A Mobility and Congestion Resilient Data Management System for Distributed Mobile Networks

Ze Li and Haiying Shen
Department of Electrical and Computer Engineering
Clemson University, Clemson, SC 29631
Email: {zel,shenh}@clemson.edu

Abstract**—Data sharing is one of the most popular applications that dominate 70% of data traffic on the Internet. This application has been penetrating wireless mobile networks with dramatic speed, which allows the sharing of data whenever and wherever. However, traditional client-server data sharing model suffers from single point failure and low-scalable transmission in a highly dense and dynamic scenario. MANET is a promising alternative structure for flexible and distributed data sharing. However, the topology-based flooding employed in MANET for data routing and querying prevents the data sharing system from achieving high scalability and robustness. Building a DHT on a MANET reduces data query overhead, but the inconsistency between the overlay and the underlying topology degrade system performance. Geographic routing based data sharing in MANET reduces the data routing overhead. However, it has more requirements (e.g. GPS) on the nodes. This paper presents a LOcality-based distRibuted Data sharing system (LORD) for mobility and congestion resilient data management. LORD consists of a DHT-based data index and retrieval protocol and a locality-aware region-based data routing protocol. It provides highly efficient, scalable and dynamism-resilient data sharing with low overhead. Moreover, it offers similarity data searching function. Theoretical analysis and simulation results show the superiority of LORD compared with other data sharing systems in terms of scalability, overhead and dynamism-resilience in a highly dense and dynamic MANET.**

I. INTRODUCTION

Data sharing is one of the most popular applications on the Internet, which dominates 70% of data traffic. People exchange and browse information through MSN, Facebook, CNN News and so on. The number of US on-line video viewers increased rapidly from 52.3 million in 2003 to 155.2 million in 2008 and has a projection about 183.0 million in 2011 [1]. At the same time, the technical improvements in microelectronics and wireless communications increase the availability of mobile applications in human life. Due to the restrictions of wired communication, the data sharing applications have been penetrating wireless mobile networks with dramatic speed. Low cost laptops and notebooks, powerful Personal Digital Assistants (PDAs), intelligent cellular phones and communication-enabled vehicles that facilitate people to "working while walking" are recently prevalent all over the world. We envision that there will be omnipresent wireless devices in the near future, and some areas such as urban cities will be covered by ubiquitous mobile nodes with high mobility and density. An efficient data sharing system suitable for a highly dynamic and dense wireless network especially the mobile ad hoc networks(MANET) is increasingly needed.

Currently, most wireless data sharing systems use client-server model [2]. That is, nodes query data from a centralized server, which collects all data information from nodes. If the density of the nodes in an area is low, the servers can provide satisfactory performance. However, as the density of the mobile devices increases, the traditional system may suffer from potential drawbacks such as single point of failure, hot spots, low scalability, distant centralized database [3], and high energy consumption [4].

A distributed data sharing system is a promising alternative to deal with the problems in the client-server model. Current distributed data sharing approaches on MANETs can be classified into two categories based on the underlying data routing protocols they employ. One approach is based on the on demand topological routing algorithms [5–10] and the other is based on the geographic routing algorithms [11–15]. In the topological routing protocols, before a source node transmits or retrieves data message, it uses broadcasting to find a route from itself to the destination node. Due to the high volume of the overhead generated, the algorithms are not suitable for large-scale and high dynamic distributed system. In addition, too many messages will generate interference in a highly dense MANET. In contrast, geographic routing protocols have been proved scalable without explicit end-to-end route establishment and maintenance. The protocols forward messages based on the location information of each node provided by positioning systems. On the other hand, in order to reduce the data querying overhead, many geographic routing-based data sharing systems [12–15] map data to a location spot, store the data to the node closest to the location, and rely on geographic routing to publish and retrieve the data. However, most such geographic routing-based system can only be used in low mobility MANETs and are not suitable for highly dynamic MANETs. High node mobility leads to frequent data mapping location updates, generates high overhead and deteriorates system performance. A delayed location mapping update may lead to a query failure. In addition, node localization needed in the geographic routing adds overhead burden and exacerbates the energy consumption. To localize nodes, some systems

978-1-4244-5113-5/09/$25.00

use Global Position System (GPS) [12, 13, 15, 16] while others [14, 17–19] rely on virtual coordinates. GPS receivers, functioning as additional high power hardwares, generate extra burden on the nodes' precious energy resource. Meanwhile, there are many situations where location information is not available at the nodes (e.g. indoors) [17]. In addition, the virtual coordinate methods need periodical coordinator updates, which also produces high overhead in a highly dynamic system.

In order to build an efficient distributed data sharing system for highly dense and dynamic MANETs, we propose a LOcality-based distRibuted Data sharing system (LORD). LORD consists of two main components: (1) Region-based data storage, and (2) Region-based geographic data delivery. A *metadata* is used to record keywords of its associated file and the current location of the file holders. LORD divides the entire MANETs area into a number of geographic regions and maps the metadata of a file to a region under DHT (distributed hash table) hashing policies. All nodes in a region hold a copy of the metadata. When a node moves to a new region, it updates the regions having all of its published metadata, drops the metadata it holds that belongs to old region, and retrieves the metadata from the new region. After a requester retrieves the metadata, it asks for the file directly from the physically close file holders based on the region information stored in the metadata. After receiving the file, the node registers its location to the metadata of the received file and regards the metadata as its own published metadata for its region tracking. In order to reduce the system overhead, a Region-based Geographic Routing (RGR) algorithm is proposed to forward data query and response efficiently. In RGR, each node only knows the physical region it resides based on landmarks beacons. The packets are forwarded based on the relationship between the physical regions. We summarize the contributions of this paper as follows:

- An efficient and congestion resilient region-based data index and retrieval hashing protocol is proposed which generates small overhead for a highly dynamic MANET and provides a similarity data query function for file retrieval.
- An energy efficient and mobility resilient region-based locality routing protocol, which combines topological routing and geographic routing is proposed. In addition, a parallel data transmission algorithm is proposed to further enhance the efficiency of file retrieval.
- A node localization protocol is proposed to facilitate the localization of mobile nodes.
- Comprehensive theoretical analysis and simulations demonstrate the superiority of LORD in comparison with previous topological routing-based and geographic routing-based data sharing systems.

The remainder of this paper is organized as follows. Section II summarizes related work of data sharing systems in MANETs. In Section III, we present the overview and design of LORD with an emphasis on its data index and retrieval protocol and RGR protocol. In Section IV, the performance of LORD is studied in comparison with other approaches. Section V concludes the paper with remarks on our plans for future work.

II. RELATED WORK

In mobile wireless networks, the data sharing service is mainly implemented with a client-server model [2, 20]. In this model, mobile nodes publish and retrieve data through centralized servers. However, because of its centralized nature, the client-server model suffers from single point of failure, hot spots, high-energy consumption, and low channel usage, especially in a large-scale network.

In order to resolve the problems faced by the centralized network, decentralized data sharing systems based on MANETs have been proposed recently. One group of data sharing systems includes 7DS [5], PDI [6] and ORION [7] that employ traditional MANET topological routing protocols such as AODV [8] and DSR [9] for message transmission. AODV and DSR are on-demand multi-hop routing algorithms that build a route path when a source node wants to send a message to a destination node. Basically, the methods rely on message flooding to establish a route from the source to the destination. Flooding transmission generates a tremendous number of messages and prevents the system from achieving higher scalability. These data sharing systems query data based on file name rather than node ID. Thus, a node needs to maintain a routing table for each data. In a system with vast amounts of data, a large number of routing tables will consume extremely large amounts of computing resources. The work in [21] proved that the topological routing is not applicable in a highly dynamic and dense environment.

In order to avoid the drawbacks of flooding search, some data sharing systems integrate a DHT overlay network [22–24] into the topological routing for efficient data searching. DHT overlay networks is a class of decentralized systems that partition ownership of a set of objects among participating nodes and can efficiently route messages to the unique owner of any given object. Each object and node is assigned a unique ID. An object is stored in a node whose ID equals or immediately succeeds the object's ID. DHT overlay networks provide efficient data searching that achieves a time complexity of $O(\log n)$ per lookup request by using $O(\log n)$ neighbors per node, where n is the number of nodes in the network. In the DHT-based data sharing systems, routing is conducted based on DHT routing algorithms. In each step in the routing, a node initializes a query flooding when looking for the next overlay hop. The inconsistency between the overlay and the underlying topology may cause packet transmission to take unnecessary longer path. Although VRR [25], Ekta [26] and MADPastry [10] can solve the topology consistency problem to some extent, the methods cannot significantly improve the system efficiency and scalability for two reasons. The first reason is that the methods still rely on flooding for the routing in each step. And the second reason is that the side effects brought about by the mismatch between the overlay layer and physical layer still exist.

Geographic routing is an alternative routing method to achieve higher routing scalability. It forwards the message to a node geographically closer to the destination in each step. Recently, a number of data sharing systems [12–15] have been proposed that depend on geographic routing for data searching in a large-scale wireless sensor network. Based on the DHT data allocation policy, these systems map a file to a geographical location, and the file is stored in the node closest to that location. To query a file, a node calculates the mapped location of the file, and uses geographic routing to send its query to the file host. Two methods can be used to provide location information for geographic routing: GPS-based [12, 13] and virtual coordinate-based [14, 17–19]. GPS-based localization methods consume a great deal of energy, which is a precious resource in MANETs. In some situations (e.g. indoor), the devices cannot work well. Rather than relying on GPS, virtual coordinate-based methods build virtual coordinates to compute the relative location of nodes. However, these methods need to periodically update nodes' virtual coordinates, which generates high overhead in a highly dynamic MANET. Meanwhile, in a dynamic MANET, the home node closest to a file's location frequently changes. Thus, these systems need to update the home node of each file and transfer data frequently, which lead to high overhead. Therefore, current locality based systems are mainly suitable for low dynamic or static networks.

For region-based routing, Datta et. al. [27] proposed connected dominating sets routing. It chooses a number of special nodes as dominating nodes. All non-dominating nodes connect to the dominating nodes for packet transmission. Frey and Gorgen [28] proposed a geographic cluster routing method (GCR). Routing in GCR is not performed on a per-node basis, but packets are forwarded along the edges of adjacent geographical clusters. The geographical clusters are a mesh of regular hexagons formed according to the topology of the mobile nodes. However, both of these two methods are still GPS-based routing. Meanwhile, as the topology of the network changes, the clusters in the above two methods are also changed, which increases the clusters' maintenance overhead. Z. J. Haas [29] proposed a hybrid zone-based routing protocol that combines proactive routing and reactive routing. Each node in the system maintains a zone centered on itself. For the nodes in the zone, ZRP uses proactive routing. For the nodes in other zones, reactive routing is conducted. Although such mechanism can increase the performance of pure proactive routing and reactive routing to certain extents, the topology-based routing nature is still not suited for highly dense and dynamic scenarios. Unlike the topology-based clustering, in LORD, the regions are geographically fixed. Based on the relationship between these regions, a directed region-based routing is conducted without GPS for the communication between the nodes in different regions. For the nodes within a region, LORD uses a reactive routing to identify specific nodes.

In [30], we proposed a preliminary prototype for data sharing in a highly dynamic and dense system. In this paper, we improve the prototype by introducing a directed region-based routing protocol, a parallel file transmission protocol, a location track and back-track protocol, and a similarity search protocol. More experimental results and analysis are provided for better understanding of the data sharing problems.

III. The LORD Data Management System

In LORD, the entire geographical area is divided into a number of physical regions. The metadata of a file is mapped to a region and stored by all nodes in the region. The request for this metadata will be forwarded to the region using region-based routing. After retrieving the metadata, the file requester asks for the file from the geographically close file hosts.

A. Region Generation

We consider a highly dense and dynamic MANET with nodes spreading out over an area. The area is divided into a number of regions by landmarks. The landmarks periodically emit identification beacons [31–34]. Considering the promising ubiquitous computing environment in the future, such static landmarks will not be difficult to find. The information about geographic boundaries of regions (i.e. a geographic map) is configured into a node when it joins the system. We assume each node has the capability to sense the direction and signal strength of a landmark from which it receives a signal [35]. Each region is represented by a region ID. Each node identifies its region by the received signal from a landmark in a region of any shape. Figure 1 shows the resultant regions in a MANET in LORD. The size of the basic region can be determined depending on the number of the regions m in the system, the transmission range R of the node, and the size S of the entire areas all-together. For example, if we wish the basic region can be covered by the transmission range of each node, the diameter X of each basic region should satisfy $X < R$ and $\frac{S}{m} < \pi X^2$.

Each region is identified by an assigned region integer ID and the region is confined to a virtual area. Suppose each region is a convex polygon with v vertices whose coordinates denoted as (x_i, y_i), where $1 \leq i \leq v$. Without loss of generality, we assume $(x_1 < x_2 < ... < x_n)$, where x_i and x_{i+1} are adjacent and x_1 adjacent with x_n. Therefore, each region can be represented as

$$R = \begin{cases} y = k_1 \cdot x + b_1 & (x_1 \leq x \leq x_2) \\ y = k_2 \cdot x + b_2 & (x_2 \leq x \leq x_3) \\ y = k_i \cdot x + b_i & (x_i \leq x \leq x_{i+1}) \\ y = k_n \cdot x + b_n & (x_0 \leq x \leq x_n) \end{cases} \tag{1}$$

B. Efficient Data Indexing and Retrieval

1) Metadata Publishing and Querying: To publish a file, the file host hashes the keywords of the file. The keywords, denoted as k, can be file names or the words with the highest frequency or semantic tags. Two different Locality Sensitive Hash Functions (LSH) [36], H_1 and H_2, are used for the hashing. The resultant hash values $(H_1(k), H_2(k))$ are normalized to a *virtual* vector (x_k, y_k), which is used as the

ID of the file. The metadata is mapped to a region in the map that contains the virtual (x_k, y_k). The ID of the mapped region is the ID of the physical region that will keep the metadata.

After locating the destination region, the data host then publishes the metadata to the destination region based on RGR algorithm. The metadata includes the IDs of the destination region, current region of the file holder as well as the keywords of the file. The routing algorithm will be introduced in Section 2. When the metadata reaches a node in the destination region, the node stores the metadata and broadcasts the metadata to all other nodes in this region. LSH [36] is an algorithm for solving approximate and exact near neighbor search in high dimensional spaces. If two keywords are similar, they will be hashed to close values with high probability by LSH. For example, two keywords “computer network” and “computer communication network” may be hashed to the same region. Thus, metadata with similar keywords will be stored into the same region.

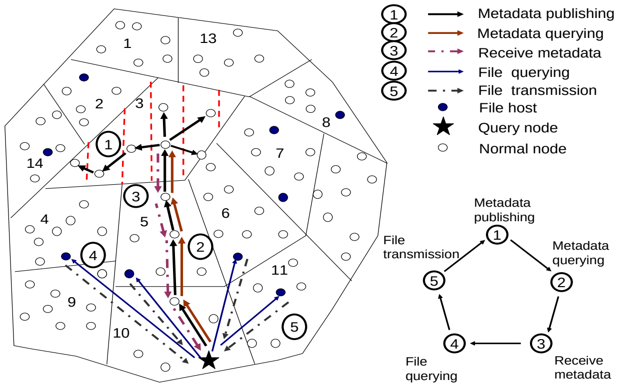

Fig. 1. The publishing and querying path in LORD.

In a dynamic MANET, it is important to maintain the mapping between data and regions by *mapping update*. In LORD, when a node moves from a region to another region, it retrieves all the metadata of the new region and drops the metadata of the old region reactively. In the mean time, the node updates its current new residing region to all of its published metadata. Recall that the metadata serves as the location-tracking server in LORD, which stores the locating region of its publishers.

When a mobile node wants to query a file, using the same process as publishing, the requester gets the vector ID (x, y) of the metadata of the file and sends a metadata request to the destination region containing (x, y). The request message also contains a similarity requirement of the result. The similarity threshold indicates the similarity degree of the requester's desired keywords with the queried keywords. The similarity between the keywords of a file k and the queried keywords k_q is calculated as $\frac{|N(k_q \bigcap k)|}{N(k_q)}$, where $N(k)$ denotes the number of keywords in k. According to the ID, the query message is routed to a region storing the desired metadata. Only the metadata sharing the similarity with the queried keywords no less than the threshold will be sent back to the requester.

2) Comparative Performance Analysis: This metadata publishing and querying algorithm has three distinctive features. First, unlike other works that only offer exact keyword searching, the LSH-based regional data publishing and retrieval enable similarity data searching in MANETs. Second, rather than relying on a proactive mapping update, LORD uses a region-based reactive mapping update to ensure the querying success by other nodes. Third, unlike location-based data sharing system [13, 15, 16] that maps a file to a location and stores the file to a single node closest to that location, LORD maps a file to a region for load balance and easy file consistency maintenance.

Specifically, in a mobility resilient location-based system, the home node of a file (i.e. the node that stores the file) periodically sends out updating messages inquiring the current closest node to the file's mapped location. The message is circulated among the neighbors of the home node. We use n to denote the number of nodes in the MANET, d to denote the average diameter of a region, and v to denote the average moving speed of a node. The average update frequency in LORD, denoted by f_R, is the number of times that a node drops old metadata and receives new metadata during a unit period. The average update frequency in location-based data sharing systems, denoted by f_L, is the number of times a home node initiates updating messages for location mapping updates. Mapping updates is used to ensure the data are stored in the correct nodes. Proposition 3.1 and its proof show that LORD generates less mapping updates overhead than a location-based data sharing system. Moreover, multiple nodes storing metadata increases LORD's mobility resilience, and avoids query congestion and single point of failure. For a system with the nodes of diverse mobility, the reactive location mapping updates reduce the query failure uncertainty.

Proposition 3.1: To guarantee a successful data querying, location-based data sharing systems need higher mapping update frequency than a region-based LORD data sharing system.

Proof: In LORD, the average update frequency is $f_R = v/d$ in a unit of time. In a location-based data sharing system, suppose a node is currently the closest node to location (x, y), T is a random variable that denotes the time interval in which another node moves to a location that is closer to (x, y). That is, an update frequency as $\frac{1}{T}$ can guarantee an accurate mapping update. T conforms to the exponential distribution [37] represented as $T \sim \text{Exponential}(f_L)$. The expected value is $E(\frac{1}{T}) = f_L$. Then $P(T < t) = e^{-f_L \cdot t} \Rightarrow t = \frac{1}{f_L} \cdot log^{\frac{1}{P(T<t)}}$. Therefore, a high update frequency is needed to achieve successful data query and $\lim_{p \to 1} t = 0$. The formula indicates that in order to guarantee that no query will be lost due to mis-mapping, the updating frequency of location-based mapping is infinite. However, the formula also shows that in a comparative static scenario where the lower bound of f_L is $inf(f_L) = 0$ and the upper bound of t is $sup(t) = 1$. It shows location based mapping works well in a stable scenario. ■

LORD is also characterized by metadata storage instead of data storage. Most current wireless data sharing systems use data storage. Admittedly, data storage methods avoid one more file query after locating the file owners in metadata storage methods. However, in a highly dynamic MANET, metadata storage produces more benefits than data storage. Since the overhead caused by a message transmission is directly related to the message size and the distance the message travelled, we use the product of the two factors to represent the overhead denoted by C.

Theorem 3.2: In a highly dynamic MANET, a LORD metadata-based data sharing system generates less update overhead than a data-based data sharing system.

Proof: We use S_d, S_m and S_q to denote the average size of a file, a file's metadata and query message respectively. Assume the average data transmission path length is $O(\sqrt{n})$. Therefore, the total overhead for one data querying operation during T is $C_d = \frac{T\cdot v}{d} \cdot S_f + (S_f + S_q) \cdot \sqrt{n}$ in a data-based system and is $C_m = \frac{T\cdot v}{d} \cdot S_m + (S_f + S_m + 2S_q) \cdot \sqrt{n}$ in a metadata-based system. Since $S_f \gg S_m \gg S_q$ and $v > 0$, $C_d - C_m = \frac{T\cdot v}{d}(S_f - S_m) - (S_m + S_q)\sqrt{n} \approx \frac{T\cdot v\cdot S_f}{d} > 0$. ■

3) Parallel Data Transmission Algorithm: After receiving the metadata of the queried file, the requester knows the region of the file holders. It then sends a file request message to the nodes in the nearest regions according to the region ID indicated in the metadata. In order to reduce data transmission latency, a parallel transmission algorithm is used. The data requester chooses geographically close data hosts among the located data hosts, and asks each data host to transmit a portion of the data. The data segments are transmitted simultaneously, which reduces data transmission latency. A question is how to determine the length of a file segment to minimize the transmission latency.

We use V to denote the expected channel propagation rate, W to denote the expected channel transmission rate, and $\overline{d}$ to denote the expected distance between two routing hops. The expected value can be calculated based on empirical data. We assume the total length of a file is L, and the number of selected data hosts is m. We use L_i, T_i and d_i to denote the length of the file segment, transmission latency, and the distance to the requester of the selection node $i(1 \leq i \leq m)$. Therefore, the average number of hops between two nodes with distance d_i is $\frac{d_i}{\overline{d}}$. Then, the expected latency for data segment transmission is $\frac{Li}{W} \cdot \frac{d_i}{\overline{d}} + \frac{d_i}{V}$. Since V is much larger than W, $\frac{d_i}{V} \approx 0$. Thus $T_i = \frac{L_i}{W} \cdot \frac{d_i}{\overline{d}}$. To minimize the file transmission latency, we desire $T_1 = T_2 = T_3 .... = T_m$, assuming there are m segments. Therefore, $L_i \cdot d_i = L_{i+1} \cdot d_{i+1}, i \in [0, m]$. Since $L_1 + L_2 + ...L_m = L$, we can get

$$L_i = \frac{L}{(\frac{1}{d_1} + \frac{1}{d_2} + \frac{1}{d_3} + ... + \frac{1}{d_n}) \cdot d_i}$$

According to the formula, a requester can determine the length of a file segment transmitted by a data host based on its distance from the requester.

C. Region-based Geographic Routing

To ensure a reliable, economical, and scalable locality-based routing, we propose a Region-based Geographic Routing algorithm (RGR) for node communication in the LORD. In contrast to the traditional region-based routing, RGR does not need exact location information. RGR consists of two components: inter-regional geographic routing and intra-region reactive routing. Inter-regional geographic routing forwards messages to the destination region based on inter-region direction. Intra-regional routing is further divided into two situations. For metadata publication, the metadata message is broadcasted to the nodes in the region, while an AODV-based routing algorithm is used for the transmission of the actual data within the region.

Since each node in the system can only sense the moving direction of its neighbor nodes, in order to route the messages with a comparative shorter path to the destination region, the message holders always choose the next hop node within a certain direction range. More specifically, each pair of regions in LORD has left-side and right-side angle ranges determined by the region boundary. The left-side angle range is the angle between the left most vertex of the region to the leftmost and rightmost vertices of the other region. Similarly, the right-side angle range is the right most vertex of a region to the leftmost and rightmost vertices of the other region. For example, in Figure 2, region 10 and region 3 have left-side angle range $[\alpha, \beta]$ and right-side angle range $[\theta, \tau]$. These two angle ranges serve as tight bounds of message transmission direction towards the destination region. For example, assume a node in region 10 intends to forward a packet to region 3. If the transmitting node stays at the left side of the it's region landmark, it chooses the next hop node within $[\alpha, \beta]$. If the transmitting node stays at the right side of the it's region landmark, it chooses the next hop node within direction between $[\theta, \tau]$. The messages are always forwarded closely towards to the destination region. When the nodes exchange "Hello" message, the transmission signal strength from these neighbor nodes can be detected. The farthest node is chosen as the next hop node. The forwarding processes are repeated until the packets are forwarded to a node that has neighbors in the destination region. In this situation, the node in the region with least query load will be chosen as the next hop. When the message arrives in the destination region, the message will be either piggybacked in the *Hello* message and broadcasted to all nodes in this region if it is in the metadata storage stage, or forwarded to the destination node based on the ADOV algorithm if it is in the data transmission stage. *Hello* message is periodically sent out by each node in a MANET for neighbor identification. RGR algorithm only uses AOA [35] devices for the angle identification. Therefore, RGR does not need location information, and it can be applied in indoor scenario that GPS cannot be used, which reflects the energy efficiency and routing flexibility features of LORD system.

Back-tracking Algorithm A data requester incorporates the ID of the region (i.e. source region ID) in the request message when it generates requests for querying metadata or data. After a node receives the request message, it sends the required metadata or data back to the requester based on the RGR algorithm. Because of the high dynamism feature of MANETs,

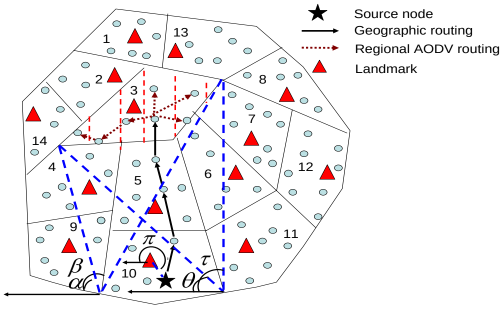

Fig. 2. Region-based geographic routing in LORD

the requester may move out of its region or even travel a number of regions before the response arrives at the source region. LORD has a back-tracking algorithm to keep track of the movement of the requester. In the algorithm, if a requester moves out of its current region during the time after sending out a query message and before receiving response, it sends a back-tracking message to the source region. The message indicates the node's current region. This message is piggybacked in the “Hello” message between neighbor nodes. Thus, each node in the source region keeps a back-tracking message of the requester. Using this message, the response can be forwarded to the requester after it moves out of the source region.

Proposition 3.3: In LORD, a response can always reach the data requester with $O(\sqrt{n})$ amount of overhead.

Proof: Suppose the area of a MANET is a plane with k regions, and the n nodes are independent and identically-distributed (i.i.d) in the plane. Therefore, the average number of nodes in a region is n/k. The overhead of regional flooding is $O(n/k)$. The overhead for the routing of metadata query is $O(\sqrt{n})$ and the query response to the source region is $O(\sqrt{n})$. If requester i doesn't move out of the source region, the total overhead for a metadata query is $O(2\sqrt{n})+O(n/k)=O(\sqrt{N})$ assuming $k=O(\sqrt{n})$. If node i moves out of the source region, the query overhead is $O(4\sqrt{n})=O(\sqrt{n})$. It includes the overhead for flooding a back-tracking message $O(\sqrt{n})$, for query flooding in the new region $O(\sqrt{n})$ and for query and response transmission $O(2\sqrt{n})$. ■

Theorem 3.4: The probability of the query node moving out of the current region to a neighbor region before receiving the metadata reply packet is

$$\frac{\frac{\lambda_d t^{vt}}{(vt)!}\cdot e^{-\lambda_d t}}{\pi}(D\cdot\arccos\frac{D}{vt}-\sqrt{(vt)^2-D^2}+vt),$$

where t is the time period between sending out a query message and receiving a reply message, v is the average moving speed of a nodes, D is the distance between the node and its region boundary, and λ_d^{-1} is the average distance from its position during time t to its original position.

Proof: We assume the basic region of LORD is a grid for computation convenience. Since the movement of each node in the system is i.i.d., the distance from the position of the node at time t to its original position conforms to a Poison distribution, then:

$$P(X=vt)=\frac{\lambda_d t^{(vt)}}{(vt)!}\cdot e^{-\lambda_d t}$$

Meanwhile, the probability that the nodes will stay out of the region at time t is $P(Y=S_d)=\frac{\arccos\frac{d}{vt}}{\pi}$ where S_d denote the event the node stay out of the region. Therefore, the probability that the node moves out of the region from shaded region is

$$\begin{aligned}&P(Y=S_d|X=vt)P(X=vt)\\&=\int_0^d\frac{\arccos\frac{x}{vt}}{\pi}\cdot\frac{\lambda_d t^{vt}}{(vt)!}\cdot e^{-\lambda_d t}d(x)\\&=\frac{\frac{\lambda_d t^{vt}}{(vt)!}\cdot e^{-\lambda_d t}}{\pi}(d\cdot\arccos\frac{d}{vt}-\sqrt{(vt)^2-d^2}+vt)\end{aligned}$$

■

Theorem 3.4 indicates that the probability that a node moves out of its region increases with t. Suppose the transmission delay is 1s, the average moving speed of the nodes is 10m/s, the diameter of the region is 200m, then the probability that the node moves out of the source region before receiving the reply message is 5%.

IV. PERFORMANCE EVALUATION

The NS-2 simulator is limited to a system size of hundreds of nodes. In order to simulate a high density network, we conducted simulation on an event-driven simulator ONE [38]. We evaluated the performance of LORD in comparison with GHT [13] and GLS [12], which are the representative locality-based data sharing protocols. We also tested the performance of LORD with AODV [8] topological routing rather than RGR. We use AODV to denote this data sharing system. In GLS, the entire geographic area is recursively divided into a hierarchy of increasingly smaller squares. A node's files and location are mapped to several home nodes in a number of squares based on their virtual IDs. A message is routed based on the virtual node IDs, and geographic routing is employed in each routing step. GLS has an updating distance. If the distance that a node moves reaches the updating distance, it notifies its home node for information update. As in [12], we set the updating distance of GLS to 50m. As in [13], in the experiment, we set the updating interval of GHT to 2s.

In the experiments, all nodes in the simulation move within a 2200m*2200m grid. The packet transmission speed of nodes was set to 250kbit/s. All the nodes move with their certain pattern in the system [39] with 0 pause time, in which three categories of movement speeds are selected uniformly at random within [0.5-2.5]m/s, [1-5]m/s and [20-30]m/s respectively to represent of movement of walkers, bikers and cars in real life. The ratio of the number of nodes in the three groups of nodes was initially set to 4:3:3. The number of nodes was set to 1000. 400 files were randomly assigned to the nodes initially. We set the transmission range of each mobile node to 150m, and set the size of a message to 2kb. The simulation time was set to 400s, and the warm up time was set to 100s. In the experiments, every message was transmitted once without retransmission in order to evaluate the data delivery ability of each topology.

We evaluate the performance of LORD with the following metrics:

- Query success rate. It is the ratio of the number of queried files received by a data requester to the number of file queries initiated by the requester. This metric represents the performance of a data sharing system in terms of successful data retrieval.
- Query path length. It is the number of hops for routing a data query. This metric represents the efficiency of a routing protocol.
- The number of messages. It is the total number of messages for querying, replying and mapping updating. This metric represents the overhead generated in a data sharing system.
- Query/reply hop ratio. It is the ratio of the path length of query to the path length of reply. This metric shows the efficiency of routing algorithm.
- The number of queries received. It is the total number of queries received by nodes in different areas with in a region. This metric is used to show the load balance of LORD by comparing the number of queries received by nodes in different area within a region.

A. *Scalability*

We varied the number of nodes in a network and measured the query success rate versus the number of nodes in different data sharing systems. Figure 3 shows that the query success rate of each system decreases as the network size increases. It is because more nodes in the network lead to higher channel contention which results in more message loss. The figure also shows that LORD leads to the highest success rate, GHT produces higher success rate than GLS, and AODV results in the lowest success rate. In a medium-size network with no more than 1000 nodes, GLS and LORD lead to almost 100% query success rate. Recall that GHT stores a file in the node closest to the file's location. Thus, a file's home node is always changing with node mobility in a dynamic MANET. If the mapping update rate is slow, GHT cannot maintain the correct mapping between files and nodes in high node mobility. As a result, a file query arrives at its new home node before the file is transferred to the new home node. Therefore, GHT has lower query success rate than LORD. GLS copes with node mobility by requiring a node to update its location information in all its home nodes when it moves out of a square. Therefore, its home nodes can always keep track of the movement of the node. These location update messages result in more load for data transmission, exacerbating traffic congestion. Therefore, GLS generates lower success rate. LORD also uses a back-tracking algorithm for location update in which a node reports its current region to its source region when it moves out, but it occurs only after a node has not received its queried file before leaving. In addition, as indicated by Theorem 3.4, the probability that a node moves to another region during the time after it sends a query and before it receives response is small. Therefore, the back-tracking algorithm in LORD does not lead to high overhead. It is also obvious to note that, with the increase of nodes in the system, the query success rate of AODV drops sharply. This is due to the flooding based on-demand routing in AODV's. When a route from a source to the destination is determined, it is likely that a node in the observed route moves before a message is forwarded to it, resulting in routing failure and hence query failure. Moreover, flooding makes the congestion situation worse, leading to more message drops. The results show that LORD has the highest query success rate, and it can maintain the high success rate in a large-scale MANET.

Figure 4 shows the average path length for metadata/data querying versus the network size in different data sharing systems. The results show that GLS leads to the longest hop length. This is because GLS's routing is based on node virtual ID. in a hierarchical structure. This leads to a longer travel path to the destination. In addition, the virtual next hop in routing may not be the geographically closest node. Moreover, unlike other systems, the path length of GLS increases rapidly as the network size grows. It means that relying on an overlay for routing, GLS is not able to achieve high scalability. Relying on geographic routing, which forwards a message to a node geographically closer to the destination in each step, GHT achieves much shorter path length than GLS. Since the RGR used in LORD also forwards a message in the direction of the destination, it is intriguing to see that LORD has slightly shorter path length than GLS. This is because the metadata in LORD are maintained by a group of nodes, and the first node in the destination region receiving a query message replies to the data requester. In GHT, after the message reaches the destination area, it will be further forwarded to the destination node. The results also show that the path length of AODV is shorter than GHT but marginally longer than LORD. It is because, rather than greedily transmitting a message to the node closest to the destination as others, AODV chooses the path with the shortest transmission latency, which is not necessarily the path with the smallest number of hops.

These results imply that LORD can achieve comparable data transmission efficiency and scalability compared to geographic routing without consuming high energy with the use of GPS.

B. *Mobility Resilience*

This experiment shows the performance of the different data sharing systems in a dynamic environment. Figure 5 shows the query success rate versus node average moving speed. The results show that LORD and GLS exhibits approximately the same mobility-resilience ability. GHT leads to much lower success rate when nodes move in a faster speed, and AODV produces significantly lower success rate than others. To ensure a file or metadata is stored in the correct mapped node based on mapping updates plays an important role to guarantee the successfulness of query in node mobility. In LORD, a metadata host drops its metadata before moving

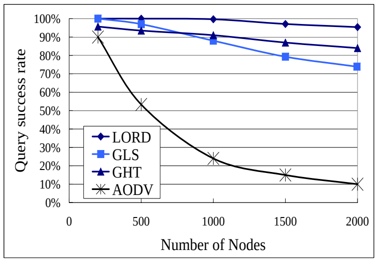

Fig. 3. Success rate versus network size

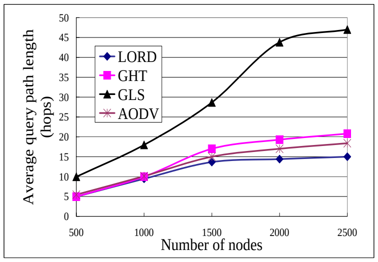

Fig. 4. Path length versus network size

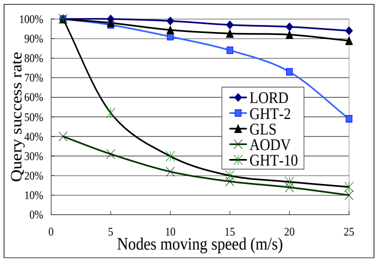

Fig. 5. Success rate versus mobility rate

out of its region and receives new metadata when moving into a new region. Therefore, the mapping update occurs only when a node moves out of a region. In addition, the back-tracking algorithm further helps to guarantee that data can be forwarded to the requester. In GLS, when the distance that a node moves reaches a pre-defined threshold, it will update its location information by sending update messages to its home hosts. In GHT, the mapping update is executed periodically. Fast mapping update helps to guarantee high success rate but at a high cost of overhead, while slow mapping update may fail to ensure the query successfulness. In the experiment, GHT-2 and GHT-10 are used to denote GHT with mapping update interval 2s and 10s respectively. Although with the increasing movement speed of nodes, the periodical mapping update cannot guarantee that a file is always stored in the node closest to the file's location, which leads to a sharp decrease in success rate of both GHT-2 and GHT-10. However, since GHT-2 has a higher update rate than GHT-10, the performance of GHT-2 is better. The MANET is more likely to be partitioned with faster node moving speed. Therefore, AODV topological routing results in dropped messages in a highly dynamic environment.

The next experiment evaluated the message overhead including request and reply messages and location mapping update overhead. Mapping update overhead is measured by the number of mapping update messages versus the node movement speed. The results are collected in the first 100s of the experiment. The results in figure 6 show that update overhead of GHT remains constant regardless of the node movement speed. GHT-2 generates the highest mapping update overhead than GHT-10 because of slower updating frequency. A short update interval of GHT leads to more updating messages. However, Figure 5 shows that a 2s updating interval still leads to a lower query success rate than GLS and LORD. Hence, GHT is not appropriate in a highly dynamic environment due to its high mapping update overhead and low query success rate. The figure also demonstrates that GLS generates higher mapping update overhead than LORD, especially in a highly dynamic environment. This is because a node has a number of home nodes, which store its location in GHT. Thus, a number of updating messages are needed for one mapping update. In LORD, a node receives new metadata and drops old metadata when it moves from one region to another. In addition, a data requester generates one location update message before it leaves its source region if it has not received the response. Therefore, LORD generates less update messages than GLS.

C. *Routing Efficiency*

Figure 7 shows the querying/replying hop rate. The result shows that the query/reply hop ratio of GHT and LORD are slightly less than 1. GPS-based GHT has high routing efficiency because its geographic routing can always forward data along geographically short path to the destination with the accurate location information offered by GPS. Thus, it hardly generates detour routing, resulting in query/reply hop ratio close to 1. The results imply that the efficiency and effectiveness of the RGR algorithm in LORD is comparable to GHT, but it consumes much less energy without depending on GPS. The reason that both query/reply hop ratios are less than 1 is because after sending out a query message, the requester may move away from the original position, which increases the routing path length of the reply message. We also observe that the query/reply hop ratio of LORD is slightly larger than GHT. This is because for the same destination location, the path of LORD is shorter since the querying process is completed when the query message meets a node in the destination region. The figure also shows that the query/reply hop ratio of GLS is much larger than 1. It is because the query routing in GLS is based on virtual ID, and the path is not graphically short path due to the mismatch between the overlay layer and physical layer. Since the destination node knows the position of the requester, it can send the response message back to the requester directly. Therefore, the response path length should be the shortest geographically short path, leading to rate larger than 1.

D. *Load Balance*

Recall that in LORD, all nodes in a region have the same metadata. If a node is overloaded when receiving a data query, it forwards the query to a lightly loaded node in its region. This experiment tests the performance of LOAD in achieving load balance and avoiding overloaded nodes. A geographic region of a MANET consists of a number of concentric circles with different radius, and the total number of queries received by the nodes along the different circles is recorded. In the experiment, 40 query messages were sent to the region at the same time from different directions. Figure 8 shows that the total number of queries received by nodes in different circles with different radius increases linearly with the increase of the radius. The results indicate load distribution is balanced

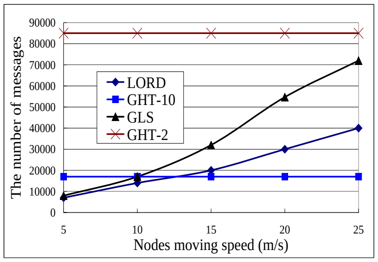

Fig. 6. Protocol message versus mobility rate

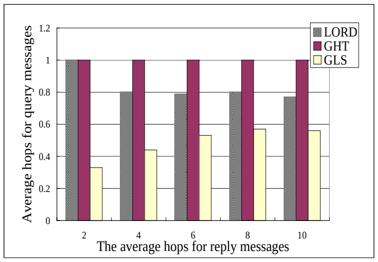

Fig. 7. Query/reply ratio versus reply hops

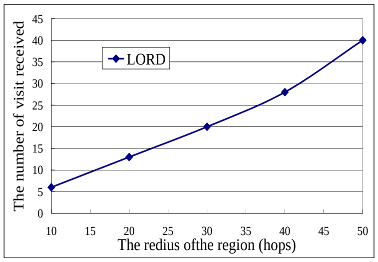

Fig. 8. Load balance

in different circles in the region. Since the mobile nodes are uniform distributed in the region, the load to individual nodes should also be uniform distributed. This result demonstrates the advantage of redundant metadata replicas, which enables to queries request to be distributed to different nodes in one region. Though redundant metadata generates more maintenance overhead, it helps to avoid overloading nodes and enhances mobility resilience.

V. CONCLUSIONS

With the advancements in data management system and wireless technology over the past decade, traditional wireless data sharing based on infrastructure wireless networks cannot achieve higher scalability due to their centralized control. Current decentralized wireless data sharing systems rely either on topological routing or geographic routing. The former fails to achieve high scalability due to flooding-based routing while the latter is not resilient to high node mobility with high overhead and energy consumption. In this paper, we propose a LOcality-based distRibuted Data sharing system (LORD) for large-scale highly dynamic MANETs. LORD consists of a data index and retrieval protocol and region-based geographic routing protocol. LORD divides the MANET area into physical regions. Without flooding, it maps the metadata of similar files to the same region for similarity data retrieval. Without relying on GPS or virtual coordinates as in traditional geographic routing, LORD region-based geographic routing forwards data in the direction of its destination with much less cost. Back-tracking algorithm further enhances the successfulness of data querying. LORD provides highly efficient, scalable and dynamism-resilient data sharing. Theoretical and experiment results show the superiority of LORD over other data sharing systems in terms of scalability, efficiency and mobility-resilience. Its distinguishing features are particularly attractive to the deployment of highly dense and dynamic MANETs.

ACKNOWLEDGMENTS

The authors are grateful to the anonymous reviewers for their valuable comments and suggestions. This research was supported in part by U.S. NSF grants CNS-0834592, CNS-0832109 and CNS-0917056.

REFERENCES

[1] List of video sharing sites and impact of video marketing. http://www.hybridsem.com/blog/2007/07/01/list-of-video-sharing-sites-and-impact-of-video-marketing/.

[2] S. S. Kulkarni and S. L. Ali. Redefining connectivity: Unifying mobile communication with the internet. In *Proc. of CCNC*, 2006.

[3] A. Leon-Garcia and I. Widjaja. Communication networks. *McGraw-Hill Science/Engineering/Math*, 2003.

[4] H. Y. Hsieh and R. Sivakumar. Performance comparison of cellular and multi-hop wireless networks: A quantitative study. In *Proc. of SIGMETRIC*, 2001.

[5] M. Papadopouli and H. Schulzrinne. Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices. In *Proc. of MobiHoc*, 2001.

[6] C. Lindemann and O. Waldhorst. A distributed search service for peer-to-peer file sharing in mobile applications. In *Proc. of P2P*, 2002.

[7] A. Klemm, C. Lindemann, and O. P. Waldhorst. A special purpose peer to peer file sharing system for mobile ad hoc networks. In *Proc. of VTC*, 2003.

[8] C. Perkins, E. Belding-Royer, and S. Das. RFC 3561: Ad hoc on demand distance vector (AODV) routing, 2003.

[9] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. *IEEE Mobile Computing*, 1996.

[10] T. Zahn and J. Schiller. Performance evaluation of a dht-based approach to resource discovery in mobile ad hoc networks. In *Proc. of IEEE/IFIP WONS*, 2007.

[11] S. M. Das, H. Pucha, and Y. C. Hu. Performance comparison of scalable location services for geographic ad hoc routing. In *Proc. of Infocom*, 2005.

[12] J. Li, J. Decouto, and R. Morris. A scalable location service for geograghic ad-hoc routing. In *Proc. of MobiCom*, 2000.

[13] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu. Data-centric storage in sensornet with ght: A geographic hash table. In *Proc. of MONET*, 2003.

[14] Y. Zhao, Y. Chen, and S. Ratnasamy. Load balanced and efficient hierarchical data-centric storage in snesor networks. In *Proc. of SECON*, 2008.

[15] X. Li, Y. J. Kim, R. Govindan, and W. Hong. Multi-

dimensional range queries in sensor networks. In *Proc. of ACM Sebsys*, 2003.

[16] B. Greenstein, D. Estrin, R. Govindan, S. Ratnasamy, and S. Shenker. Difs: A distributed index for features in sensor networks. In *Proc. of SNPA*, 2003.

[17] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica. Geographic routing without location information. In *Proc. of Mobicom*, 2003.

[18] A.Caruso, S. Chessa, S. De, and A. Urpi. GPS free coordinate assignment and routing in wireless sensor networks. In *Proc. of Infocom*, 2005.

[19] R. Fonseca, S. Ratnasamy, J. Zhao, and C. T. Ee. Beacon vector routing: scalable point-to-point routing in wireless sensornets. In *Proc. of NSDI*, 2005.

[20] S. S. Kulkarni and S. L. Ali. Minimizing energy for wireless web access with bounded slowdown. In *Proc. of Mobicom*, 2002.

[21] J. Li, C. Blake, D. J. Couto, H. I. Lee, and R. Morris. Capacity of wireless ad hoc networks. In *Proc. of Mobicom*, 2001.

[22] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Proc. of ACM SIGCOMM*, pages 329–350, 2001.

[23] I. Stoica, R. Morris, and et al. Chord: A scalable peer-to-peer lookup protocol for Internet applications. *TON*, 1(1):17–32, 2003.

[24] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, 2001.

[25] M. Caesar, M. Castro, G. Oshea E. Nightingale, and A. Rowstron. Virtual ring routing: Network routing inspired by dhts. In *Proc. of SIGCOMM*, 2006.

[26] H. pucha, S. M. Das, and Y. C. Hu. Ekta: An efficient dht substrate for distributed applications in mobile ad hoc networks. In *Proc. of WMCSA*, 2004.

[27] S. Datta, I. Stojmenovic, and J. Wu. Internal node and shortcut based routing with guaranteed delivery in wireless networks. In *Proc. of WNMC*, 2001.

[28] H. Frey and D. Gorgen. Planar graph routing on geographical clusters. *Ad hoc network*, 2004.

[29] Z. J. Haas, M. R. Pearlman, and P. Samar. The zone routing protocol (ZRP) for ad hoc networks. In *internet-Draft*, 2002.

[30] Z. Li and H. Shen. LORD: A locality-based distributed data sharing system for highly dense and dynamic manets. In *Proc. of MoViD*, 2009.

[31] M. Maroti, P. Volgyesi, S. Dora, B. Kusy, A. Nadas, A. Ledeczi, G. Balogh, and K. Molnar. Radio interferometric geolocation. In *Proc. of SenSys*, 2005.

[32] M. Rudafshani and S. Datta. Localization in wireless sensor networks. In *Proc. of IPSN*, 2007.

[33] D. Niculescu and B. Nash. Ad hoc positioning system (APS) using aoa. In *Proc. of Inforcom*, 2003.

[34] B. Kusy, A. Ledeczi, M. Maroti, and L. Meertens. Node density independent localization. In *Proc. of IPSN*, 2006.

[35] D. Niculescu and B. Nath. Ad hoc positioning system (aps) using aoa. In *Proc. of INFOCOM*, 2003.

[36] A. Andoni and P. Indyk. Near-optimal hashing algorithms for near neighbor problem in high dimensions. In *Proc. of FOCS06*, 2006.

[37] E. T. Jaynes and G. L. Bretthorst. Probability theory: The logic of science. *Cambridge University Press*, 2003.

[38] The "ONE" Simulator . http://www.netlab.tkk.fi/tutkimus/dtn/theone/.

[39] B. Xie, A. Kumar, and D. Cavalcanti. Mobility and routing management for heterogeneous multi-hop wireless networks. 2005.