_Timer* is set to three times of *hello* interval, (assumption: congestion-free network with no packet loss). Similarly, node v will also detect network partitioning. In case of network partition, nodes across the critical link recover the lost LS for reusing in the disjoint partitions.

LS recovery and reusing mechanisms depend on the logical structure used and on the way how to implement it . For example in Chord structure [31, 32], in case of partition detection, nodes across the critical link recover the lost LS by changing their LIDs to S and E based on their relationship i.e predecessor/successor. The predecessor node in the logical chord recovers LS by changing its LID to E and the successor node in the chord recovers LS by changing its LID to S. This results complete (i.e. uniformly distributed LS) cords in the disjoint parts of the partitioned network. Similarly, in tree type structures [1, 36], nodes across the critical link recover the lost LS by electing new roots in the disjoint partition.

For this purpose, node with lowest LID in the disjoint parts is elected as root. In a 3D structure[2] nodes across the critical links recover the lost LS by changing their dimensions i.e. *dim* value.

The proposed algorithm is fully distributed and relies on the local knowledge of neighbors available to nodes without network wide dissemination of control information. In the simplest implementation (for k=2, i.e., relying on the existing *hello* messages used for LID computation in DHT networks), the algorithm ensures partition detection event. Since, a global critical node/link remains critical for any value of k. The algorithm declares globally critical nodes/links as critical by using only 2-hop topological information, because globally critical nodes/links must be locally critical as well [56]. For network merging, we are using the existing techniques in [6].

Algorithm 4.2: Partition_Detection & LS_Recovery

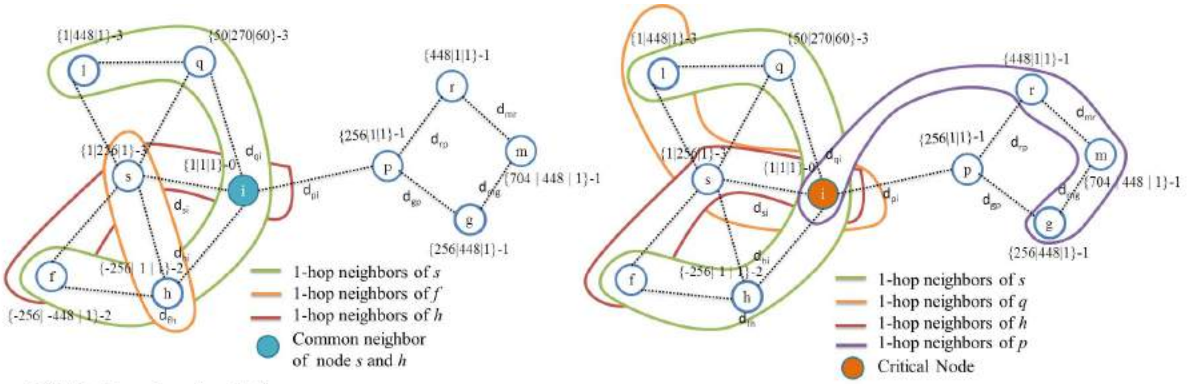
Required: Each node in the network exchanges the list of its k-hop neighbors in the periodic *hello* message.

```
Each node $u$ periodically broadcasts its closed neighborhood $N_k[u]$ and waits to listen messages from other nodes.
for each node $u$ in the network    do
    for each node $v \in adj[u]$    do
        If ( ( $N_k[u] \cap (N_k[v] \setminus u,v)$ = {} && Partition_Timer_Expired == True )
            Trigger Partition Event and
            Recover/Reuse the lost LS
    end-for  end-if
end-for
```

4.4 A Case Study: 3D-RP

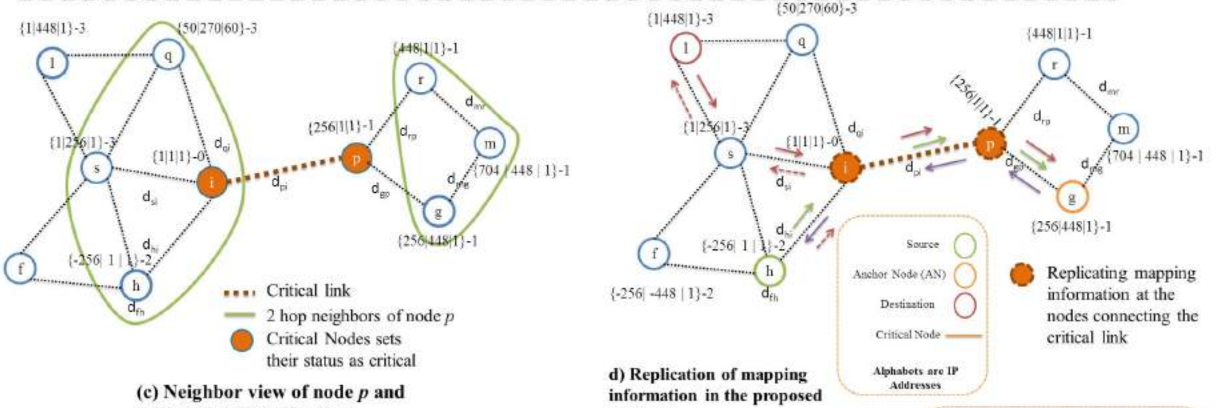
To better understand the working of our proposed mechanisms, we are conducting a case study using 3D-RP [2] protocol in DHT-based MANET. We implement the proposed distributed partition detection and replication(DDR) algorithm in 3D routing protocol (3D-RP), we call it 3DDR (i.e. 3D-RP with DDR). A detail overview of 3D-RP is given in Chapter 2. A 3D-RP based scenario is shown in Figure 4.2 to illustrate the working of 3DDR.

3DDR first identifies the critical nodes/links using the existing *hello* messages exchanged in 3D-RP. Each node in the network periodically shares its 1-hop neighbor information through the *hello* messages besides the other information. Exchanging the 1-hop neighboring information provides each node a network wide view of up to 2-hops that assists in finding the critical nodes/links of up to 2-hop disconnected network. For instance, in the Figure 4.2, 1-hop neighboring nodes of f, s, and h are $\{s, h\}$, $\{l, q, i, h, f\}$, and $\{s, f, i\}$, respectively. Upon receiving the neighbors information from the nodes s and h, the node f would have information of up to 2-hop neighboring nodes. In order to declare its status as *critical or non-critical*, node f checks for any common neighboring node among the 1-hop neighbors of fs adjacent neighboring nodes, i.e., node s and node h excluding itself. Figure 4.2 shows that there exists a node i common to the 1-hop neighboring nodes of s and h. This implies that the removal of f leaves a connected sub-graph $\{l, q, i, h, s\}$ and the node f is 2-hop non-critical. So, f declares its status as *non-critical*. Similarly, the node i receives 1-hop neighbor information, i.e., $\{s, l\}$,$\{q, l, f, h\}$,$\{s, f\}$,$\{r, g\}$, $\{r, m, g\}$ from its immediate neighboring nodes q, s, h, p, respectively. From Figure 4.2(b), it can be observed that the neighboring nodes q, s, h of i share the common nodes and form a connected sub-graph excluding the node i, however, p's 1-hop neighboring nodes are not common to the connected sub-graph of q, s, h. This makes the nodes i and p to declare their status as critical. A link between the two critical



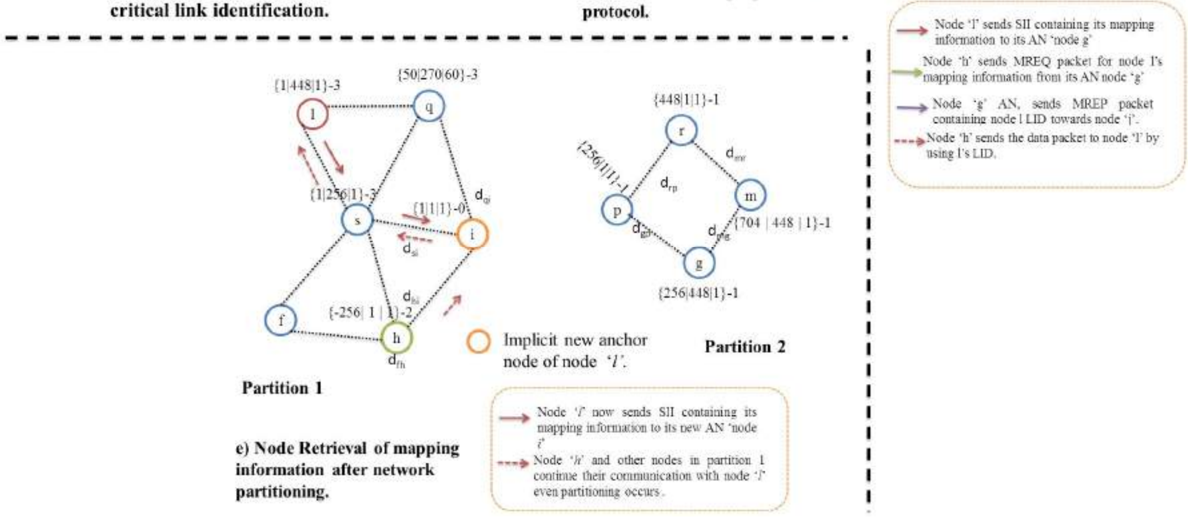
(a) Neighbor view of node *f*

(b) Neighbor view of node *i*.



(c) Neighbor view of node *p* and critical link identification.

d) Replication of mapping information in the proposed protocol.



e) Node Retrieval of mapping information after network partitioning.

Figure 4.2: Handling network partitioning using 3DDR

nodes is also critical. In this case link $i \leftrightarrow p$ is critical. Similarly, each node in the network periodically updates its status as *critical* or *non-critical*.

After computing its LID, a newly joining node, say l, forwards the store mapping information (SMI), also known as store index information (SII) message to its AN in order to store its mapping information/index information. In case if SMI passes through the critical node i, then i stores a copy of the mapping information contained in SMI and forwards it further. Thus the critical nodes i and p connecting the link $i \leftrightarrow p$ keeps the replica of the information contained in SMI as shown in Figure 4.2(d). In case if the network is partitioned, then the communicating nodes in the disjoint partitions obtain the node ls mapping information from i. In this way, the communication within the partition would remain uninterrupted even the critical link fails. For instance, consider a scenario where a source node h wishes to communicate with a destination node l as shown in Figure 4.2(c).

The source node h retrieves the node l's mapping information from g, i.e., an AN for l. In case if the critical link $i \leftrightarrow p$ fails, the mapping information would not be available to the source node h. 3DDR avoids such issues and guarantees the availability of mapping information in case of the partitioning event. Taking pre-partitioning measures such as the critical nodes/links identification and placing the replicas enable the source node h to obtain the node l's mapping information from the node i despite of the network partitioning or in case the AN fails/moves as shown in Figure 4.2(d). This also, significantly reduces longer lookup delay in the address resolution/Anchor_Request.

Furthermore, 3DDR triggers the partitioning event when the critical nodes across a critical link do not hear each other for a certain time interval, i.e., *Partition_Timer*. However, the critical nodes must be 2-hop as well as 1-hop critical, i.e., excluding a node(s) that is 2-hop critical but 1-hop non-critical. In Figure 4.2(d), when the critical nodes i and p across the critical link $i \leftrightarrow p$ do not hear each other and the *Partition_Timer* expires then the partitioning event triggers and the corresponding critical nodes recover/reuse the lost LS. In case if the critical node is the anchor node, then it also replicates its mapping information across the critical link (please see Chapter 3 for detail). The LS recovery with replica placement allows uninterrupted communication in the disjoint partitions when both the source/destination nodes reside in the same partition, but the corresponding AN remains in the other partition. The recovered LS can be reused in the disjoint partitions and may result into an evenly distributed disjoint LSs despite of the network partitioning. The LS recovery procedure varies from a case to another case and depends primarily on the logical structure of the LS, i.e., 3D-structures, Cord, Ring, Tree etc. A detail on the LS recovery is given in 4.3.2.

In general, DHT-based routing protocols in MANETs suffer from the longer lookup delays and the performance degrades as the network scales. Our scheme exploits MANET dynamicity, i.e., relative connectivity (2-hop topological information) and local network variation (1-hop connectivity information) using existing periodic *hello* messages without any extra control overhead, which ensure network availability, but also effectively reduce end-to-end delay for lookup requests.

4.5 Chapter Summary

We have proposed distributed mechanisms for critical node(s)/link(s) identification, partition detection and LS recovery. It is advocated that curtailing the information loss on the control and data planes due to network partition is imperative for the reliable communication in a DHT-based routing protocol over mobile ad hoc networks (MANETs). Both limited transmission range and the mobility of nodes cause recurrent network partitioning in MANETs that leads to inaccessibility of nodes mapping information (i.e., logical identifier (LID) and loss of LID space in a DHT-based routing protocol) and partitioning of the DHT structures. The proposed mechanisms address these problems in a fully distributed manner. Furthermore, it is discussed that smooth merging needs uniformly distributed LS(s) in the disjoint instances of similar type.

Chapter 5

Performance Evaluation and Results Analysis

5.1 About the Chapter

This chapter deals with performance evaluation and simulation details of the proposed mechanisms. We evaluate the performance of the proposed mechanisms and compare it with existing state-of-the-art protocols in DHT-based MANETs. We consider standard performance metrics for results comparisons. We are using NS2.35[58] by considering standard simulation parameters for the implementation of the proposed mechanisms. The chapter is based on the results published in [3].

5.2 Simulation Model and Parameters

We are using NS2 (version 2.35)[58] for implementation and results comparison of the proposed algorithms. The standard values for both the link and physical layers are exploited to simulate IEEE 802.11 with Two-Ray Ground as the propagation model. We have used BonnMotion2 [59] to develop the mobility scenarios that use random way-point as the mobility model and tunes the mobility parameters to stimulate moderate mobility(i.e. the node moving speeds are observed in the range from 0.5 to 2 m/s). The mobility scenarios using BonnMotion2 keep track of the physical network partitioning. Table 1 illustrates the simulation parameters in detail.

The proposed distributed partition detection and replication mechanism (DDR) is implemented in VCP[31, 32] and 3D-RP[2] to further stress the impact of DDR mechanism in other schemes. We implement DDR in 3D-RP and name it 3DDR. Similarly, we implement DDR in VCP and name it VCP-with-our-approach (VCP-WOA). The simulation results of 3DDR are compared with VCP, 3D-RP and VCP with approach (VCP-WOA). We have used CBR flows to model the data traffic and Random Traffic Model is used as the data pattern.

Table 5.1: Simulation parameters

Parameters	Their Values
Total Nodes	[50-200]
Radio range	100m
Simulation Area	1000*1000m
Data transmission	64pps
Total Time of Simulation	500 sec
Node Moving Speed	0.5 to 2 m/s
Traffic Model	Random Traffic pattern
No. of flows	12
Topology Connectivity	Bonnmotion 2
Propagation Model	Two-Ray Ground

TCP is not used at the transport layer to circumvent the elasticity effects on routing due to the TCP flow control mechanism[60]. To elude the packet drops due to congestion, the traffic load in the network is maintained at 64 pkts/sec. Following metrics are examined under the various network sizes to investigate the performance of protocols.

- Lookup Success Ratio (SR): The ratio between the total mapping request packet (MREQ) initiated and the total MREQ entertained successfully by receiving the mapping reply packets (MREP).

$$\textit{Lookup Success Ratio} = \frac{\textit{Total Lookup-Requests Initiated}}{\textit{Total lookup-Requests Successfully entertained}}$$

- End-to-end Lookup delay (E2E): The average time elapsed between when the source node initiates MREQ and the source node gets MREP.

$$\textit{End-to-End Lookup Delay} = \frac{\textit{Total Lookup Delay}}{\textit{Total Lookup-Requests Successfully entertained}}$$

  End-to-End lookup delay is measured in seconds.

- Normalized Overhead (NO): Total number of transmissions at the network layer divided by the total number of lookup queries successfully entertained. In other words, the total routing overhead divided by the total MREQ for which MREP is received.

$$\textit{Normalized Overhead} = \frac{\textit{Total Number of Transmissions at the Network Layer}}{\textit{Total Lookup-Requests Successfully entertained}}$$

Normalized Overhead shows routing overhead (i.e. no. of routing packets) per successful lookup request.

5.2.1 End-to-End Lookup Delay

The end-to-end lookup delay is an important metric for any DHT-based routing protocol. A fare analysis could not be possible if we do not consider the impact of network size and the node moving speed on end-to-end lookup delay. So, we have analyzed the end-to-end lookup delay by varying the network size and the node moving speed.

Figures 5.1 and 5.2 illustrate the average end-to-end lookup delay of 3DDR, VCP-WA, 3D-RP, and VCP with respect to the node moving speed from 0.5m/s to 2m/s with varying number of nodes in the network. The results in Figures 5.1and 5.2 show significant gains in terms of reducing the end-to-end lookup delay of 3DDR and VCP-WOA compared to VCP and 3D-RP. The main factor in improving the end-to-end lookup delay is the proposed distributed partition detection and replication mechanism that identifies the critical nodes and replicates the mapping information for seamless communication. The placement of additional replicas reduces the lookup delay by responding from a nearby replica instead of forwarding MREQ to the original AN. By dealing with the nearby replica instead of the original AN against a MREQ reduces the traffic overhead in the proposed approach. The resulting effect of minimizing the traffic overhead decreases the contention to access the medium at MAC layer in IEEE 802.11, which helps further in reducing the end-to-end

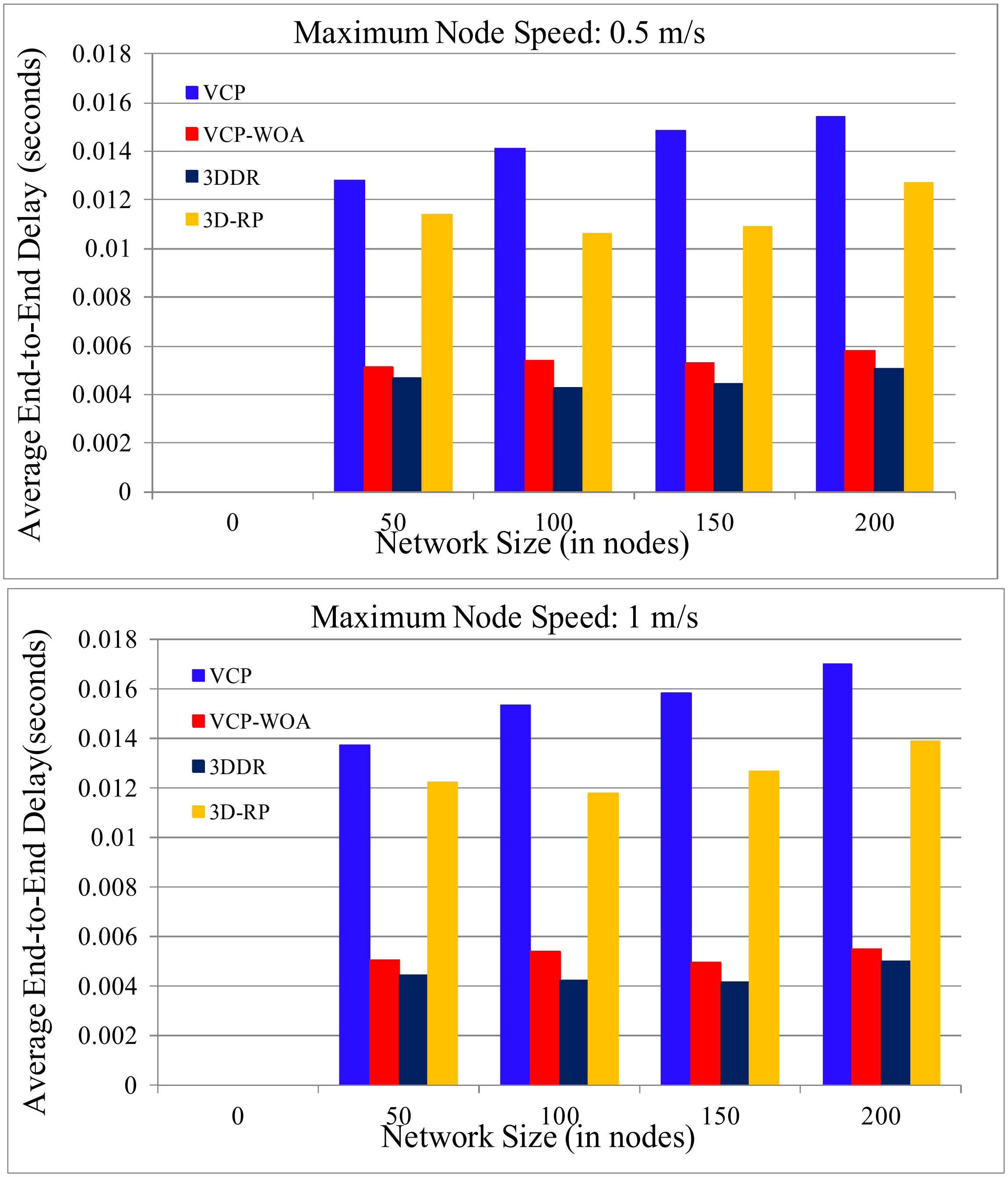
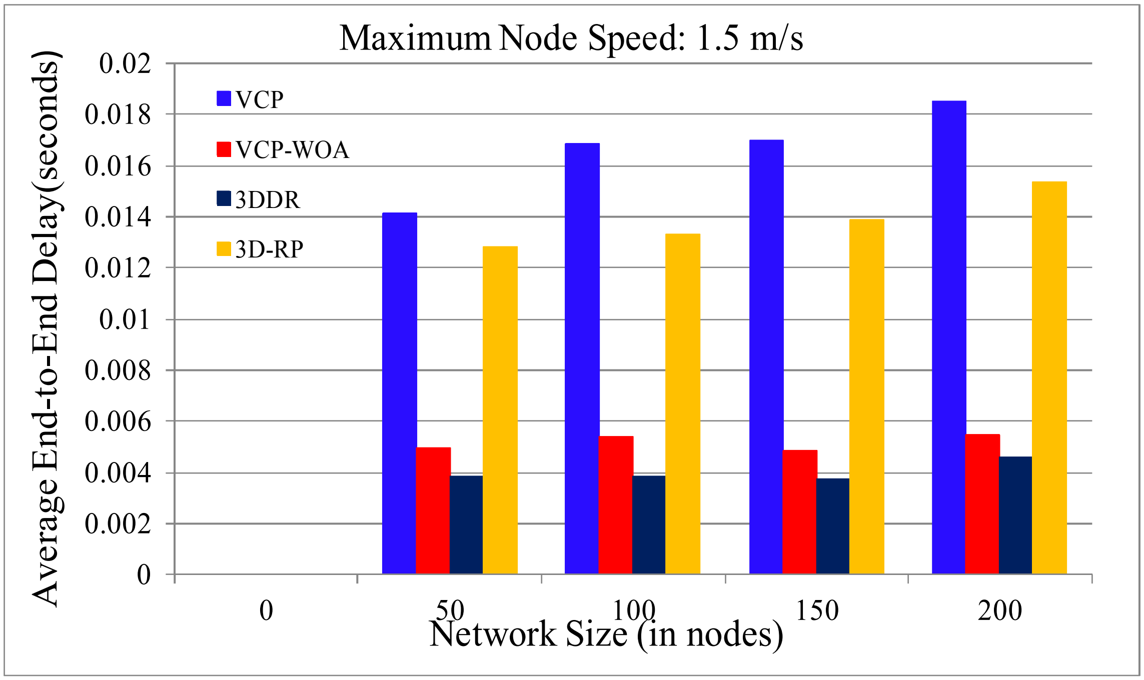


Figure 5.1: Average End-to-End delay as a function of the node number



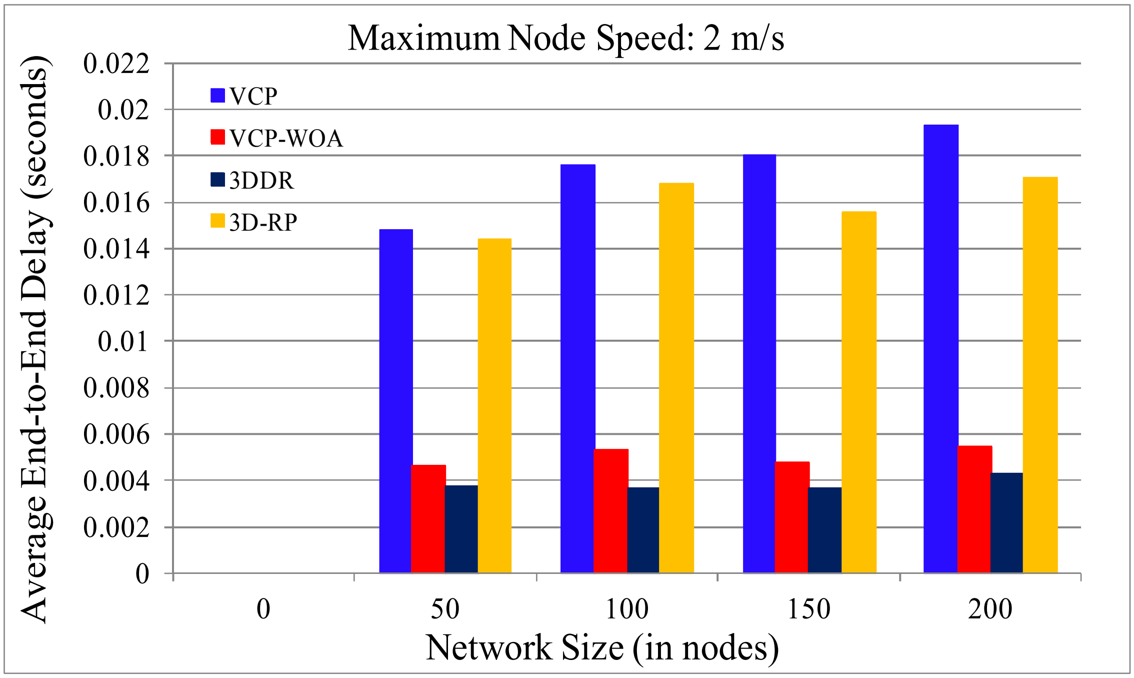


Figure 5.2: Average End-to-End delay as a function of the node number

lookup delay in the proposed approach.

The improvement in end-to-end lookup delay shown by 3DDR over VCP-WOA demonstrates the effectiveness of 3DDR against the mismatch problem. An increase in the node moving speed raises the probability of having the critical nodes in the network. Moreover, as the network frequently changes its topology due to node mobility, the lookup and routing traffic in the network increase due to the recurrent execution of the distributed partition detection and replication mechanism, which elevates the end-to end lookup delay for all protocols. But, the impact is less in 3DDR and VCP-WA as illustrated in Figures 5.1 and 5.2, because 3DDR and VCP-WOA both exploit the critical nodes for placing replicas across the critical links and thus more effective in reducing the end-to-end lookup delay compared to VCP and 3D-RP. But, 3DDR is more promising in reducing the end-to-end delay compared to VCP because of its resilient 3D logical network that reflects the physical proximity of nodes exactly in the logical network, which minimizes the impact of the mismatch problem that results in the optimized routes and reduced end-to-end lookup delay. Since our proposed mechanism exploits the critical nodes for deploying the replica, it significantly reduces the lookup delay with an increase in the nodes moving speed as shown in Figures 5.1 and 5.2.

Figures 5.7 and 5.8 illustrate the percentage improvement in the average end-to-end lookup delay for 3DDR at various node speeds compared to VCP, VCP-WA and 3D-RP. As shown in Figures 5.7 and 5.8, the end-to-end lookup delay improvement of 3DDR over VCP-WA, 3D-RP, and VCP is between 9 and 22%, 58 and 72%, 62 and 71%, respectively at various node moving speeds. The end-to-end lookup delay increases with the increase in number of nodes because expanding the network in terms of nodes amplifies the impact of partitioning and merging that increases the end-to-end lookup delay as demonstrated in the results illustrated in Figures 5.1 and 5.2.

Similarly, Figures 5.9 and 5.10 show percentage improvement in the average end-to-end lookup delay for VCP-WOA with different node speeds compared to VCP and 3D-RP. It can be inferred from the figures that end-to-end delay improvement of VCP-WOA over VCP and 3D-RP is between 60 and 73% and 49 and 69% respectively with different speed values.

5.2.2 Normalized Overhead

The normalized overhead is an important factor in defining network scalability and shows routing overhead per successful lookup request. Figures 5.3 and 5.4 illustrate the normalized overhead of 3DDR, VCP-WOA, 3D-RP, and VCP with respect to node moving speeds 0.5m/s - 2m/s with varying network sizes. Figures 5.3 and 5.4 show significant gains in term of reducing the normalized overhead of 3DDR and VCP-WOA compared to VCP and 3D-RP at various node moving speeds. The dominant factor in improving the normalized overhead is an adoption of DDR mechanism that identifies the critical nodes and replicates the mapping information, thus reducing the lookup overhead.

Figures 5.7 and 5.8 illustrate the percentage improvement in normalized overhead for 3DDR over VCP-WOA, 3D-RP, and VCP is between 4 and 8%, 30 and 41%, 33 and 40%, respectively at various node moving speeds. In Figures 5.3 and 5.4, the normalized overhead for 3DDR and VCP-WOA is a bit high, but still better compared to VCP and 3D-RP and is controlled with respect to the increase in the network size. The increased overhead is mainly due to the adoption of DDR mechanism by those protocols. The normalized overhead increases with the node moving speed as the network frequently changes its topology, generating more lookup and routing traffic in the network by performing the operations, e.g., displacement of the anchor nodes, the LS recovery, storing the mapping Information at ANs, and the re-assignment of LIDs. The normalized overhead of 3DDR is lower as shown in Figures 5.3 and 5.4 ,because of the proposed mechanism for detecting the critical nodes and replicating

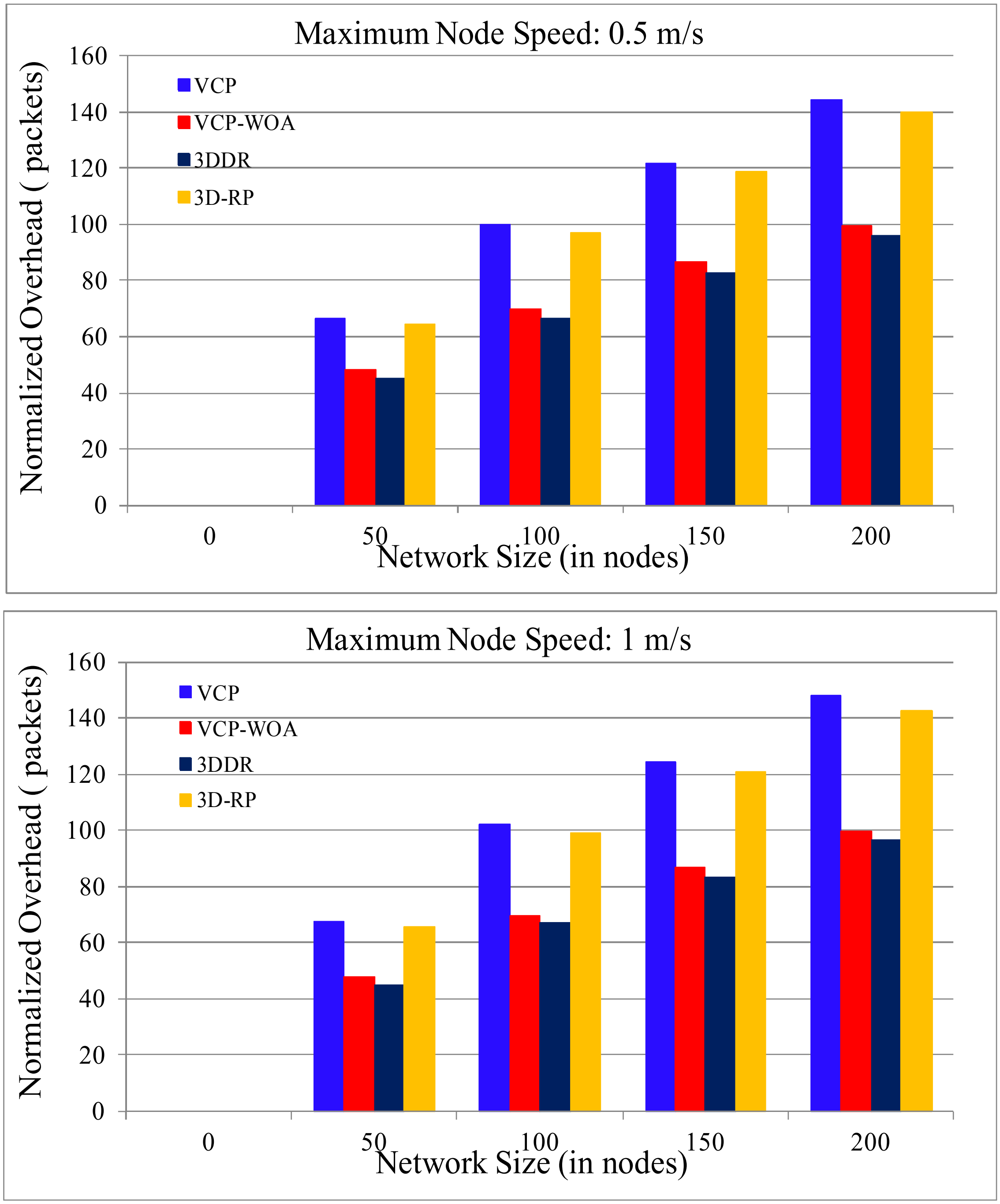


Figure 5.3: Normalized Overhead as a function of the node number

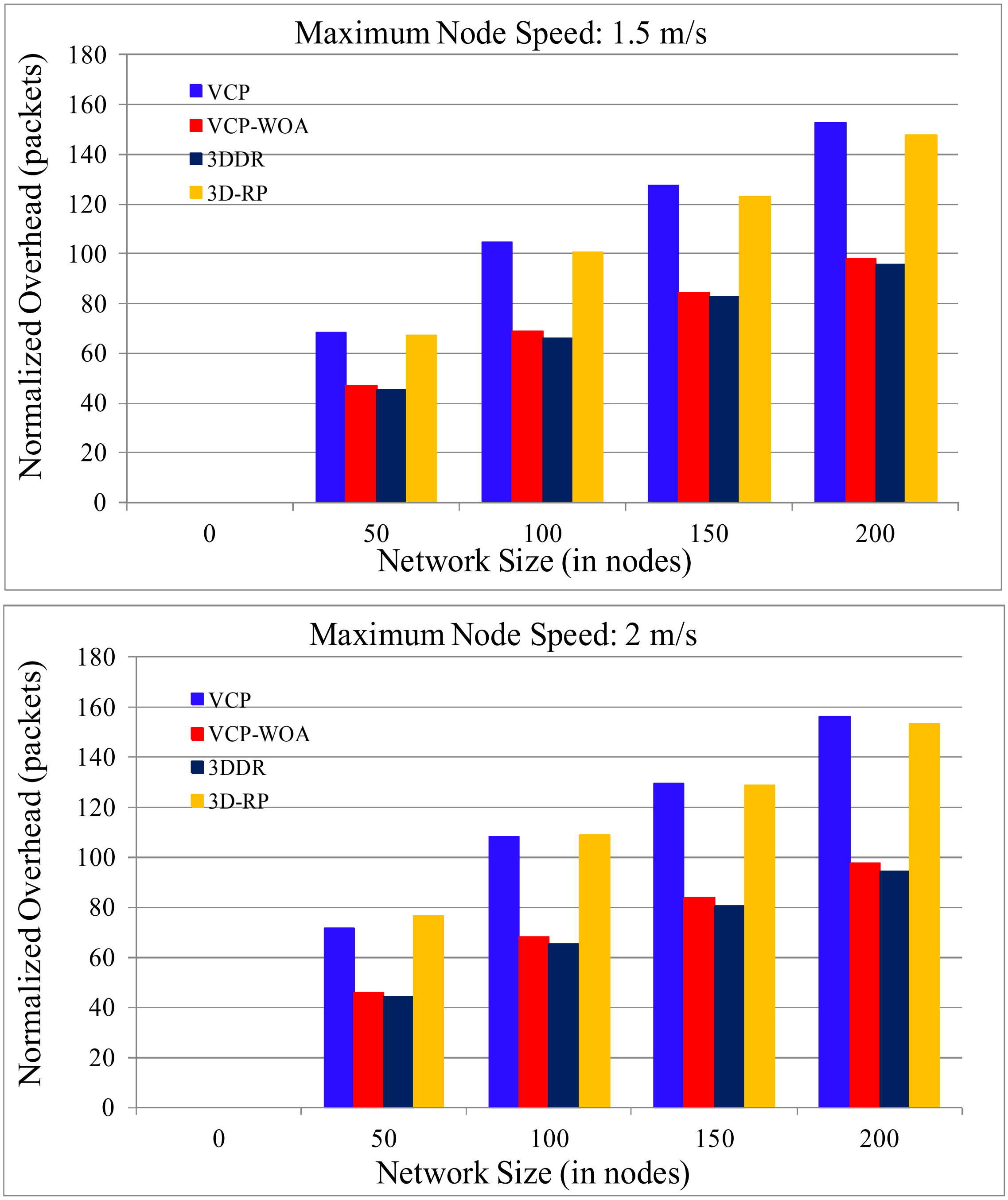


Figure 5.4: Normalized Overhead as a function of the node number

the mapping information. In addition, 3DDR employs resilient 3D logical network that is proved to be effective in handling the mismatch problem, the partitioning and merging of the network.
Similarly, the percentage improvement in normalized overhead of VCP-WOA over VCP and 3D-RP is shown in Figures 5.9 and 5.10 with various speed values. The percentage normalized overhead improvement of VCP-WOA over VCP and 3D-RP is between 28 and 39% and 26 and 40%, as shown in Figure 5.9 and 5.10.

5.2.3 Lookup Success Ratio

The lookup success ratio reflects the capability of a DHT routing protocol to successfully resolve lookup queries. The lookup success ratio for a routing protocol decreases as the network size increases. Because the increased traffic in the network causes packet collisions at the MAC layer in IEEE 802.11, i.e., the loss of MREQ and MRPY increases due to the packet collision. Similarly, the large network size increases the average hop count between the communicating nodes, which leads to more number of transmissions, increases the packet in-network delay and raises the chances of packet being collide at the MAC layer. This subsequently affects the successful delivery of MREQ and MRPY.

Figures 5.5 and 5.6 show the impact of increasing network size and node moving speed on success ratio is less in 3DDR compared to VCP, 3D-RP, and VCP-WOA. 3DDR outperforms its competitors in terms of success ratio, which verifies the effectiveness and the capability of 3DDR in delivering the packets in the large network. The proposed partition detection and replication mechanism further assists further in maintaining and improving the success ratio of 3DDR. 3DDR replicates the mapping information across all the critical links along the path to AN. Therefore, in case of network partitioning, the mapping information readily

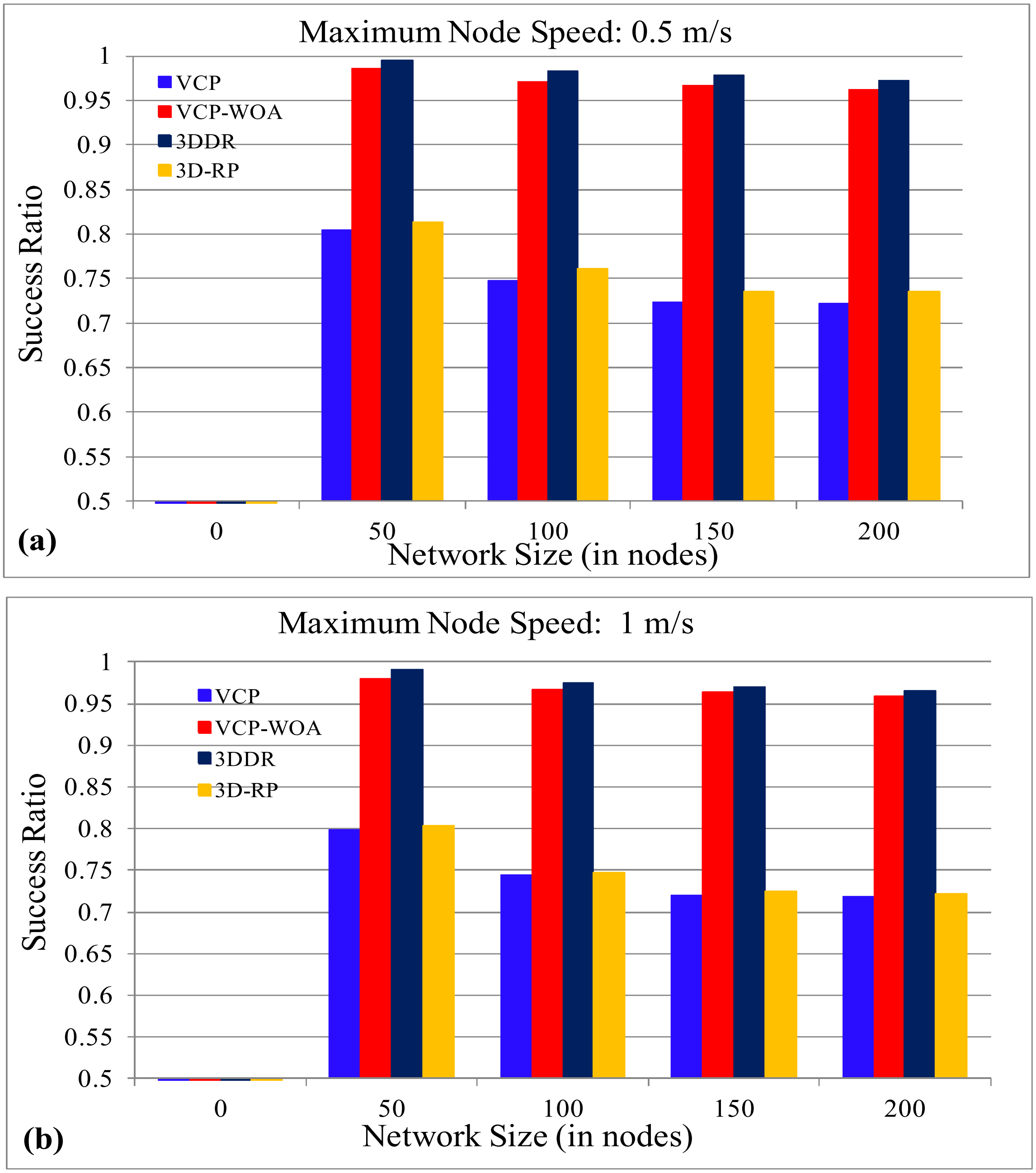


Figure 5.5: Success Ratio as a function of the node number

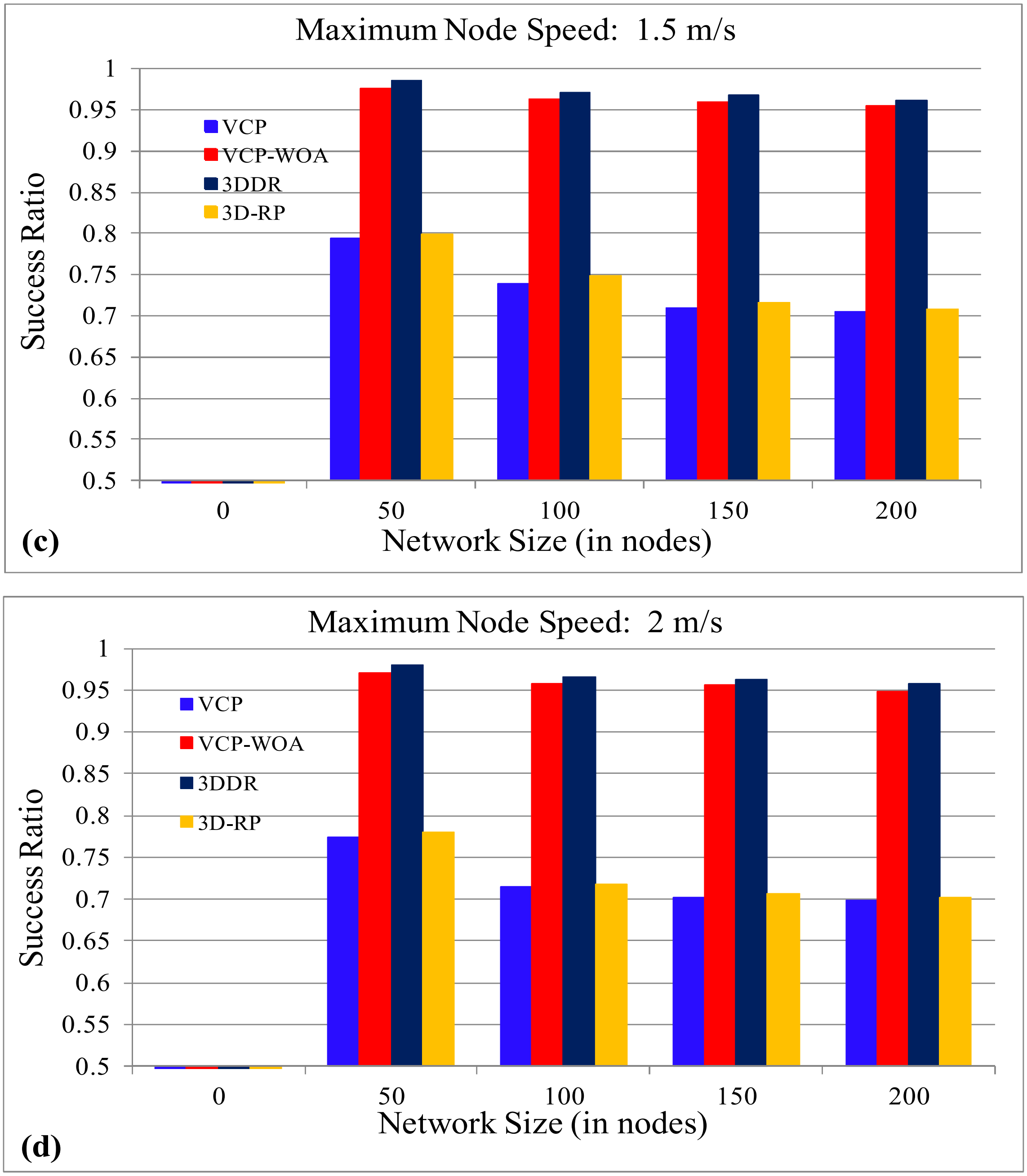


Figure 5.6: Success Ratio as a function of the node number

available in the disjoint partitions and subsequently contributes the high success ratio in the proposed protocol. Moreover, an increase in the node moving speed causes frequent network partitioning and merging. In such scenarios, retrieving the mapping information may not be possible using 3D-RP and VCP if the corresponding AN resides in the disjoint partition. Secondly, maintaining the replica in 3DDR and VCP-WOA reduces the traffic overhead, which decreases the probability of packet collision at the MAC layer, thus contributes in maintaining the higher success ratio. Furthermore, VCP and VCP-WOA suffer from the mismatch problem as Cord logical structure is not flexible to reflect the physical intra neighbor relationships of nodes exactly in the logical Cord network, resulting in long routes and redundant traffic. This increase the probability of packet collision at the MAC layer, thus reducing the success ratio of both VCP and VCP-WOA compared to 3DDR as shown in Figures 5.5 and 5.6.

Figures 5.7 and 5.8 illustrate the percentage improvement in success ratio of 3DDR at various node moving speeds compared to VCP, VCP-WOA and 3D-RP. The improvement in success ratio of 3DDR over VCP-WOA, 3D-RP, and VCP is between 2 and 4%, 24 and 38%, 21 and 35%, respectively for various network sizes and node moving speeds. The increase in network size and node moving speed cause the network topology to change more frequently, which result in an increased lookup and routing overhead due to frequent execution of the recovery and partition detection operations. This causes congestion at the MAC layer as packets are delayed in the queue and drop eventually. The impact of congestion on 3DDR is lower compared to the rest.

Similarly, the percentage improvement of VCP-WOA with respect to VCP and 3D-RP in lookup success ratio at various speed values is shown in Figures 5.9 and 5.10. The improvement in success ratio of VCP-WOA over VCP and 3D-RP is between 21 and 37%, and 22 and 36%.

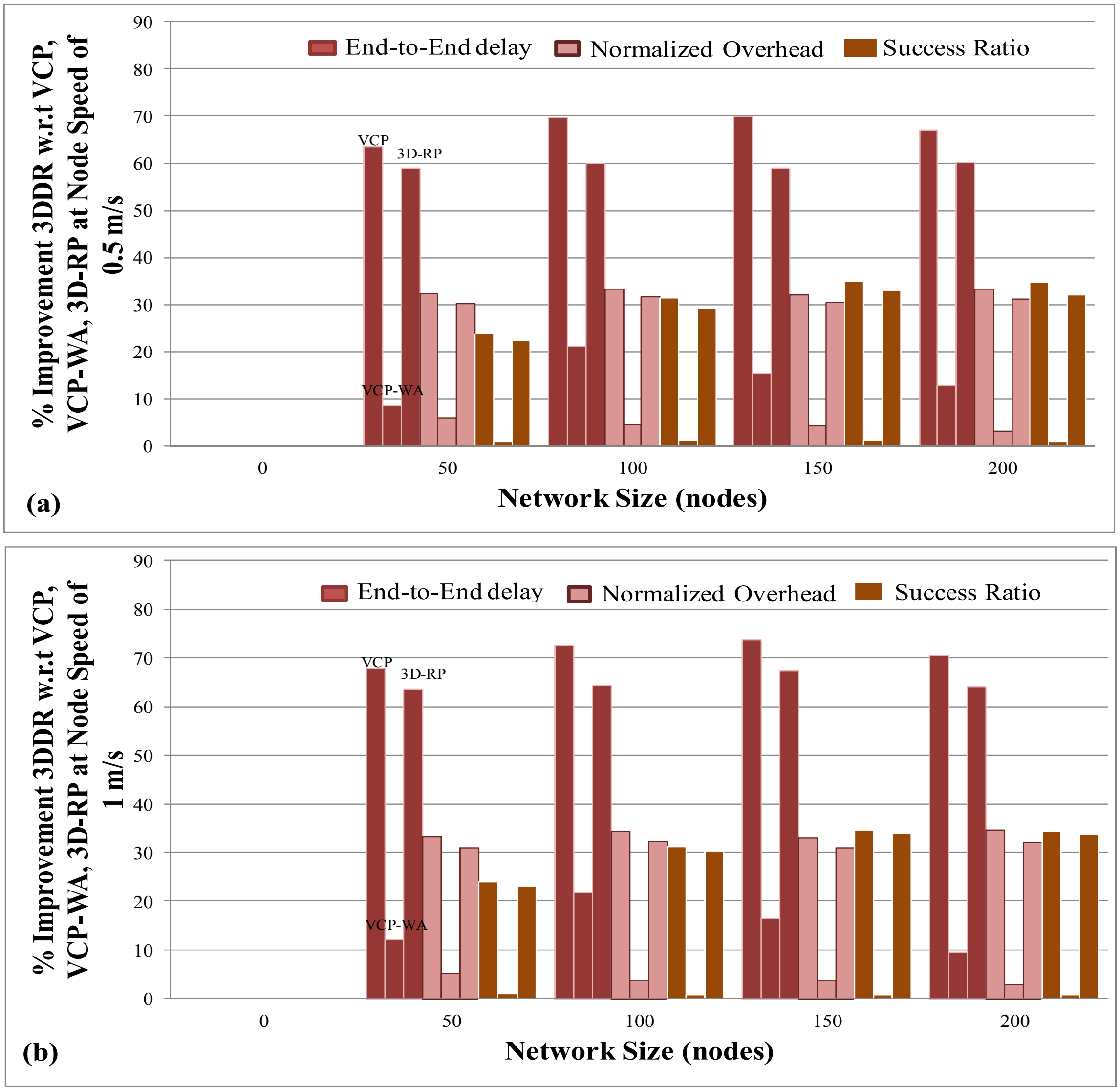
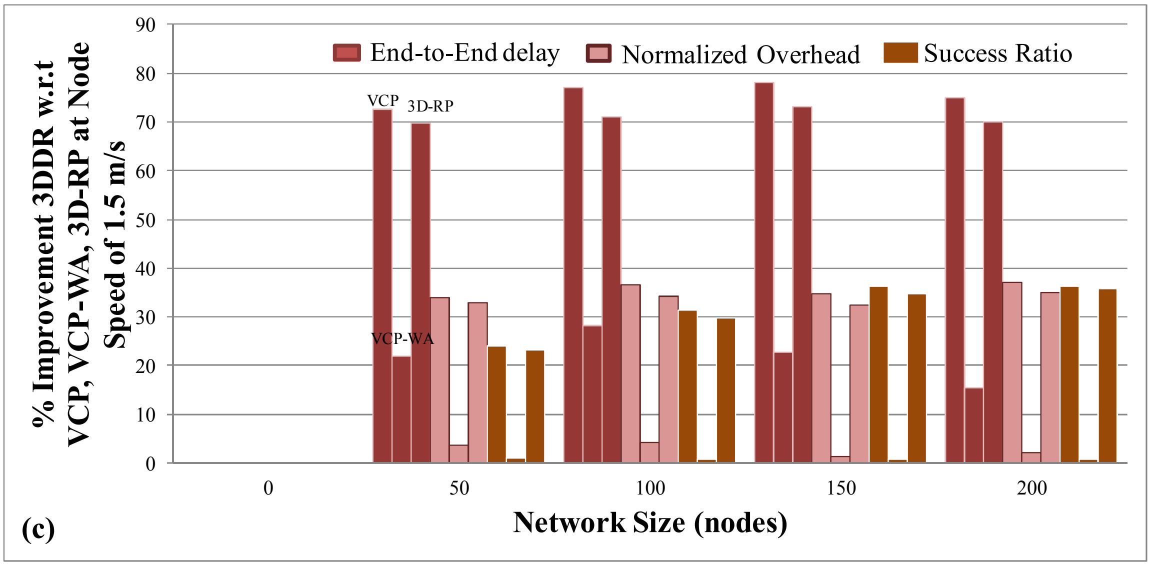


Figure 5.7: % Improvement of 3DDR over VCP, VCP-WOA, 3D-RP in terms of E2E, NO, and SR. The protocol name is displayed on first three bars of each graph and the pattern is same for the rest in each graph.



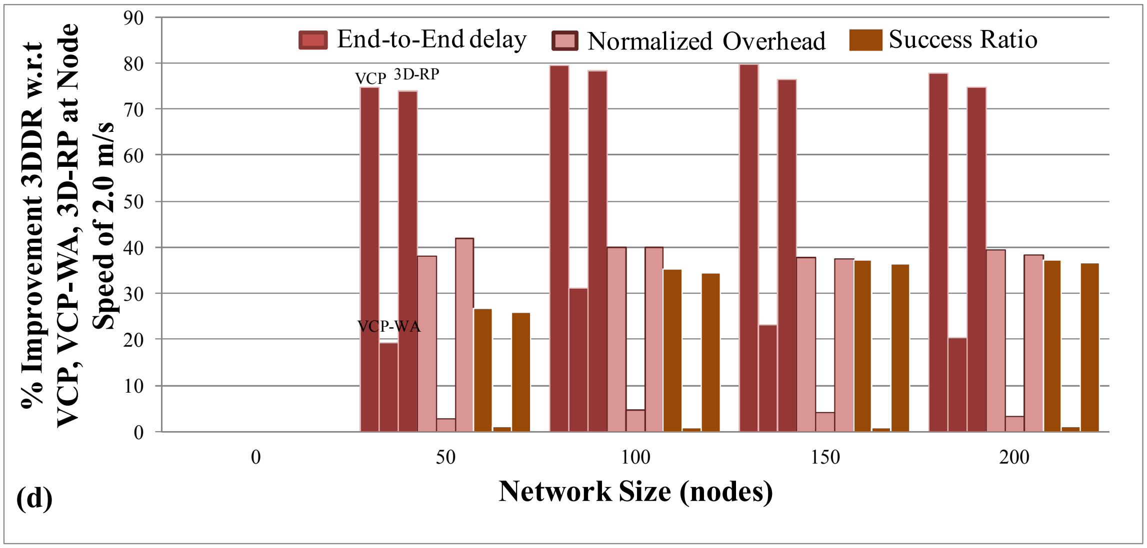


Figure 5.8: % Improvement of 3DDR over VCP, VCP-WOA, 3D-RP in terms of E2E, NO, and SR. The protocol name is displayed on first three bars of each graph and the pattern is same for the rest in each graph.

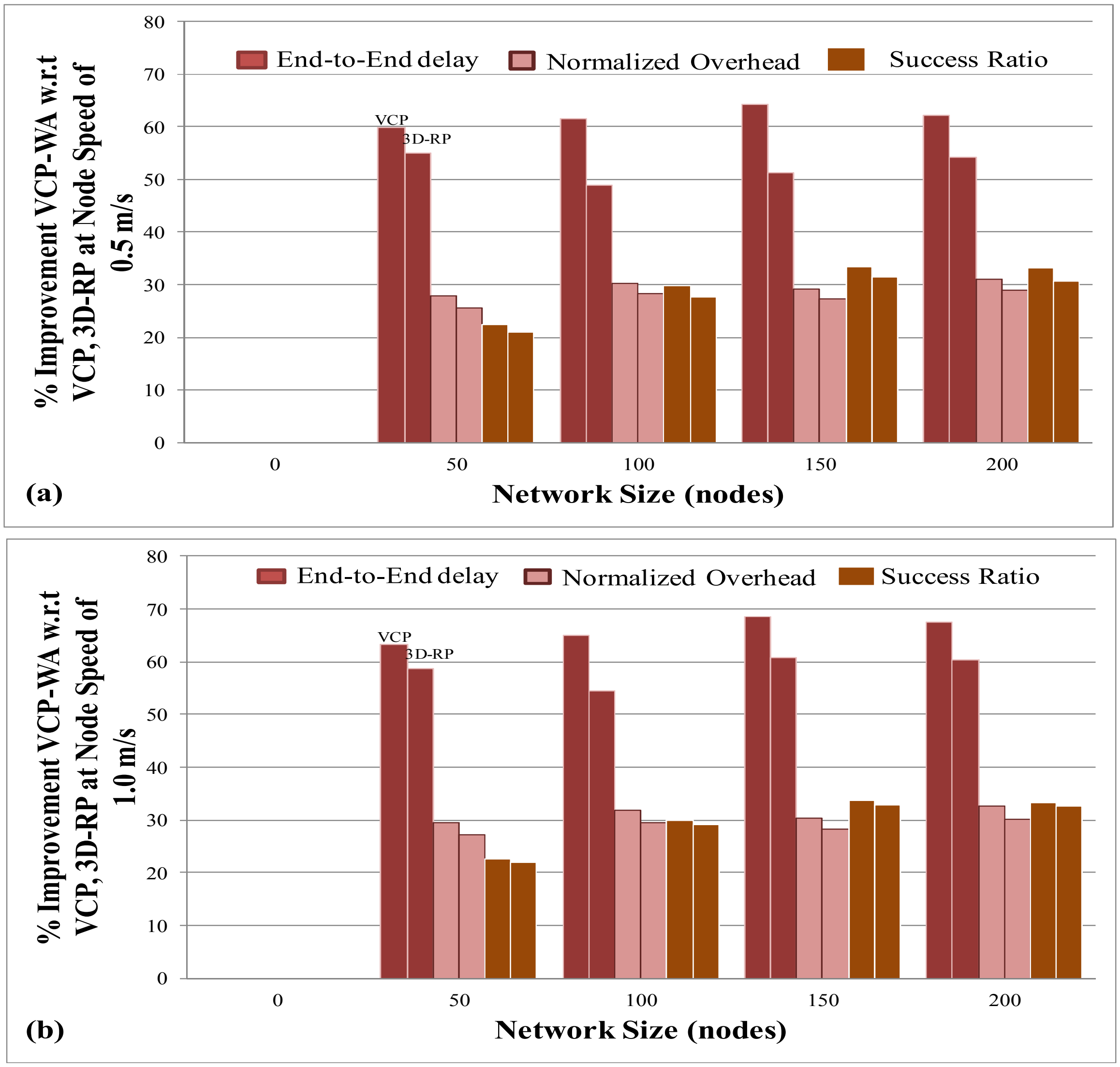


Figure 5.9: % Improvement of VCP-WOA over VCP, 3D-RP in terms of E2E, NO, and SR. The protocol name is displayed on first three bars of each graph and the pattern is same for the rest in each graph.

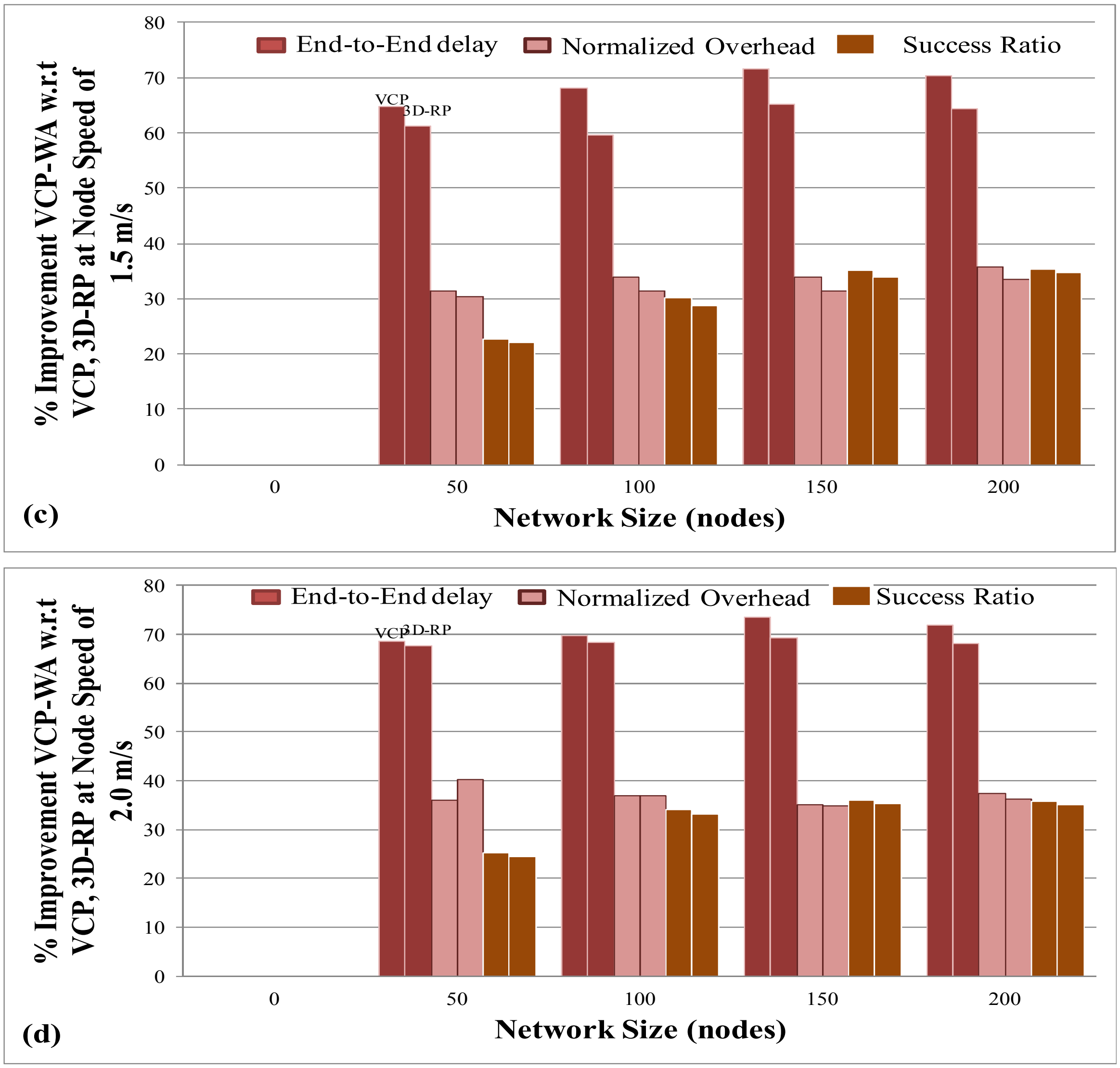


Figure 5.10: % Improvement of VCP-WOA over VCP, 3D-RP in terms of E2E, NO, and SR. The protocol name is displayed on first three bars of each graph and the pattern is same for the rest in each graph.

Chapter 6

Conclusions and Future Work

6.1 About the Chapter

This chapter concludes the thesis. Significance of the thesis contribution along with future possible extensions are given as the end of this chapter.

6.2 Conclusions

This thesis studies the impact of network dynamics (i.e. node(s)/link(s) failure, network partitioning, merging etc.) on DHT-based routing protocols in MANETs. It is found that network dynamics severely damage the functionality of DHT protocols in MANETs and raise issues such as anchor node (AN) failure, LS loss, longer lookup etc. Therefore, distributed solutions under the scalability constraints are needed.

The fist part of this thesis (i.e. Chapter 3) addresses the problems of anchor node (AN) failure, index information recovery and longer lookup delay in DHT-based MANETs. A novel address publication (also called, Anchor_Request) mechanism is devised that tolerates faults in the network. The proposed mechanism exploits k-hop topological information in a fully distributed manner and replicates index information across the critical regions in the network. This confirms resolutions of the lookup requests with a significant improvement in lookup delay. The solutions are based on the existing *hello* messages used for LID computation in DHT networks, thus our approach causes no extra traffic. Simulation results confirm the effectiveness of the proposed solutions.

The problems of critical node(s)/link(s) identification, partition detection, LS recovery and reusing cover the second part of this thesis (i.e. Chapter 4). Network partition elicits the problems of anchor node (AN) failure and LS loss. Moreover, it is explored that the LS loss further elicits two sub-problems i) Disrupted LS(s) in the partitioned network's instances. ii) prohibits smooth merging because merging needs uniformly distributed identical structures (i.e. LS(s)). The proposed algorithm 4.1 detects critical node/link in the network. The critical nodes/links play a vital role in the partition detection event and LS recovery. Algorithm 4.2 recovers and reuses the lost LS, due to network partition, and makes sure uniformly distributed LS(s) in the partitioned network instances. This sets ground for smooth merging of the disjoint instances of the partitioned network. The algorithms solve the mentioned problems in a fully distributed manner.

A case study is conducted to provide an up-close, in-depth, and detailed examination of the proposed distributed partition detection and replication algorithms in order to enhance the understanding of various related contextual conditions and to evaluate the functionality of the proposed mechanisms. We evaluate the performance of the proposed solutions using simulation. The simulation results and comparison with existing approaches endorse the effectiveness of the proposed solutions in terms of End-to-End lookup delay, lookup success ratio and normalized overhead.

6.3 Significance of Contributions

For the first time, this thesis explores the problems raised by network dynamics and provides novel solutions to the identified problems in DHT-based MANETs. Therefore, the thesis contributions are two fold, in the problem domain as well as in the solution domain. Main contributions of the thesis (i.e. solution domain) are summarized below.

- A novel Address publication (also known as, Anchor_Request) is proposed.
- A dynamic replica deployment mechanism based on *k*-hop topological information is proposed.

- Distributed mechanism for k-hop critical node(s)/link(s) detection is provided.
- Distributed partition detection and LS recovery/reusing mechanisms based on the local information available to nodes is devised. The LS recovery and reusing mechanism uniformly distribute LS(s) in the network instances after the network gets partitions. This, also, sets grounds for smooth network merging.
- Simulation results show significant gains in term of lookup delay and lookup resolution success ratio.
- It worthy to mention that the proposed mechanisms do not incur any extra transmission overhead and are purely based on the existing *hello* messages used for LID computation in DHT networks.

6.4 DHT-based Routing and Future Trends

As a future work, we have planned to extend our work; i) using other mobility models and a unified approach for handling both the merging and partitioning detection; ii) existing schemes use k=2-hop topological information (i.e. only based on the existing *hello* messages used for LID computation in DHT-based MANETs). We have planned to use different variation of k and to study its impact on the network performance iii) to corroborate the feasibility of such protocols in the emerging domains such as the Content Centric Networking, Internet-of-Things and software defined networks. Detail description is listed in the subsequent sections.

6.4.1 Content Centric Networking

In the recent years Content Centric Networking(CCN) has emerged as hot research area. CCN has a promising future for Internet as well as for MANETs. CCN works on named data despite of hosts identifiers(IP addresses etc.) [61–63]. CCN decouples contents from source and is capable of accessing content by name rather than identifiers. CCN tolerates faults and avoids dependency on end-to-end connectivity. However, CCN degrades performance with growing network size [63]. In CCN, DHT can be used to address the scalability problems. DHT decouples location from identity in a scalable manner. Therefore, DHT best suites to the functionality and requirements of CCN for Internet as well as for MANETs. We have planned to extend/implement our work in the domain of CCN for Internet and MANETs.

6.4.2 Internet of Things (IoT)

IoT consists of identifiable objects(i.e. electronic devices, sensors etc.), with computing and networking facilities, that cooperate with each other and provide value added services [64, 65]. The main challenge in the deployment of IoT is scalability [66]. Addressing/naming, the address space and address resolution due to huge number of nodes are major challenges to address scalability in IoT. DHT-based routing and lookup mechanisms cab be exploited to address the scalability issues in IoT.

Bibliography

[1] J. Eriksson, M. Faloutsos, and S. V. Krishnamurthy, “Dart: Dynamic address routing for scalable ad hoc and mesh networks,” *IEEE/ACM Transactions on Networking (TON)*, vol. 15, no. 1, pp. 119–132, 2007.

[2] S. A. Abid, M. Othman, and N. Shah, “Exploiting 3d structure for scalable routing in manets,” *IEEE Communications Letters*, vol. 17, no. 11, pp. 2056–2059, 2013.

[3] S. Zahid, S. A. Abid, N. Shah, S. H. A. Naqvi, and W. Mehmood, “Distributed partition detection with dynamic replication management in a dht-based manet,” *IEEE Access*, vol. 6, pp. 18 731–18 746, 2018.

[4] S. A. Abid, M. Othman, and N. Shah, “A survey on dht-based routing for large-scale mobile ad hoc networks,” *ACM Computing Surveys (CSUR)*, vol. 47, no. 2, p. 20, 2015.

[5] E. M. Belding-Royer, “Multi-level hierarchies for scalable ad hoc routing,” *Wireless Networks*, vol. 9, no. 5, pp. 461–478, 2003.

[6] S. A. Abid, M. Othman, N. Shah, O. Sabir, M. Ali, J. Shafi, S. Ullah *et al.*, “Merging of dht-based logical networks in manets,” *Transactions on Emerging Telecommunications Technologies*, vol. 26, no. 12, pp. 1347–1367, 2015.

[7] I. Chlamtac, M. Conti, and J. J.-N. Liu, “Mobile ad hoc networking: imperatives and

challenges," *Ad hoc networks*, vol. 1, no. 1, pp. 13–64, 2003.

[8] E. M. Belding-Royer, "Multi-level hierarchies for scalable ad hoc routing," *Wireless Networks*, vol. 9, no. 5, pp. 461–478, 2003.

[9] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (aodv) routing," Tech. Rep., 2003.

[10] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile computing*. Springer, 1996, pp. 153–181.

[11] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *Multi Topic Conference, 2001. IEEE INMIC 2001.* IEEE, 2001, pp. 62–68.

[12] M. Jiang, "Cluster based routing protocol (cbrp)," *IETF Internet-draft*, 1999.

[13] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE Journal on Selected areas in Communications*, vol. 15, no. 7, pp. 1265–1275, 1997.

[14] O. Younis and S. Fahmy, "Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on mobile computing*, no. 4, pp. 366–379, 2004.

[15] M. Ni, Z. Zhong, and D. Zhao, "Mpbc: A mobility prediction-based clustering scheme for ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 9, pp. 4549–4559, 2011.

[16] W.-d. Yang, "Weight-based clustering algorithm for mobile ad hoc network," in *Proceedings of 2011 Cross Strait Quad-Regional Radio Science and Wireless Technology Conference*, vol. 1. IEEE, 2011, pp. 787–791.

[17] R. B. Muhammad, "A distributed graph algorithm for geometric routing in ad hoc wireless networks," *Journal of Networks*, vol. 2, no. 6, pp. 50–57, 2007.

[18] B. Karp and H.-T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM, 2000, pp. 243–254.

[19] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *IEEE network*, vol. 15, no. 6, pp. 30–39, 2001.

[20] Y. Xue, B. Li, and K. Nahrstedt, "A scalable location management scheme in mobile ad-hoc networks," in *Proceedings LCN 2001. 26th Annual IEEE Conference on Local Computer Networks*. IEEE, 2001, pp. 102–111.

[21] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, *Mobile ad hoc networking, Ad Hoc Wireless Networking, ch: Position based routing algorithms for ad hoc networks"*. John Wiley & Sons, 2004.

[22] X. Lin, "Geographic distance routing in ad hoc wireless networks," *Technical Report in University Ottawa*, vol. 98, 1998.

[23] J. Urrutia, "Routing with guaranteed delivery in geometric and wireless networks," *Handbook of wireless networks and mobile computing*, vol. 18, pp. 393–406, 2002.

[24] N. Shah, "Efficient overlay routing for peer to peer network over mobile ad hoc network," Ph.D. dissertation, Ph. D. Dissertation. School of Computer Science and Engineering, Beihang , 2011.

[25] N. Shah and D. Qian, "An efficient unstructured p2p overlay over manet using underlying proactive routing," in *2011 Seventh International Conference on Mobile Ad-hoc and Sensor Networks*. IEEE, 2011, pp. 248–255.

[26] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, *A scalable content-addressable network*. ACM, 2001, vol. 31, no. 4.

[27] F. Dabek, E. Brunskill, M. F. Kaashoek, D. Karger, R. Morris, I. Stoica, and H. Balakrishnan, "Building peer-to-peer systems with chord, a distributed lookup service," in *Proceedings Eighth Workshop on Hot Topics in Operating Systems*. IEEE, 2001, pp. 81–86.

[28] A. Rowstron and P. 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*. Springer, 2001, pp. 329–350.

[29] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," in *International Workshop on Peer-to-Peer Systems*. Springer, 2002, pp. 53–65.

[30] Y. C. Hu, S. M. Das, and H. Pucha, "Exploiting the synergy between peer-to-peer and mobile ad hoc networks." in *HotOS*, 2003, pp. 37–42.

[31] A. Awad, C. Sommer, R. German, and F. Dressler, "Virtual cord protocol (vcp): A flexible dht-like routing service for sensor networks," in *2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*. IEEE, 2008, pp. 133–142.

[32] A. Awad, R. German, and F. Dressler, "Exploiting virtual coordinates for improved routing performance in sensor networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 9, pp. 1214–1226, 2011.

[33] A. C. Viana, M. D. de Amorim, S. Fdida, and J. F. de Rezende, "Indirect routing using distributed location information," in *Proceedings of the First IEEE International Con-*

ference on Pervasive Computing and Communications, 2003.(PerCom 2003). IEEE, 2003, pp. 224–231.

[34] D. Sampath and J. J. Garcia-Luna-Aceves, "Prose: scalable routing in manets using prefix labels and distributed hashing," in *2009 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*. IEEE, 2009, pp. 1–9.

[35] Q. Xu, L. Sun, and J. Liu, "Topology-aware kademlia based on distributed clustering in self-organizing mode," in *2010 2nd International Conference on Computer Engineering and Technology*, vol. 1. IEEE, 2010, pp. V1–267.

[36] M. Caleffi and L. Paura, "M-dart: multi-path dynamic address routing," *Wireless communications and mobile computing*, vol. 11, no. 3, pp. 392–409, 2011.

[37] S. Shin, U. Lee, F. Dressler, and H. Yoon, "Motion-mix dht for wireless mobile networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 12, pp. 3100–3113, 2016.

[38] H. Wirtz, T. Heer, R. Hummen, and K. Wehrle, "Mesh-dht: A locality-based distributed look-up structure for wireless mesh networks," in *2012 IEEE International Conference on Communications (ICC).* IEEE, 2012, pp. 653–658.

[39] A. Tahir, S. A. Abid, and N. Shah, "Logical clusters in a dht-paradigm for scalable routing in manets," *Computer Networks*, vol. 128, pp. 142–153, 2017.

[40] S. Cherbal, A. Boukerram, and A. Boubetra, "A survey of locality-awareness solutions in mobile dht systems," in *2015 12th International Symposium on Programming and Systems (ISPS).* IEEE, 2015, pp. 1–7.

[41] G. Fersi, W. Louati, and M. B. Jemaa, "Clever: cluster-based energy-aware virtual ring

routing in randomly deployed wireless sensor networks," *Peer-to-Peer Networking and Applications*, vol. 9, no. 4, pp. 640–655, 2016.

[42] N. Shah, A. Ahmad, W. Mehmood, D. Qian, and R. Wang, "An efficient and scalable routing for manets," *Wireless personal communications*, vol. 75, no. 2, pp. 987–1004, 2014.

[43] H. Shen and Z. Li, "A hierarchical account-aided reputation management system for manets," *IEEE/ACM Transactions on Networking (TON)*, vol. 23, no. 1, pp. 70–84, 2015.

[44] J. J. Garcia-Luna-Aceves and D. Sampath, "Scalable integrated routing using prefix labels and distributed hash tables for manets," in *2009 IEEE 6th International Conference on Mobile Adhoc and Sensor Systems*. IEEE, 2009, pp. 188–198.

[45] E. Baccelli and J. Schiller, "Towards scalable manets," in *2008 8th International Conference on ITS Telecommunications*. IEEE, 2008, pp. 133–138.

[46] B. Zhao, Y. Wen, and H. Zhao, "Kdsr: An efficient dht-based routing protocol for mobile ad hoc networks," in *2009 Ninth International Conference on Hybrid Intelligent Systems*, vol. 2. IEEE, 2009, pp. 245–249.

[47] M. Caesar, M. Castro, E. B. Nightingale, G. O'Shea, and A. Rowstron, "Virtual ring routing: network routing inspired by dhts," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, pp. 351–362, 2006.

[48] S. A. Abid, M. Othman, N. Shah, M. Ali, and A. Khan, "3d-rp: A dht-based routing protocol for manets," *The Computer Journal*, vol. 58, no. 2, pp. 258–279, 2014.

[49] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information sys-

tem based on the xor metric," in *International Workshop on Peer-to-Peer Systems*. Springer, 2002, pp. 53–65.

[50] H. Ritter, R. Winter, and J. Schiller, "A partition detection system for mobile ad-hoc networks," in *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*. IEEE, 2004, pp. 489–497.

[51] A. Derhab, N. Badache, and A. Bouabdallah, "A partition prediction algorithm for service replication in mobile ad hoc networks," in *Second Annual Conference on Wireless On-demand Network Systems and Services*. IEEE, 2005, pp. 236–245.

[52] K. H. Wang and B. Li, "Efficient and guaranteed service coverage in partitionable mobile ad-hoc networks," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2. IEEE, 2002, pp. 1089–1098.

[53] M. Hauspie, J. Carle, and D. Simplot, "Partition detection in mobile ad-hoc networks using multiple disjoint paths set," in *International Workshop on Objects models and Multimedia technologies*, 2003, p. 15.

[54] B. Milic, N. Milanovic, and M. Malek, "Prediction of partitioning in location-aware mobile ad hoc networks," in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*. IEEE, 2005, pp. 306c–306c.

[55] N. Shah, A. Ahmad, B. Nazir, and D. Qian, "A cross-layer approach for partition detection at overlay layer for structured p2p in manets," *Peer-to-Peer Networking and Applications*, vol. 9, no. 2, pp. 356–371, 2016.

[56] I. Stojmenovic, D. Simplot-Ryl, and A. Nayak, "Toward scalable cut vertex and link

detection with applications in wireless ad hoc networks," *IEEE network*, vol. 25, no. 1, pp. 44–48, 2011.

[57] M. Jorgic, M. Hauspie, D. Simplot-Ryl, and I. Stojmenovic, "Localized algorithms for detection of critical nodes and links for connectivity in ad hoc networks," in *Mediterranean Ad Hoc Networking Workshop*, 2004, p. 12.

[58] T. V. Project, "The ns manual:notes and documentation."

[59] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn, "Bonnmotion: a mobility scenario generation and analysis tool," in *Proceedings of the 3rd international ICST conference on simulation tools and techniques*. ICST (Institute for Computer Sciences, Social-Informatics and , 2010, p. 51.

[60] G. Holland and N. Vaidya, "Analysis of tcp performance over mobile ad hoc networks," *Wireless Networks*, vol. 8, no. 2-3, pp. 275–288, 2002.

[61] S. Y. Oh, D. Lau, and M. Gerla, "Content centric networking in tactical and emergency manets," in *2010 IFIP Wireless Days*. IEEE, 2010, pp. 1–5.

[62] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. ACM, 2009, pp. 1–12.

[63] H. Liu, X. De Foy, and D. Zhang, "A multi-level dht routing framework with aggregation," in *Proceedings of the second edition of the ICN workshop on Information-centric networking*. ACM, 2012, pp. 43–48.

[64] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[65] N. Chilamkurti, S. Zeadally, and H. Chaouchi, *Next-generation wireless technologies: 4G and beyond*. Springer Science & Business Media, 2013.

[66] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad hoc networks*, vol. 10, no. 7, pp. 1497–1516, 2012.