IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 4, 2018, accepted March 5, 2018, date of publication March 12, 2018, date of current version April 23, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2814017

Distributed Partition Detection With Dynamic Replication Management in a DHT-Based MANET

SALEEM ZAHID[1], SHAHBAZ AKHTAR ABID[2], NADIR SHAH[2], SYED HUSNAIN ABBAS NAQVI[1], AND WAQAR MEHMOOD[2]

[1]Department of Computer Science and Software Engineering, International Islamic University, Islamabad 44000, Pakistan
[2]Department of Computer Science, COMSATS Institute of Information Technology, Islamabad 45550, Pakistan

Corresponding author: SALEEM ZAHID (szahid77@gmail.com)

ABSTRACT Curtailing the information loss due to network partitioning is imperative for the reliable communication in a distributed hash table (DHT)-based routing protocol for mobile ad hoc networks (MANETs). Both the limited transmission range and mobility of nodes cause recurrent network partitioning in MANETs that leads to an inaccessibility of the nodes' mapping information, loss of the LID space, and partitioning of the DHT structure. To address the above-mentioned issues, this paper proposes a protocol that employs pre-partitioning measures, such as the critical link/node detection, the replication of the nodes' mapping information, and so on, to augment the inter-node communication, but also reflects substantial improvement in the lookup delay. The proposed protocol effectively exploits the local neighbor information in a 3-D logical structure to identify/detect the critical nodes by employing a localized distributed algorithm. Simulation results show the significance of the proposed protocol in terms of end-to-end lookup delay, routing overhead, and success ratio.

INDEX TERMS Flooding-free routing, partition detection, replication, DHT.

I. INTRODUCTION

In the past few years, several DHT-based routing protocols for MANETs have been proposed to eliminate flooding in the route discovery phase, thus enhancing the network reliability and scalability [1]–[7], [10], [20], [22]. In a DHT-based routing protocol, a node is assigned a logical identifier (LID) in addition to a universal identifier (UID), i.e., MAC/IP address, based on the LIDs of its physical neighboring nodes. The LID is drawn from a predefined logical identifier space (LS). A node maintains a disjoint portion of the whole LS referred as the logical space portion (LSP). Moreover, each node keeps track of its logical neighboring nodes having LIDs close to its own LID by following a ring, a chord, or a multidimensional structure. Thus, a logical network is built up over the physical network. Forwarding of packets on both the control and data planes is carried out using nodes LIDs in the logical network.

To enable the inter-node communication, each node stores its mapping information (i.e., LID, UID pair) by sending a store mapping information (SMI) message at a certain node in a network referred as an anchor node (AN). Table 1 illustrates the definition of the terms used inn this paper. Each node determines its AN by applying a hash function over its UID that generates a hashed value, say $h(v)$. The $h(v)$ is drawn from the same LS that has been used to assign the LIDs to nodes. A node p acts as the AN for a node q if either $h(v)$ falls in the LSP of p or the node p's LID is closest to the $h(v)$. To send a data packet, a source node s retrieves the destination node q's LID from q's AN. For this purpose, the node s applies a hash function over q's UID that generates the LID of q's AN, i.e., node p. Using the generated LID, node s forwards a mapping request message (MREQ)towards the node p in order to obtain the mapping information of the node q. Upon receiving the mapping information of q in a mapping reply message (MRPY), node s then forwards the data packet towards q using the received LID of the node q. The next hop of the data packet among the 1-hop/2-hop logical neighboring nodes is determined based on which neighboring node has the LID closest to the destination node's LID.

Existing DHT-based routing protocols mainly focus in addressing the mismatch problem [7], [20], [22] and overlook a major issue of network partitioning that may completely halts the communication between a source and a destination node (see Section 3 for detail). Network partitioning is the breakdown of a connected topology into two or more disconnected partitions [8]. A node in one partition ceases to access a node in another partition. The self-organizing

TABLE 1. 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)	It is a unique ID that identifies a node in the Logical Identifier Structure (LIS) and it describes the relative position of the node in the LIS.
Logical Identifier Space (LS)	An address space from which each node gets its LID. For example, in VCP [2] the address space is [0-1], which means each node gets a LID between 0 and 1.
Logical Identifier Structure (LIS)	A structure that arranges nodes according to their LID is called Logical Identifier Structure, e.g. a cord [2] and a ring.
Logical Network (LN)	The interconnection of nodes based on their LIDs is called Logical Network.
LS Portion (LSP)	Each node in the LN has a disjoint subset of the whole LS, which termed the LS Portion of that node.
Universal Identifier (UID)	It refers to an identifier of a node that is unique and remains the same throughout the network lifetime. It could be the IP or MAC address of a node.
T_{ix}	Refers to the tuple of the LID of a node *i* in *x* axis.
T_{iy}	Refers to the tuple of the LID of a node *i* in *y* axis.
T_{iz}	Refers to the tuple of the LID of a node *i* in *z* axis.
LSP*ix*+.	LS portion of node *i* in positive *x*-axis
LSP*ix*-.	LS portion of node *i* in negative *x*-axis
LSP*iy*+.	LS portion of node *i* in positive *y*-axis
LSP*iy*-.	LS portion of node *i* in negative *y*-axis
LSP*iz*+.	LS portion of node *i* in positive *z*-axis
LSP*iz*-.	LS portion of node *i* in negative *z*-axis
L_{nbr}	Logical neighbors of a node
T_{kx} T_{ky} T_{kz}	Refers to three tuples of the node *k* in *x*, *y*, and *z* axis
W_{mk}	Weight assigned by the joining node *m* to its neighboring node *k*
W_{mj}	Weight assigned by the joining node *m* to its neighboring node *j*

nature, the limited transmission range, and the mobility of nodes cause frequent network partitioning in MANETs. For DHT-based routing protocols, the network partitioning raises several core issues, i.e., the inaccessibility of the nodes' mapping information, loss of the LID space, and partitioning of the DHT structure (see Section 3 for detail). The impact of those issues gets more severe in case of a DHT-based routing protocol because the communication among the nodes is performed using the LIDs/LSPs of the nodes rather using their UIDs. The information about the destination node's LID depends on the availability of its mapping information at the AN. To improve the performance of a DHT-based routing protocol, it is imperative to identify/detect the critical links/nodes that instigate partitioning in the physical network. Timely detection of the critical links/nodes lessens the information loss and the communication disruption in case of the network partitioning.

A critical link refers to a link whose failure/disconnection leads to network partitioning whereas the critical node refers to a node whose failure results in the network partitioning. Figure 1 elaborates both scenarios. A distributed way of identifying a critical link or a critical node is through its common *x*-hop neighboring nodes. A link $s \leftrightarrow b$ as shown in Figure 1 is said to be *x*-hop critical if and only if both the nodes *s* and *b* has disjoint neighbor set assuming that the link between the nodes *s* and *b* does not exist. For $x = 1$, the $s \leftrightarrow b$ link is 1-hop critical as the 1-hop neighbor set of the nodes *s* and *b* are disjoint. For $x = 2$, the $s \leftrightarrow b$ link is 2-hop critical as no common neighbor exists between the 2-hop neighboring nodes of *s* and *b*. The link $s \leftrightarrow b$ in Figure 1 is a globally critical link because even if the value of *x* is increased, there exists no common neighbor between the nodes *s* and *b*. Similarly, node *e* in Figure 1 is the globally critical node as its neighboring nodes can be divided into two disjoint sets of nodes.

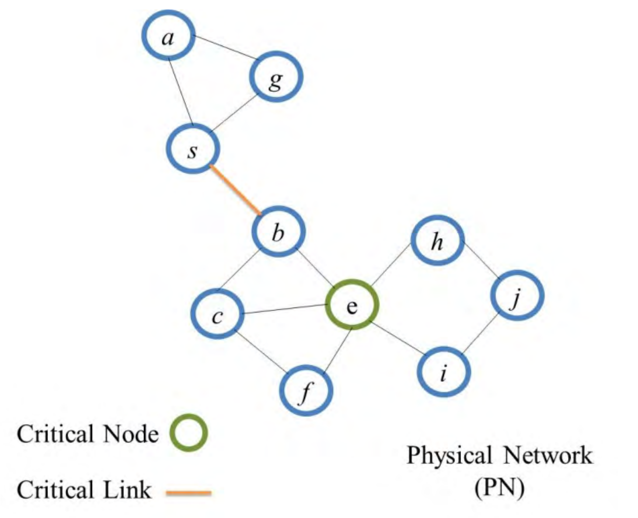


FIGURE 1. ***x*-hop critical link and critical node scenario.**

Uptil now none of the existing DHT-based routing protocols has identified/discussed/addressed the challenges related to the network partitioning. A solution to those challenges is imperative. This paper devises a novel protocol, i.e., 3D distributed partition detection and dynamic replication (3DDR), which exploits the local neighbor information of each node in a 3D-logical structure (3D-LIS) [5], [8], [10] to detect/identify the critical links in the physical network and proposes an effective replication mechanism that assists in minimizing the information loss and communication disruption. Moreover, to avoid the loss of LID space in case of the network partitioning, a LID space recovery mechanism is proposed. To the best of our knowledge, this paper is the first attempt to handle the network partitioning in context of the DHT-based routing protocols in MANETs.

The rest of this paper is organized as follows: Section II provides a review of the related work. Section III describes the problem statement. The proposed protocol is explained in Section IV. Section V discusses the simulation results and Section VI concludes the paper.

II. RELATED WORK

Several DHT-based protocols have been proposed for MANETs that mainly focus the mismatch problem between the logical and the physical topologies. To reduce the impact of the mismatch problem, various structures have been proposed to arrange nodes in the logical network [2]–[11], [17], [20], [21]. This section provides a review of the few existing DHT-based routing protocols.

Virtual Cord Protocol (VCP) [2] organizes nodes into a cord structure. Each node is given a LID in addition to its UID from a pre-defined LS that ranges [0-1]. The LID describes the relative position of the node in the cord structure. A newly joining node computes its LID based on the LIDs of its 1-hop physical neighboring nodes using various decision choices. Similarly, each node proactively maintains its logical neighboring nodes in the chord along with its physical neighbors. This makes the routing table of size $O(m)$ at each node, where m refers to the logical (predecessor/successor nodes) and the one-hop physical neighbors of a node. VCP employs greedy forwarding that uses both the logical and physical neighbors of a node to ensure the successful delivery of the data packet. VCP is not reliable against the network partitioning. In case of partitioning, the logical chord structure breaks into two disconnected cords that results into a disrupted LIS. VCP is unable to handle these problems.

MDART [6], a multi-path DHT-based routing protocol, exploits a logical tree structure for routing in MANETs. MDART proactively maintains all the existing routes via its next hop neighboring nodes to reach the destination node in the sibling tree. However, the protocol does not considers/handles the network partitioning.

The network merging and its impact on the mismatch-problem has been studied in [8]. Reference [8] advocates two key observations: i) the connecting order of the logical structure is vital to avoid the mismatch-problem; ii) the multi-dimensional structures are more resilient for the smooth merging of the disconnected networks compared to the tree, ring, cord, etc. However, the work considers only the post-partitioning measures while ignoring the pre-partitioning issues/considerations such as the LS recovery, the mapping information recovery and their impact on the network merging.

3D routing protocol (3D-RP) in [10] is primarily designed to address the mismatch problem. Each node in 3D-RP envisions its neighboring nodes in a three-dimensional coordinate system while keeping itself at the origin, forming a 3D logical identifier structure (3D-LIS) that divides the space into three planes having six dimensions and eight octants. The core idea of using the 3D-LIS is to logically interpret the physical inter/intra-neighbor relationship of nodes. Each node in the 3D-LIS computes its LID in an ordered three tuple {x|y|z} that reflects its physical proximity with respect to its neighboring nodes. Each tuple of the LID is an M-bit identifier drawn from a pre-defined three-dimensional logical identifier space (3D-LS). 3D-LS ranges from 1 to $\pm\ 2^M$ for each dimension, i.e., x, y, and z. Each node is responsible for maintaining a disjoint portion of the 3D-LS in each of its LID's x, y, and z dimensions, called logical space portion(LSP). A node stores the mapping information of other nodes with respect to its LSP [7]. Each node uses 1-hop *hello* messages to construct and maintain the 3D-LS. In addition to its LID, a node computes a dimension parameter (*dim*) that 3D-RP uses to group nodes with respect to the different dimensions. In 3D-RP, each node computes the distances from its neighboring nodes using the received signal strength (RSS). In addition, a weight is assigned to each link based on their distances providing the connectivity to its neighboring nodes. The assigned weights are used in the interpolation method to calculate the relative position of a node with respect to its neighboring nodes.

If a joining node j does not have any neighboring nodes except the node i, the node j computes its LID through Eq(1) by exploiting the first dimension of the node i as shown in Case 1 of Figure 2. T_{ix}, T_{iy}, and T_{iz} refer to the tuples of node i's LID, and $LSP_{ix}+$ is the maximum range of node i's LSP in $+x$-dimension. By using this formula, node j obtains 3/4 of node i's $LSP_{ix}+$. The purpose here is to allocate more LSP to the corner nodes so that they can accommodate new nodes in future. Similarly, the nodes h and s compute their LIDs by utilizing the next two dimensions of i. The decision choices in Case 1 is to map the physical intra-neighbor relationships among a node and its neighboring nodes in the 3D-LIS. For instance, If two neighboring nodes of a joining node are not in contact/transmission range of each other, it infers that the neighboring nodes exist physically in two different dimensions of the joining node. Each of the node j, h, and s obtains LIDs in different dimensions of i because none of those is physically connected to each other. The joining nodes r and q compute their LIDs using the interpolation Eq(2) (see Case 2 of Figure 2) after checking the adjacency with their existing neighboring nodes i and j. The decision choices in Case 2 is to handle a node's physical adjacency with its neighboring nodes and to compute a LID of the node with respect to its adjacent neighboring nodes. Similarly, node p computes its LID with respect to its non-adjacent neighboring nodes r and s using Eq(3) as shown in Case 3 of Figure 2. The decision choices in Case 3 is to handle the physical non-adjacency of the neighboring nodes and to compute a LID of the node relative to its non-adjacent neighboring nodes. In Case 4, the joining node c computes its LID after checking the contiguity of its neighboring nodes q, h, and i.

Figure 3 illustrates the local 3D-LIS of node i and its physical links with the neighboring nodes in the logical network. The black dashed lines refer to the physical links connecting the nodes. The three planes of the local 3D-LIS of i are shown by the blue dashed lines. It can be analyzed from the Figures 2 and 3 that the node i is logically close to all of its physically adjacent neighboring nodes. This reduces the redundant traffic and the long routes that effectively minimize the end-to-end delay caused by the mismatch problem. To forward a data packet towards the destination (say, LID {x|y|z}-dim), 3D-RP utilizes 1-hop logical neighboring nodes (L_{nbr}). A node selects the next hop from its L_{nbr} having the same *dim* value to that of the destination LID and offers the least sum of difference (LSD) to the destination

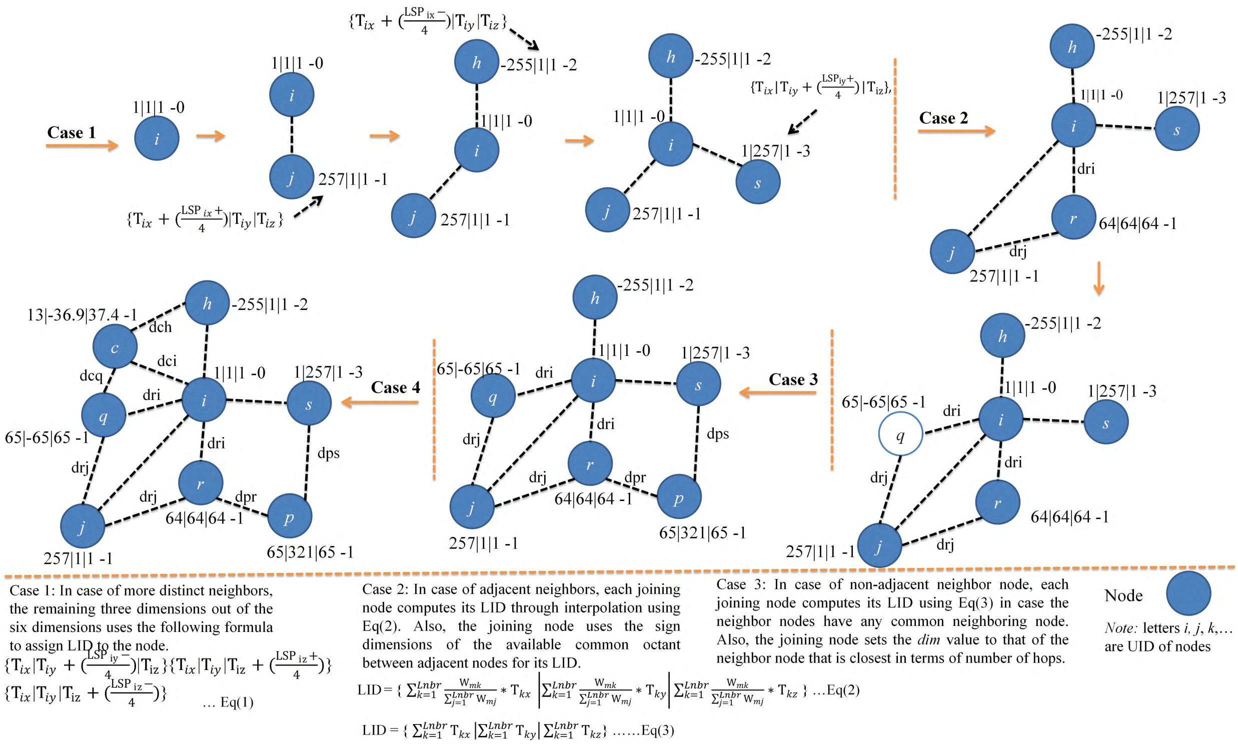


FIGURE 2. The node-joining process in 3D-RP. Black dashed lines are the physical links between neighbor nodes in the physical network. In Eq(1) $LSP_{iy}-$,$LSP_{iz}+$, and $LSP_{iz}-$ are disjoint logical space portions of node *i* in negative *y*, positive z and negative z dimensions, respectively. In Eq(2) *m* is the newly joining node; $L_{nbr} >= 2$ 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 neighbors LID. In Eq(3), L_{nbr} is the number of logical neighbors.

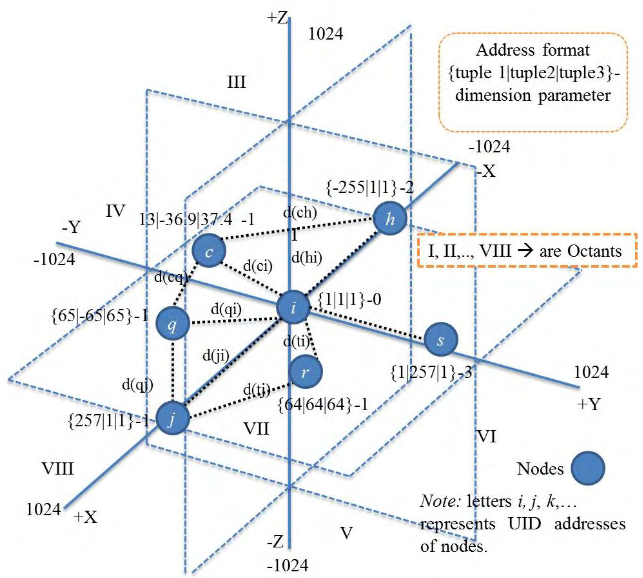


FIGURE 3. A logical view of the physical arrangement of neighbor nodes in the local 3D-LIS of node *i* maintained by the 3D-RP.

nodes LID. If a logical neighbor with the same *dim* value is unavailable, the node forwards the packet to its base node. A 'base node' refers to those neighboring nodes that are involved in the computation of the joining nodes LID. The assignment of the *dim* value to the joining node depends on the *dim* values of its base node(s). Although 3D-RP is designed to avoid the mismatch problem, however, it does not handle the network partitioning. Section 3 elaborates the limitations of 3D-RP.

Motion-Mix [11] targets the scenarios with node mobility. For data packet delivery, Motion-Mix exploits the past node mobility patterns i.e., 1-hop encounter records of the mobile nodes. Motion-Mix connects only the 1-hop nodes in the logical network and does not maintain the global logical network. Thus, Motion-Mix reduces the node joining/leaving cost as well as the lookup cost for routing the data packets. The protocol is suitable for the networks with low mobility and may not perform well in the high node mobility scenarios.

Mesh-DHT [17] attempts to reduce the ill-matching between the physical and the logical networks by using a 2-dimensional (2D) structure. Mesh-DHT builds the 2D structure of the link graph, where the physically closer neighboring nodes obtains closer 2D coordinates and the physically distant nodes obtains distant 2D coordinates. In this way, Mesh-DHT ensures that a joining node obtains its coordinates close to the coordinates of its 1-hop logical neighboring nodes and it periodically informs each of its direct neighboring nodes about its current coordinates and the coordinates of its 1-hop neighboring nodes. Thus, each node iteratively improves its 2D coordinates with respect to its 2-hop away neighboring nodes. Mesh-DHT is slightly effective against the mismatch problem, but does not discuss/handle the network partitioning.

The work in [24] contributes an energy efficient design for multiple-input-multiple-output (MIMO) transmissions in a cognitive vehicular network. The sole purpose of the work is to develop a robust energy efficient beam forming design (by assuming imperfect interference channel state information (CSI), and by satisfying the transmission power constraint and minimum transmission rate constraint) for MIMO transmissions in a cognitive vehicular network. The authors formulate the problem of energy efficiency in MIMO transmissions as an optimization problem under certain constraints and iteratively solve the problem. The proposed distributed algorithm iteratively converges to an optimal solution (i.e. optimal beam forming) that best satisfy the constraints. In other words, the proposed algorithm progressively obtains an optimal trade-off between sum of data rates and power consumption for MIMO broadcast channels in cognitive vehicular network.

Similarly, in [25], the problem of energy efficient cooperative communication in the presence of selfish nodes in ad hoc wireless networks (cooperative communication) is investigated. It is advocated that there exists an optimal energy-efficient network state (by considering a good trade-off between energy consumption and network performance) despite the presence of selfish nodes in the cooperative network. The proposed mechanism learns and exploits the cooperative behavior of nodes in the cooperative network. The work models the behavior of selfish nodes into a non-cooperative multi-player automata game and uses it in the relay selection process. The game theoretic mapping of the problem makes possible a proper and efficient distributed learning-based decision making process for relay selection, in the presence of selfish nodes. The proposed solution iteratively converges to an optimal solution (i.e. balancing the energy consumptions and average rewards). In other words, playing the game repeatedly converges the cooperative network.

DHT-based routing protocols are specifically designed to reduce the routing/control traffic overhead by removing flooding from the network, thus saves energy at each mobile node. DHT-based routing protocols are energy efficient compared to the conventional routings and can be a viable option for the resource constrained networks and various sub-domains of MANETs, like wireless sensor networks(WSNs)/wireless sensor and actuator networks(WSANs). Moreover, researchers have recently exploited the benefits associated with DHT functionalities in software defined networks (SDNs) [26], [27].

We believe that handling the network partitioning is crucial to ensure uninterrupted communication among the nodes in a DHT-based routing protocol. Pre-partitioning measures can play an important role in eluding the information loss and critical nodes/links. There exist several algorithms in the literature [12]–[16] to deal with the network partitioning problem in MANETs. However, these approaches cannot be directly applied to DHT-based routings because the problem severely interrupts/haults the communication of nodes in the DHT-based logical network and entails additional efforts to confront with such a problem. A solution to such a problem could be a distributed approach that depends merely on the localized knowledge at each node in the network and employs a mechanism for the distributed partition detection and dynamic replication of the mapping information.

III. NETWORK PARTITIONING IN A DHT-BASED PARADIGM FOR ROUTING

Network partitioning elicits two major issues for the DHT-based routing protocols, which adversely affect their performance: i) *Unavailability of an anchor node in both partitioned and un-partitioned network.*; ii) *Loss of LID space.* Another correlated issue is the longer *lookup-delay*. Reducing the lookup delay in the current setup is also a major challenge.

A. UNAVAILABILITY OF AN ANCHOR NODE IN BOTH PARTITIONED AND UN-PARTITIONED NETWORK

In a DHT-based routing protocol, the anchor node (AN) holds the mapping information of various other nodes in the network and a source requires the mapping information to communicate with a destination node. Unavailability of the destination node's AN completely disrupts the communication between the source and the destination node, i.e., the availability of the destination node's AN is imperative for the uninterrupted communication. Inaccessibility of AN occurs when a DHT-based network is partitioned into two disconnected partitions such that both the source and the destination nodes are in the same network partition, but the corresponding AN remains in a disjoint partition. In this case, despite that the source and the destination nodes are reachable, but unable to communicate due to inaccessibility of the destination node's AN.

Figure 4(a) illustrates a connected network with a critical link $i \leftrightarrow p$. The link $i \leftrightarrow p$ is globally critical because the neighboring nodes between the nodes i and p are disjoint. The destination node l stores its mapping information at its anchor node g. To communicate with node l, the source node h retrieves the mapping information of l from g. After receiving the mapping information, node h directly starts its communication with node l. For instance, the critical link $i \leftrightarrow p$ disconnects and the network partitioning occurs as shown in Figure 4(b). The source node h and the destination node l are in the same partition, but the corresponding AN, i.e., node g remains in the disjoint partition. In such a case, node h is unable to retrieve the mapping information of the node l from its AN, i.e., node g, which would disrupt the communication between the nodes h and l. Moreover, a new lookup request for the node ls mapping information would not

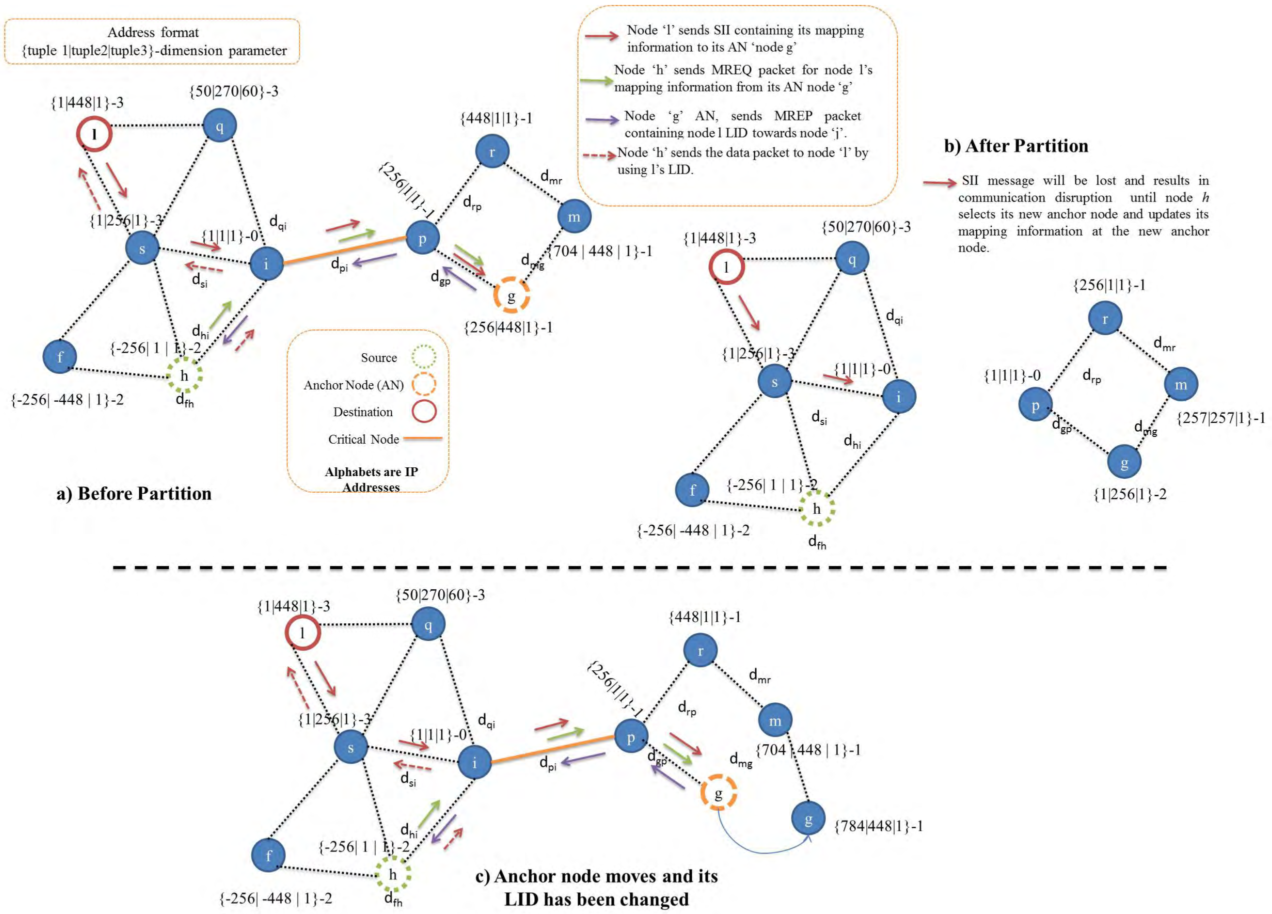


FIGURE 4. Network Partitioning in a DHT-based Routing Protocol.

be resolved until *l* selects a new AN and updates its mapping information there. This would cause the longer lookup delays and information loss.

Similarly, when an existing AN fails/move (with or without network partitioning), it is crucial/challenging to ensure the availability/recovery of the mapping information stored at that AN. In Figure 4(c), the anchor node *g* moves away from the node *p* and obtains a new LID. In this case, node *l* would select a new AN in order to store its mapping information. Meanwhile, the in-transit lookup queries might get lost and would result in communication disruption, information loss and longer lookup delays.

B. LOSS OF LID SPACE

When a network is partitioned into two disconnected networks, the LS space of one partition can be reused in the other partition. This results in an evenly distributed LS and enables the communication in the disjoint partitions. Thus, the LID space recovery in case of the network partitioning is a correlated issue in a DHT-based routing protocol for MANETs. We believe that pre-partitioning measures such as timely detection of the partitioning event prior to the actual partitioning is desirable for an efficient LS recovery. Therefore, designing a distributed partitioning detection mechanism is a potential sub-problem in this domain that is imperative to be addressed.

Moreover, another correlated issue is the longer lookup delays that exacerbates when the network gets partitioned. Reducing the lookup delay in a distributed manner using only the localized knowledge available at each nodes, is a challenging task.

To address the above problems, 3D distributed partitioning detection and replication protocol (3DDR) based on the localized neighbor information is proposed, which best suit the scalability constraints of a DHT-based routing paradigm. More specifically, following contributions covers the solution domain.

- A dynamic replication management algorithm is proposed. The algorithm ensures the availability of the mapping information in the disjoint partitions.
- A distributed partition detection algorithm is devised that plays a vital role in taking the pre-partitioning measures such as the critical nodes/links detection and lost LS recovery.
- The impact of dynamic replication management algorithm on the lookup delay is studied and finds significant gains.

3DDR considers the relative connectivity, i.e., 2-hop neighbor information and the local network variation, i.e., nodes degree to detect/identify the critical links/nodes and for replication of the mapping information. The proposed protocol ensures not only the availability of mapping information in case of the network partitioning (i.e., reduces information loss), but also significantly reduces the lookup delay without any additional control overhead.

IV. 3D DISTRIBUTED PARTITION DETECTION AND REPLICATION (3DDR)

To design 3DDR, we have exploited *hello* messages for the dynamic replica management and distributed partitioning detection. Each node in 3DDR periodically exchanges *hello* messages with its 1-hop neighboring nodes after a predefined *hello interval*. A *hello* message of a node x contains the list of x's 1-hop neighboring nodes in addition to its LID, UID, and LSP. By exchanging the *hello* messages, each node in 3DDR maintains the connectivity information of up to 2-hops physical neighboring nodes. 3DDR identifies/detects a link $x_1 \leftrightarrow x_2$ between the nodes x_1 and x_2 as critical if the sets of 1-hop neighboring nodes of x_1 and x_2 are disjoint after removing the link $x_1 \leftrightarrow x_2$. If a node x_1 is 1-hop critical with respect to x_2 implies that the 1-hop neighboring nodes of x_1 would be inaccessible to x_2 if x_1 moves/fails. The link $x_1 \leftrightarrow x_2$ is therefore marked as 1-hop critical. Similarly, the link $x_1 \leftrightarrow x_2$ would be considered 2-hop critical if 2-hop neighboring nodes of x_1 and x_2 are disconnected upon the removal of the link. In 3DDR, A critical node or the nodes across the critical link sets their status as *critical*. Otherwise, the nodes status remains as *non-critical*. Each node announces its status (*critical/non-critical*) to its 1-hop neighboring nodes through the *hello* messages.

The following sub sections discuss in detail about the various components, i.e., dynamic replica deployment and distributed partition detection of the proposed 3DDR.

A. DYNAMIC REPLICA DEPLOYMENT

To replicate the mapping information of a node at various ANs based on the location of the critical links; 3DDR employs a dynamic replication mechanism and identifies the critical links using 2-hop neighbor information of a node as follows: Each node publishes its mapping information (i.e. LID and UID) by sending SMI message to its AN. When SMI is received at a node k from s, the node k performs the replication and forwarding decisions based on the outcome of the function f as shown in Eq(4). Let U be the set of nodes in the network and $M_k \subseteq U$ are 1-hop neighboring nodes of k excluding the node k,i.e.,open neighborhood, where $M_k^+ = M_k \cup \{k\}$, M_k^+ refers to the closed neighborhood of k,i.e., including the node k.

$$f(M_k^+, AN) = \begin{cases} k, \text{ if } d(k, AN) = min_{j \neq s \in M_k^+}(d(j,AN)), \; else \\ \exists_i \neq s | i \in M_k \wedge d(i, AN) = min_{j \neq s \in M_k}(d(j,AN)) \end{cases} \quad (4)$$

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

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

ii) If k is *critical* with respect to $p \in M_k$, then the mapping information in SMI is stored at the nodes k and p across the critical link $k \leftrightarrow p$.

Algorithm 1 illustrates the dynamic replication algorithm for replicating the mapping information. The mechanism ensures the availability of mapping information in disjoint partitions. This would avoid the communication disruption within partitions, but also reduces the lookup delay to retrieve the mapping information even in case the critical link fails. Moreover, the proposed algorithm significantly reduces the lookup delay even network partitioning does not occur because the nodes would easily be able to retrieve the mapping information from the deployed replicas besides the actual AN.

B. DISTRIBUTED PARTITION DETECTION

The proposed partition detection and LIS recovery mechanism of 3DDR exploits k-hop topological information to detect the critical nodes in the network. To achieve this, each node periodically exchanges a graph of its (k-1)-hop neighboring nodes with its directly connected neighboring nodes via *hello* messages. For $k = 2$, each node exchanges 1-hop neighbor list in the periodic *hello* messages beside other information, i.e., LID, UID, LSP. In such a way, each node maintains 2-hop topological information.

Formally, 1-hop neighbors of a node can be expressed as its neighborhood. Therefore, for $G = (V,E)$, the neighborhood of a vertex $v \in V$ is the induced sub-graph of G consisting of all the vertices adjacent to v and all the edges connecting those vertices. Similarly, the open neighborhood, $N(v)$, of the vertex v consists of the set of vertices adjacent to v excluding v, i.e. $N(v) = \{w \in N : vw \in E\}$ and the closed neighborhood of v is $N[v] = N(v) \cup \{v\}$. For any two adjacent nodes $u,v \in V$ across the link $u \leftrightarrow v \in E$, 1-hop neighboring nodes of u excluding u, i.e., open neighborhood of node u $N(u)$. As described earlier, each node exchanges its neighbor lists in the

Algorithm 1 Dynamic Replication Algorithm

Required: When S/l message is received at node k from node. v along the path to intended AN, where $M_k <= U, M_k M_k U\{kj\}$ denotes 1 -hop neighbors oT node k excluding k and including k respectively and U is the set of all nodes in the network. Function $f(M_k^+ AN)$ makes forwarding decisions.

On the reception of *SII* at a node k from node $s \in M_k^+$
if $((M_k^+, AN)$ *returns* $k)$ **then**
  if k *is non-critical* **then**
    $|k \leftarrow SII$ // *Store mapping info, and*
  end-if
      // *node k acts as designated AN*
  if (k *is critical w.r.t* $p \in M_k$) **then**
    $k \leftarrow SII$ // *p can be either s or any other node*
    $p \leftarrow SII$ // *replicate S/l across the critical link* $k \leftrightarrow p$
  end-if
end-if
if ($f(M_k^+ AN)$ *returns* $i \in M_k$) **then**
  if k *is non-critical* **then**
    Forward SI I to node / w.o.t making any replica,
  end-if
  if k *is critical w.r.t node I* **then**
    $k \leftarrow SII$ // *store mapping information across*
      // *the critical link(k,i)*
    $i \leftarrow Sll$
  end-if
end-if

Algorithm 2 Distributed Partition Detection

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

Each node u periodically broadcasts its closed neighborhood $N[u]$ and waits to
listen messages from other nodes.
for *each node u in the network* ***do***
  for *each node* $v \in adj[u]$ ***do*** /* *For any two adjacent neighbors (u, v) across the link (u,v), where N[u] and N[v] denotes open neighborhood of the adjacent nodes u and v*
    if $(N[u] \bigcap (N[v] \backslash u, v) = \{\})$
      STATUS $(u) \leftarrow$ *critical*
      STATUS $(v) \leftarrow$ *critical*
      link(u, v) $\leftarrow$ *critical*
    else-if $(N[u] \bigcap (N[v] \backslash u, v) \neq \{\})$
      STATUS $(u) \leftarrow$ *non-critical*
      STATUS $(v) \leftarrow$ *non-critical*
    end-if
  end-for
end-for
for *each node u in the network* ***do***
  for *each node* $v \in adj[u]$ ***do***
    if $(STATUS(v) == critical\ \&\&\ Partition\_Timer\_Expired == true)$
      Trigger partition event and
      Rccovcr/Rcusc the lost LIS.
    end-if
  end-for
end-for

periodic *hello* messages, therefore, nodes u and v periodically exchange *N(u)* and *N(v)*, respectively. If there exist common node(s) among *N(u)* and *N(v)*, i.e., $(N(u) \cap (N(v)) \neq \{\}$ or $(N[u] \cap N[v] \backslash u, v) \neq \{\}$, then the nodes u and v declare their status as *non-critical*, otherwise *critical*. A link between the two critical nodes would be treated as *critical*.

The identification of the critical nodes/links are vital to timely detect the network partitioning. For any two nodes x_1 and x_2 across a critical link $x_1 \leftrightarrow x_2$, the proposed distributed partition detection mechanism triggers the partitioning event and takes measures regarding the LIS alignment. Moreover, if the critical node x_1 across the critical link $x_1 \leftrightarrow x_2$ does not hear from x_2 for a certain time interval, referred as the *Partition_Timer*, then it means that the network has partitioned. *Partition_Timer* is set to three times the *hello* interval. Similarly, x_2 would also detect the network partitioning. In case of network partitioning, nodes across the critical link recover the lost LIS for reusing in the disjoint partitions. LIS recovery mechanism depends on the logical structure used. In the 3D structure each node across the critical links recover the lost LIS by changing its dimension value, i.e., *dim*. Similarly, in the cord structure nodes across the critical links recover the lost LIS by changing their LIDs to S and E based on their relationship, i.e., predecessor/successor. The predecessor node in the logical cord recovers the LIS by changing its LID to E and the successor node in the cord recovers the LIS by changing its LID to S. The lost LIS recovery and using the same results into an evenly distributed LISs in the disjoint partitions. Algorithm 2 provides an overview of the proposed mechanism for distributed partition detection. The proposed mechanism is fully distributed and relies on the local knowledge of the neighboring nodes without involving the network wide dissemination of control information. We are using the existing approach in [8] for network merging.

C. 3D-RP: A CASE STUDY

For illustration and evaluation, we implement the proposed distributed partition detection and replication algorithm in 3D routing protocol (3D-RP) [10]. A 3D-RP based scenario is shown in Figure 5 to illustrate the working of 3DDR. 3DDR first identifies the critical nodes/links using the existing *hello* messages exchanged in 3D-RP. Each node in the network periodically shares its 1-hop neighbor information through the *hello* messages besides the other information. Exchanging the 1-hop neighboring information provides each node a network wide view of up to 2-hops that assists in finding the critical nodes/links of up to 2-hop disconnected network. For instance, in the Figure 5, 1-hop neighboring nodes of f, s,

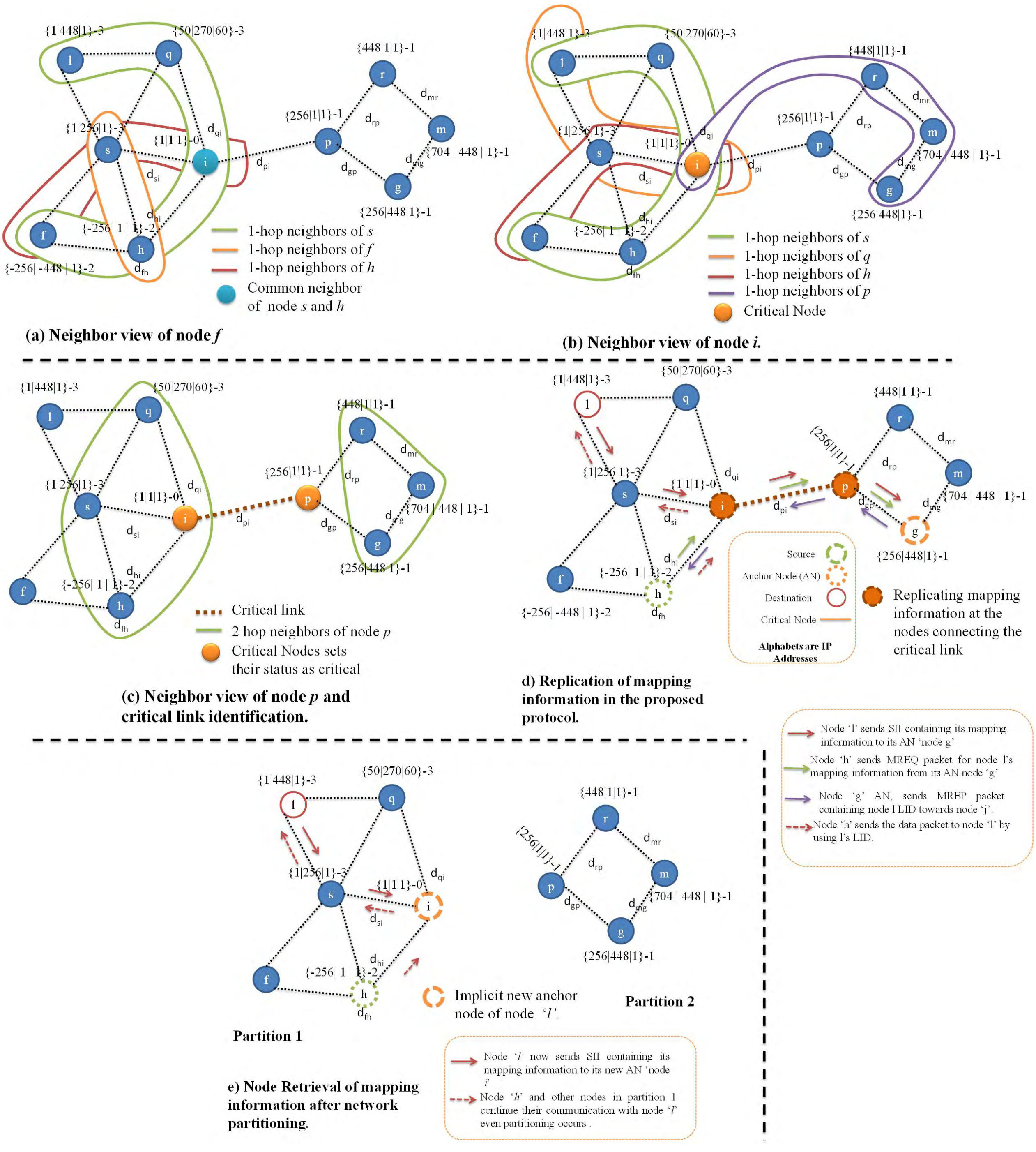


FIGURE 5. Handling network partitioning using 3DDR.

and *h* are *{s, h}*, *{l, q, i, h, f}*, and *{s, f, i}*, respectively. Upon receiving the neighbors information from the nodes *s* and *h*, the node *f* would have information of up to 2-hop neighboring nodes. In order to declare its status as *critical/non-critical*, node *f* checks for any common neighboring node among the 1-hop neighbors of *f*s adjacent neighboring nodes, i.e., node *s* and node *h* excluding itself. Figure 5 shows that there exists a node *i* common to the 1-hop neighboring nodes of *s* and *h*. This implies that the removal of *f* leaves a connected sub-graph *{l, q, i, h, s}* and the node *f* is 2-hop non-critical. So, *f* declares its status as *non-critical*. Similarly, the node *i* receives 1-hop neighbor information, i.e., *{s, l}*,*{q, l, f, h}*, *{s, f}*,*{r, g}*, *{r, m, g}* from its immediate neighboring nodes *q, s, h, p*, respectively. From Figure 5(b), it can be observed that the neighboring nodes *q, s, h* of *i* share the common nodes and form a connected sub-graph excluding the node *i*, however, *p*'s 1-hop neighboring nodes are not common to the connected sub-graph of *q, s, h*. This makes the nodes *i* and *p* to declare their status as critical. A link between the two critical nodes is also critical. In this case link $i \leftrightarrow p$ is critical. Similarly, each node in the network periodically updates its status as *critical* or *non-critical*.

After computing its LID, a newly joining node, say l, forwards the store mapping information (SMI) message to its AN in order to store its mapping information. In case if SMI passes through the critical node i and it is not forwarded by any other critical node, then i stores a copy of the mapping information in contained in SMI and forwards it further. Thus the critical nodes i and p connecting the link $i \leftrightarrow p$ keeps the replica of the information contained in SMI as shown in Figure 5(d). In case if the network is partitioned, then the communicating nodes in the disjoint partitions obtain the node ls mapping information from i. In this way, the communication within the partition would remain uninterrupted even the critical link fails. For instance, consider a scenario where a source node h wishes to communicate with a destination node l as shown in Figure 5(c).

The source node h retrieves the node l's mapping information from g, i.e., an AN for l. In case if the critical link $i \leftrightarrow p$ fails, the mapping information would not be available to the source node h. 3DDR avoids such issues and guarantees the availability of mapping information in case of the partitioning event. Taking pre-partitioning measures such as the critical nodes/links identification and placing the replicas enable the source node h to obtain the node l's mapping information from the node i despite of the network partitioning or in case the AN fails/moves as shown in Figure 5(d).

Furthermore, 3DDR triggers the partitioning event when the critical nodes across a critical link do not hear each other for a certain time interval, i.e., *Partition_Timer*. However, the critical nodes must be 2-hop as well as 1-hop critical, i.e., excluding a node(s) that is 2-hop critical but 1-hop non-critical. In Figure 5(d), when the critical nodes i and p across the critical link $i \leftrightarrow p$ do not hear each other and the *Partition_Timer* expires then the partitioning event triggers and the corresponding critical nodes recover/reuse the lost LIS. In case if the critical node is the anchor node, then it also replicates its mapping information across the critical link. The LIS recovery with replica placement allows uninterrupted communication in the disjoint partitions when both the source/destination nodes reside in the same partition, but the corresponding AN remains in the other partition. The recovered LIS can be reused in the disjoint partitions and may result into an evenly distributed disjoint LISs despite of the network partitioning. The LIS recovery procedure varies from a case to another case and depends primarily on the logical structure of the LIS, i.e., 3D-structures, Cord, Ring, Tree, etc.

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

V. SIMULATION MODEL AND PARAMETERS

To corroborate the validity and performance of 3DDR, the protocol is implemented in NS2 (version 2.35) [18]. The standard values for both the link and physical layers are exploited to simulate IEEE 802.11 with Two-Ray Ground as the propagation model. We have used BonnMotion2 [19] to develop the mobility scenarios that use random way-point as the mobility model and tunes the mobility parameters to stimulate moderate mobility, i.e., the node moving speeds are observed in the range from 1 to 2 m/s. The mobility scenarios using BonnMotion2 keep track of the physical network partitioning. Table 2 illustrates the simulation parameters in detail.

TABLE 2. Simulation parameters.

Parameters	Their Values
Total Nodes	[50-200]
Radio range	100m
Simulation Area	1000*1000m
Data transmission	64pps
Total Time of Simulation	500 sec
Node Moving Speed	1 to 2 m/s
Traffic Model	Random Traffic pattern
No. of flows	12
Topology Connectivity	Bonnmotion 2
Propagation Model	Two-Ray Ground

The simulation results of 3DDR are compared with VCP, 3D-RP and VCP with approach (VCP-WA). In VCP-WA, the proposed distributed partition detection and replication mechanism (DDR) is implemented in VCP to further stress the impact of DDR mechanism in other schemes. We have used CBR flows to model the data traffic and Random Traffic Model is used as the data pattern. TCP is not used at the transport layer to circumvent the elasticity effects on routing due to the TCP flow control mechanism [23]. To elude the packet drops due to congestion, the traffic load in the network is maintained at 64 pkts/sec. Following metrics are examined under the various network sizes to investigate the performance of protocols.

- Success Ratio (SR): The ratio between the total mapping request packet (MREQ) initiated and the total MREQ entertained successfully by receiving the mapping reply packets (MREP).
- End-to-end Lookup delay (E2E): The average time elapsed between when the source node initiate MREQ and the source node gets MRPY.
- Normalized Overhead (NO): Total routing overhead divided by the total MREQ for which MREP is received, i.e., the routing overhead per successful MREQ.

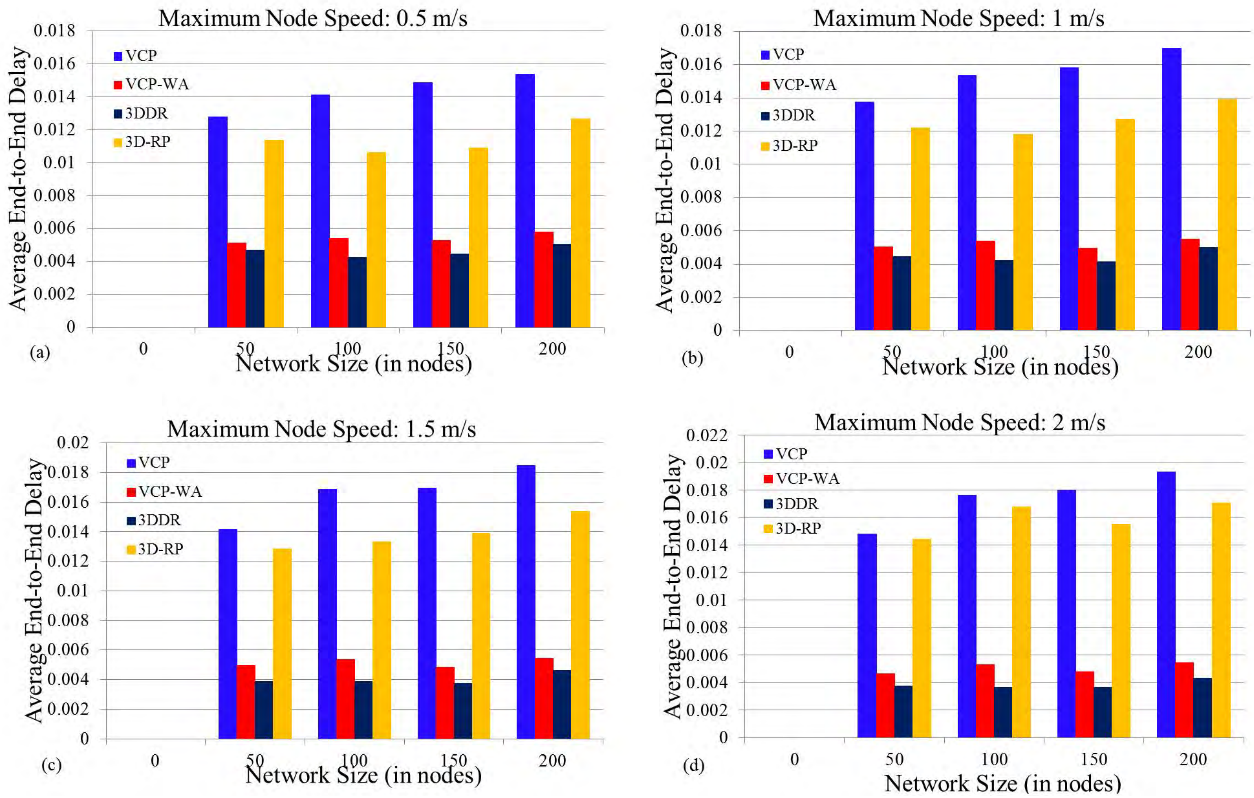


FIGURE 6. **Average end-to-end lookup delay as a function of the node number.**

A. END-TO-END LoOKUP DELAY

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

Figure 6 illustrates the average end-to-end lookup delay of 3DDR, VCP-WA, 3D-RP, and VCP with respect to the node moving speed from 0.5m/s to 2m/s with varying number of nodes in the network. The results in Figure 6 shows significant gains in terms of reducing the end-to-end lookup delay of 3DDR and VCP-WA compared to VCP and 3D-RP. The main factor in improving the end-to-end lookup delay is the proposed distributed partition detection and replication mechanism that identifies the critical nodes and replicates the mapping information for seamless communication. The placement of additional replicas reduces the lookup delay by responding from a nearby replica instead of forwarding MREQ to the original AN. By dealing with the nearby replica instead of the original AN against a MREQ reduces the traffic overhead in the proposed approach. The resulting effect of minimizing the traffic overhead decreases the contention to access the medium at MAC layer in IEEE 802.11, which helps further in reducing the end-to-end lookup delay in the proposed approach.

The improvement in end-to-end lookup delay shown by 3DDR over VCP-WA demonstrates the effectiveness of 3DDR against the mismatch problem. An increase in the node moving speed raises the probability of having the critical nodes in the network. Moreover, as the network frequently changes its topology due to node mobility, the lookup and routing traffic in the network increase due to the recurrent execution of the distributed partition detection and replication mechanism, which elevates the end-to end lookup delay for all protocols. But, the impact is less in 3DDR and VCP-WA as illustrated in Figure 6 because, 3DDR and VCP-WA both exploit the critical nodes for placing replicas across the critical links and thus more effective in reducing the end-to-end lookup delay compared to VCP and 3D-RP. But, 3DDR is more promising in reducing the end-to-end delay compared to VCP because of its resilient 3D logical network that reflects the physically proximity of nodes exactly in the logical network, which minimizes the impact of the mismatch problem that results in the optimized routes and reduced end-to-end lookup delay.

Since our proposed proposed exploits the critical nodes for deploying the replica, it significantly reduces the lookup delay with an increase in the nodes moving speed as shown in Figure 6. Figure 9 illustrates the percentage improvement in the average end-to-end lookup delay for 3DDR at various node speeds compared to VCP, VCP-WA and 3D-RP. As shown in Figure 9, the end-to-end lookup delay improvement of 3DDR over VCP-WA, 3D-RP, and VCP is between 9% and 22%, 58% and 72%, 62% and 71%, respectively at various node moving speeds. The end-to-end lookup delay increases with the increase in number of nodes because expanding the network in terms of nodes amplifies the impact of partitioning and merging that increases the end-to-end lookup delay as demonstrated in the results illustrated in Figure 6.

B. NORMALIZED OVERHEAD

The normalized overhead is an important factor to defining the network scalability and shows per lookup request overhead. Figure 7 illustrates the normalized overhead of 3DDR, VCP-WA, 3D-RP, and VCP with respect to node moving speeds 0.5m/s - 2m/s with varying network sizes. Figure 7 shows significant gains in term of reducing the normalized overhead of 3DDR and VCP-WA compared to VCP and 3D-RP at various node moving speeds. The dominant factor in improving the normalized overhead is an adoption of DDR mechanism that identifies the critical nodes and replicates the mapping information, thus reducing the lookup overhead. Figure 9 illustrates the percentage improvement in normalized overhead for 3DDR over VCP-WA, 3D-RP, and VCP is between 4% and 8%, 30% and 41%, 33% and 40%, respectively at various node moving speeds. In Figure 7, the normalized overhead for 3DDR and VCP-WA is a bit high, but still better compared to VCP and 3D-RP and is controlled with respect to the increase in the network size. The increased overhead is mainly due to the adoption of DDR mechanism by those protocols. The normalized overhead increases with the node moving speed as the network frequently changes its topology, generating more lookup and routing traffic in the network by performing the operations, e.g., displacement of the anchor nodes, the LIS recovery, storing the mapping Information at ANs, and the reassignment of LIDs. The normalized overhead of 3DDR is lower as shown in Figure 7, because of the proposed mechanism for detecting the critical nodes and replicating the mapping information. In addition, 3DDR employs resilient 3D logical network that is proved to be effective in handling the mismatch problem, the partitioning and merging of the network.

C. SUCCESS RATIO

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

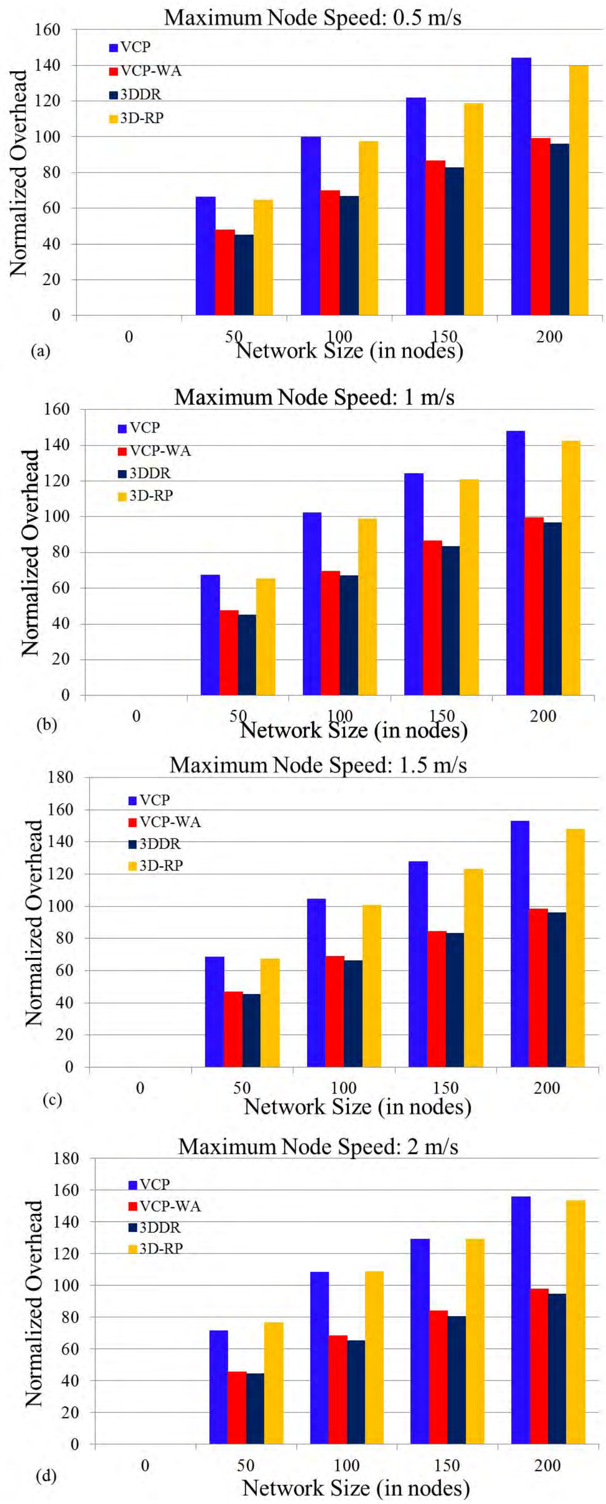


FIGURE 7. Normalized Overhead as a function of the node number.

Figure 8 illustrates the impact of increasing the network size and the node moving speed on the success ratio is less in 3DDR compared to VCP, 3D-RP, and VCP-WA. 3DDR outperforms its competitors in terms of success ratio, which verifies the effectiveness and the capability of 3DDR in

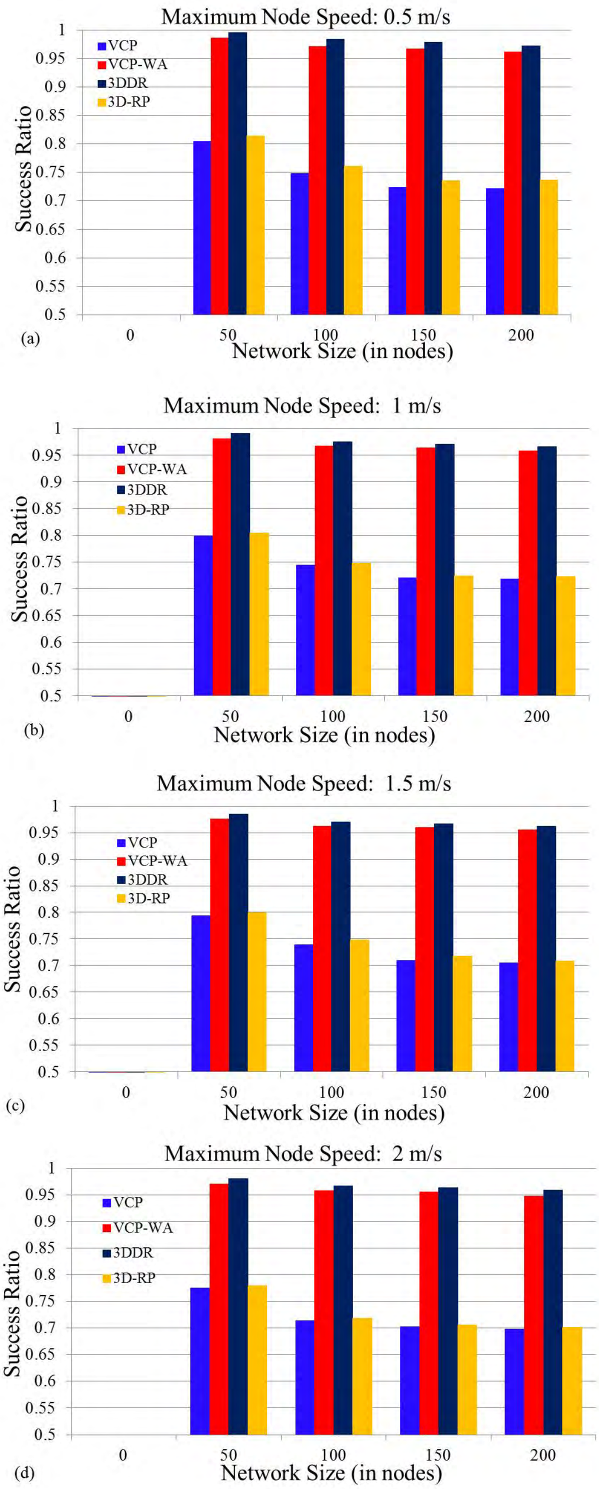


FIGURE 8. Success Ratio as a function of the node number.

delivering the packets in the large network. The proposed partition detection and replication mechanism further assists further in maintaining and improving the success ratio of 3DDR. 3DDR replicates the mapping information across all the critical links along the path to AN. Therefore, in case of network partitioning, the mapping information readily available in the disjoint partitions and subsequently contributes the high success ratio in the proposed protocol. Moreover, an increase in the node moving speed causes frequent network partitioning and merging. In such scenarios, retrieving the mapping information may not be possible using 3D-RP and VCP if the corresponding AN resides in the disjoint partition. Secondly, maintaining the replica in 3DDR and VCP-WA reduces the traffic overhead, which decreases the probability of packet collision at the MAC layer, thus contributes in maintaining the higher success ratio. Furthermore, VCP and VCP-WA suffer from the mismatch problem as Cord logical structure is not flexible to reflect the physical intra neighbor relationships of nodes exactly in the logical Cord network, resulting in long routes and redundant traffic. This increase the probability of packet collision at the MAC layer, thus reducing the success ratio of both VCP and VCP-WA compared to 3DDR as shown in Figure 8.

Figure 9 illustrates the percentage improvement in success ratio of 3DDR at various node moving speeds compared to VCP, VCP-WA and 3D-RP. The improvement in success ratio of 3DDR over VCP-WA, 3D-RP, and VCP is between 2% and 4%, 24% and 38%, 21% and 35%, respectively for various network sizes and node moving speeds. The increase in network size and node moving speed cause the network topology to change more frequently, which result in an increased lookup and routing overhead due to frequent execution of the recovery and partition detection operations. This causes congestion at the MAC layer as packets are delayed in the queue and drop eventually. The impact of congestion on 3DDR is lower compared to the rest.

D. ERROR ESTIMATION IN DISTANCE CALCULATION USING RECEIVED SIGNAL STRENGTH INDICATOR (RSSI)

Distance calculation among nodes is a complex process and there are number of factors, for instance the transmission range, node placement, node churns, etc. can influence the value of the measured distance using RSSI. In order to see the impact of error in distance calculation using RSS and the behavior of 3DDR in that case, we conduct experiments by keeping the number of nodes and the data rate constant 100 and 64 pkts/sec, respectively. We introduced error in distance measurement using RSS at a particular node n in order to observe the impact of this error in calculating the LID of node n and its recurrent effect on the routing of packets from the source to the destination node, where node n may acts as an intermediate node towards the destination node depending upon its LID. The range of error is introduced on distance estimation, at the parameter setting (mean = 0, standard deviation [5, 10, 15, 25, 30, 35, 40, 45, 50]. Upon running the simulation, we observed that the path-stretch ratio(the ratio of the length of the traversed path by a routing algorithm to the available shortest path. It describes the resilience of the routing algorithm to find the shortest route towards a destination node in the network) of 3DDR increases with the

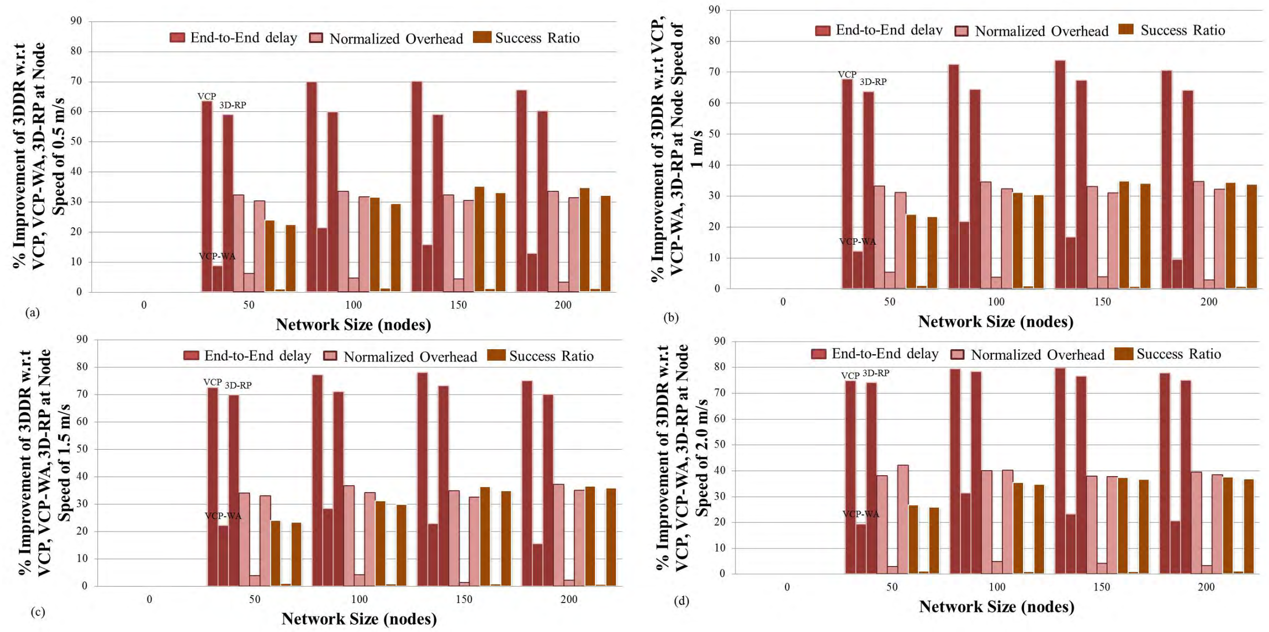


FIGURE 9. % Improvement of 3DDR over VCP, VCP-WA, 3D-RP in terms of E2E, NO, and SR. The protocol name is displayed on first three bars of each graph and the pattern is same for the rest in each graph.

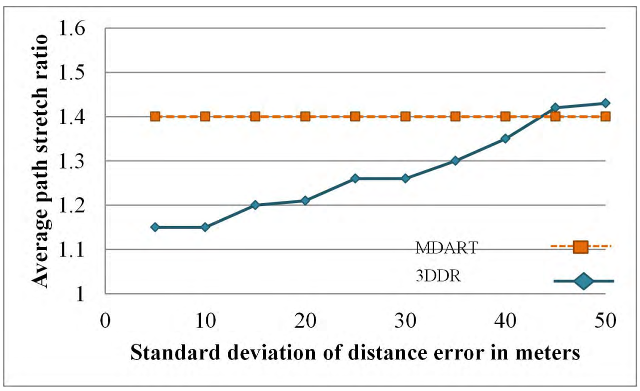


FIGURE 10. Impact of error in distance estimations on path-stretch ratio.

increase of error in distance measurement, but it remains less than the path stretch ratio up to the error value of 40 meters. On further increasing the error in distance measurement, the path-stretch ratio obtained for 3DDR exceeds the path stretch ratio of MDART [6] as shown in Figure 10.

From the results reported in Figure 10, it can be inferred that when assuming a Gaussian error on distance estimation, at parameter setting (mean = 0, standard deviation [5, 10, 15, 25, 30, 35, 40, 45, 50], number of nodes = 100), the proposed protocol shows a significant performance improvement compared to MDART for standard deviation of the distance error up to 40 meters.

VI. CONCLUSION

The network partitioning detection in a DHT-based routing protocol for MANETs under the scalability constraints is a challenging task and causes the unavailability of mapping information, disrupted logical identifier space, and longer lookup delay, which may completely halt the communication in the network. Similarly, the timely detection of the partitioning event before the actual network partitioning allows the induction of pre-partitioning measures such as the replication of mapping information, LIS recovery, etc.

In this paper, we propose mechanisms to handle the distributed partition detection, dynamic replication management and exploring the possibilities to minimize the lookup and routing delay for a DHT-based routing paradigm in MANETs. We have exploited the network dynamics to timely detect the partitioning event and for taking the pre-partitioning measures. Our proposed scheme identifies the critical nodes/links using the localized neighboring knowledge available to nodes without creating the additional traffic overhead. A case study is conducted to provide an up-close, in-depth, and detailed examination of the proposed distributed partition detection and replication algorithms in order to enhance the understanding of various related contextual conditions and to evaluate the functionality of the proposed mechanisms.

Performance comparisons with the existing approaches endorse the effectiveness of the proposed mechanism. As a future work, we have planned to extend our work; i) using other mobility models and a unified approach for handling both the merging and partitioning detection; ii) to corroborate the feasibility of such protocols in the emerging domains such as the in-band software defined networks, smart grids, etc.

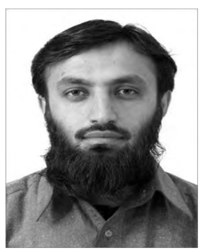
REFERENCES

[1] S. Jain, Y. Chen, and Z.-L. Zhang, "Viro: A scalable, robust and namespace independent virtual ID routing for future networks," in *Proc. INFOCOM*, Apr. 2011, pp. 2381–2389.

[2] A. Awad, R. German, and F. Dressler, "Exploiting virtual coordinates for improved routing performance in sensor networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 9, pp. 1214–1226, Sep. 2011.
[3] J. Eriksson, M. Faloutsos, and S. V. Krishnamurthy, "DART: Dynamic address routing for scalable ad hoc and mesh networks," *IEEE/ACM Trans. Netw.*, vol. 15, no. 1, pp. 119–132, Feb. 2007.
[4] Q. Xu, L. Sun, and J. Liu, "Topology-aware Kademlia based on distributed clustering in self-organizing mode," in *Proc. 2nd Int. Conf. Comput. Eng. Technol. (ICCET)*, vol. 1. Apr. 2010, pp. V1-267–V1-271.
[5] S. A. Abid, M. Othman, and N. Shah, "Exploiting 3D structure for scalable routing in MANETs," *IEEE Commun. Lett.*, vol. 17, no. 11, pp. 2056–2059, Nov. 2013.
[6] M. Caleffi and L. Paura, "M-DART: Multi-path dynamic address routing," *Wireless Commun. Mobile Comput.*, vol. 11, no. 3, pp. 392–409, 2011.
[7] S. A. Abid, M. Othman, and N. Shah, "A survey on DHT-based routing for large-scale mobile ad hoc networks," *ACM Comput. Surv.*, vol. 47, no. 2, p. 20, 2015.
[8] S. A. Abid *et al.*, "Merging of DHT-based logical networks in MANETs," *Trans. Emerg. Telecommun. Technol.*, vol. 26, no. 12, pp. 1347–1367, 2015.
[9] I. Stojmenovic, D. Simplot-Ryl, and A. Nayak, "Toward scalable cut vertex and link detection with applications in wireless ad hoc networks," *IEEE Netw.*, vol. 25, no. 1, pp. 44–48, Jan. 2011.
[10] S. A. Abid, M. Othman, N. Shah, and M. A. Khan, "3D-RP: A DHT-based routing protocol for MANETs," *Comput. J.*, vol. 58, no. 2, pp. 258–279, 2015.
[11] S. Shin, U. Lee, F. Dressler, and H. Yoon, "Motion-MiX DHT for wireless mobile networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 12, pp. 3100–3113, Dec. 2016.
[12] H. Ritter, J. Rolf, and J. Schiller, "A partition detection system for mobile ad-hoc networks," in *Proc. 1st Annu. IEEE Commun. Soc. Conf. Sensor Ad Hoc Commun. Netw. (SECON)*, Oct. 2004, pp. 489–497.
[13] A. Derhab, N. Badache, and A. Bouabdallah, "A partition prediction algorithm for service replication in mobile ad hoc networks," in *Proc. 2nd Annu. Conf. Wireless On-Demand Netw. Syst. Services*, Jan. 2005, pp. 236–245.
[14] K. H. Wang and B. Li, "Efficient and guaranteed service coverage in partitionable mobile ad-hoc networks," in *Proc. 21st Annu. Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM)*, vol. 2. Jun. 2002, pp. 1089–1098.
[15] M. Hauspie, J. Carle, and D. Simplot, "Partition detection in mobile ad-hoc networks using multiple disjoint paths set," in *Proc. Int. Workshop Objects Models Multimedia Technol.*, 2003, p. 15.
[16] B. Milic, N. Milanovic, and M. Malek, "Prediction of partitioning in location-aware mobile ad hoc networks," in *Proc. 38th Annu. Hawaii Int. Conf. Syst. Sci.*, vol. 2. Jan. 2005, p. 306.
[17] H. Wirtz, T. Heer, R. Hummen, and K. Wehrle, "Mesh-DHT: A locality-based distributed look-up structure for wireless mesh networks," in *Proc. IEEE Ad-Hoc Sensor Netw. Symp. (ICC)* Ottawa, ON, Canada, Jun. 2012, pp. 653–658.
[18] K. Fall and K. Varadhan, Eds., *The ns Manual (Formerly ns Notes and Documentation)* The VINT Project A Collaboration Between Researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC, Nov. 2011. [Online]. Available: https://www.isi.edu/nsnam/ns/doc/
[19] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn, "BonnMotion: A mobility scenario generation and analysis tool," in *Proc. 3rd Int. ICST Conf. Simulation Tools Techn. (SIMUTools)*, Torremolinos, Spain, 2010, p. 51.
[20] N. Shah, S. A. Abid, D. Qian, and W. Mehmood, "A survey of P2P content sharing in MANETs," *Comput. Elect. Eng.*, vol. 57, pp. 55–68, Jan. 2017.
[21] R. Nawaz, S. A. Hussain, S. A. Abid, and J. Shafi, "Beaconless multihop routing protocol for wireless sensor networks," in *Proc. IEEE 3rd Int. Conf. Commun. Softw. Netw. (ICCSN)*, May 2011, pp. 721–725.
[22] A. Tahir, S. A. Abid, and N. Shah, "Logical clusters in a DHT-paradigm for scalable routing in MANETs," *Comput. Netw.*, vol. 128, pp. 142–153 Dec. 2017. [Online]. Available: https://doi.org/10.1016/j.comnet.2017.05.033
[23] G. Holland and N. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," *Wireless Netw.*, vol. 8, nos. 2–3, pp. 275–288, 2002.
[24] D. Tian, J. Zhou, Z. Sheng, and V. C. M. Leung, "Robust energy-efficient MIMO transmission for cognitive vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3845–3859, Jun. 2016.
[25] D. Tian, J. Zhou, Z. Sheng, and Q. Ni, "Learning to be energy-efficient in cooperative networks," *IEEE Commun. Lett.*, vol. 20, no. 12, pp. 2518–2521, Dec. 2016.
[26] A. Potluri and G. Prathyusha, "An efficient DHT-based elastic SDN controller," in *Proc. 9th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, 2017, pp. 267–273.
[27] B. Dumba, H. Mekky, S. Jain, G. Sun, and Z. L. Zhang, "A virtual ID routing protocol for future dynamics networks and its implementation using the SDN paradigm," *J. Netw. Syst. Manage.*, vol. 24, no. 3, pp. 578–606, 2016.

SALEEM ZAHID received the M.S. degree from IIU Islamabad, Pakistan, where he is currently pursuing the Ph.D. degree. His research area is computer networks.

SHAHBAZ AKHTAR ABID received the M.S. (CS) degree in data communication networks and distributed systems from the COMSATS Institute of Information Technology, Pakistan, and the Ph.D. degree (Hons.) in computer networks (computer sciences) from the University of Malaya, Malaysia. He is currently an Assistant Professor with the COMSATS Institute of Information Technology. He is an HEC approved Supervisor. His research interests include communication and security issues in self-organized networks and distributed systems. He received the fully funded performance-based scholarship during his M.S. (CS) from COMSATS and the fully funded scholarship during his Ph.D. under the Bright Spark Scholarship Program from University of Malaya.



NADIR SHAH received the B.Sc. and M.Sc. degrees in computer science from Peshawar University, Peshawar, Pakistan, in 2002 and 2005, respectively, the M.S. degree in computer science from International Islamic University, Islamabad, Pakistan, in 2007, and the Ph.D. degree from the Sino-German Joint Software Institute, Beihang University, Beijing, China. He was a Lecturer with the Department of Computer Science, COMSATS Institute of Information Technology, Abbottabad, Pakistan, from 2007 to 2008, and an Assistant Professor with COMSATS Wah Campus from 2011 to 2015. He has been an Associate Professor with the COMSATS Institute of Information Technology, Pakistan, since 2011. His current research interests include computer networks, distributed systems, and network security. He has published several research papers in international journals/conferences, such as *ACM Computing Surveys* and the IEEE Communication Letters. He has been serving as a Reviewer for several journals/conferences, including ICC, *Computer Networks* (Elsevier), and *The Computer Journal*.).

HUSNAIN ABBAS NAQVI received the M.S. degree (Hons.) from MAJU, Islamabad, and the Ph.D. degree in sensor networks from The University of Auckland, New Zealand. He is currently an Assistant Professor/Acting Chairman with IIU Islamabad, Pakistan. His research interest includes statistical software engineering, stochastic process, ad-hoc/sensor networks, collaborative/cooperative communications, and information security.

Waqar Mehmood received the master's degree in software engineering from CUST, Islamabad, and the Ph.D. degree in computer science (software engineering) from the University of Insbruck, Austria, in 2011. He is currently an Assistant Professor with the COMSATS Institute of Information Technology, Wah Cantt, Pakistan. His research interests include model-driven engineering, software configuration management, and software testing. He received the CUST 100% Fee Waiver Scholarship for his M.S. degree and the HEC Oversees Scholarship for his Ph.D. degree.