

# A Survey on DHT-based Routing for Large Scale Mobile Ad hoc Networks

Shahbaz Akhtar Abid, University of Malaya  
Mazliza Othman, University of Malaya  
Nadir Shah, COMSATS Institute of Information technology

Mobile ad hoc networks (MANETs) are infrastructure-less and distributed communication systems that require sophisticated approaches to routing in order to cope with node mobility and heterogeneous application requirements. In the last few years, Distributed Hash Table (DHT) has come forth as a useful additional technique to the design and specification of spontaneous and self-organized networks. Researchers have exploited its advantages by implementing it at the network layer and developing scalable routing protocols for MANETs. The implementation of DHT-based routing in a MANET requires different algorithms and specifications compared to routing in the Internet because a MANET has its unique characteristics, e.g., node mobility, spontaneous networking, decentralized architecture, limited transmission range, dynamic topology, and frequent network partitioning/merging.

In this article, we present a comprehensive survey of researches related to DHT-based routing that aim at enhancing the scalability of MANETs. We present a vivid taxonomy of DHT-based routing protocols and the guidelines to design such protocols for MANETs. We compare the features, strengths and weaknesses of existing DHT-based routing protocols and highlight key research challenges that are vital to address. The outcome of the analysis serves as a guide for anyone willing to delve into research on DHT-based routing in MANETs.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Routing Protocols

General Terms: Classification, Challenges, Comparisons, Future Directions

Additional Key Words and Phrases: Wireless Networks, Distributed Hash Tables, Routing, Logical Network, Overlays, Scalability, Mismatch Problem

## ACM Reference Format:

Shahbaz Akhtar Abid, Mazliza Othman, Nadir Shah, XXXX. A Survey on DHT-based Routing for Large Scale Mobile Ad hoc Networks. *ACM Comput. Surv.* 0, 0, Article 00 (0000), 47 pages.

DOI: <http://dx.doi.org/10.1145/0000000.0000000>

## 1. INTRODUCTION

A mobile ad hoc network (MANET) provides instant, low cost, and flexible communication between groups of people that may not be within transmission range of one another. Each node in a MANET acts as host (for sending/receiving data) and router (maintains the routing information to forward data to other nodes). Nowadays, most people use mobile devices, such as cell phones, PDAs and laptops, which have larger

---

This work is supported by the Institute of Research Management and Monitoring (IPPP), under grant PG122-2012B.

Author's addresses: S.A. Abid and Mazliza Othman, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur Malaysia; Nadir Shah, Department of Computer Science, COMSATS Institute of Information technology, Wah Cantt, Pakistan;

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or [permissions@acm.org](mailto:permissions@acm.org).

© 0000 ACM 0360-0300/0000/-ART00 \$15.00

DOI: <http://dx.doi.org/10.1145/0000000.0000000>

memory, higher processing capability, and richer functionality compared to five years ago [Statistics 2012; 2013]. Users can store more audio, video, text and images on them. Equipped with Bluetooth [Haartsen 2000] or Wi-Fi [Wi-Fi.org ], these devices can communicate with each other without using any communication infrastructure (e.g., cellular infrastructure) and form a self-organizing MANET.

There are several application scenarios for a MANET ranging from campus and conference scenarios, to emergency operations (like natural disasters and political unrests), to military scenarios. The number of users in each application scenario ranges from a handful of people in an emergency situation, to tens and hundreds of people in campus and conference scenarios, to thousands and tens of thousands of people in political unrest and military applications [Belding-Royer 2003]. Due to these reasons, a scalable routing protocol is critical for any application that is intended to support a large number of users in a MANET.

The primary goal of a routing protocol is to establish an efficient route between the source and the destination nodes, and to provide dynamic topology maintenance and loop prevention, so that messages can be delivered in a timely manner with minimal traffic and processing overheads [Deng et al. 2002; Junhai et al. 2009]. A significant amount of researches have been done in order to address the scalability issue of routing protocols in MANETs, and have partially succeeded in addressing it by controlling flooding, lowering traffic overhead, and reducing the size of the routing table. Some of these routing protocols are implemented on test-beds and have been used in real world applications [Kieess and Mauve 2007; Kulla et al. 2012]. Based on the role of routing nodes and the organization of the network, we classify the existing routing protocols for MANETs into five categories as shown in Fig. 1.

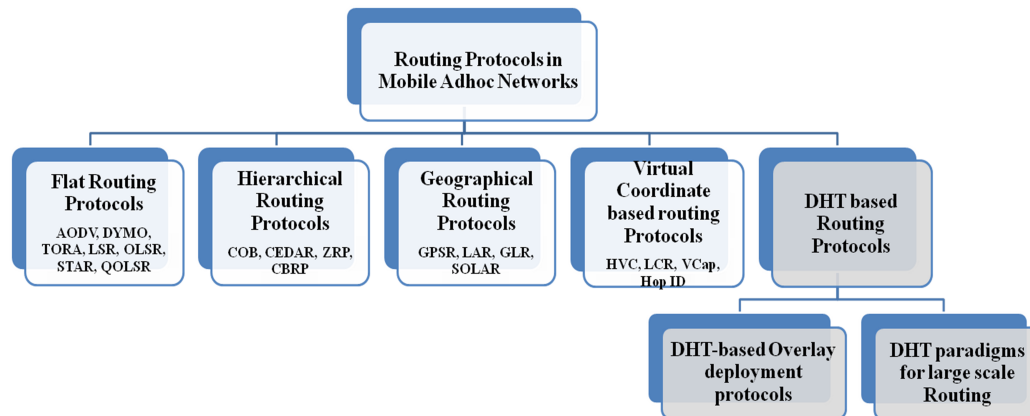


Fig. 1. Classification of Routing Protocols in MANETs

### 1.1. FLAT AND HIERARCHICAL ROUTING PROTOCOLS FOR MANETS

In a flat routing protocol, each node has the same role and the network has a flat structure [Belding-Royer 2003; Rajaraman 2002]. These protocols are suitable for small networks and their performance degrades as the network size grows [Caleffi and Paura 2011; Awad et al. 2011; Eriksson et al. 2007]. Flat routing protocols can be further classified into reactive or source initiated and proactive or table driven based on how the protocol reacts to network topology.

Reactive routing protocols establish routes on-demand, i.e., these protocols find the route to a destination only when there is data to be sent. In this way, reactive protocols avoid the prohibitive cost of maintaining routing information for nodes to which there

is no data to be sent. These routing protocols tend to work well in practice for scenarios where data exchange among nodes is less frequent [Rajaraman 2002]. AODV[Perkins et al. 2003], DYMO[Chakeres and Perkins 2008], TORA[Park and Corson 1997] and LSR[Rangarajan and Garcia-Luna-Aceves 2007] are some well-known reactive routing protocols. Reactive routing protocols introduce flooding during the route discovery phase [Abolhasan et al. 2004; Belding-Royer 2003; Jacquet et al. 2001; Lee et al. 2003; Liu and Kaiser 2003]. The destination node replies to a route request either by using reverse path (in case of bidirectional link) or by flooding mechanism.

In case the route to the destination node changes frequently due to node mobility or because a source is communicating with multiple destinations, redundant transmission during route discovery increases the amount of traffic significantly, hence increasing the probability of packet collisions. Reactive routing protocols also introduce a route acquisition latency, or a period of waiting to acquire a route prior to sending the data, resulting in longer delays [Abolhasan et al. 2004; Belding-Royer 2003].

On the other hand, each node in a proactive routing protocol maintains an up-to-date routing information to all other nodes in the network, regardless of whether or not there is data to be sent [Belding-Royer 2003; Rajaraman 2002]. As a result, an up-to-date route to any other node in the network is immediately available. [Jacquet et al. 2001; Garcia-Luna-Aceves and Spohn 1999; Murthy and Garcia-Luna-Aceves 1996; Munaretto and Fonseca 2007] are some well-known proactive routing protocols. The dissemination of routing information via a flooding mechanism and the unnecessary route discovery (hence, unnecessary traffic overhead) consumes a major portion of the bandwidth. Therefore, these protocols introduce the traffic overhead complexity of an  $O(n^2)$ , resulting in low scalability [Abolhasan et al. 2004; Belding-Royer 2003; Liu and Kaiser 2003]. The performance of this kind of protocols would degrade as the network size increases, which means it fails to meet the basic requirement, i.e., scalability.

To improve routing scalability, one alternative to the flat routing protocols is clustering or hierarchical routing protocols [Belding-Royer 2003]. In this approach, nodes take different roles, such as cluster heads, anchors, root nodes, agents, and gateway nodes based on the structure used and the organization of nodes in the network [Boukerche et al. 2011; Sucec and Marsic 2002; 2004; Yang et al. 2007; Yu and Chong 2005]. The basic motivation behind these protocols is to achieve scalability by limiting the flooding within a certain region, which in turn, reduces the overall traffic overhead on the control and data planes. By grouping nodes into clusters, only selected nodes forward the route discovery packets, thus reducing redundant traffic [Belding-Royer 2003; Rajaraman 2002; Yang et al. 2007].

Many schemes have used clustering/zone and parent/child relationship to localize flooding in order to minimize the traffic overhead for better scalability. For instance, [Ritchie et al. 2006] proposed a scalable on-demand routing protocol that is based on constant density clustering, where density refers to the number of cluster-heads per unit area. This scheme forms an overlay network of cluster-heads and enables a non-head node in a cluster to reach its cluster-head in one hop. The hierarchical routing protocols are effective to an extent in achieving network scalability and minimizing flooding, but give rise to other challenges, e.g., single point of failure, long routes and centralized information management [Yu and Chong 2005]. So, in case of high node mobility, node churn rate, and link failures, these protocols are vulnerable to information loss, increased traffic overhead and network performance degradation [Abolhasan et al. 2004; Chen and Heinzelman 2007; Sucec and Marsic 2004; Yu and Chong 2005]. COB [Ritchie et al. 2006], CEDAR [Sivakumar et al. 1999], ZRP [Samar et al. 2004] and CBRP [Jiang 1999] are some well-known hierarchical routing protocols for MANETs.

## 1.2. GEOGRAPHIC ROUTING AND VIRTUAL COORDINATE ROUTING PROTOCOLS FOR MANETS

Routing protocols that utilize knowledge about the geographic location of nodes and their position in the network is called geographic routing. GPSR [Karp and Kung 2000], LAR [Ko and Vaidya 2000], GLR [Na and Kim 2006] and SOLAR [Ghosh et al. 2007] are some well-known geographic routing protocols for MANETs. The position of a node is obtained through GPS or any other external positioning system. The aim of using geographic position is to confine the route search space into a smaller estimated range and localize broadcasting of queries. Geographic routing reduces the routing overhead and scales better in terms of per router state because it operates without routing tables and node location information is maintained only at the router/relay-nodes. Unlike source routing protocols that allow a source node to partially or completely specify the route a packet takes through the network, non-source routing protocols determine the path at each node based on the packet's destination address. Although geographic routing is suitable for highly mobile ad hoc networks, it gives rise to a few new challenges. These protocols may suffer from dead ends while routing packets. Obtaining a node's coordinate information is also an expensive task. Furthermore, GPS fails to work in some circumstances, like indoor or in a tunnel [Alvarez-Hamelin et al. 2006; Mauve et al. 2001]. Another drawback is geographic routing may select long detour paths when there are voids between the source and destination [Na and Kim 2006].

In order to avoid the problems associated with using GPS, the research community has investigated different ways to determine the coordinates of nodes in the network. One of these approaches is the use of virtual coordinates, which are obtained when a node is switched on and updated each time the node changes its location. The virtual coordinate system is constructed to find an embedding of nodes into a multi-dimensional space to reflect the underlying connectivity of the network [Cao and Abdelzaher 2006; Caruso et al. 2005; Sheu et al. 2009]. The coordinates are either assigned randomly or based on the hop distance between a node to one or more local/global landmarks or anchor nodes that are selected randomly [Cao and Abdelzaher 2006]. [Zhao et al. 2007] assign a multi-dimensional Hop ID to a node based on its hop distance from all landmarks in the network. The landmark selection algorithm is controlled by a coordinator that keeps track of all landmarks in the network. The virtual coordinate based schemes generate extensive traffic overhead when selecting landmarks and assigning virtual coordinates to all nodes, especially when nodes frequently change their geographic positions and leave/join the network.

Above we have discussed the basic concepts, aims, merits and demerits of four basic classes of routing protocols in MANETs. There are several surveys and tutorials available related to routing issues and solutions regarding different aspects of MANETs, none has discussed the detailed classification and challenges related to DHT-based routing in MANETs. To the best of our knowledge, the protocols reviewed in this survey have not been discussed in this perspective. Previous articles have mainly focused on proactive, reactive and hybrid protocols that use flat addressing, cluster/zone based, hierarchical addressing, and GPS. In the following paragraphs, we give an overview of a few existing surveys related to routing protocols in MANETs in order to distinguish our contribution in this paper.

An overview of the state of the art position-based routing protocols for MANETs is provided by [Mauve et al. 2001] that compares different types of location services, and concludes that GLS [Li 2001] and Home zone [Stojmenovic 1999] provide useful location services. Later, [Liu and Kaiser 2003] discuss and compare routing protocols based on how they structure and delegate the routing tasks, exploit network metrics, and evaluate topology. Similarly, [Abolhasan et al. 2004] conclude that proactive proto-



cols based on flat addressing can be made more scalable using GPS. Also, in hierarchical routing, the overhead for location management and a single point of failure can be controlled using GPS. The authors further conclude that the major problem with these schemes is the traffic overhead for location management.

[Yu and Chong 2005] provide a fairly comprehensive overview of clustering and cluster-based routing protocols in MANETs by classifying these into dominating set-based, low maintenance, mobility-aware, energy efficient, load balancing, and hybrid clustering protocols. [Hanzo and Tafazolli 2007] summarize issues in diverse QoS routing solutions for MANETs and classify the routing protocols based on their interaction with the MAC layer. Similarly, [Chen and Heinzelman 2007] emphasize on considering bandwidth/delay estimation, overhead in route discovery, and in-band signaling for resource reservation in designing a MANET routing protocol that supports QoS. Moreover, the authors conclude that cross layer design is the key to provide QoS to applications running on MANETs. On the other hand, [Li and Wang 2007] compare diverse routing protocols and related mobility models in VANETs on the basis of node position information, structure used and the way these protocols are evaluated. The authors conclude that position-based routing and geo-casting are more promising than other routing protocols.

[Marwaha et al. 2009] review a variety of ant-based routing proposals and conclude that distributed cooperative mobile agent can reduce control overhead compared to proactive routing protocols. After analyzing a number of routing protocols in MANETs, [Shrivastava et al. 2011] also conclude that congestion-adaptive routing is more promising than congestion aware routing. [Anand and Prakash 2010] compare different MANET routing proposals by considering energy efficiency as the key performance indicator. Similarly, [Boukerche et al. 2011] provide a taxonomy of routing protocols in MANETs and uncover the requirements of different protocols. [Thanh et al. 2009] review a few DHT-based protocols that are designed to work in WSN. The survey does not provide any classification, challenges, and concludes nothing. [Gurmukh Singh 2012] provides a very precise review of only three DHT-based protocols for routing. It does not provide any classification, potential challenges, and comparisons of DHT-based protocols. [Fersi et al. 2013] investigate mostly those DHT-based protocols that are designed for data management at the application layer in WSN. The survey classifies DHT-based protocols into flat and hierarchical protocols and concludes that sensors dynamism, asymmetric link detection, and bootstrapping are rarely considered and need to be researched. None of the challenges discussed in Section 2.2.4 are explained by [Fersi et al. 2013]. Moreover, the classification of DHT-based protocols and the literature reviewed in this survey are significantly different from the same in [Fersi et al. 2013].

The surveys discussed above mainly aim at classifying and comparing MANET routing protocols based on different attributes and performance indicators, none focuses on DHT-based routing protocol that are designed primarily to conduct routing at network layer. In this paper, we present a comprehensive survey on DHT-based routing protocols for MANETs proposed in the past ten years. This paper differs from previous surveys as follows: i) to the best of our knowledge, this survey is the first that attempts to review comprehensively and discusses critically the most prominent DHT-based routing protocols developed for MANETs; ii) it presents a fine-grained taxonomy of DHT-based routing protocols based on how these protocols use DHT; iii) it compares the features and limitations of existing DHT-based routing protocols and highlights key research challenges that are vital to be addressed in order to achieve scalability in MANETs.

The rest of this paper is organized as follows: In Section 2, we discuss in detail the basic concepts, detailed classification, potential challenges, and shortcomings of

existing protocols related to DHT-based routing for MANETs. Section 3 discusses a few emerging fields of research and the implications of DHT-based routing in those fields. Lastly, Section 4 concludes the paper.

## 2. DHT-BASED ROUTING IN MANETS

As an increasing number of users would like to use MANET to share data (text, audio, video, news etc.) with other people, one major requirement for the MANET applications is to support a large number of users (nodes), which is possible only if the core routing protocol is scalable. We use the term 'traditional protocols' to refer routing protocols discussed in Section 1.1 and Section 1.2.

In traditional protocols, the IP address is used to identify a node in the network and for routing. Therefore, the node identity is equal to the routing address of the node (static addressing). This assumption is not valid for MANETs because the node changes its location. In MANETs, the node should have a routing address that reflects its relative position with respect to its neighbors nodes (dynamic addressing) [Caleffi and Paura 2011; Caleffi et al. 2007]. The routing protocols that use MAC or IP addresses as node identifiers to perform routing rely on flooding or network-wide dissemination of routing information because these identifiers are independent of the relative location of nodes in the network.

The traditional protocols in Section 1.1 suffers from redundant transmissions due to flooding during route discovery, which affects scalability. Clustering mechanism or hierarchical routing protocols are effective to an extent in localizing flooding, but suffer from a single point of failure, long routes, and centralized information management [Abolhasan et al. 2004; Chen and Heinzelman 2007; Sucec and Marsic 2004; Yu and Chong 2005].

Similarly, the traditional routing protocols in Section 1.2, are effective in controlling flooding, but introduces new challenges, e.g., these protocols suffer from dead ends while routing packets, obtaining coordinate information via GPS is expensive and it does not work indoors or in a tunnel [Alvarez-Hamelin et al. 2006; Mauve et al. 2001]. Moreover, these protocols introduce long detour paths when there are voids between the source and the destination [Eriksson et al. 2007; Na and Kim 2006]. Also, the assignment of virtual coordinates using landmark results in extensive traffic overhead, especially in application scenarios, where nodes frequently change their geographic positions and leave/join the network.

The shortcomings of traditional protocols are the key factor that limits the network scalability. It would be possible to support a large network if we could eliminate network-wide flooding and minimize routing overhead.

In order to achieve this goal, for the past few years, researches have focused on utilizing a DHT structure as a scalable substrate in order to provide a diverse set of functionalities, like information distribution, location service and location-independent identity, with which various self-organized applications can be built [Das et al. 2008; Frey 2004; Viana et al. 2005]. In a self-organized system, the identity and location of nodes are considered separately because nodes are mobile and the network topology continuously changes. In this context, providing a scalable location service in a situation, where there is a relationship between the location and identity of a node is a challenging task. This challenge evolves the concept of dynamic addressing, where a node changes its address according to its location. DHT provides a scalable way to decouple node location from its identity and facilitate general mapping between them.

Before discussing the detailed classification and challenges of DHT-based routing in MANETs, the following section explains the basic DHT concepts that would be helpful in understanding the whole idea of integrating DHT at the network layer (see Section 2.2.2) for the purpose of routing.

Table I. Definitions of important terms related to DHT-based Routing in MANETs

Anchor Node (AN)	A node that holds the mapping information of other nodes with respect to its logical identifier space portion (LSP). Any node in the logical network can act as an Anchor Node.
Logical Identifier (LID)	A unique ID that identifies a node in the Logical Identifier Structure (LIS) and it describes the relative position of that node in the LIS.
Logical Identifier Space (LS)	A user defined address space from which each node obtains its LID.
Logical Identifier Structure (LIS)	A logical network that arranges nodes according to their LID following some structure, e.g. a cord [Awad et al. 2011] and a ring [Caesar et al. 2006].
Logical Network (LN)	The interconnection of nodes based on their LIDs is called Logical Network.
LS Portion (LSP)	A subset of the entire LS that is disjoint from that of other nodes.
Universal Identifier (UID)	An identifier of a node that is public, unique, and remains the same throughout the network lifetime. It could be the IP or MAC address of a node.

## 2.1. DISTRIBUTED HASH TABLES (DHTs) AND DHT-BASED LOGICAL IDENTIFIER STRUCTURE (LIS)

DHT supports a scalable and unified platform for managing application data. It provides a logical identifier-based indirect routing and location framework [Eriksson et al. 2007]. Moreover, it offers a simple application programming interface for designing a protocol that can be used for a variety of applications [Baccelli and Schiller 2008; Eriksson et al. 2007]. Table. I lists the definition of important terms to clarify the concepts related to DHT-based routing. DHT maps application data/values to keys, which are  $m$ -bit identifiers drawn from the LS. A node participating in DHT is assigned a UID and a LID. The LID is drawn from the same LS [Shah et al. 2012]. Each node has a disjoint subset of the whole LS, called LSP, which is used to store the database of keys of application data/values to resolve address resolution queries. A data item itself or its index information is stored at node  $P$  if the key of the data item falls in the LSP of  $P$ . DHTs provide two-methods, namely  $\text{Insert}(k,v)$  and  $\text{Lookup}(k)$ , where  $k$  and  $v$  represent the key and its value, respectively. A DHT defines how the LIS is fabricated (i.e., it defines the logical addressing of nodes), how node state is maintained (i.e., lookup procedure) and how communications between nodes is carried out in LN (i.e., routing). All these operations depend upon the structure in which these nodes are interlinked with each other. Fig. 2 illustrates an example of the basic concepts related to DHT-based addressing, look-up and routing. The range of the LS is 0 to  $2^m$ , where  $m=3$ . The alphabets refer to the UID of nodes, while the numeric digits refer to the LID of nodes. The nodes are arranged in a ring shaped LN with an increasing order of their LIDs. Each node maintains its 1-hop logical neighbors ( $L_{nbr}$ ) in the ring, i.e., its predecessor and successor nodes and physical neighbors to perform routing on both control and data planes. A greedy routing approach is adopted in which a neighbor with the closest LID compared to the destination node's LID becomes the next hop towards the destination node. A physical network of six nodes with its corresponding ring-LN is illustrated in Fig. 2. Below is an explanation of the operations in a logical network:

**a) LID Addressing:** To join a network, a node is assigned a LID either by hashing the UID of the node, or based on the LIDs of its neighbor nodes. For example, a node with UID  $f$  obtains its LID 5 from its logical neighbor node  $e$  with LID 4 as shown in Fig. 2(a). In addition to its LID, node  $f$  obtains its corresponding LSP (5-8) that is a subset of the whole LS.

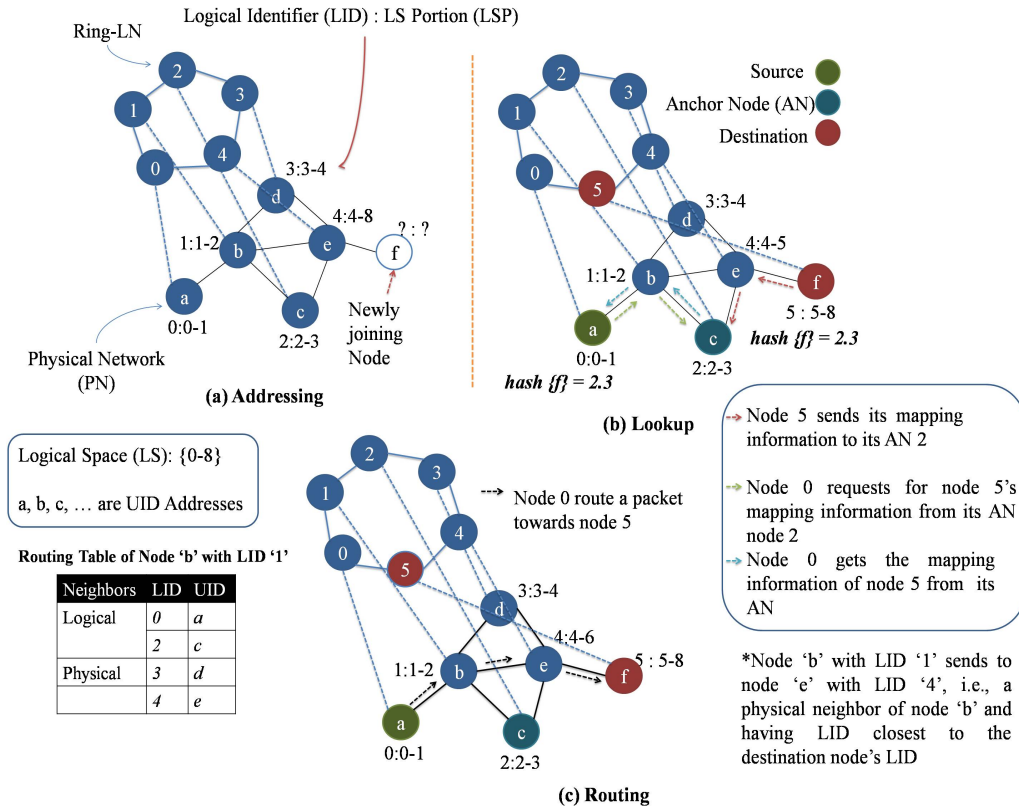


Fig. 2. An example of DHT-based routing

**b) Lookup:** After computing its LID, a node computes its anchor node (AN) in order to store its own mapping information. For this purpose, a consistent hashing function, e.g., SHA-1, is used that takes the UID of the joining node as input and generates a hashed value  $h(v)$  within the range of LS. LIDs of nodes and  $h(v)$  are drawn from the same logical identifier space (LS). A node whose LID is closest to the  $h(v)$  becomes the AN for the joining node's LID. Referring to Fig. 2(b), node 5 computes the LID of its AN by applying the hash function on its UID as  $hash(f) = 2.3$ . The resulting hashed value (2.3) is closest to node with LID 2 and also falls in its LSP, which is 2-3. This means that node 2 acts as an anchor for node 5. So, node 5 then stores its mapping information (LID, UID and LSP) at node 2. For this purpose, node 5 selects one its logical and physical neighbor nodes with LID closest to the hashed value, i.e., 2.3. Similarly, each intermediate hop repeats the same process until the mapping information arrives at node 2 as shown by the red dotted arrows in Fig. 2(b). Let's say, node 0 wants to send a data packet to node 5. The first step is then to locate the AN of node 5 by applying a hash ( $f$ ), which results clearly in hashed value, i.e., 2.3 that is closest to node with LID 2. A request query is then routed towards node 2 as shown by the green dotted arrows in Fig. 2(b). node 2 responds with the reply containing the mapping information (i.e., LID and LSP) of node 5 (see the light blue dotted arrows in Fig. 2(b)), which allows node 0 to communicate directly with node 5 as shown in Fig. 2(c).

**c) Routing:** To route a data/control packet to any destination, a source node forwards the data packet to one of its neighbor nodes, which has the closest LID to that of the destination LID in the packet. This process repeats until the data packet arrives

at the destination node. The route traversed by a data packet from node 0 to node 5 using its LID and LSP is given by the black dotted arrows in 2(c).

A LIS/overlay/LN is a layer on top of the physical network (PN) [Shah et al. 2012][Abid et al. 2014a]. Therefore, a direct link between two nodes in the LIS may span multi-hops in the PN [Shah 2011], as shown in Fig. 3. Each node stores information about a certain number of logical neighbors, depending on the specification of the routing algorithm, and employs a deterministic algorithm to route the query for key  $k$  from the requesting node to the destination node. This lookup is achieved in  $O(f(n))$  logical hops, where  $f(n)$  is a function of the number of neighbors a node has in the LIS. Now that we have introduced the basic terms and concepts of DHT-based rout-

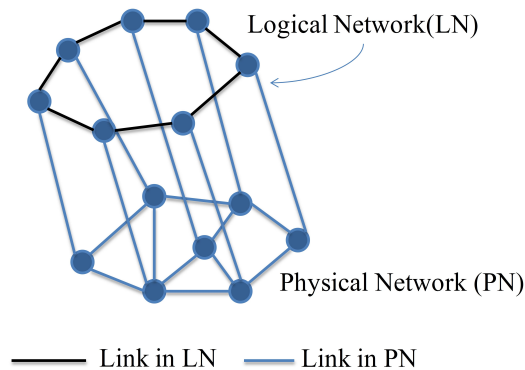


Fig. 3. Logical Network over Physical Network

ing and location services, the following sections describe in detail the classification, challenges, and features of DHT-based routing protocols, followed by a critique of the existing work.

## 2.2. CLASSIFICATION OF DHT-BASED ROUTING PROTOCOLS

The DHT-based approaches were initially proposed to work at the application layer for peer-to-peer (P2P) overlay over the Internet. Later on, researchers have exploited these protocols to work with MANETs, which have a totally different network architecture compared to the Internet. DHT-based LIS is investigated for MANETs in two ways:

- (i) Due to advances in wireless and mobile technology, P2P overlays can also be deployed over MANETs and several approaches have been proposed to do so, we call these approaches DHT-based overlay-deployment protocols. These approaches are designed to work at the application layer and rely on the underlying routing protocol at the network layer. An overview of these approaches is given in Section 2.2.1.
- (ii) Both DHT-based P2P overlays and MANET share common characteristics such as self-organization, decentralized architecture, and dynamic topology. There is a synergy between P2P overlays and MANET [Hu et al. 2003], which can be exploited for large scale routing. In the past few years, DHT-based overlays have been adopted for large scale MANET routing protocols by implementing DHT directly at the network layer [Awad et al. 2011; Awad et al. 2008; Caesar et al. 2006; Caleffi and Paura 2011; Caleffi et al. 2007; Eriksson et al. 2007; Garcia-Luna-Aceves and Sampath 2009; Sampath and Garcia-Luna-Aceves 2009; Jain et al. 2011; Zhao et al. 2009]. We name these approaches DHT-based paradigm for large scale routing. An overview of these approaches is presented in Section 2.2.2.

**2.2.1. DHT-based Overlay Deployment Protocols.** We first discuss a few state-of-the-art P2P overlay based resource/service discovery mechanisms. Then, we discuss several schemes that have been proposed for P2P networks over MANETs.

[Stoica et al. 2001] propose a decentralized overlay deployment protocol, named Chord, that stores key-value pairs for distributed contents. Chord assigns a  $m$ -bit LID to each node from a pre-defined identifier space  $0$  to  $2^m - 1$  by applying a hash function over node's UID. Similarly, each content is assigned a key  $k$  by hashing the content name from the same identifier space that is used to assign LIDs to nodes. Chord arranges nodes in a ring structure in order of increasing their logical identifiers. The key-value pairs are placed at the first Chord node, whose LID is equal to or greater to the value of the key. This node is called the successor node of  $k$ . When a new node  $n$  joins the network, certain keys are assigned to  $n$  that previously assigned to its successor. In case node  $n$  leaves the Chord, all of the key-value pairs stored at  $n$  are transferred to  $n$ 's successor. Fig. 4(a) illustrates a Chord ring of  $m = 6$  with ten nodes. Lookup queries involve the matching of key and node LID.

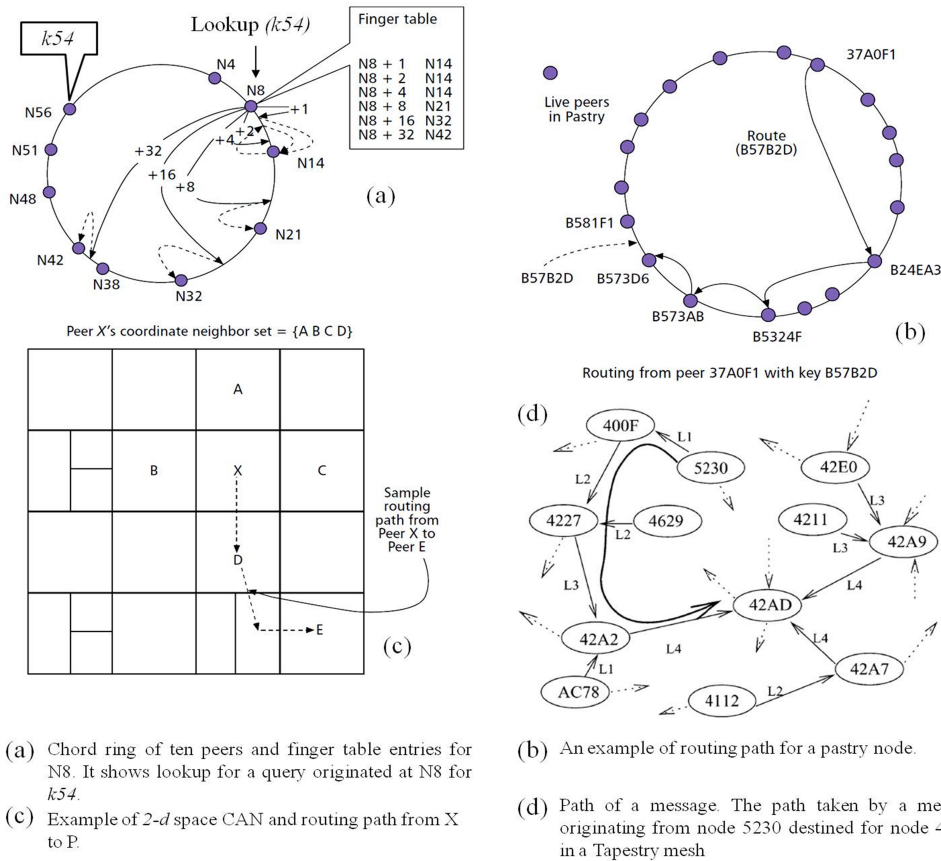


Fig. 4. Functionalities of a) Chord, b) Pastry, c) CAN, d) Tapestry [Eng Keong, Crowcroft, Pias, Sharma and Lim 2005]

For instance, node 8 in Fig. 4(a) performs a lookup for  $k=54$ . Node 8 maintains a routing table with up to  $m$  entries, called a finger table as shown in Fig. 4(a). The first entry in the finger table of node 8 points to node 14 in the Chord ring, as node 14 is the

first node that succeeds  $(8+2^0) \bmod 2^6 = 9$ . Similarly, the last entry in the finger table of node 8 points to node 42, i.e., the first node that succeeds  $(8 + 2^5) \bmod 2^6 = 40$ . In this way, each node maintains a finger table and store information about a small number of other nodes in the Chord ring. Node 8 initiates the lookup operation for  $k=54$ , which eventually returns the successor of  $k=56$ , i.e. node 56 using the path node 8 -> node 42 -> node 51 -> node 56. The response to the lookup query is returned along the reverse of the path.

In the steady state, each Chord node maintains routing information about  $O(\log N)$  other nodes, and resolves all lookups via  $O(\log N)$  messages to other nodes, where  $N$  is the number of nodes in the network. To update the routing table or in case of nodes leaving and joining the ring, Chord requires  $O(\log_2 N)$  messages [Meshkova et al. 2008]. The LIDs to nodes are of their physical topology. The LIDs are assigned to nodes without taking the physical topology into account, which means a single hop in the overlay network would be multiple hops long in the physical network.

[Rowstron and Druschel 2001] propose a decentralized object location and routing protocol, named Pastry, which randomly assigns each pastry node a 128-bit LID from a circular logical space that ranges from 0 to  $2^{128}-1$  such that the resulting set of nodes' LIDs is uniformly distributed in the 128-bit logical space. The LIDs and keys are a sequence of digits with base  $B$  value. Pastry is a hybrid protocol, where lookup for a key-value pair is performed either in a tree-like structure or a ring like manner similar to Chord. In Pastry, a message is routed towards a node whose LID is numerically closest to the given key  $k$ . Pastry uses prefix routing in which a node  $p$  forwards the message to a node  $q$  whose LID is at least one digit (or  $b$  bits) longer than the prefix that  $k$  shares with the  $p$ 's LID. Fig. 4(b) illustrates route from pastry node *37A0F1* for key *B57B2D*.

Pastry takes into account the physical proximity of nodes in the overlay network. A pastry node maintains three tables (a routing table, a neighborhood set, and a leaf set) to assist the routing process. The routing table complexity is  $O(\log_B N)$ , where  $B$  is typically equal to  $2^b$  with  $b=4$  [Meshkova et al. 2008]. Each entry in the routing table of node  $p$  contains the UID of a pastry node whose LID shares the  $p$ 's LID in the first  $n$  digits, but whose  $(n+1)th$  digit has one of the  $B-1$  possible values other than the  $(n+1)th$  digit in the  $p$ 's LID [Lua et al. 2005]. The neighborhood set of a pastry node  $p$  contains the LIDs and UIDs of  $B$  or  $2B$  pastry nodes that are closest in proximity to  $p$ . Pastry uses UID routing geographic distance as the scalar proximity metric. The leaf set of a pastry node  $p$  consists of pastry nodes with  $B$  or  $2B$  numerically closest larger LIDs and  $B$  or  $2B$  numerically smaller LIDs with respect to the  $p$ 's LID. Pastry guarantees delivery of messages with good reliability and fault resiliency even with concurrent pastry nodes fails, unless  $B/2$  or  $2B/2$  pastry nodes with adjacent LIDs fail simultaneously.

[Ratnasamy et al. 2001] propose a distributed content addressable P2P infrastructure called CAN. CAN is designed around a virtual multi-dimensional coordinate space on a multi-torus. This coordinate space is randomly partitioned into zones and each node has its own distinct zone. A CAN node keeps information (i.e., UID and logical coordinate zone) of its  $2d$  neighbors, where  $d$  is the dimensions of the logical space. When a new node joins the system, an existing CAN node splits its zone into two halves. The existing CAN node retains first half and allocates the other half to the newly joining node. In addition, the existing CAN node handovers the key-value pairs corresponding to the other half, i.e., allocates to the newly joining node. After obtaining its zone, the new peer learns the UID of its neighbor nodes. In case a CAN node leaves the system, CAN ensures that one of its neighbor node takes over its zone. A CAN node uses soft-state updates to ensure that all of its CAN neighbor nodes learn

about the changes occurred in its routing information and update their neighbor tables accordingly.

In CAN, the key-value pairs are mapped uniformly on the multi-torus by using a hash function and each node stores the key-value pairs that are allocated to its zone. CAN uses greedy routing strategy, where a message is routed to the neighbor of a node that is closer to the required location. For  $N$  number of nodes in the network and multi-torus with  $d$  dimensions, the lookup complexity of CAN is  $O(d \cdot N^{1/d})$  [Meshkova et al. 2008]. The Fig. 4(c) illustrates a simple routing path from CAN node  $X$  to CAN node  $E$ . The average routing path length in a CAN's  $d$  dimensional logical space partitioned into  $z$  zones is  $(d/4) \cdot (z^{1/d})$  hops.

[Zhao et al. 2004] propose a tree-based P2P overlay, named Tapestry, which employs decentralized randomness to achieve both load distribution and routing locality. Tapestry supports a logical network for locating named content and assigns multiple roots to each content in order to avoid a single point of failure. It differs from Pastry in handling content replication and network locality. It uses the correlation between Tapestry node's LID and content's key  $k$  to route a message. Tapestry uses suffix lookup and routing in which next Tapestry hop is the one that shares a suffix of at least length one with the destination LID. Fig. 4(d) illustrates the path taken by a message from Tapestry node  $5230$  destined for Tapestry node  $42AD$ . Tapestry guarantees the delivery of messages in  $O(\log_B N)$  hops, where  $N$  is the number of nodes and  $B$  is the base value. Each Tapestry node maintains a routing table that consists of levels, where each level  $l$  contains pointers to a set of Tapestry nodes that matches the suffix for that level. Each Tapestry node maintains  $\log_B N$  entries, where  $B=4$ .

Below is a description of a few schemes for DHT-based overlays over MANETs that have been proposed recently.

[Pucha et al. 2004] integrate the functionality of the DHT protocol operating in a logical namespace with an underlying MANET routing protocol operating in a physical namespace. However, the protocol does not consider the hop count between nodes in the physical network, which causes undesirable long end-to-end latency. Furthermore, [Zahn and Schiller 2005] provide an explicit consideration of locality by arranging nodes that have a common logical ID prefix in the same cluster so that they are likely to be physically close. This approach of clustering also helps to reduce control overhead. They use AODV as the underlying protocol and modified it from network-wide broadcast to cluster-wide broadcast. By meeting these requirements, packets take a shorter route in the overlay network as well as in the physical network.

[Kummer et al. 2006] improve Chord [Stoica et al. 2001] over MANET by maintaining a peer's physically adjacent peers along with its logical neighboring peers. A lookup query from peer  $P$  is forwarded to the closest logical neighboring peer among  $P$ 's physically adjacent neighboring peers. This approach also has some limitations. *First*, maintaining the physical adjacent neighboring peers along with the logical neighboring peers generates redundant routing traffic. *Second*, the physically adjacent peers are not necessarily the logical neighboring peers that lead to a random distribution of the DHT structure rather than a systematic one in the network, resulting in a larger file lookup delay. Another approach by [da Hora et al. 2009] to improve the performance of Chord over MANET uses redundant transmissions of the file-lookup query to avoid frequent loss of query packets due to packet collision. This approach suffers from a large file retrieval delay. Also, it does not attempt to construct an overlay that matches the physical network and may perform poorly in MANET.

[Shin and Arbaugh 2009] adopt a different approach by proposing the Ring Interval Graph Search (RIGS) that is suitable for static scenarios. RIGS is not a distributed approach as it requires the topology information of the entire network to construct the spanning tree containing all peers in the physical network for building up RIGS. Sim-



ilarly, [Sözer et al. 2009] use DHT and the topology-based tree-structure to store the file index and the routing information, and unify the lookup and routing functionalities. The limitation of this scheme is that peers (nodes that are participating in P2P overlay) cannot communicate if they are separated by some intermediate non-peer(s) (nodes other than peers in P2P overlay), resulting in P2P network partition. A network partition may also occur at the overlay layer if two peers do not have a parent-child relationship even though they are within communication range in the physical network.

Later, [Shah and Qian 2011] introduce a root-peer in the P2P network. In this approach, each peer stores a disjoint portion of the ID space such that the peer closer to the root-peer has a lower portion of the ID space. This scheme introduces heavy traffic overhead in exchanging information when the node's distance to the root-peer changes.

A more recent approach to P2P overlay proposed by [Shah et al. 2012] focus mainly on the locality of the node and ensuring that neighbors in the overlay network are physically close. Moreover, the LS portions of each directly connected neighboring peers should be consecutive in the overlay. The distribution of LS ensures that physically adjacent peers are also close to each other in the overlay topology. This approach has two limitations. *First*, due to node mobility, peers frequently exchange information about their LID space and the index of the stored files when their distances to the root-peer change, generating heavy network traffic. *Second*, a peer  $P$  (except the root-peer) has at least one directly connected neighboring peer, say  $P1$ , such that  $P1$  is closer to the root-peer than  $P$ , and  $P$  stores the portion of the identifier space higher than  $P1$ 's identifier space. The peer  $P$  might not be a neighbor in the DHT structure (logical space) to all of its physical adjacent peers. Thus, this approach might result in an overlay that does not match the physical network and may perform poorly. Moreover, in [Shah et al. 2012], the LID space distribution among peers is inconsistent, and the peers are not placed in a proper structure (like ring, chord, multi-dimensional spaces, etc.) for the overlay network.

From the above discussion, we identify that the main problems in applying DHT-based P2P overlays in MANETs are: i) lack of explicit consideration of locality; ii) frequent route breaks caused by node mobility and superfluous application level routing due to broadcast in the underlying routing protocols; iii) high maintenance overhead incurred by maintaining the DHT routing structures; and iv) a need for an explicit mechanism to detect the partitioning and merging of P2P overlays at the application layer [Shah and Qian 2010b; 2010a].

Researchers also try to apply the DHT-based overlay-deployment protocols directly at the network layer. Unfortunately, those protocols are designed for the application layer and cannot be used directly at the network layer for routing because they assume that reachability of nodes in the underlying network through the routing protocol. Also, these protocols do not consider network topology changes in the underlying network.

*2.2.2. DHT Paradigm for Large Scale Routing.*: DHT distributes the LS and node location information throughout the network by providing a mapping mechanism that decouples the identification of a node from its location. This characteristic motivates the research community to use DHT to devise large scale routing protocols for MANETs that can be used directly at the network layer. DHT at the network layer is used in three ways:

- (a) **DHT for Addressing** DHT is applied to assign unique LID from the LS, which is used for routing in LN. The LID could be location dependent (i.e., the LID changes with the location of a node and shows its relative position in the LIS; termed as locators [Sampath and Garcia-Luna-Aceves 2009] or location independent (i.e., the LID does not change with the location of the node and is retained for the entire

network lifetime; termed fixed LIDs [Caesar et al. 2006]. LID can be assigned to a node either by hashing its UID from LS (e.g., VRR[Caesar et al. 2006]) or on the basis of LIDs of its neighbor nodes (e.g., VCP [Awad et al. 2011]).

- (b) **DHT as a Location Service** DHT is used to provide a location service to look up the location or mapping information of a node. It provides a distributed location structure to maintain the mapping information of nodes [Viana et al. 2005]. After a node is assigned coordinates using either GPS or GPS-free positioning system [Caruso et al. 2005; Ratnasamy et al. 2003], it advertises its mapping information (i.e., both coordinates and UID) to its AN. For instance, in [Blazevic et al. 2001; Hubaux et al. 2001; Li 2001; Morris et al. 2003; Robert Morris 2002; Xue et al. 2001; Ratnasamy et al. 2003] DHT is used only for location services.

DHT at network layer Protocols	DHT for Addressing				DHT as a Location Service	DHT for Routing	
	Fixed LIDs		Locator			Logical Neighbor Info.	Logical + Physical Neighbor Info.
	By hashing	Via Logical Neighbors	By hashing	Via Logical Neighbors			
L+[Chen and Morris 2002]	-	-	-	✓	✓	-	
Tribe[Viana et al. 2004]	-	-	-	✓	✓	-	
DFH [Alvarez-Hamelin, Viana and de Amorim 2006]	-	-	-	✓	✓	✓	
VRR[Caesar, Castro, Nightingale, O'Shea and Rowstron 2006]	✓	-	-	-	-	✓	
DART[Eriksson, Faloutsos and Krishnamurthy 2007]	-	-	-	✓	✓	-	
ATR[Caleffi, Ferraiuolo and Paura 2007, Caleffi and Paura 2011]	-	-	-	✓	✓	-	
VCP [Awad, German and Dressler 2011, Awad, Sommer, German and Dressler 2008]	-	-	-	✓	✓	✓	
EMP[Jha et al. 2008]	-	-	-	✓	✓	-	
AIR[Garcia-Luna-Aceves and Sampath 2009, Sampath and Garcia-Luna-Aceves 2009]	-	-	-	✓	✓	-	
KDSR[Zhao, Wen and Zhao 2009]	✓	-	-	-	✓	-	
VIRO[Jain, Chen and Zhang 2011, Lu, Jain, Chen and Zhang 2008]	-	-	-	✓	✓	✓	
3D-RP [Abid et al. 2013, Abid et al. 2014]	-	-	-	✓	✓	✓	

Fig. 5. Classification of DHT-based routing protocols based on how they use DHT

- (c) **DHT for Routing** DHT is used to disseminate information (data packets, control packets and mapping advertisements) in LN at both the control and data planes. The routing decisions are made in two ways:

- (i) **Logical information:** The packet forwarding is decided by utilizing only logical neighbors of the node in the LN. The number of logical neighbors depends on the connection order of the LIS. A node determines the next hop among its logical neighbors on the basis of the LIDs of its logical neighbors (1-hop/2-hop). For example, in [Caleffi et al. 2007; Eriksson et al. 2007; Sampath and Garcia-Luna-Aceves 2009; Zhao et al. 2009] the routing decision for a packet is made by utilizing only a node's logical neighbor information.
- (ii) **Logical and physical information:** The routing decision for a packet utilizes logical neighbor information (LIDs and LSPs) as well as physical neighbor information of the node. Here, the physical neighbor information comprises of LIDs and LSPs of physical neighbors that are not adjacent to the node in LIS. A node determines the next hop of a packet based on the LID of both its logical and physical neighbor nodes (1-hop/2-hop). For example, in [Awad et al. 2011; Awad et al. 2008] the routing decision is made at the node by considering both its logical and physical neighbor information.

Fig. 5 summarizes how different protocols use DHT at the network layer in the ways mentioned above.

DHT-based protocols that are mainly designed to work at the network layer in MANETs can be further classified into three categories based on how they implement DHT as described in Fig. 5. *First*, DHT is used for addressing and routing without using lookup services [Caesar et al. 2006; Zhao et al. 2009] referred to as DHT-like protocols (see Section 2.2.5). In these protocols, a node is assigned a fixed LID from the LS by hashing its UID. This LID uniquely identifies a node in the network and is used to perform data routing among nodes. *Second*, DHT is used only for lookup services [Blazevic et al. 2001; Hubaux et al. 2001; Li 2001; Robert Morris 2002; Ratnasamy et al. 2003; Xue et al. 2001]. In these protocols, the node addressing is performed by using either geographical means via GPS or any other position assignment mechanism [Caruso, Chessa, De and Urpi 2005] and DHT provides a distributed location structure to maintain the mapping information of nodes. *Third*, DHTs define the addressing and routing mechanism in addition to location services (see Section 2.2.6).

In this paper, our focus is mainly on the protocols that are related to the *first* and *third* categories because the challenges discussed in Section 2.2.4 are related to the protocols that fall into these two categories. The protocols related to the *second* category do not maintain LIS and do not assign LIDs to nodes from the LS. In this category, the addressing and location services are completely independent. Moreover, the routing decisions are performed at the node based on the addresses (geographic coordinates) obtained by GPS or any other positioning system. Such protocols only utilize DHT to locate the geographic coordinates of the destination in the network. Fig. 6 shows the detailed classification of DHT-based routing protocols. Before describing in detail the challenges that are critical to address in order to design a DHT-based large scale routing in MANETs, the following section briefly describes the advantages and disadvantages of using DHT services for routing in MANETs.

*2.2.3. Advantages and Disadvantages of Implementing the DHT Services:* The following are the advantages of using DHT for routing in MANETs:

- (i) DHT impose a structure on the logical network that enables to choose routing table entries satisfying a certain criteria depending on the respective DHTs [Castro et al. 2010]. This structure allows DHTs to introduce an upper bound of  $O(\log N)$  on the number of hops, where  $N$  is the number of nodes, which means a node needs to coordinate with only a few other nodes in the logical structure to reach the destination node that removes flooding and reduces routing overhead [Gerla et al. 2005].

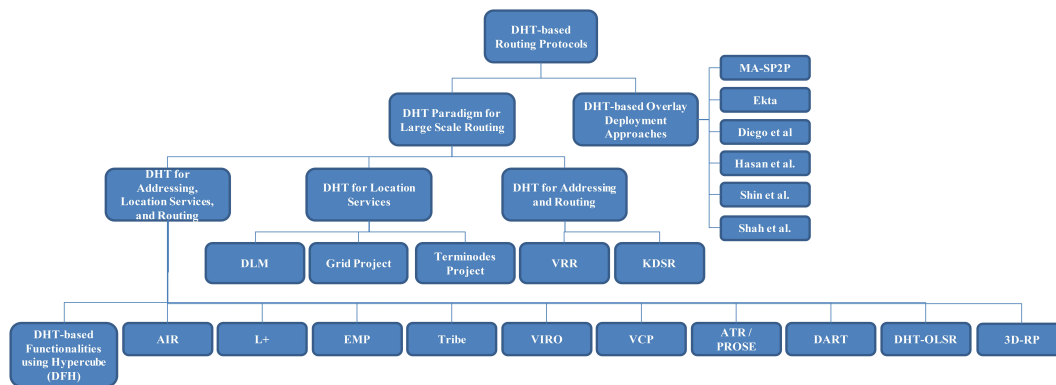


Fig. 6. Classification of DHT-based Protocols

DHT-based approaches outperform non DHT-based approaches when the number of nodes, the number of objects, or the query rate increases, since they do not introduce flooding in the network [Awad et al. 2011; Caleffi and Paura 2011; Eriksson et al. 2007].

- (ii) DHT-based approaches introduces autonomy and decentralization in the system that allow nodes to communicate with each other without any central coordination. This enhances the fault tolerance of the system when nodes continuously joining, leaving, and failing, resulting in a scalable network that functions efficiently even with hundreds and thousands of nodes [Awad et al. 2011; Caleffi and Paura 2011; Eriksson et al. 2007].
- (iii) Unlike non-DHT based approaches (e.g., AODV, DSR), DHT-based approaches for routing similar to the content sharing approaches (e.g., P2P over MANETs) ensure that if the requesting node does not receive reply from an anchor node, then it is either a lookup query or reply to the lookup query has lost in the network due to packet collision. Because, upon receiving the lookup query if the anchor node does not have the LID of the destination node, it sends *NULL* value in the reply to the requesting node. This ensures if the requesting node does not receive reply, then it is either the lookup query or reply to the lookup query has lost in the network due to packet collision.

On the other hand, DHT-based services impose the following disadvantages:

- (i) Unlike traditional proactive routing protocols (e.g., OLSR), the route to the destination node is not immediately available in DHT-based routing protocols. In these protocols, a source node  $s$  first obtains the LID of the destination node  $d$  from  $d$ 's anchor node and then  $s$  sends a packet towards node  $d$  using the  $d$ 's LID. This introduces a delay at the requesting node in order to obtain the LID of the destination node before data is to be sent to the destination node. This introduces delay that can be avoided or reduced by using a caching/replication mechanism.
- (ii) Connectivity of nodes in LIS is the minimal requirement to the functionality of a DHT-based routing protocol that introduces routing traffic in the network. In case of high mobility, the network topology changes more frequently that leads to higher maintenance and routing overhead in the network. Support towards high mobility in DHT-based routing protocols for MANETs is itself a major challenge, which needs an immediate attention.

Before describing in detail the working, features, and shortcomings of protocols that utilize DHT for addressing and DHT for routing in Section 2.2.5 and Section 2.2.6, respectively, the following Section 2.2.4 describes the challenges that are critical to address in order to design a DHT-based large scale routing in MANETs.

*2.2.4. Challenges and Requirements to Develop DHT-based Large Scale Routing Protocols for MANETs:* Now that we have introduced the basic terms, concepts, and detailed classification of DHT-based routing and location services, in this section we describe the challenges that are critical to address in order to design a DHT paradigm for large scale routing in MANETs.

#### (a) Mismatch between Logical and Physical Topologies

In DHT-based LIS, each node is assigned an LID from the LS and is responsible for maintaining a disjoint portion of the LS, i.e., LSP. Also, the node maintains a connection to each neighbor that has an LID close to its own LID. These neighbors are called logical neighbors of the node and can be different from its physical neighbors. The LIS in Fig. 7(a) and Fig. 7(b) describes the logical interpretation of physical topology illustrated in Fig. 7(s). We assume that each node in LIS maintains information about 1-hop logical neighbors. The mismatch between logical and physical topologies, also known as mismatch/ill-match problem, can be analyzed in the following two ways.

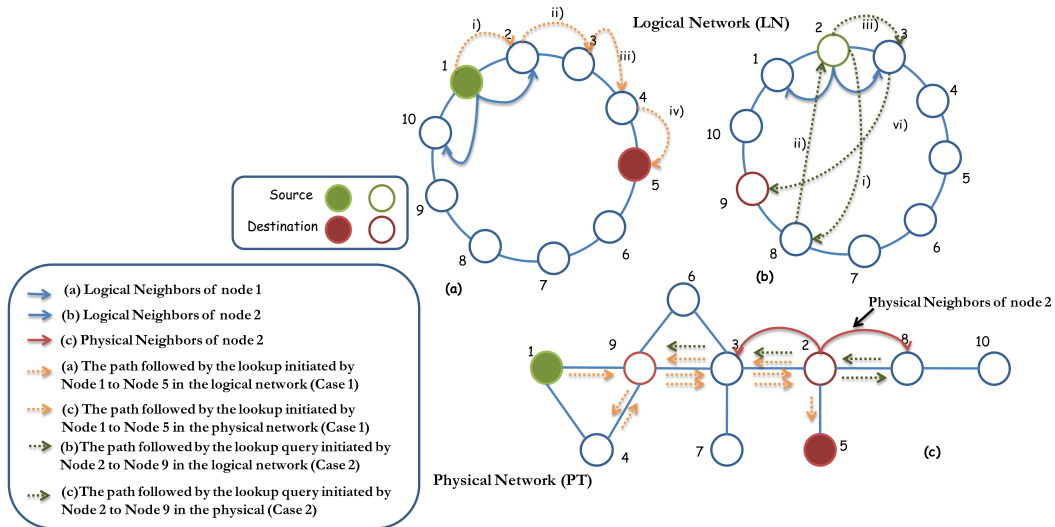


Fig. 7. An example of path-stretch penalty caused by uncorrelated Logical Address Space and Physical Network

**Case 1:** A node's logical neighbors may not be its physical neighbors, resulting in an ill-match between the LIS and physical topology (PT) [Baccelli and Schiller 2008; Shah et al. 2012]. It has a more negative impact in MANETs, especially when LIS is implemented directly at the network layer. Fig. 7 illustrates the ill-match problem between LIS and PT, which causes redundant traffic and high lookup latency.

Fig. 7(a) and Fig. 7(c) show that the 1-hop neighbors of node 1 in the LIS, i.e., node 2 and node 10, are not its adjacent neighbors in the PT (the physical neighbors of node 1 are node 4 and node 9). This results in a mismatch between LIS and PT. Suppose node 1 initiates a query for node 5. The protocol forwards the query to node 2 in the LIS because node 2 is closer to the destination node 5. This produces three

transmissions in PT after passing through links 1-9, 9-3 and 3-2. Upon receiving the query, node 2 forwards the query toward node 3, which is one of its logical neighbors and closer to the destination node 5. This produces one transmission on the link 2-3 in PT. Similarly, node 3 then forwards the query to its logical neighbor node 4 in LIS. This produces two more transmissions in PT, 3-9 and 9-4. Node 4 has node 5 as its logical neighbor in the LIS, which is the final destination of the query. So, node 4 forwards the query to node 5 in the LIS. This produces four more transmissions in PT, 4-9, 9-3, 3-2 and 2-5. The overall transmission for a query from node 1 to node 5 in the LIS produces four transmissions as shown as brown dotted arrows in Fig. 7(a). However, the same produces ten transmissions in PT, shown as the brown dotted arrows in Fig. 7(c). In this example, we can see that the query passes through links 2-3, 3-9 and 4-9 more than once, resulting in redundant traffic as well in larger end-to-end latency.

Based on the problem identified above, the primary requirement in designing a large scale, DHT-based routing protocol is that neighbor nodes in the LIS should also be adjacent in the PT to reduce the end-to-end latency and redundant traffic at both the control and data planes.

**Case 2:** A few approaches [Alvarez-Hamelin et al. 2006; Awad et al. 2011; Awad et al. 2008; Caesar et al. 2006; Werneck et al. 2000] maintain a node's adjacent neighbors in PT along with its logical adjacent neighbors in LIS in an attempt to avoid the mismatch problem in Case 1. This approach is also not effective in completely avoiding the ill-match between the LIS and PT as shown in Fig. 7(b). For example, node 2 initiates a query towards node 9. Node 2 has nodes 1 and 3 as logical neighbors in the LIS, while its physical neighbors are node 3 and node 8 as shown in Fig. 7(b) and Fig. 7(c), respectively. Node 2 selects node 8 as its next hop toward destination node 9 among its physical and logical neighbors (i.e., nodes 1, 3, 8) because node 8 is numerically closest to node 9 by using the greedy routing approach. This moves the query away from node 9 in PT by generating one transmission. After receiving the query, node 8 forwards the query towards node 9 because it is the closest among the logical and physical neighbors of node 8. This further produces three transmissions in the PT, on links 8-2, 2-3, and 3-9, shown as green dotted arrows in Fig. 7(b) and Fig. 7(c). So, to deliver the query from node 2 to node 9, the total number of transmissions in the PT is four, which is higher because there is a shorter route in PT from node 2 to node 9 through links 2-3, 3-9 in PT, which requires only two transmissions (see Fig. 7(c)).

Based on the problem identified above, the second requirement in designing a large scale, DHT-based routing protocol is that a node in LIS should be logically close to all its physically adjacent nodes. This reduces the number of transmissions when forwarding a query/packet to a destination, thus, reducing both end-to-end latency and redundant traffic at the control and data planes.

(b) **High Maintenance Overhead**

The DHT maintenance procedure ensures routing convergence and efficiency in terms of the number of hops in LIS. As network topology continuously changes in MANETs, each node periodically runs some procedures to ensure consistent and up-to-date information in its routing table. Each operation may require a route discovery. The traffic overhead incurred by such procedures is high for bandwidth constrained networks like MANETs. For reactive routing protocols, the overhead is up to  $O(n)$ , where  $n$  is the number of nodes in the network [Shen et al. 2010].

Proactive routing requires periodic flooding of topology control messages, which is particularly costly in MANETs. It is also difficult to achieve convergence in MANETs as frequent topology changes may trigger multiple route discoveries. Furthermore, the ill-match between LIS and PT would worsen this issue because more

bandwidth would be consumed in obtaining routes that are unnecessarily long. The situation could be even worse than simple flooding in resolving requests for data items.

In order to overcome this problem, the third requirement for designing a large scale, DHT-based routing protocol is that a node should control the traffic overhead by carefully calling the DHT maintenance procedures to reduce redundant traffic at both the control and data planes.

(c) **Selection of LIS Structure**

The structure interconnecting the nodes in the LS is another challenge to the performance of a DHT-based routing protocols in MANETs. Different protocols have used different structures, such as cord [Awad et al. 2008], ring [Caesar et al. 2006], [Stoica et al. 2001], hypercube [Alvarez-Hamelin et al. 2006], and binary tree [Caleffi et al. 2007; Eriksson et al. 2007; Caleffi and Paura 2011; Viana et al. 2004] to organize nodes in the LS.

The resilience of a protocol in terms of route selection strongly depends on the shape of LS structure and there is always a tradeoff between robustness and complexity in choosing the LS structure [Gummadi et al. 2003]. For example, tree, ring and cord structures are less complex and easy to maintain. Unfortunately, these structures offer low flexibility in route selection that directly degrades the routing performance and eventually results in poor resilience towards link failures and node mobility [Alvarez-Hamelin et al. 2006]. Moreover, the parent-child relationship in a tree structure inherently suffers from longer routes and the parent node is responsible for maintaining most of the information. This makes the network more centralized.

On the other hand, using multidimensional Cartesian Space structures, such as a sphere or hypercube, for LS can enhance the resilience towards node failure and node mobility, which provides more flexibility in route selection [Viana et al. 2005]. These structures also help in even distribution of the LS among nodes, resulting in a balanced traffic at each node and inefficient bandwidth utilization. Moreover, this type of structure provides a means to map the PT to LIS in such a way that the logical distance between two nodes is close to their physical distance, resulting in shorter forwarding routes between the nodes.

Therefore, the fourth requirement for designing a large scale, DHT-based routing protocol is that the LS structure selection should support flexible route selection. This is an important issue because it directly affects routing performance in terms of path length, traffic concentration and resilience to link failure.

(d) **Address Space Utilization**

Efficient utilization of the LS is one of the major concerns in the design of a large scale, DHT-based routing protocol. The LS should be evenly distributed among all nodes in the LIS. As mentioned in Section 2.1, each node in the LIS holds a portion of the whole LS and stores information about other nodes or data. The LS portions (LSPs) allocated to each node should be equal in capacity so that it results in relatively equal handling of information on each node. This implies that the load to each node should be distributed evenly and each node has an equal opportunity to store information. The benefit of maintaining such a structure is that minimum information has to be transferred in case a node leaves the network, which might directly affect the traffic overhead at both the control and data planes. Also, the traffic overhead can be reduced by effective replication or caching schemes, which are vital for any DHT-based routing schemes. One more element that plays a vital role in distributing the LS is the shape of the LIS.

(e) **Partitioning and Merging**

The limited transmission range of nodes and their mobility can cause both network partitioning and network merging in MANETs. Network partitioning is the breakdown of a connected topology into two or more disconnected parts [Ritter et al. 2004]. A node in one partition cannot access a node in another partition. Network merging is the merging of two or more disconnected topologies into one topology after nodes come into transmission range of each other. In DHT-based protocols, nodes are arranged in a tree, cord or ring, where paths are limited by some hierarchical structure that allows only one path between any two nodes, resulting in low flexibility when selecting routes – this is unlike the greater flexibility offered by the multi-dimensional approaches. There is a higher chance of LIS partitioning, which directly depends on the structure of the LS. As discussed before, if the structure is resilient in terms of route selection because it maintains multiple routes to a node, it would avoid unnecessary route discovery/recovery. If a route to a node is lost due to network partitioning, another route to the node can be utilized provided the node is accessible in the network. Similarly, when two physical networks merge, then their LIS would be disjointed [Shah and Qian 2010b; 2010a]. To detect this situation and merge the LNs, a DHT-based protocol should support seamless merging of LIS, which is a great challenge. Most protocols discussed in Section 2.2.5 and Section 2.2.6 have not addressed the merging of partitioned networks, which is a major concern, especially in DHT-based LIS.

(f) **Replication Strategy/Replica Management**

Effective replication/replica management strategy is crucial to the efficiency of a DHT-based routing protocol in MANETs. The anchor nodes are critical nodes and store the mapping information of other nodes in the network. In case an anchor node  $A$  of a node  $Q$  moves/fails, it would result in an information loss that is stored at node  $A$ . Moreover, a new lookup request for  $Q$ 's LID would not be resolved until  $Q$  selects a new anchor node and updates its mapping information there. In addition, replica management is also a core concern when using any replication strategy. For instance: i) the location of the replica; ii) the overhead related to replicating the information on other nodes; ii) interval to update the replica. The replica management

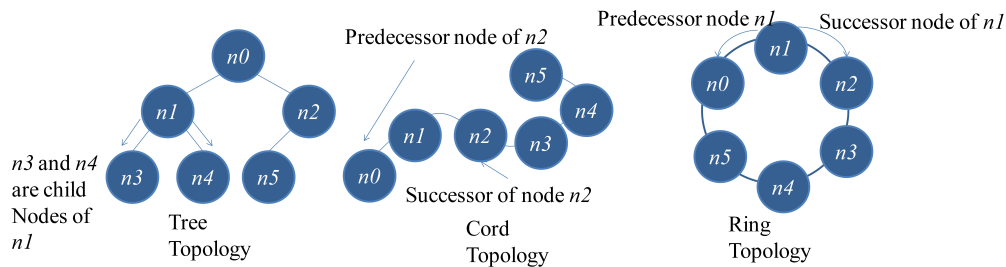


Fig. 8. The Connection Order of Different Logical Structures

and logical neighbors of a node are strongly depend upon the connecting order of a node in the LIS. More flexible is the connecting order of the LIS, resulting in flexible replication strategy. The node  $n1$ 's 1-hop neighbors in both ring and cord LISs are its predecessor and successor that may be used to place replica are shown in Fig. 8. Similarly, the child nodes in the tree-based LIS could be potential locations to place replica. The resilience of the LIS in terms of connecting order of nodes would be helpful in deriving an effective replication strategy for DHT-based routing protocols in MANETs.



To the best of our knowledge, the detailed and effective replication/replica management strategy for the DHT-based routing in MANETs has not been discussed in the existing approaches.

In summary, below are *seven* requirements that must be fulfilled in order to design a scalable DHT-based routing protocol:

- The neighbor nodes in the LIS should also be adjacent in the PT.
- A node in LIS should be close to all its physically adjacent nodes.
- The DHT maintenance procedures should incur minimal traffic overhead.
- The LS structure selection should support flexible route selection.
- The LS should be evenly distributed among all the nodes in LIS.
- The protocol should address the issue of merging partitioned network.
- The protocol should be equipped with an effective replication strategy and replica management policy.

These requirements are matters of great concern and affect the overall route resilience, end-to-end latency, traffic overhead, network throughput, and path-stretch penalty. The path-stretch penalty is the ratio of the path length between two nodes traversed by the routing algorithm to the length of the shortest path available in the network. The existing work discussed in the following Section 2.2.5 and Section 2.2.6 fails to overcome these requirements and suffers from major problems that are yet to be addressed in order to obtain the optimal network performance.

*2.2.5. DHT for Addressing in MANETs:* In this section, we elaborate on routing protocols that perform routing by exploiting the logical identifiers of nodes assigned using DHT-based LS. These protocols do not use DHT-based location services.

[Caesar et al. 2006] propose a DHT-based Virtual Ring Routing (VRR) protocol for MANETs. It is a proactive unicast routing protocol. The proposed scheme organizes the nodes into a virtual ring (LIS) in an increasing order of their LIDs. Each node maintains information about  $r/2$  logical neighbors on each side of the ring (clockwise and anti-clockwise) in a virtual neighbor set (*vset*), where  $r$  represents the cardinality of the *vset* and the value of  $r$  depends on the number of bits assigned to the LID.

Each node also maintains a physical neighbor set (*pset*), which consists of neighbors that are physically close. The link quality at the node towards these physical neighbors must be above a certain threshold value. Each node keeps track of all *vset*-paths to its logical neighbors, including the node itself. The routing table complexity of VRR is  $O(r*p)$ , where  $r$  is the number of virtual paths and  $p$  is the average virtual path length. Fig. 9 illustrates the *vset* with 12-bit identifier (*8F6*) in radix 16, where  $r$  is 4. It also shows the mapping of nodes in the virtual ring to their corresponding location in the physical topology (PT). A node's routing table entry consists of LIDs of the endpoints of the path, the LID of the physical neighbor that could be used as the next hop towards each endpoint and the identifier of each *vset*-path. A newly joining node first initializes its *pset* and *vset* by using its physical neighbors as proxies to forward messages. Forwarding in VRR is simple as the next hop is the one with the numerically closest LID to the destination node's LID. VRR employs a DHT-based randomly hashed LID assignment that produces LIS, which is completely independent of the physical network. Forwarding in VRR is based on the logical distance to the LID of the destination, incurring a path-stretch penalty (which is unbounded in the worst case). VRR detects both node and path failures using only direct communication between physical neighbors. VRR also introduces a symmetric failure detection procedure, which ensures that if node  $n1$  marks a neighbor node  $n2$  as faulty, node  $n2$  would also mark node  $n1$  as faulty. The link/node failures and node dynamics (node joining/leaving and its movement in the network) in VRR might induce a network-wide effect, as two logically

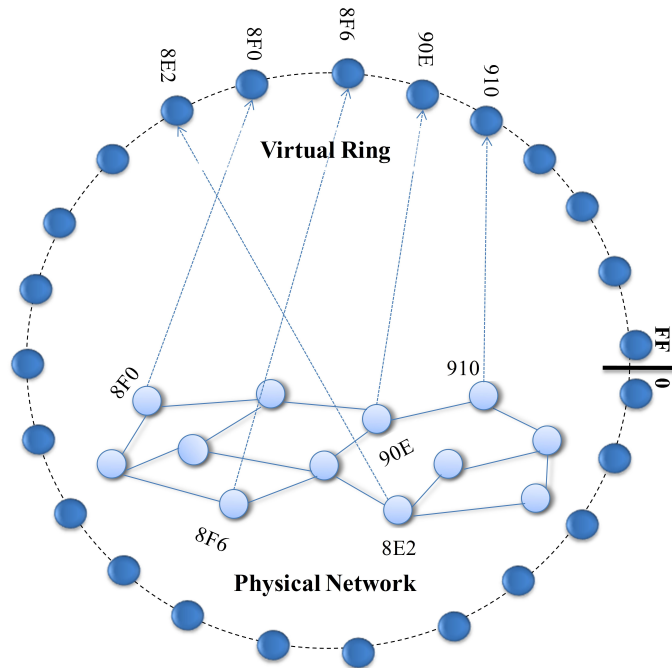


Fig. 9. Relationship between Virtual Ring and Physical Topology

close nodes may be far away in the underlying physical network. The VRR scheme partially addresses partitioning and merging of the ring structures that occurs due to link/node failure. The merging of two disconnected topologies (rings) after coming into each other's transmission range is achieved by selecting one node as a representative of each ring that has an LID close to zero. Each node maintains a route to these representatives and keeps the LID of the representative in its *vset* by exchanging the setup messages. The routes to the representative nodes ensure that the messages can be routed to other ring partition. The protocol achieves a routing complexity of  $O(\log n)$  for  $n$  number of nodes.

In VRR, adjacent neighbors in the virtual ring (LIS) might not be physically close in PT because the LIDs are assigned to nodes without taking the physical topology into account, which leads to Case 1 described in Section 2.1. Moreover, because a node maintains its physical neighbors along with its logical neighbors, it might cause the problem discussed in Case 2 described in Section 2.1. The routing table overhead might be significant because a node maintains all routes to its logical and physical neighbors. Additionally, a node in VRR also maintains routes to destinations for which it is an intermediate node. It also suffers from the partitioning and merging problem that is partially addressed. VRR does not support high node mobility because it produces significant routing overhead in this situation.

A different approach to is taken by [Zhao et al. 2009], called Kademlia-based Dynamic Source Routing (KDSR) that integrates the functionality of both Kademlia [Maymounkov and Mazières 2002] and Dynamic Source Routing (DSR) [Johnson et al. 2007] at the network layer. KDSR is a reactive routing protocol that provides an efficient indirect routing primitive in MANETs. It employs a DHT-based randomly hashed LID assignment that produces LIS and LS that are completely independent of the underlying network topology. Nodes in KDSR store the contact information to each other using  $k$ -buckets. Each node keeps a list of  $k$ -buckets for nodes of distance

between  $2^i$  and  $2^i+1$  from itself, where  $0 \leq i \leq 160$ . To obtain information about logical neighbors, each newly joining node sends a packet to its own LID using non propagating route request. The distance between any two nodes is defined by the bitwise *XOR* of their LIDs. Each entry in the  $k$ -bucket stores a vector of source routes to reach the destination. KDSR not only uses explicit route discovery, but also relies on the implicit route discovery by snooping and overhearing packets in order to find the freshest route to the destination node. KDSR uses the least recently discovered replacement algorithm to update  $k$ -buckets.

To route a packet from the source node  $n1$  to the destination node  $n2$ , node  $n1$  generates the LID of node  $n2$  by hashing  $n2$ 's UID and sends the packet by using the *XOR*-based routing algorithm. Forwarding in KDSR is based on *XOR* distance to the LID of the destination, which might incur high path-stretch penalty in the worst case. KDSR maintains a route cache, created by using the node's  $k$ -bucket, in order to find direct routes to the destination before executing the *XOR*-based routing algorithm. To minimize the route discovery overhead, KDSR uses a non-propagating route request, whose hop limit is 1, if an intermediate node does not find any node to progress in the LS. The basic aim of sending the non-propagating route request is to determine whether the destination node is currently a neighbor of the initiator, or if any of its neighbors has a direct source route, or if there is a closer  $k$ -bucket entry for the destination node. KDSR inherits all the route maintenance features of DSR. In case of a link failure, the node attempts one of the following two options before dropping the packet. The *first* option is the node finds an alternative route from its route cache for the destination. The *second* option is the node sends the packet to the next logical hop using *XOR* distance.

KDSR might introduce extensive traffic overhead in case of link/node failures and node dynamics because two logically close nodes may be far away in the underlying physical network, resulting in unbounded path-stretch penalty in the worst case. It combines the features of traditional routing protocols with DHT to improve performance in terms of short routes. However, KDSR also inherit the limitations of traditional routing protocol as discussed above.

*2.2.6. DHT for Routing in MANETs:* In this section, we discuss in detail routing protocols that use DHT-based LS to provide location services and perform routing in the network based on the logical identifiers assigned to nodes from the same LS.

[Chen and Morris 2002] propose a proactive routing algorithm, named L+, which is designed to enhance the original Landmark system proposed in [Tsuchiya 1988]. L+ uses DHT to implement the location service, landmark hierarchy and routing algorithm to achieve scalability and support node mobility. Each node has a UID and LID that are used for routing. The node's LID is a concatenation of the node's identifier, followed by its ancestor's LIDs, until the root node LID is reached in the logical identifier space. L+ nodes are arranged in a tree-based LIS and the LID of a node describes its relative position in the LIS. The leaves of the tree are called level 0 landmarks. Every node starts out as level 0 landmark.

Each level  $i$  landmark (L+ logical nodes at level  $i$  of the hierarchy) picks the nearest upper level  $i+1$  landmark as its parent within a radius of  $r_i$  hops, where the radius at level 0 is 2, i.e.,  $r_0$  and it doubles every level. If no such landmark node is available, level  $i$  landmark increases its landmark level by one, i.e., level  $i$  landmark is moved to level  $i+1$ . Similarly, level  $i$  landmark decrements its level by one when all level  $i-1$  landmarks can be covered by another level  $i$  landmark. A landmark node keeps information about nodes that are  $2r_i$  hops away from it for level  $i$  landmark.

Each L+ node keeps multiple ANs at exponentially increasing distances. A node sends update information to each level  $i$  landmark whose address is numerically clos-

est to its hashed UID value. Then, the level  $i$  landmark sends the update information to its child nodes that are at level  $i-1$  downward in the hierarchy, and this process continues until the information reaches the leaf nodes. To deliver a packet to a destination node  $n2$ , the source node  $n1$  takes the following steps. *First*, node  $n1$  applies a hash function on node  $n2$ 's UID. This gives the AN address where the LID of node  $n2$  is stored. *Second*, node  $n1$  forwards the query to the AN. *Third*, the AN returns the LID of node  $n2$  to node  $n1$ . *Finally*, node  $n1$  sends the packet to node  $n2$  based on  $n2$ 's LID.

In addition to the shortest path to the destination node, each node keeps information about all other paths with distance one-hop more than the shortest one. To forward a packet to the destination, the node looks for each component of the destination's LID in its own routing table. While scanning the LID from left to right, the leftmost entry (lowest level) is used if it corresponds to a valid node in the structure. Otherwise, the second entry (component) of the LID is used. If a routing failure occurs also when using the second entry, the packet is dropped. The per node communication cost is  $O(\log n)$ , where  $n$  number of nodes.

L+ is limited by the hierarchical tree structure as there exist only one path between any two nodes, which may degrade performance in terms of path length, traffic concentration, and resilience to failures. L+ focuses primarily on the design of scale-free systems. Thus, node mobility may result in lower throughput, extensive traffic overhead, or lost of system stability. [Viana et al. 2004] propose Tribe, a DHT-based proactive protocol for scalable unicast routing in MANET. In Tribe, each node holds the LSP such that physically close nodes in the network also manage consecutive LSPs in the LS. By doing so, the logically close nodes would also be physically close, thus reducing control traffic by avoiding the mismatch problem [Shah et al. 2012]. Each node has a global UID, its AN's LID, and its own LID that describes its relative position in the LIS. LID is an  $m$ -bit identifier drawn from the same LS. Each node keeps information about its 1-hop logical neighbors.

A new node joins the network by broadcasting a request packet to its 1-hop physical neighbors. These physical neighbors reply by sending their LSPs along with other information to the new node. Then, the new node sends a joining request packet to a neighbor with the largest LSP. On receipt of a joining request, the neighbor node splits its LSP into half and assigns the upper half portion to the new node. Tribe follows a tree-like LIS in which descendants of a node  $n1$  have LSPs that are subset of  $n1$ 's LSP. The routing table complexity of Tribe is  $O(k)$ , where  $k$  is the number of 1-hop neighbors. Each node maintains one or more ANs to store its mapping/index information. To find the LID of node  $n2$ , node  $n1$  applies the hash function on node  $n2$ 's UID. This gives the AN's LID for node  $n2$ . Node  $n1$  forwards the query to AN, which then returns the LID of node  $n2$  to node  $n1$ . Node  $n1$  sends the data packet to node  $n2$  by forwarding to one of its 1-hop logical neighbors whose LID is close to node  $n2$ 's LID. These logical neighbors of the node  $n1$  can be one of its children or its parent, or the nodes in different sub trees of the LIS. The forwarding preference among these logical neighbors at node  $n1$  is as follows. *First*, node  $n1$  examines if the LID of node  $n2$  corresponds to one of its children. If so, node  $n1$  forwards the packet to one of its children that is closest to the LID of node  $n2$ . But, if the LID of  $n2$  corresponds to a neighbor of node  $n1$  in a different sub-tree, node  $n1$  forwards the packet to a neighbor with LID closest to  $n2$ 's LID. If both fail, node  $n1$  forwards the packet to its parent. The protocol has the routing complexity of  $O(\log n)$  for  $n$  number of nodes in the network.

Tribe may suffer from longer routes and critical node problem due to the inherent parent-child relationship. This problem is exacerbated if the parent-child address space portions are not contiguous. Moreover, Tribe uses flooding to find a node with a contiguous portion of LS to that of the leaving node, which could produce extensive routing overhead in both the control and data planes. Furthermore, Tribe clones ad-

addresses, which is unsuitable for networks with high mobility because it may lead to extensive routing overhead. Tribe is more suitable for MANETs with low mobility and churn rate.

[Alvarez-Hamelin et al. 2006] propose a DHT-based protocol, referred to as DFH, for unicast routing in MANET based on a hypercube structure in order to increase the number of multiple paths between two nodes. The protocol can work in either proactive mode or reactive mode. Each node has a unique identifier UID and a  $d$ -bit LID in binary form, where  $d$  is the dimensions of the hypercube. The total number of nodes supported in the network is  $2^d$  for  $d$ -dimensional hypercube. A node  $n1$  is logically connected to all nodes whose LIDs differ only in one dimension from that of node  $n1$ , e.g., a node with LID  $0000$  is linked to nodes with LIDs  $0100$ ,  $0010$ ,  $0001$ , and  $1000$  as shown in Fig. 10. A newly joining node broadcasts a request packet to its 1-hop physical

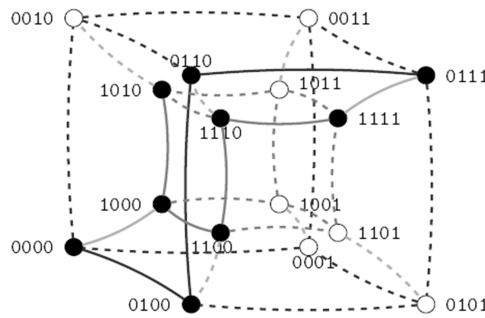


Fig. 10. Hypercube with  $d=4$  [Alvarez-Hamelin et al. 2006]

neighbors to obtain their LIDs. Based on these LIDs, the joining node computes its own LID such that its LID is close to the LIDs of its physical neighbors—this minimizes the mismatch problem between physical and logical topologies. The newly joining node finds its AN by applying the hash function on its UID. Then, the joining node stores its LID and its corresponding LSP at the AN.

In addition to LID, a node also obtains a Secondary Logical Identifier (SLID) if some of its physical neighbors are not adjacent in the LIS, so that a mismatch between LIS and PT can be reduced. The routing table complexity is  $O(d+s)$ , where  $d$  is the dimension of the hypercube and  $s$  is the number of non-adjacent nodes. The LSP of a node is determined by taking the logical AND of its LID and the mask (represented by the number of 1's from left side). The hypercube is said to be incomplete if a node in the LIS (hypercube) is not logically connected to all of its physical neighbors [Shah et al. 2012], which can lead to a mismatch problem between physical and logical topologies. The protocol has partially addressed the mismatch between LIS and PT by assigning SLID to a node when some of its physical neighbors are logically non-adjacent.

The lookup process for AN is similar to the routing of a packet towards the destination node except that in the routing process, the destination's SLID cannot be used. But in lookup process, both LID and SLID of AN can be used as AN's identifier. Let's take Fig. 11 as an example. Node  $n1$  with LID  $0110m3$  wants to obtain the mapping information of node  $n2$  with LID  $1011m3$ . By applying a hash function on the UID of node  $n2$ , node  $n1$  obtains the LID of node  $n2$ 's AN, say for example,  $\text{hash}(n2\text{'s UID}) = 1101m2$ . The hashed value  $1101$  is not managed by node  $n1$  with LID  $0110m3$  as shown in Fig. 11, so it forwards the request packet to one of its neighbors as follows.

The first entry in the routing table of node  $n1(0110m3)$  is  $1100m2 \rightarrow 1111m4$ , as shown in Fig. 11, and this entry matches with  $1101$  because both have  $11$  as their most significant bits. Therefore, node  $n1(0110m3)$  forwards the request packet to node with

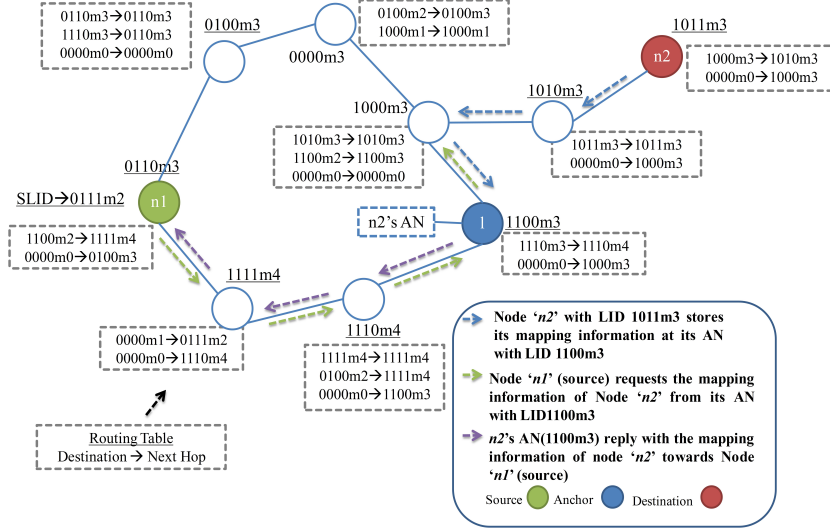


Fig. 11. Spontaneous Network: physical position of nodes

LID  $1111m4$ . After receiving the request packet, the node with LID  $1111m4$  examines the first entry in its routing, i.e.,  $0000m1 \rightarrow 0111m2$  and finds that this entry does not match with  $1101$  (the LID in the request packet). Then, the node with LID  $1111m4$  examines the second entry (i.e.,  $0000m0 \rightarrow 1110m4$ ) in its routing table, which is the default routing entry. Therefore, the request packet at the node with LID  $1111m4$  is forwarded to node with LID  $1110m4$ . This procedure is repeated at every node along the path until the request packet reaches the node with LID  $1100m3$  that holds the address  $1101$  in its LSP. Therefore, node with LID  $1100m3$  sends a reply packet to node  $n1(0110m3)$  in response to the request packet in order to provide the LID of node  $n2$ , i.e.,  $1011m3$ . After receiving node  $n2$ 's LID, node  $n1$  can directly communicate with node  $n2$  using  $n2$ 's LID.

In order to ensure connectivity between two nodes, DFH partially overcomes the mismatch problem by assigning multiple coordinates to a node in order to provide better adjacency among nodes. But, maintaining physical neighbors at a node by using secondary LIDs might lead to Case 2 of the mismatch problem in Section 2.2.4. Moreover, it does not evenly distribute LS among all nodes. Hence, there is a possibility of extensive information loss in case a critical node fails. The protocol is more suitable for networks with low churn rates and node mobility.

[Eriksson et al. 2007] propose DART, a dynamic address unicast routing protocol to deal with the routing scalability issue in MANETs. The main idea is to use dynamic addressing instead of static or flat addressing, which is one of the basic hindrances in achieving routing scalability. DART is an attempt to handle the challenges of dynamic address allocation and address lookup by using DHT. Each node has a UID and an  $L$ -bit LID. The LID of a node reflects the relative position of the node with respect to its neighbors in the logical network. This means that nodes that are close in the physical network topology share a common LID prefix by forming a subgraph in the network topology. DART arranges LIDs in the form of a binary tree with  $L+1$  levels. A

leaf of the tree represents the nodes and their LIDs in the LIS. Each inner node in the tree represents a sub tree that consists of nodes whose LIDs share a common prefix with the inner node. These nodes form a sub graph in the network topology as shown in Fig. 12. The level  $K$  sub tree shares the prefix of  $(L-K)$  bits among the nodes. For example, in the 3-bit LS, the level-1 sub tree can only consist of two nodes, which share the  $(L-1)$  prefix (e.g., 3-1=2 as  $L=3$  in Fig. 12). Two nodes with a longest common prefix would have a shorter physical distance between them in the physical network. DART

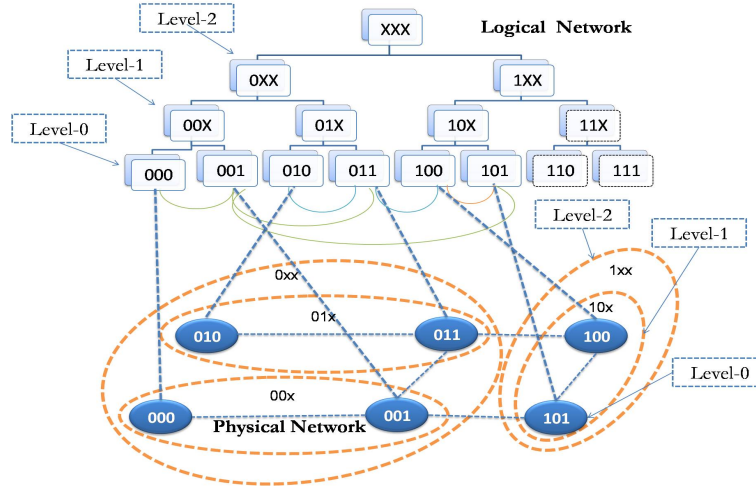


Fig. 12. DART Logical Address Tree and Corresponding Physical Network

proactively maintains routing information and incurs  $O(\log n)$  routing table complexity for  $n$  number of nodes in the network. The newly joining node obtains the unoccupied LID based on the largest set of available LIDs among its physical neighbors. Then, the new node applies a hash function on its UID and stores its LID on the node with LID close to the hashed value of the node's UID. The node that keeps the mapping information acts as AN for the corresponding node. To send a packet to a destination node  $n2$ , the source node  $n1$  obtains  $n2$ 's AN by applying the hash function to  $n2$ 's UID, which gives the AN's LID. Then, node  $n1$  sends a request packet to AN in order to obtain  $n2$ 's LID. This request packet is forwarded in the network as follows:

Node  $n1$  finds the entry in its routing table that has the longest prefix match with AN's LID. If this entry points to one of node  $n1$ 's sibling tree, node  $n1$  forwards the request to the node in that sibling tree. In this routing process, a packet may visit a sub-tree more than once, which could lead to looping. However, DART avoids looping by restricting the forwarding of packets as follows. Each node maintains a route login, where a bit  $k$  is used to ensure that the route update arrives at the node via level- $k$  sibling. This routing procedure is repeated at each intermediate node until the request packet reaches the AN. After receiving the request packet, AN sends a reply packet to the requesting node  $n1$ , containing  $n2$ 's LID along with other information. The reply packet is forwarded to node  $n1$  in the same way the request packet is routed from node  $n1$  to  $n2$ 's AN. After obtaining node  $n2$ 's LID, node  $n1$  can send data packets to  $n2$  according to DART routing. The limitation of DART is its low fault tolerance because only one path is maintained between a node and its siblings, which degrades resilience to failures. This scheme could be vulnerable if either the next hop towards the destination fails or the network is partitioned. The tree-based LIS in DART suffers from

a single point of failure and congestion due to the presence of critical nodes. DART, like L+, focuses primarily on the design of scale-free systems. Thus, node mobility in these approaches may result in lower throughput, extensive traffic overhead, or loss of system stability.

To overcome the limitations of DART, [Caleffi and Paura 2011] propose a DHT-based hierarchical multi-path routing protocol, named M-DART. M-DART exploits augmented tree-based address space structure to achieve scalability, to gain resilience against node churn/mobility, and to avoid link congestion/instability in MANETs. Unlike DART, M-DART proactively maintains all possible routes via its next hop neighbor nodes to reach a destination node in the sibling tree without incurring any additional communication or coordination overhead. In DART, a newly joining node obtains a LID from one of its physical neighbors with the largest unused LSP. This process could result in invalid address assignment and slower convergence [Caleffi and Paura 2011]. However, in M-DART, if a new node obtains an invalid LID from its neighbor because the neighbor's routing table is not updated, the new node examines its other physical neighbors to obtain a valid LID. Furthermore, M-DART uses a caching technique to minimize the traffic overhead associated with the node lookup. This cache mechanism also provides fault tolerance to M-DART's routing process.

Each node in M-DART keeps a subset of pairs in the form of (identifier (UID), network address (LID)) that is assigned to the node based on the hash function. Suppose that node  $n2$  with UID  $id2$  joins the network and picks up the LID  $add2$ . Then, node  $n2$  sends a Network Address Update (NAUP) packet to its AN whose LID is equal to the hashed value of  $n2$ 's  $id2$ , e.g., the LID of AN is  $add3 = \text{hash}(n2\text{'s } id2)$ . M-DART adopts the unicast routing procedure of DART in addition to multi-path routing and caching mechanism. While forwarding the NAUP packet towards AN with LID  $add3$ , every intermediate node along the path also caches the pair  $\langle id2, add2 \rangle$  of node  $n2$ . In case AN with LID  $add3$  does not exist in the network, the NAUP packet is routed to a node with a LID that is at least greater than  $add3$ .

Similarly, to send a data packet to the destination node  $n2$ , the sending node  $n1$  applies a hash function to  $id2$  and obtains the LID of  $n2$ 's AN (say  $add3$ ). Node  $n1$  sends a Network Address Request (NARQ) packet to  $n2$ 's AN to obtain the LID of node  $n2$ . Here, the routing of NARQ is similar to the routing of NAUP. The AN returns the LID  $add2$  of node  $n2$  in the reply to NARQ from node  $n1$ . Node  $n1$  then forwards the data packet to node  $n2$  based on its LID  $add2$ . If node  $n1$  gets multiple paths towards node  $n2$ , it selects the shortest one in terms of the number of hops. In case of a route failure, node  $n1$  resends the data packet through an alternative shortest path.

In M-DART, despite maintaining all routes towards a destination, the scheme does not fulfill the requirement in Case 1 of the mismatch problem in Section 2.2.4 because the LIS does not ensure adjacency of neighbors between LIS and PT.

[Baccelli and Schiller 2008] propose a hybrid protocol called DHT-OLSR that maintains a regular OLSR [Jacquet et al. 2001] routing table along with DHT support that enables DHT-OLSR to provide an efficient and low delay unicast routing. In DHT-OLSR, each node runs OLSR locally within a cluster, which confines the signaling of nodes to a local scope by limiting the TTL of Topology Control packet to two hops. This effectively places each node at the center of its own OLSR cell/cluster with a diameter of four hops. To send a packet, a node first examines the route for the destination in its OLSR routing table. If the route is available, the node sends the packet according to OLSR routing. Otherwise, the node switches to DHT-based routing, which is based on a modified MADpastry [Zahn and Schiller 2005; 2006]. In this mode of routing, the packet is routed based on the node LID drawn from the MADpastry's LS instead of UID. DHT-OLSR uses a unicast scheme to resolve node addresses to their corresponding LIDs as follows:



Each node obtains its AN's LID by applying a hash function on its UID and it sends its mapping information to its AN. In this way, DHT-OLSR reduces routing overhead compared to pure OLSR routing. DHT-OLSR has two limitations. *First*, DHT-OLSR does not address the mismatch problem between LIS and PN that results in path-stretch penalty. *Second*, DHT-OLSR does not consider the node churns that is common in every network.

DHT-OLSR combines the features of a traditional routing protocol with DHT to improve performance for short routes. However, DHT-OLSR also inherits the limitations of traditional routing protocol. It may introduce extensive traffic overhead in case of link/node failures and node dynamics because two logically close nodes may be far away in the underlying physical network, resulting in unbounded path-stretch penalty in the worst case.

[Awad et al. 2011] propose the Virtual Cord Protocol (VCP) in an attempt to achieve routing scalability in MANETs. In VCP, nodes are organized into a cord structure with respect to their LID in the logical identifier space (LS), i.e.,  $[0-1]$ . Each node has a UID and a LID. The LID describes the relative position of the node in the cord structure. In addition to its 1-hop logical neighbors, each node proactively keeps information of its 1-hop physical neighbors. Hence, the routing table size is  $O(k)$ , where  $k$  is the sum of its logical and 1-hop physical neighbors.

A newly joining node obtains its LID based on the LIDs of its 1-hop physical neighbors. If a new node has two 1-hop physical neighbors that are logically adjacent in the cord structure (i.e., these two physical neighbors have adjacent LIDs), it obtains the LID that is in between the LIDs of these two physical neighbors. If the new node has only one 1-hop physical neighbor, it obtains the LID between the LIDs of the physical neighbor and a virtual node that is created by its physical neighbor.

A node forwards the packet to one of its next-hop neighbors with the closest LID to the destination node's LID among the node's logical predecessor and successor, and the node's 1-hop physical neighbors. In case of link failure to the next hop at an intermediate node, the packet is dropped if the next hop is the final destination. Otherwise, the intermediate node creates a No-path interval ( $NP-I$ ), consisting of LIDs that the failed node was responsible for and sends a no path ( $NP$ ) packet containing  $NP-I$  to another active node among its neighbors as shown in Fig. 13. Each node receiving a  $NP-I$  either forwards it to the destination by using a greedy approach or continues to send  $NP$  to another active node in its neighbors. If a node receives a duplicate  $NP$ , it sends a no path back ( $NPB$ ) packet to avoid loops.

In order to improve the reliability of VCP in case of node or link failure, the scheme uses integrated replication strategies. In this approach, VCP exploits the virtual cord to place the replicas at a few logical neighbors along the cord in both directions, which would produce traffic overhead that is twice the number of neighbors to create and manage replicas.

The limitation of VCP is its low fault tolerance. A node failure could split a cord into two disconnected logical partitions, resulting in packet loss and increased end-to-end delay. The protocol maintains both logical neighbor and physical neighbor information, which may lead to the problem discussed in Case 2 described in Section 2.2.4. VCP is unsuitable for networks with high churn rates and high node mobility.

[Garcia-Luna-Aceves and Sampath 2009] propose an approach called Automatic Incremental Routing (AIR), which is a DHT-based proactive approach for both unicast and multicast routing in MANETs. This scheme focuses on two major routing issues, namely flooding and scalability. Each node has a UID and obtains its LID in such a way that the nodes in the logical network form a Labeled Directed Acyclic Graph (LDAG). This LDAG structure is built with reference to a designated node, called the root node. The LID of a node shows its relative position with respect to the root node

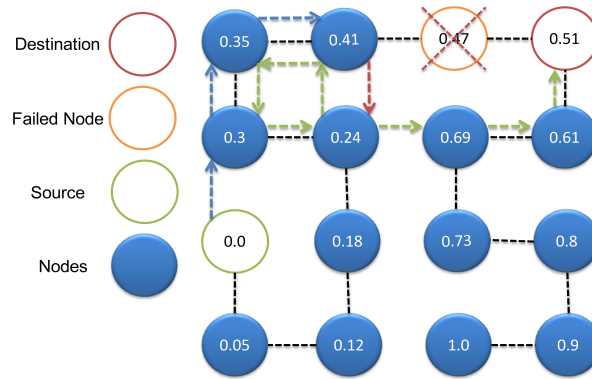


Fig. 13. Node(0.0) sends a packet towards node(0.51). The black dotted line represents a logical cord. The blue dotted line shows the route of the packet. A dead end is detected as node 0.47 fails. The Green dotted line represents the NP packet to find an alternative route. The red dotted line is the NBP to avoid loops. Nodes use greedy forwarding to send packet toward the destination node 0.51

in the LDAG structure. LDAG is established by periodically exchanging hello packets among the nodes, which are propagated in a breadth-first manner from the root node.

Each node maintains information of its 1-hop and 2-hop logical neighbors in two separate tables. After obtaining a LID, a node computes the LID of its AN by applying a hash function on its UID. To store LID at its AN, the node forwards a request packet to one of its neighbor nodes up to two hops away whose LID has the closest prefix matching to the LID of the AN. This routing procedure is repeated at each intermediate node until the request packet reaches the AN. Fig. 14 illustrates the lookup and routing procedures using AIR.

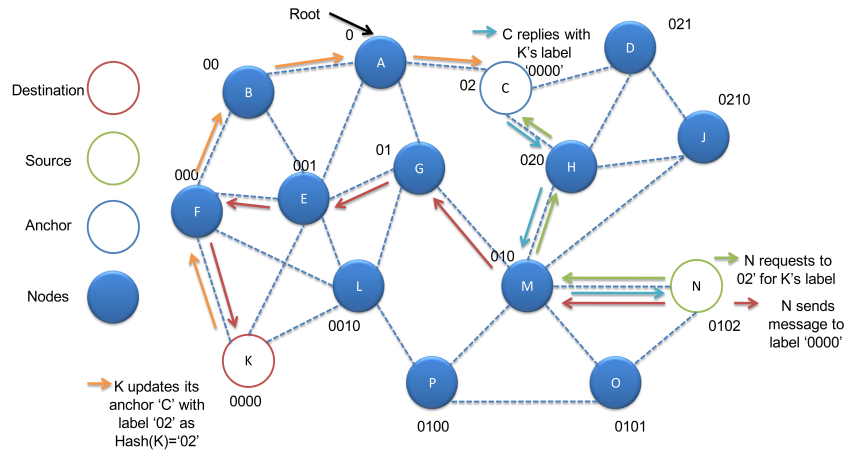


Fig. 14. Node N sends data to node K. Node C acts as the anchor for K

AIR rely on tree-based logical identifier structure that keeps only one path between a node and its siblings. The failure of a next hop towards a sibling node would break the connectivity, leaving the destination set of nodes in the sibling tree disconnected from the forwarding node. Also, the failure of a critical node might cause the re-assignment of all the siblings, thus, increasing the traffic overhead. One of the requirements for any DHT-based routing is that LS resulted from a DHT function mappings should be fixed and static. PROSE and AIR do not assume fixed and static LS.

[Jha et al. 2008] proposed a DHT-based unicast routing protocol called Enhanced Mobile Party (EMP). EMP improves the mobile party (MP) protocol [Sabeur et al. 2007] by an enhanced scheme for maintaining LIDs when a node joins or leaves a network. Similar to MP, the nodes are arranged in a logical tree structure in EMP. Each node has a UID and a LID, which is based on its parent's LID. Each node proactively updates its routing table and maintains only information about its 1-hop logical neighbors. EMP incurs routing table complexity of  $O(k)$ , where  $k$  is the number of its 1-hop logical neighbors. Each node is responsible for a portion of the LS depending on its LID. An LID is a  $k$ -digit decimal number ( $ak-1\dots a0$ ). The first node in the network is called the root-node and it obtains LID  $00\dots 0$ . The 1-hop neighbors of the root node are referred as *level-1* nodes and they are assigned LIDs by flipping the first digit of the root-node's LID, i.e., their LIDs would be  $100\dots 0$  to  $900\dots 0$ . In the same way, the *level-1* nodes assign the LID to their child nodes by flipping the second leftmost digit in the LS, i.e., LIDs  $110\dots 0$  to  $990\dots 0$  would be assigned to their child nodes. These child nodes are called *level 2* of the LS. In this way, all the nodes are arranged in the LIS.

EMP supports unicast routing. To forward a packet, the node searches the list of its 1-hop neighbors to find a node whose LID shares the longest prefix to the LID of the destination node in the packet. If the node succeeds, the packet is forwarded to that 1-hop neighbor. Otherwise, the node forwards the packet to its parent node. When the parent of a node in EMP is lost or fails, the node obtains a new LID from one of the available 1-hop neighbors in the network. If its parent's LID changes, the child nodes' LID also change. EMP uses a tree-based structure and is vulnerable to network partition and extensive information loss in case of critical node failure, which would affect the network throughput and end-to-end delay. Also, it does not provide any explicit mechanism for avoiding loops and keeps only the shortest routes to its neighbors.

Jain, [Jain et al. 2011] propose a scalable DHT-based unicast routing algorithm, named virtual identifier routing paradigm (VIRO). The idea is to introduce a topology-aware structured virtual id (Vid) space to which both the UIDs as well as higher layer addresses/names of the nodes are mapped. This is an attempt to eliminate flooding at both the data and control planes. The proposed scheme consists of three major phases: LID assignment, VIRO routing, and LID lookup and forwarding.

The LID of a node can be assigned either in a centralized (top-down) or distributed (bottom-up) fashion. The LIS forms a Kademlia binary tree [Maymounkov and Mazzières 2002]. A node's LID is a  $L$ -bit identifier that is based on its distance from the root node. The LIDs are arranged in a logical tree structure with  $L$  levels for  $L$ -bit identifier. In VIRO, the leaves of the tree represent the nodes and their LIDs. The LIDs are assigned to nodes according to the following two criteria. *First*, if two nodes are close in the LS, then they would also be close in the physical topology. *Second*, there should be at least one node in a sub-tree that has a link to a node in the other sub-tree. To join the network, a node obtains its LID based on the physical neighbor's LID. For  $L$ -bit LID, each node has a routing table of  $L$  entries. Hence, the routing table size is  $O(\log n)$ , where  $n$  is the number of nodes in the network.

VIRO proactively builds its routing tables by discovering nodes at each level. It avoids loops by selecting a gateway at each level. It handles node failures using a withdraw update mechanism in which a node adjacent to the failed node, say the gateway, notifies the appropriate rendezvous point(s) by withdrawing its previously published connectivity information. Upon receiving the withdraw notification, the rendezvous point notifies all nodes in the affected sub-tree about the gateway failure and suggests an alternative gateway. If the rendezvous node fails, a neighboring node would take over and serves as a new rendezvous node.

The VIRO protocol is designed to work in static networks or networks with low mobility of only end nodes. The tree-based LIS in VIRO may suffer from extensive information loss and network partitioning in case of critical node failure, leaving a set of nodes disconnected. This protocol does not address the network partitioning and merging problem, which may make the network vulnerable to node churns and critical node failures, thus affecting network throughput and end-to-end delay.

[Abid et al. 2013; Abid et al. 2014b] propose a scalable DHT-based routing protocol, named 3D routing protocol (3D-RP), which is primarily designed to address the mismatch problem. The basic idea is that each node envisions its neighbors in a 3D rectangular coordinate system, i.e, local 3D-LIS consisting of three planes that divide the space into six dimensions and eight octants. Each node acts as the origin of its local 3D-LIS. In local-3D LIS of a node, each neighbor obtains its LID that reflects its relationship with other neighbors. The basic motivation behind using 3D-LIS and decision

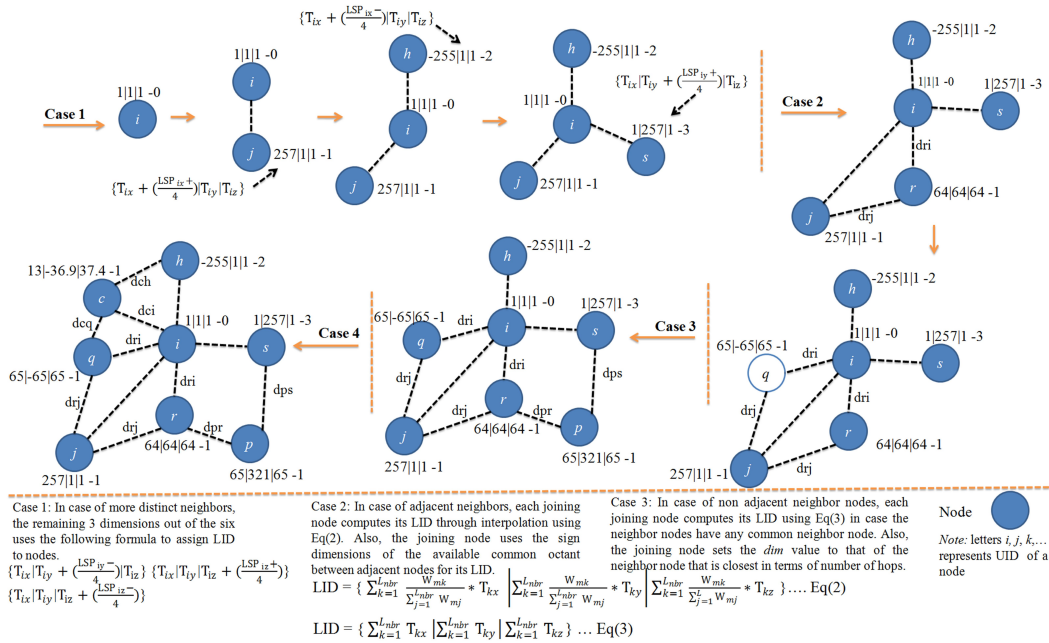


Fig. 15. The node-joining process. Black dashed lines are the physical links between neighbor nodes in the physical network. In Eq(2)  $m$  is the newly joining node;  $L_{nbr} \geq 2$  are 1-hop neighbor logical neighbors of  $m$ ;  $W_{mk}$  and  $W_{mj}$  are the weights assigned by  $m$  to its logical neighbor nodes using inverse distance function;  $T_{kx}$ ,  $T_{ky}$ , and  $T_{kz}$  are the corresponding tuples in  $x$ ,  $y$ , and  $z$  dimensions of logical neighbor's LID. In Eq(3),  $L_{nbr}$  is the number of logical neighbors

choices in 3D-RP is to logically interpret the physical intra-neighbor relationship of a node. To achieve this goal, each node in 3D-RP computes a LID in the form of an ordered three tuple  $\{x|y|z\}$ , where each tuple is an  $M$ -bit identifier calculated from a pre-determined 3D-LS. The 3D-LS ranges from 1 to  $\pm 2^M$  for each axis, i.e.,  $x$ ,  $y$ , and  $z$ . The protocol uses 1-hop hello messages to maintain the 3D-LIS, i.e., it relies on local information. Each node periodically transmits a hello message that contains the LID, UID, LSP, and its logical 1-hop neighbor information corresponding to its local 3D-LIS. In addition to the LID at each node, a dimension parameter (*dim*) is maintained to group nodes with respect to different dimensions, which is helpful while routing packets. In 3D-RP, each node computes the distance between itself and its neighbor nodes using

Received Signal Strength (RSS) method. Weights are assigned to each link, providing connectivity to its neighbors, on the basis of their distances. A node uses these weights to calculate its relative position with respect to its neighbors by using interpolation method.

If a joining node  $j$  has node  $i$  as its only neighbor,  $j$  checks node  $i$ 's neighbors information received in the hello message. If node  $i$  does not have any neighbor except  $j$ , node  $j$  calculates its LID using the first available dimension of node  $i$  as shown in Case 1 of Fig. 15. Similarly, nodes  $h$  and  $s$  calculate their LIDs in two different dimensions of  $i$ . Nodes  $j$ ,  $h$ , and  $s$  have LIDs corresponding to different dimensions of node  $i$  because these nodes are not physically connected. The joining nodes  $r$  and  $q$  compute their LIDs in Case 2 of Fig. 15 using interpolation (Eq2) after checking the adjacency with their existing neighbor nodes  $i$  and  $j$ . Similarly, node  $p$  computes its LID with respect to its non-adjacent neighbor nodes  $r$  and  $s$  using Eq3 as shown in Case 3 of Fig. 15. In Case 4, the joining node  $c$  computes its LID after checking the contiguity of its neighbor nodes  $q$ ,  $i$ , and  $h$ .

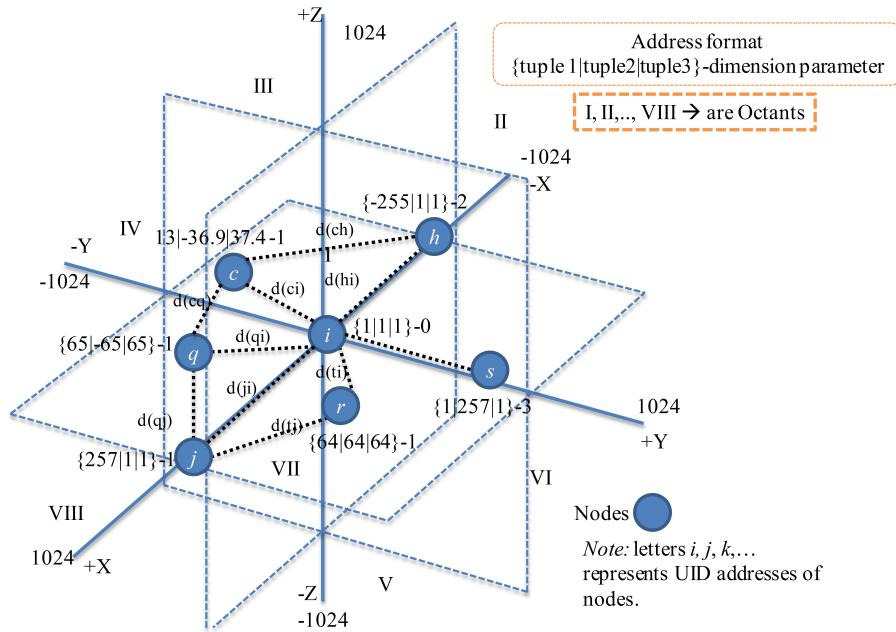


Fig. 16. A logical view of the physical arrangement of neighbor nodes in the local 3D-LIS of node  $i$  maintained by the 3D-RP

Fig. 16 illustrates the local view of the 3D-LIS of node  $i$  and its neighbor nodes in the logical network that is built according to the joining process of 3D-RP. The black dashed lines are the physical links between the nodes. The blue dashed lines are the three planes of the local 3D-LIS of node  $i$ .

Fig. 16 describes the logical mapping of the physical relationships of node  $i$  with its 1-hop neighbor nodes shown in Fig. 15. This relationship is expressed in terms of LIDs and logical dimensions of nodes in  $i$ 's 3D-LIS that allows the nodes to calculate their LIDs such that the physically close nodes have close LIDs. It can be analyzed from Fig. 15 and Fig. 16 that the neighbors of node  $i$  in the logical network are adjacent in the physical network and  $i$  is logically close to all its physically adjacent neighbor nodes. This helps to avoid long routes and redundant traffic overhead, and decrease

the end-to-end delay caused by the mismatch problem. To route a message with destination LID  $\{x|y|z\}$ - $dim$ , each node uses information about its 1-hop logical neighbors ( $L_{nbr}$ ) and forwards the query to one of its  $L_{nbr}$  that has same  $dim$  to that of  $\{x|y|z\}$ - $dim$  and offers the closest position in every tuple of its LID with respect to  $\{x|y|z\}$ - $dim$ , i.e., with least sum of difference (LSD) to the  $\{x|y|z\}$ - $dim$ . If such neighbor does not exist, the node simply forwards the message to its base node. The term 'base node' refers to nodes that are involved in the computation of new node's LID. The assignment of  $dim$  value to a joining node depends on  $dim$  values of its base node(s). This protocol does not address the network partitioning and merging problem, which may make the network vulnerable to node churns and critical node failures, thus affecting network throughput and end-to-end delay. Moreover, 3D-RP is designed for networks with low mobility.

*2.2.7. Summary of Discussion.* We end Section 2 with a summary of the main design issues discussed above: mismatch problem, selection of LS structure, address space utilization, the handling of network partitioning and merging, and the replication strategy.

The mismatch problem is important to consider in the deployment of DHT at the network layer in MANETs because it affects performance in terms of path-stretch and end-to-end delay. All protocols discussed above suffer from this problem. Some attempt to resolve the issue by maintaining the physical neighbors of a node in addition to its logical neighbors, but it results in a high path-stretch penalty and larger end-to-end delay.

VRR, ATR, DART, DHT-OLSR, VCP, AIR, KDSR, L+, Tribe, DFH, and EMP suffer from the mismatch problem discussed in Case 1 and Case 2. The address assignment mechanism of these protocols does not ensure contiguity among the neighbor's identifier space portions (LSPs) nor the adjacency of neighbors in LIS and PT. This makes them vulnerable to high path-stretch penalty, which in turn, produces larger end-to-end delay. DFH and VCP try to address the problem in Case 1, described in Section 2.2.4, by maintaining information about both the physical and logical neighbors. Unfortunately, this also leads to the mismatch problem in Case 2 described in Section 2.2.4.

3D-RP tries to address the mismatch problem by considering the intra-neighbor relationships of nodes when computing LID of a joining node. The solution reduces the impact of the mismatch problem, but does not avoid it completely. When a new node, for instance  $p$ , comes in contact with two non-adjacent neighbors (say,  $p1$ ,  $p2$ ) with different  $dim$  (dimensions) values and there is no common neighbor, then the new node  $p$  would obtain an LID using available dimensions of either  $p1$  or  $p2$ , depending on which one is closer in terms of distance. So, in this case, the LID of the new node  $p$  would only show its relative position in the 3D-LIS with respect to that neighbor from which it obtains its LID. This can cause a slight mismatch problem in 3D-RP.

We have carefully analyzed the mismatch problem and proffer that an optimal solution to the mismatch problem would only be possible if the physical proximity of nodes is interpreted exactly into the LIS and all physically close nodes are assigned LIDs that reflect their proximity. The solution in 3D-RP is towards in the right direction, but still much improvement is required.

The second issue identified is the shape of the LS structure, which plays a vital role in avoiding a high path-stretch penalty caused by the mismatch problem and in maintaining multiple routes to the destination. Several protocols discussed in Section 2.2.5 and Section 2.2.6 exploit different structures to arrange nodes according to their LIDs. Routing paths in tree-based, cord-based and ring-based structures are constrained by the connection order of the nodes that result in low flexibility when selecting a route

towards a destination. Also, these structures are not flexible in fulfilling the conditions to avoid the mismatch between LIS and PT, which in turn, leads to high path-stretch penalty. This problem is aggravated in case of node/link failure.

The LIS in DART, AIR, L+, Tribe, and EMP maintains only one path between any two nodes, which may degrade performance in terms of path length, traffic concentration, and resilience to failures. The LIS in VCP and VRR can only interpret the relationship of a node with up to two adjacent physical neighbors. These structures are inflexible when interpreting the physical relationship of a node in the LIS if the node has more than two physical neighbors.

DFH takes a different approach by using a hypercube to provide greater flexibility in route selection in order to enhance the resilience towards node failure and node mobility. The hypercube structure partially overcomes the mismatch problem by assigning multiple coordinates to a node in order to provide better adjacency with its physical neighbors. The drawback of this approach is that in a dense network, the number of connections per node could be high and may lead to Case 1 (see Section 2.2.4) because the hypercube dimensions are fixed and must be defined at startup time. Also, maintaining information about physical neighbors by using secondary LIDs might lead to Case 2 described in Section 2.2.4. Another drawback is the addressing and location services of a hypercube are more complex compared to a tree-based, cord-based, and ring-based structure.

3D-RP uses a 3D structure that also provides greater flexibility in route selection. Here, dimensions are not utilized as they are used in DFH. Six dimensions do not mean that a node can only accommodate six neighbors, but a node can accommodate up to six neighbors that are not in the transmission range of each other. But, more flexible structures along with the notion of intra-neighbor relationship and interpolation can be utilized to provide more flexibility in handling the mismatch problem.

The third important issue to consider is the address space utilization when allocating addresses and distributing LS among nodes in MANETs. The aim of distributing LS evenly or assigning LSPs in equal capacity to nodes is so that each node has an equal opportunity to store any information about other nodes or data, resulting in a balanced load. The benefit of maintaining such a structure is that minimal information has to be transferred in case a node leaves the network. The amount of information transferred may directly affect information loss and traffic overhead at both the control and data planes. Almost all protocols discussed in Section 2.2.5 and Section 2.2.6 have partially succeeded in their attempt to distribute the LS space evenly among nodes because they are either constrained by the LS structure or the addressing strategy adopted to allocate the addresses to nodes.

In VCP, the creation of virtual nodes hampers equal distribution of LS among all nodes, which may lead to extensive information loss and high traffic overhead in case of node failure. Also, in PROSE and AIR, the dual LID assignment when a root node moves or fails is similar to that of Tribe. The solution given in PROSE and AIR is not optimal as it may cause uneven utilization of LS and increase the number of nodes with dual LIDs. If there is no suitable node to hold the root/parent node's LID, it may result in the reassignment of LIDs for the whole sub-tree. Similarly, the LS structure in ATR, DART, KDSR, L+, EMP, VIRO and DHT-OLSR hampers equal distribution of LS among all nodes. DFH's approach of maintaining physical neighbors information by using secondary LIDs leads to an uneven distribution of LS among all nodes, hence, the possibility of extensive information loss in case a critical node fails. 3D-RP assigns more LSP to corner nodes so that they can accommodate new nodes in the future as shown in Case 1 of Fig. 15, but this makes these nodes critical. Still, much research is required to address the issue of load balancing in DHT-based routing protocols.

The network partitioning and merging issue is an open challenge. It is caused by either limited transmission range of nodes or node mobility. The logical partitioning and merging highly depends on the flexibility of the LS structure. In ATR, DART, AIR, L+, Tribe and EMP, paths are constrained by their tree-based logical identifier structure that allows only one path between any two nodes. This may result in the partitioning of a sub-tree when a parent fails - this is unlike the greater flexibility offered by the multi-dimensional approaches. Compared to tree-based routings, though VRR and VCP have partially addressed the network partitioning and merging of a network, these protocols do not provide a comprehensive and viable solution. None of the protocols discussed in Section 2.2.5 and Section 2.2.6 have addressed the network partitioning and merging that may result in extensive information loss and communication disruption between two disconnected physical topologies.

Lastly, effective replication/replica management strategy plays a key role in case an anchor node moves/fails or a network gets partitioned. Most of the existing protocols discussed in Section 2.2.5 and Section 2.2.6 do not discuss the replication/replica management. VCP uses its successor and predecessor nodes to replicate information. Similarly, 3D-RP introduces a concept of secondary anchor node that stores the replica of the mapping information that is stored at the primary anchor node. These replication strategies would not be effective in case of network partitioning and merging. A more sophisticated replication/replica management strategy is required to avoid extensive information loss and communication disruption when network partitioning occurs.

In a nutshell, in this section we discuss in detail the basic concepts related to DHT and pinpoint the key challenges and requirements as a guideline for researchers who intend to design a DHT-based routing protocol. Furthermore, we classify DHT-based routing into two major categories, namely DHT-based overlay deployment approaches and DHT-paradigms for large scale routing in MANETs, followed by an explanation of the criteria that distinguish them. We also classify the DHT paradigms for large scale routing in MANETs into three categories and elaborate on the routing protocols related to these. Lastly, the features of the discussed protocols are summarized in Fig. 17, where each protocol is analyzed against important metrics that would be helpful to people working in this area.



Metric	Protocols	L+	Tribe	DFH	VRR	DART	ATR	VCP	EMP	AIR	KDSR	VIRO	3D-RP
Routing Philosophy (reactive, proactive, on demand)		Proactive	Proactive	Proactive/Reactive	Proactive	Proactive	Proactive	Proactive	Proactive	Proactive	Reactive	Proactive	Proactive
Routing Metric		Beside Shortest path keeps other routes	Shortest path	Keeps all possible routes	Shortest Path	Shortest path	Keeps all possible routes	Shortest path	Shortest path	Shortest path based prefix label matching	Shortest path	Shortest path	Beside Shortest path keeps other routes
Scalable		Partially	Yes	Partially	Partially	Yes	Partially	Yes	Partially	Partially	No	Yes	Yes
Routing Table size for n number of nodes in the network		$O(k)$ , where k is the number of immediate neighbors of a node	$O(d+r)$ , where d is the dimensions of the hypercube and s is number of non-adjacent nodes	$O(rp)$ , where r is the number of virtual paths and p is the average path length	$O(\log n)$	$O(n)$	$O(k)$ , where k is the number of one hop (logical + physical) neighbors.	$O(k)$ , where k is the number of one-hop logical neighbors.	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(k)$ , where k is the number of one hop logical neighbors
Control Overhead		Medium	Medium	Medium	High	High	High	Medium	Medium	Medium	Medium	Medium	Medium
Routing Stretch/Path- Stretch penalty'		Medium	High	Medium	High	High	Medium	High	High	Medium	Medium	High	Low
Logical Structure		Tree-like	Tree-like	Hyper-cube	Ring	Tree	Tree	Cord	Tree	Tree	XOR based Tree	Tree	3D
DHT based		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Routing-Table Information scope <sup>3</sup>		$2r_i$ No of hops where value of r is 2 for $i=0$ .	One-hop Neighbors	One-hop neighbors	$(r/2)$ neighbors clockwise and anti-clockwise in a ring, where r is the virtual neighbor set) + one hop physical neighbors	One Hop Neighbors	Multi-hop neighbors	One-hop logical and physical neighbors	One hop neighbors	Two-hop logical neighbor information	Nodes of XOR distance b/w $2^i$ and $2^{i+1}$ $0 \leq i \leq 160$	One-hop neighbors	One-hop neighbors + One hop neighbors of a neighbor
Mobility Support		Low	Low	Yes	Low	Yes	Yes	Yes	Yes	Yes	Low	No (only for host nodes)	Low

Metric	Protocols	L+	Tribe	DFH	VRR	DART	ATR	VCP	EMP	AIR	KDSR	VIRO	3D-RP
Network Merging detection		No	No	No	Partially	Partially	No	Partially	No	No	No	No	No
Considering Physical Topology/ New or Extension	No/Extension of Landmark Routing System [54]	Yes/New	Yes/ New	Yes/ New	No/New	Yes/ New	Yes/ Extension of DART[21]	Yes/New	No/ Extension of MP [68]	Yes/ New	Yes/ Extension of DSR [6] with Kademia[72]	Yes/New	Yes/New
Routing forwarding based on	Logical neighbors	Logical neighbors	Logical+ physical neighbors	Logical+ physical neighbors	Logical+ physical neighbors	Logical neighbors	Logical neighbors	Logical+ physical neighbors	Logical neighbors	Logical neighbors	Logical neighbors	Logical+ Physical neighbors	Logical neighbors
Routing Complexity for n number of nodes	O (log n)	O(log n)	N/A	O(log n)	O(log n)	O(log,n)	N/A	O(log n)	N/A	O(log n)	O(log n)	O(log n)	O(log n)
Support(Unicast/ Multicast)	Unicast	Unicast	Unicast	Unicast	Unicast	Unicast	Unicast	Unicast	Unicast	Unicast and Multicast	Unicast	Unicast	Unicast
Loop Avoidance Addressed(Yes/No)	No	No	N/A	Yes	Yes	Yes	Yes	Yes	No	N/A	N/A	Yes	Yes
Multipath Support(Yes/No)	Yes	No	Yes	Yes	Yes	No	Yes	Yes	No	Yes	Yes	Yes	Yes
Flood Control (Yes/No)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Overall Complexity <sup>2</sup> (High/Medium/Low)	High	Medium	High	High	High	Medium	High	Medium	Medium	Medium	Medium	Medium	Medium
Single point failure	Yes	Yes	Yes	Yes	No	Yes	Yes	No	Yes	Yes	Yes	Yes	No
Duplicate logical ID existence	N/A	No	N/A	No	No	Yes	Yes	No	No	Yes	N/A	N/A	No

**1. Routing Stretch (RS) or Path-stretch penalty:** The ratio between the length of the path traversed by a routing algorithm and the shortest path available in the network [Awad, Sommer, German and Dressler 2011, Awad, Sommer, German and Dressler 2008]. We consider the routing stretch in the worst case. Scale: It is low if  $1 < RS \leq 1.25$ . It is medium if  $1.25 < RS \leq 1.5$ . It is high if  $RS > 1.5$ .

**2. Overall Complexity:** We consider the overall complexity in terms of routing information dissemination, route updating, routing table size, and routing overhead in case of link failure.

**3. Routing Table Information Scope:** It refers to the extent of information a node keeps about its neighbors in its routing table. The scope could be 1-hop, 2-hop, or according to the protocol specification. A node uses this information to select the next hop in order to forward the packet towards the destination.

Fig. 17. Summarized features of DHT-based protocols for scalable routing in MANETs

### 3. FUTURE TRENDS AND DHT-BASED ROUTING

Here, we discuss some of the emerging fields of research and the applicability of DHT-based lookups and routings in these fields.

#### 3.1. CONTENT CENTRIC NETWORKING (CCN)

Recently, a content centric networking (CCN) paradigm, which is promising not only for the Internet but also for MANETs, has emerged as a hot research topic. CCN is based on named data rather than host identifiers (UID) for routing [Jacobson et al. 2009; Liu et al. 2012; Oh et al. 2010]. It is capable of accessing and retrieving content by name. It decouples content from its producer/source/owner at the network layer. CCN is effective for disruption tolerant networks and avoids dependency on end-to-end connectivity. However, it might suffer from scalability and efficiency challenges in global deployments [Liu et al. 2012]. In such scenarios, a DHT structure may be used to achieve scalability in CCN for both Internet and MANET because DHT provides not only location-independent identity, but it also provides a scalable substrate to manage contents and distribute information in the network.

#### 3.2. DEVICE-TO-DEVICE (D2D) COMMUNICATION

D2D communication is a technology component that allows transmitting data signals between user equipment over a direct link using the cellular resources, thus bypassing the base station (BS) [Doppler et al. 2009; Lin et al. 2013; Lingyang Song 2014]. D2D introduces new opportunities for proximity based commercial services, particularly social network applications for LTE-A. D2D users communicate directly while remaining users are controlled under the BS. Spectrum sharing between D2D users and BS controlled users is one of the key challenges. D2D is classified into: i) in-band in which D2D uses the cellular frequency band; ii) out-band in which D2D uses the other frequency band, like 2.4 GHz ISM band [Lin et al. 2013]. The in-band is further classified into: i) overlay D2D in which both D2D and cellular transmitters use a statistically unrelated frequency band; ii) underlay D2D in which both cellular and D2D transmitter access the frequency band in an opportunistic manner [Lin et al. 2013]. Communication in D2D underlay can be in a single-hop or multi-hops, depending upon the location of the destination and transmission power of the source device. DHT-based routings can be applied to multi-hop D2D communications, thus result in less routing overhead and more network scalability and longevity.

#### 3.3. INTEGRATED MANET AND INTERNET

In recent years, cellular networks are used not only for voice communications, but also for data communication (Internet access). A mobile user needs data communication mainly for content-sharing, emails, staying connected to social-networks (like, Facebook etc), etc. Fourth-generation (4G) wireless system connects mobile users to the Internet through heterogeneous connecting technologies (e.g. Cellular, wireless LAN, mobile ad hoc network (MANETs) etc.) [Al Shidhani and Leung 2010; Cavalcanti et al. 2005; Ding 2008] which introduces several challenges in order to integrate these heterogeneous networks [Ding 2008]. One can find several advantages of the integrated MANETs and Internet. *First*, it would extend the coverage of infrastructure based wireless networks (e.g. Cellular network). *Second*, a mobile user in the MANET can access the Internet via another user connected to the Internet [King 2011; Online 2011]. *Third*, it can avoid the dead-zone. Supporting a large MANET integrated into Internet requires the underlying routing protocol for MANET to be scalable. The existing traditional routing protocols for MANETs are not scalable because these protocols are based on flooding mechanism [Caleffi and Paura 2011; Eriksson et al. 2007; Garcia-

Luna-Aceves and Sampath 2009]. Therefore, deploying a DHT-based routing protocols would make MANETs more scalable [Eriksson et al. 2007; Jain et al. 2011], which in turn would allow a larger disconnected community to be connected with Internet [King 2011, Reporter 2011].

### 3.4. INTERNET OF THINGS (IOT)

IoT refers to a smart world of identifiable objects, such as devices, sensors, actuators, and mobile phones with ubiquitous computing and networking and cooperating with their neighboring objects to provide value added services [Atzori et al. 2010; Chilamkurti et al. 2013]. Scalability in IoT is one of the core issues of concern. Scalable identification, naming, name resolution, and addressing space and structure—due to the sheer size of the resulting system and scalable data communication and networking—due to the high level of interconnection among a large number of objects are a few major concerns related to scalability in IoT [Chaouchi et al. 2013; Miorandi et al. 2012]. The analysis and design of IoT cannot overlook aspects related to networking technologies such as routing protocols, flow control robustness, and synchronization. The distributed implementation of routing protocols is a key issue for any networked systems and for IoT in particular [Chaouchi et al. 2013]. DHT-based lookup and routing technologies can be adopted for proximity communications whenever possible in case of large deployments in IOT.

### 3.5. MACHINE-TO-MACHINE (M2M) COMMUNICATIONS

M2M communication refers to data communication between autonomous machines without human intervention [Antón-Haro et al. 2013]. These machines could be smart sensors, mobile devices, or computers that can communicate autonomously using different network technologies, like Zigbee, Bluetooth, and Wi-Fi to wide area network such as wired. IoT concepts can be seen as a superset of functionalities necessary to the design of M2M as IoT involves different other technologies such as nanotechnology, robotics, artificial intelligence, etc. [Bourgeau et al. 2013]. M2M traffic raises a wide range of requirements on mobility, latency, reliability, security, and power consumption. Extensive communication overhead depletes energy resources of machines. This can be reduced by carefully applying algorithmic and distributed computing techniques to design efficient communication protocols, e.g., routing protocols [Chang et al. 2011; Lu et al. 2011]. DHT-based lookup and routing technologies can be adopted for energy efficient communications in case of an increase in data volumes and number of connections due to large deployments in M2M.

## 4. CONCLUSIONS

One of the basic design issues in implementing large scale MANETs is scalability, which is heavily influenced by the routing protocol. Instead of modifying or optimizing the traditional routing protocols for MANETs, the DHT or DHT-like technologies can be used for routing in MANETs. Maintaining a DHT-based structure for a highly dynamic MANET environment has introduced a number of new research issues. This paper highlights some major challenges that are raised by direct adoption of DHT-based or DHT-like strategies for implementing the logical identifier space at the network layer.

We classify the existing DHT-based protocols into three major categories: DHT for location services, DHT for addressing and routing, and DHT for addressing, routing and location services. In the *first* category, the DHT-based location service is coupled with a geographic addressing space defined by some positioning system. The protocol defines addressing and routing by utilizing the geographic addressing space, while the distribution of the node location information is based on DHT.

In the *second* category, the protocol deploys a DHT-based structure that is used only for addressing and routing. Nodes have fixed LIDs throughout the network lifetime and routing is performed based on the LIDs. This category does not use DHT-based location service. In the *third* category, a DHT-based structure is used for location services in addition to addressing and routing. Contrary to the *first* and *second* categories, the location services, routing, and addressing are dependent on each other and any changes in one aspect would influence the others.

We review the existing approaches related to DHT-based routing paradigm for MANETs by comparing the performance of different protocols against various parameters. We then identify the shortcomings of these protocols in the light of critical challenges discussed in Section 2.2.4. The requirements summarized in Section 2.2.4 are vital to the optimal design of a scalable DHT-based routing protocol in MANETs. By carefully analyzing the addressing schemes and LIS structures offered by different DHT-based protocols, we conclude that there are two major correlated issues that require immediate attention, namely the mismatch problem and the selection of the LS structure, which directly or indirectly cause immense overhead, unequal LS utilization, and network partitioning. An optimal solution to the mismatch problem would only be possible if: i) the physical relationship of nodes is mapped exactly into the LIS; ii) a node takes into account the physical intra-neighbor relationship before computing its LID; and iii) all physically close nodes are assigned LIDs that reflect their physical proximity. This can only be accomplished if the LIS is flexible.

Merging of partitioned network is a potential research issue, especially, in DHT based large scale routing protocols for MANETs. None of the existing literature in Section 2.2.5 and Section 2.2.6 has addressed the issues related to merging detection and merging of two distinct logical identifier structures over MANETs. Information loss and overhead at both the control and data planes in case of network partitioning can be reduced by an effective replication/replica management strategy. None of the existing literature in Section 2.2.5 and Section 2.2.6 has discussed replication/replica management in context of network partitioning and merging.

We believe that a better understanding of the current approaches paves the basis for designing a new DHT-based routing protocol that can satisfy all the requirements presented by this survey. We intend to evaluate the performance of these protocols by implementing them in a real world application scenario.

## REFERENCES

- S.A. Abid, M. Othman, and N. Shah. 2013. Exploiting 3D Structure for Scalable Routing in MANETs. *IEEE Communications Letters* 17, 11 (November 2013), 2056–2059. DOI: <http://dx.doi.org/10.1109/LCOMM.2013.091113.131256>
- S.A. Abid, Mazliza Othman, and Nadir Shah. 2014a. 3D P2P overlay over MANETs. *Computer Networks* 64, 0 (2014), 89 – 111. DOI: <http://dx.doi.org/10.1016/j.comnet.2014.02.006>
- S.A. Abid, Mazliza Othman, Nadir Shah, Mazhar Ali, and A.R. Khan. 2014b. 3D-RP: A DHT-Based Routing Protocol for MANETs. *Comput. J.* (2014). DOI: <http://dx.doi.org/10.1093/comjnl/bxu004>
- Mehran Abolhasan, Tadeusz Wysocki, and Eryk Dutkiewicz. 2004. A review of routing protocols for mobile ad hoc networks. *Ad hoc networks* 2, 1 (2004), 1–22.
- A. Al Shidhani and V. C M Leung. 2010. Secure and Efficient Multi-Hop Mobile IP Registration Scheme for MANET-Internet Integrated Architecture. In *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*. 1–6. DOI: <http://dx.doi.org/10.1109/WCNC.2010.5506193>
- José I Alvarez-Hamelin, Aline C Viana, and Marcelo D de Amorim. 2006. DHT-based functionalities using hypercubes. In *Ad-Hoc Networking*. Springer, 157–176.
- D.K. Anand and S. Prakash. 2010. A Short Survey of Energy-Efficient Routing Protocols for Mobile Ad-Hoc Networks. In *Advances in Recent Technologies in Communication and Computing (ARTCom), 2010 International Conference on*. 327–329. DOI: <http://dx.doi.org/10.1109/ARTCom.2010.103>

- Carles Antón-Haro, Thierry Lestable, Yonghua Lin, Navid Nikaein, Thomas Watteyne, and Jesus Alonso-Zarate. 2013. Machine-to-machine: an emerging communication paradigm. *Transactions on Emerging Telecommunications Technologies* 24, 4 (2013), 353–354.
- Luigi Atzori, Antonio Iera, and Giacomo Morabito. 2010. The internet of things: A survey. *Computer Networks* 54, 15 (2010), 2787–2805.
- A. Awad, R. German, and F. Dressler. 2011. Exploiting Virtual Coordinates for Improved Routing Performance in Sensor Networks. *Mobile Computing, IEEE Transactions on* 10, 9 (Sept 2011), 1214–1226. DOI: <http://dx.doi.org/10.1109/TMC.2010.218>
- A. Awad, C. Sommer, R. German, and F. Dressler. 2008. Virtual Cord Protocol (VCP): A flexible DHT-like routing service for sensor networks. In *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on*. 133–142. DOI: <http://dx.doi.org/10.1109/MAHSS.2008.4660079>
- E. Baccelli and J. Schiller. 2008. Towards scalable MANETs. In *ITS Telecommunications, 2008. ITST 2008. 8th International Conference on*. 133–138. DOI: <http://dx.doi.org/10.1109/ITST.2008.4740243>
- Elizabeth M Belding-Royer. 2003. Multi-level hierarchies for scalable ad hoc routing. *Wireless Networks* 9, 5 (2003), 461–478.
- L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, J-P Hubaux, and J.-Y. Le Boudec. 2001. Self organization in mobile ad hoc networks: the approach of Terminodes. *Communications Magazine, IEEE* 39, 6 (Jun 2001), 166–174. DOI: <http://dx.doi.org/10.1109/35.925685>
- Azzedine Boukerche, Begumhan Turgut, Nevin Aydin, Mohammad Z. Ahmad, Ladislau BĂŪlĂŪni, and Damla Turgut. 2011. Routing protocols in ad hoc networks: A survey. *Computer Networks* 55, 13 (2011), 3032–3080. DOI: <http://dx.doi.org/10.1016/j.comnet.2011.05.010>
- Thomas Bourgeau, Hakima Chaouchi, and Pinar Kirci. 2013. Machine-to-Machine Communications. In *Next-Generation Wireless Technologies*, Naveen Chilamkurti, Sherali Zeadally, and Hakima Chaouchi (Eds.). Springer London, 221–241. DOI: [http://dx.doi.org/10.1007/978-1-4471-5164-7\\_11](http://dx.doi.org/10.1007/978-1-4471-5164-7_11)
- Matthew Caesar, Miguel Castro, Edmund B. Nightingale, Greg O’Shea, and Antony Rowstron. 2006. Virtual Ring Routing: Network Routing Inspired by DHTs. *SIGCOMM Comput. Commun. Rev.* 36, 4 (Aug. 2006), 351–362. DOI: <http://dx.doi.org/10.1145/1151659.1159954>
- Marcello Caleffi, G. Ferraiuolo, and L. Paura. 2007. Augmented Tree-based Routing Protocol for Scalable Ad Hoc Networks. In *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE International Conference on*. 1–6. DOI: <http://dx.doi.org/10.1109/MOBHOC.2007.4428727>
- Marcello Caleffi and Luigi Paura. 2011. M-DART: multi-path dynamic address routing. *Wireless Communications and Mobile Computing* 11, 3 (2011), 392–409.
- Qing Cao and Tarek Abdelzaher. 2006. Scalable logical coordinates framework for routing in wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)* 2, 4 (2006), 557–593.
- A. Caruso, S. Chessa, S. De, and A. Urpi. 2005. GPS free coordinate assignment and routing in wireless sensor networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, Vol. 1. 150–160. DOI: <http://dx.doi.org/10.1109/INFOCOM.2005.1497887>
- MarcelC. Castro, AndreasJ. Kessler, Carla-Fabiana Chiasserini, Claudio Casetti, and Ibrahim Korpoglu. 2010. Peer-to-Peer Overlay in Mobile Ad-hoc Networks. In *Handbook of Peer-to-Peer Networking*, Xuemin Shen, Heather Yu, John Buford, and Mursalin Akon (Eds.). Springer US, 1045–1080. DOI: [http://dx.doi.org/10.1007/978-0-387-09751-0\\_37](http://dx.doi.org/10.1007/978-0-387-09751-0_37)
- D. Cavalcanti, D. Agrawal, Carlos Cordeiro, Bin Xie, and A. Kumar. 2005. Issues in integrating cellular networks WLANs, AND MANETs: a futuristic heterogeneous wireless network. *Wireless Communications, IEEE* 12, 3 (June 2005), 30–41. DOI: <http://dx.doi.org/10.1109/MWC.2005.1452852>
- Ian D Chakeres and Charles E Perkins. 2008. *Dynamic MANET on-demand routing protocol*. Technical Report. IETF Internet Draft, draft-ietf-manet-dymo-12. txt.
- Kim Chang, Anthony Soong, Mitch Tseng, and Zhixian Xiang. 2011. Global wireless machine-to-machine standardization. *IEEE Internet Computing* 15, 2 (2011), 64–69.
- Hakima Chaouchi, Thomas Bourgeau, and Pinar Kirci. 2013. Internet of Things: From Real to Virtual World. In *Next-Generation Wireless Technologies*, Naveen Chilamkurti, Sherali Zeadally, and Hakima Chaouchi (Eds.). Springer London, 161–188. DOI: [http://dx.doi.org/10.1007/978-1-4471-5164-7\\_9](http://dx.doi.org/10.1007/978-1-4471-5164-7_9)
- Benjie Chen and Robert Morris. 2002. *L: Scalable Landmark Routing and Address Lookup for Multi-hop Wireless Networks*. Technical Report. Tech. Rep. MIT-LCS-TR-837, MIT.
- Lei Chen and Wendi B Heinzelman. 2007. A survey of routing protocols that support QoS in mobile ad hoc networks. *Network, IEEE* 21, 6 (2007), 30–38.
- Naveen Chilamkurti, Sherali Zeadally, and Hakima Chaouchi. 2013. *Next-Generation Wireless Technologies: 4G and Beyond*. (2013).

- Diego N da Hora, Daniel F Macedo, Leonardo B Oliveira, Isabela G Siqueira, Antonio AF Loureiro, José M Nogueira, and Guy Pujolle. 2009. Enhancing peer-to-peer content discovery techniques over mobile ad hoc networks. *Computer Communications* 32, 13 (2009), 1445–1459.
- Saumitra M Das, Himabindu Pucha, and Y Charlie Hu. 2008. Distributed hashing for scalable multicast in wireless ad hoc networks. *Parallel and Distributed Systems, IEEE Transactions on* 19, 3 (2008), 347–362.
- Hongmei Deng, Wei Li, and Dharma P Agrawal. 2002. Routing security in wireless ad hoc networks. *Communications Magazine, IEEE* 40, 10 (2002), 70–75.
- Shuo Ding. 2008. A survey on integrating MANETs with the Internet: Challenges and designs. *Computer Communications* 31, 14 (2008), 3537–3551.
- Klaus Doppler, Mika Rinne, Carl Wijting, Casio Ribeiro, and Klaus Hugl. 2009. Device-to-device communication as an underlay to LTE-advanced networks. *Communications Magazine, IEEE* 47, 12 (2009), 42–49.
- Jakob Eriksson, Michalis Faloutsos, and Srikanth V Krishnamurthy. 2007. DART: dynamic address routing for scalable ad hoc and mesh networks. *Networking, IEEE/ACM Transactions on* 15, 1 (2007), 119–132.
- Ghofrane Fersi, Wassef Louati, and Maher Ben Jemaa. 2013. Distributed Hash table-based routing and data management in wireless sensor networks: a survey. *Wireless networks* 19, 2 (2013), 219–236.
- Hannes Frey. 2004. Scalable geographic routing algorithms for wireless ad hoc networks. *Network, IEEE* 18, 4 (2004), 18–22.
- JJ Garcia-Luna-Aceves and Dhananjay Sampath. 2009. Scalable integrated routing using prefix labels and distributed hash tables for MANETs. In *Mobile Adhoc and Sensor Systems, 2009. MASS'09. IEEE 6th International Conference on*. IEEE, 188–198.
- Jose Joaquin Garcia-Luna-Aceves and Marcelo Spohn. 1999. Source-tree routing in wireless networks. In *Network Protocols, 1999.(ICNP'99) Proceedings. Seventh International Conference on*. IEEE, 273–282.
- Mario Gerla, Christoph Lindemann, and Ant Rowstron. 2005. P2P MANET's-New Research Issues.. In *Peer-to-Peer Mobile Ad Hoc Networks*.
- Joy Ghosh, Sumesh J Philip, and Chunming Qiao. 2007. Sociological orbit aware location approximation and routing (SOLAR) in MANET. *Ad Hoc Networks* 5, 2 (2007), 189–209.
- Krishna Gummadi, Ramakrishna Gummadi, Steven Gribble, Sylvia Ratnasamy, Scott Shenker, and Ion Stoica. 2003. The impact of DHT routing geometry on resilience and proximity. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 381–394.
- Sukhvir Singh Gurmukh Singh, D. S. G. 2012. DHT based Routing Protocols for MANETs: A Survey. *International Journal of Mobile and Adhoc Network* 2, 1 (2012), 20–24.
- Jaap C Haartsen. 2000. The Bluetooth radio system. *Personal Communications, IEEE* 7, 1 (2000), 28–36.
- Lajos Hanzo and Rahim Tafazolli. 2007. A survey of QoS routing solutions for mobile ad hoc networks. *IEEE Communications Surveys & Tutorials* 9, 2 2nd Quarter (2007), 50–70.
- Y Charlie Hu, Saumitra M Das, and Himabindu Pucha. 2003. Exploiting the Synergy between Peer-to-Peer and Mobile Ad Hoc Networks.. In *HotOS*. 37–42.
- J-P Hubaux, Thomas Gross, J-Y Le Boudec, and Martin Vetterli. 2001. Toward self-organized mobile ad hoc networks: The Terminodes Project. *Communications Magazine, IEEE* 39, 1 (2001), 118–124.
- Van Jacobson, Diana K Smetters, James D Thornton, Michael F Plass, Nicholas H Briggs, and Rebecca L Braynard. 2009. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. ACM, 1–12.
- Philippe Jacquet, Paul Muhlethaler, Thomas Clausen, Anis Laouiti, Amir Qayyum, and Laurent Viennot. 2001. Optimized link state routing protocol for ad hoc networks. In *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*. IEEE, 62–68.
- Sourabh Jain, Yingying Chen, and Zhi-Li Zhang. 2011. Viro: A scalable, robust and namespace independent virtual id routing for future networks. In *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2381–2389.
- Satish Chandra Jha, Badii Jouaber, and Kazi M Ahmed. 2008. Dynamic-address-allocation based scalable routing protocol in context of node mobility. In *Wireless Hive Networks Conference, 2008. WHNC 2008. IEEE*. IEEE, 1–6.
- Mingliang Jiang. 1999. Cluster based routing protocol (CBRP). *draft-ietf-manet-cbrp-spec-01.txt* (1999).
- D Johnson, Y Hu, D Maltz, and others. 2007. The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4. *RFC4728* (2007), 2–100.
- Luo Junhai, Ye Danxia, Xue Liu, and Fan Mingyu. 2009. A survey of multicast routing protocols for mobile ad-hoc networks. *Communications Surveys & Tutorials, IEEE* 11, 1 (2009), 78–91.

- Brad Karp and H. T. Kung. 2000. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom '00)*. ACM, New York, NY, USA, 243–254. DOI: <http://dx.doi.org/10.1145/345910.345953>
- Wolfgang Kiess and Martin Mauve. 2007. A survey on real-world implementations of mobile ad-hoc networks. *Ad Hoc Networks* 5, 3 (2007), 324–339.
- Ritchie S. King. 2011. Building a Subversive Grassroots Network. (July 2011). <http://spectrum.ieee.org/telecom/internet/building-a-subversive-grassroots-network>
- Young-Bae Ko and Nitin H Vaidya. 2000. Location-Aided Routing (LAR) in mobile ad hoc networks. *Wireless Networks* 6, 4 (2000), 307–321.
- Elis Kulla, Makoto Ikeda, Leonard Barolli, Fatos Xhafa, and Jiro Iwashige. 2012. A Survey on MANET Testbeds and Mobility Models. In *Computer Science and Convergence*, James J. (Jong Hyuk) Park, Han-Chieh Chao, Mohammad S. Obaidat, and Jongsung Kim (Eds.). Lecture Notes in Electrical Engineering, Vol. 114. Springer Netherlands, 651–657. DOI: [http://dx.doi.org/10.1007/978-94-007-2792-2\\_63](http://dx.doi.org/10.1007/978-94-007-2792-2_63)
- Raphaël Kummer, Peter Kropf, and Pascal Felber. 2006. Distributed Lookup in Structured Peer-to-Peer Ad-Hoc Networks. In *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE*, Robert Meersman and Zahir Tari (Eds.). Lecture Notes in Computer Science, Vol. 4276. Springer Berlin Heidelberg, 1541–1554. DOI: [http://dx.doi.org/10.1007/11914952\\_37](http://dx.doi.org/10.1007/11914952_37)
- Sung-Ju Lee, Elizabeth M Belding-Royer, and Charles E Perkins. 2003. Scalability study of the ad hoc on-demand distance vector routing protocol. *International Journal of Network Management* 13, 2 (2003), 97–114.
- Fan Li and Yu Wang. 2007. Routing in vehicular ad hoc networks: A survey. *Vehicular Technology Magazine, IEEE* 2, 2 (2007), 12–22.
- Jinyang Li. 2001. *A scalable location service for geographic ad hoc routing*. Ph.D. Dissertation. Massachusetts Institute of Technology.
- Xingqin Lin, Jeffrey G Andrews, and Amitava Ghosh. 2013. A Comprehensive Framework for Device-to-Device Communications in Cellular Networks. *arXiv preprint arXiv:1305.4219* (2013).
- Chen Xu Lingyang Song, Zhu Han. 2014. *Resource Management for Device-to-Device Underlay Communication*. Springer New York.
- Changling Liu and Jörg Kaiser. 2003. *A survey of mobile ad hoc network routing protocols*. Universität Ulm, Fakultät für Informatik.
- Hang Liu, Xavier De Foy, and Dan Zhang. 2012. A Multi-level DHT Routing Framework with Aggregation. In *Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking (ICN '12)*. ACM, New York, NY, USA, 43–48. DOI: <http://dx.doi.org/10.1145/2342488.2342498>
- Rongxing Lu, Xu Li, Xiaohui Liang, Xuemin Shen, and Xiaodong Lin. 2011. GRS: The green, reliability, and security of emerging machine to machine communications. *Communications Magazine, IEEE* 49, 4 (2011), 28–35.
- Eng Keong Lua, Jon Crowcroft, Marcelo Pias, Ravi Sharma, and Steven Lim. 2005. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys and Tutorials* 7, 1-4 (2005), 72–93.
- Shivanajay Marwaha, Jadwiga Indulska, and Marius Portmann. 2009. Biologically Inspired Ant-Based Routing In Mobile Ad hoc Networks (MANET): A Survey. In *Ubiquitous, Autonomic and Trusted Computing, 2009. UIC-ATC'09. Symposia and Workshops on*. IEEE, 12–15.
- Martin Mauve, A Widmer, and Hannes Hartenstein. 2001. A survey on position-based routing in mobile ad hoc networks. *Network, IEEE* 15, 6 (2001), 30–39.
- Petar Maimounkov and David Mazières. 2002. Kademia: A Peer-to-Peer Information System Based on the XOR Metric. In *Peer-to-Peer Systems*, Peter Druschel, Frans Kaashoek, and Antony Rowstron (Eds.). Lecture Notes in Computer Science, Vol. 2429. Springer Berlin Heidelberg, 53–65. DOI: [http://dx.doi.org/10.1007/3-540-45748-8\\_5](http://dx.doi.org/10.1007/3-540-45748-8_5)
- Elena Meshkova, Janne Riihijärvi, Marina Petrova, and Petri Mähönen. 2008. A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks. *Computer Networks* 52, 11 (2008), 2097–2128.
- Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlamtac. 2012. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks* 10, 7 (2012), 1497–1516.
- Robert Morris, Frans Kaashoek, David Karger, Daniel Aguayo, John Bicket, Sanjit Biswas, Douglas De Couto, Jinyang Li, B Chambers, J Jannotti, and others. 2003. The grid ad hoc networking project. (2003).
- Anelise Munaretto and Mauro Fonseca. 2007. Routing and quality of service support for mobile ad hoc networks. *Computer Networks* 51, 11 (2007), 3142–3156.



- Shree Murthy and Jose Joaquin Garcia-Luna-Aceves. 1996. An efficient routing protocol for wireless networks. *Mobile Networks and Applications* 1, 2 (1996), 183–197.
- Jongkeun Na and Chong-kwon Kim. 2006. GLR: A novel geographic routing scheme for large wireless ad hoc networks. *Computer networks* 50, 17 (2006), 3434–3448.
- Soon-Young Oh, Davide Lau, and Mario Gerla. 2010. Content centric networking in tactical and emergency manets. In *Wireless Days (WD), 2010 IFIP*. IEEE, 1–5.
- Reporter Daily Mail Online. 2011. Mobile phones could soon be helping in the aftermath of disasters by becoming an ad-hoc message passing network. (2011). <http://www.dailymail.co.uk/sciencetech/article-2024585/Mobile-phones-used-disasters-pass-emergency-messages.html>
- Vincent Douglas Park and M Scott Corson. 1997. A highly adaptive distributed routing algorithm for mobile wireless networks. In *INFOCOM'97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, Vol. 3. IEEE, 1405–1413.
- Charles Perkins, E Belding-Royer, Samir Das, and others. 2003. RFC 3561-ad hoc on-demand distance vector (AODV) routing. *Internet RFCs* (2003), 1–38.
- H. Pucha, S.M. Das, and Y.C. Hu. 2004. Ekta: an efficient DHT substrate for distributed applications in mobile ad hoc networks. In *Mobile Computing Systems and Applications, 2004. WMCSA 2004. Sixth IEEE Workshop on*. 163–173. DOI: <http://dx.doi.org/10.1109/MCSA.2004.11>
- Rajmohan Rajaraman. 2002. Topology Control and Routing in Ad Hoc Networks: A Survey. *SIGACT News* 33, 2 (June 2002), 60–73. DOI: <http://dx.doi.org/10.1145/564585.564602>
- Hari Rangarajan and J.J. Garcia-Luna-Aceves. 2007. Efficient use of route requests for loop-free on-demand routing in ad hoc networks. *Computer Networks* 51, 6 (2007), 1515–1529. DOI: <http://dx.doi.org/10.1016/j.comnet.2006.08.005>
- Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. 2001. A Scalable Content-addressable Network. *SIGCOMM Comput. Commun. Rev.* 31, 4 (Aug. 2001), 161–172. DOI: <http://dx.doi.org/10.1145/964723.383072>
- Sylvia Ratnasamy, Brad Karp, Scott Shenker, Deborah Estrin, Ramesh Govindan, Li Yin, and Fang Yu. 2003. Data-centric storage in sensornets with GHT, a geographic hash table. *Mobile networks and applications* 8, 4 (2003), 427–442.
- Luke Ritchie, Hyo-Sik Yang, Andrea W Richa, and Martin Reisslein. 2006. Cluster overlay broadcast (COB): MANET routing with complexity polynomial in source-destination distance. *Mobile Computing, IEEE Transactions on* 5, 6 (2006), 653–667.
- Hartmut Ritter, Rolf Winter, and Jochen Schiller. 2004. 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, 489–497.
- David Karger Daniel Aguayo John Bicket Sanjit Biswas Douglas De Couto Jinyang Li. Robert Morris, Frans Kaashoek. 2002. The Grid Ad Hoc Networking Project. (2002). <http://pdos.csail.mit.edu/grid/>
- Antony Rowstron and Peter Druschel. 2001. Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In *Middleware 2001*, Rachid Guerraoui (Ed.). Lecture Notes in Computer Science, Vol. 2218. Springer Berlin Heidelberg, 329–350. DOI: [http://dx.doi.org/10.1007/3-540-45518-3\\_18](http://dx.doi.org/10.1007/3-540-45518-3_18)
- Mehdi Sabeur, Ghazi Al Sukkar, Badii Jouaber, Djamel Zeglache, and Hossam Affi. 2007. Mobile party: A mobility management solution for wireless mesh network. In *Wireless and Mobile Computing, Networking and Communications, 2007. WiMOB 2007. Third IEEE International Conference on*. IEEE, 45–45.
- Prince Samar, Marc R Pearlman, and Zygmunt J Haas. 2004. Independent zone routing: an adaptive hybrid routing framework for ad hoc wireless networks. *IEEE/ACM Transactions on Networking (TON)* 12, 4 (2004), 595–608.
- Dhananjay Sampath and JJ Garcia-Luna-Aceves. 2009. Prose: scalable routing in manets using prefix labels and distributed hashing. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON'09. 6th Annual IEEE Communications Society Conference on*. IEEE, 1–9.
- Nadir Shah. 2011. *Efficient Overlay Routing for Peer to Peer Network over Mobile Ad Hoc Network*. Ph.D. Dissertation. Beihang University, Beijing, China.
- N. Shah and Depei Qian. 2010a. Cross-Layer Design for Merging of Unstructured P2P Networks over MANET. In *Ubiquitous Information Technologies and Applications (CUTE), 2010 Proceedings of the 5th International Conference on*. 1–7. DOI: <http://dx.doi.org/10.1109/ICUT.2010.5677833>
- Nadir Shah and Depei Qian. 2010b. Cross-Layer Design to Merge Structured P2P Networks over MANET. In *Parallel and Distributed Systems (ICPADS), 2010 IEEE 16th International Conference on*. IEEE, 851–856.

- N. Shah and Depei Qian. 2011. An Efficient Unstructured P2P Overlay over MANET Using Underlying Proactive Routing. In *Mobile Ad-hoc and Sensor Networks (MSN), 2011 Seventh International Conference on*. 248–255. DOI : <http://dx.doi.org/10.1109/MSN.2011.15>
- Nadir Shah, Depei Qian, and Rui Wang. 2012. MANET adaptive structured P2P overlay. *Peer-to-Peer Networking and Applications* 5, 2 (2012), 143–160.
- Xuemin Shen, Heather Yu, John Buford, and Mursalin Akon. 2010. *Handbook of peer-to-peer networking*. Vol. 1. Springer Heidelberg.
- Jang-Ping Sheu, Kun-Ying Hsieh, and Ming-Lung Ding. 2009. Routing with hexagonal virtual coordinates in wireless sensor networks. *Wireless Communications and Mobile Computing* 9, 9 (2009), 1206–1219.
- Minho Shin and William A Arbaugh. 2009. Efficient Peer-to-Peer Lookup in Multi-hop Wireless Networks. *TIIS* 3, 1 (2009), 5–25.
- Laxmi Shrivastava, GS Tomar, and Sarita S Bhadauria. 2011. A survey on congestion adaptive routing protocols for mobile ad-hoc networks. *International Journal of Computer Theory and Engineering* 3, 2 (2011), 189–196.
- Raghupathy Sivakumar, Prasun Sinha, and Vaduvur Bharghavan. 1999. CEDAR: a core-extraction distributed ad hoc routing algorithm. *Selected Areas in Communications, IEEE Journal on* 17, 8 (1999), 1454–1465.
- Hasan Sözer, Metin Tekkalmaz, and Ibrahim Korpeoglu. 2009. A peer-to-peer file search and download protocol for wireless ad-hoc networks. *Computer Communications* 32, 1 (2009), 41–50.
- Statistics. 2012. The Rise of Mobile. (2012). <http://www.digitalbuzzblog.com/infographic-2012-mobile-growth-statistics/>
- Statistics. 2013. Global mobile statistics 2013. (2013). <http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats/a#subscribers>
- Ion Stoica, Robert Morris, David Karger, M Frans Kaashoek, and Hari Balakrishnan. 2001. Chord: A scalable peer-to-peer lookup service for internet applications. In *ACM SIGCOMM Computer Communication Review*, Vol. 31. ACM, 149–160.
- Ivan Stojmenovic. 1999. *Home agent based location update and destination search schemes in ad hoc wireless networks*. MIT Research Lab Technical Report TR-99-10. Computer Science, SITE, University of Ottawa, Canada.
- John Sucec and Ivan Marsic. 2002. Clustering overhead for hierarchical routing in mobile ad hoc networks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, Vol. 3. IEEE, 1698–1706.
- John Sucec and Ivan Marsic. 2004. Hierarchical routing overhead in mobile ad hoc networks. *Mobile Computing, IEEE Transactions on* 3, 1 (2004), 46–56.
- Vinh Vu Thanh, Hung Nguyen Chan, Binh Pham Viet, and Thanh Nguyen Huu. 2009. A survey of routing using dhds over wireless sensor networks. In *Proceedings of The 6th international conference on information technology and applications (ICITA 2009)*.
- P. F. Tsuchiya. 1988. The Landmark Hierarchy: A New Hierarchy for Routing in Very Large Networks. *SIGCOMM Comput. Commun. Rev.* 18, 4 (Aug. 1988), 35–42. DOI : <http://dx.doi.org/10.1145/52325.52329>
- Aline Carneiro Viana, Marcelo Dias de Amorim, Serge Fdida, and José Ferreira de Rezende. 2004. An underlay strategy for indirect routing. *Wireless Networks* 10, 6 (2004), 747–758.
- Aline Carneiro Viana, Marcelo Dias de Amorim, Serge Fdida, and José Ferreira de Rezende. 2005. Self-organization in spontaneous networks: the approach of DHT-based routing protocols. *Ad Hoc Networks* 3, 5 (2005), 589–606.
- Renato Werneck, João Setubal, and Arlindo da Conceição. 2000. (old) Finding minimum congestion spanning trees. *J. Exp. Algorithmics* 5 (2000), 11. DOI : <http://dx.doi.org/10.1145/351827.384253>
- Wi-Fi.org. Wi-Fi Technology. (????). <http://www.wi-fi.org/>
- Yuan Xue, Baochun Li, and Klara Nahrstedt. 2001. A scalable location management scheme in mobile ad-hoc networks. In *Local Computer Networks, 2001. Proceedings. LCN 2001. 26th Annual IEEE Conference on*. IEEE, 102–111.
- Hyo-Sik Yang, Luke Ritchie, Andréa W Richa, and Martin Reisslein. 2007. MANET routing with provably low complexity through constant density clustering and route request broadcast. *Wireless Personal Communications* 43, 2 (2007), 605–621.
- Jane Yang Yu and Peter Han Joo Chong. 2005. A survey of clustering schemes for mobile ad hoc networks. *IEEE Communications Surveys and Tutorials* 7, 1-4 (2005), 32–48.
- Thomas Zahn and Jochen Schiller. 2005. MADPastry: A DHT substrate for practicably sized MANETs. In *Proc. of ASWN*.

- T. Zahn and J. Schiller. 2006. DHT-based unicast for mobile ad hoc networks. In *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*. 5 pp.–183. DOI : <http://dx.doi.org/10.1109/PERCOMW.2006.42>
- Bo Zhao, Yingyou Wen, and Hong Zhao. 2009. KDSR: An Efficient DHT-based Routing Protocol for Mobile Ad Hoc Networks. In *Hybrid Intelligent Systems, 2009. HIS'09. Ninth International Conference on*, Vol. 2. IEEE, 245–249.
- Ben Y Zhao, Ling Huang, Jeremy Stribling, Sean C Rhea, Anthony D Joseph, and John D Kubiatowicz. 2004. Tapestry: A resilient global-scale overlay for service deployment. *Selected Areas in Communications, IEEE Journal on* 22, 1 (2004), 41–53.
- Yao Zhao, Yan Chen, Bo Li, and Qian Zhang. 2007. Hop ID: a virtual coordinate based routing for sparse mobile ad hoc networks. *Mobile Computing, IEEE Transactions on* 6, 9 (2007), 1075–1089.