Performance Evaluation of Information Retrieval Schemes for Multi-Attribute Queries in DTNs

M. Chuah, J. Han

Department of Computer Science & Engineering

Lehigh University
chuah@cse.lehigh.edu, jih206@lehigh.edu

***Abstract*—Mobile nodes in some challenging network scenarios suffer from intermittent connectivity and frequent partitions e.g. battlefield and disaster recovery scenarios. Disruption Tolerant Network (DTN) technologies are designed to enable nodes in such environments to communicate with one another. In an earlier work, we studied information retrieval schemes for single-attribute queries in DTNs. Our schemes disseminate replicated data copies and queries to local-neighborhood. However, data items often have multiple attributes and not every node can be trusted to store replicated data or queries. Thus, in this paper, we study the scenario where the queries have multiple attributes. In addition, we compare the effectiveness of using opportunistically encountered nodes or specially deployed index and storage points (ISPs) for storing replicated data items or indices of the data items. Specifically, we conduct extensive simulation studies to evaluate three information retrieval schemes namely (a) the Pre-determined ISP advertisement (PISA) scheme where ISPs advertise indices of data items with attribute values that fall within some pre-determined ranges, (b) the Opportunistic Regular Node Advertisement with Index Duplication (ORNA-ID) scheme which uses opportunistically encountered nodes for advertising replicated indices, and (c) the Opportunistic Regular Node Advertisement with Data Duplication (ORNA-DD) scheme which uses opportunistically encountered nodes for storing replicated data items. Our results indicate that the ORNA-ID scheme and the PISA scheme achieve similar performance. This shows that either architecture can be used. In addition, the ORNA-DD scheme provides 91% more query success rate and 128% more overall success rate in the sparsest network scenario (40 nodes distributed over 4000x4000m^2).**

***Index Terms*—disruption tolerant networking, content-based routing, multi-attribute queries, information retrieval.**

I. INTRODUCTION

With the advancement in technology, many users carry wireless computing devices e.g. PDAs, cell-phones etc. Such devices can form mobile ad hoc networks and communicate with one another via the help of intermediate nodes. Such ad hoc networks are very useful in several scenarios e.g. battlefield operations, vehicular ad hoc networks and disaster response scenarios. Many ad hoc routing schemes have been designed for ad hoc networks but such routing schemes are not useful in some challenging network scenarios where the nodes have intermittent connectivity and suffer frequent partitioning.

Disruption tolerant network technologies [1],[2] are designed to allow nodes in such extreme networking environment to communicate with one another. Several DTN routing schemes [3],[4] have been proposed.

Although routing design for such sparsely connected networks is important, it is equally important to design information retrieval schemes for extreme networking conditions which allow rapid and efficient information access. For example, in a battlefield, soldiers need to access information related to detailed geographical maps, intelligent information about enemy locations, new commands from the general, weather information etc. In addition, a particular data item may be of interests to multiple soldiers so replicating the data item at multiple nodes permits other nodes to access it more easily. This allows us to save battery power, bandwidth consumption and the data item retrieval time. Such data caching also means that the source of the data items need not know the identities of the nodes that need to access the data items.

Researches on data access and dissemination techniques in ad hoc and sensor networks are not new. For example in [7], the authors design schemes for placing the storage nodes such that the total energy cost for gathering data to the storage nodes and replying queries can be minimized. The approach in [7] is not directly applicable to our problem since they assume stationary nodes in their environments. They also do not consider how a node can discover the replicas of the data items. In [9], the authors propose three distributed caching techniques for well-connected ad hoc networks, namely CacheData, CachePath and HybridCache. However, their techniques are only useful for well-connected ad hoc networks since they assume the routes from the publisher to the receivers do not change frequently. In [5][17], we design and study information retrieval schemes for single-attribute queries. In [5],[17], we assume every node is trusted to store data or queries and we do not address multiple-attribute queries.

In this paper, we consider a system where we have queries with multiple attributes. In addition, a small number of moving index and storage point(ISP)s may be deployed to store replicated data and/or advertise indices of published data

978-1-4244-2677-5/08/$25.00

items. We compare such a system with another system that does not have ISPs but uses regular nodes encountered opportunistically to store replicated indices or data. Specifically, we present three information retrieval schemes, namely (a) the Pre-determined ISP advertisement (PISA) scheme where ISPs advertise indices of data items with attribute values that fall within different pre-determined ranges, (b) the ORNA-ID scheme which uses opportunistically encountered nodes for advertising indices, and (c) ORNA-DD scheme which uses opportunistically encountered nodes for storing replicated data. Then, via simulation studies, we compare the query performance of these three schemes. Our results indicate that the query success rates achieved by the PISA and the ORNA-ID schemes with the same number of replicated indices are similar. This indicates that either architecture can be deployed depending on the trust levels nodes have among themselves or with some specially deployed storage nodes. In addition, our results indicate that the ORNA-DD (that incorporates data duplication) scheme provides 91% gain in query success rate and 128% gain in the overall success rate in the sparsest network scenario which we studied. This shows that index duplication alone is not sufficient to achieve high query success rate in sparse network scenarios. Instead, data duplication needs to be used.

The rest of the paper is organized as follows. In Section II, we discuss related work. In Section III, we describe our system model. In Section IV, we present the various data, index and query replication techniques and two specific information retrieval schemes we study. In Section V, we describe our simulation model and present performance evaluation of the proposed schemes. In Section VI, we provide our concluding remarks and discuss future research.

II. RELATED WORK

In [7], the authors considers the storage node placement problem in sensor networks with the goal of minimizing the total energy cost for gathering data to the storage nodes and replying queries. The authors consider both fixed and dynamic tree models. For fixed tree model, they give an exact solution on how to place storage nodes to minimize total energy cost. For the dynamic tree model, they allow each sensor node to select the storage node for storage with respect to the minimal communication cost for data forwarding and query diffusion and reply once the storage nodes have been positioned.

In [9], the authors present three distributed caching schemes, namely (a) CacheData which caches data items that pass by a node, (b) CachePath which caches the path to the nearest cache, and (c) HybridCache which caches the data item if its size is small enough or lese the path to the data will be cached. LRU policy is used for cache replacement. These three approaches however do not take into considerations how nodes within a neighborhood can collaborate such that the data caching will always generate better benefits in terms of access cost. In addition, their approach also assumes well-connected adhoc networks.

In [8], the authors design a protocol called Mercury to support multi-attribute, range-based searches. There are two main components in Mercury: (a) Mercury handles multi-attribute queries by creating a routing hub for each attribute in the application schema. Each routing hub is a logical collection of nodes. Queries are passed to exactly one of the hubs corresponding to the attributes that are queried, while a new data item is sent to all hubs for which it has an associated attribute. (b) Each routing hub is organized into a circular overlay of nodes and data is placed contiguously on this ring. Their evaluation shows that Mercury is able to achieve logarithmic-hop routing and near-uniform load balancing. Mercury is designed for overlay networks with static nodes. Thus, its technique is not directly applicable to the network environments that we study. The PISA scheme we design and evaluate in this paper is an adapted version of the Mercury protocol for mobile environments.

In [15], the authors describe Snoggle, an implementation which uses information retrieval techniques to index information and process user queries. Their hierarchical system consists of indexing points (IPs), a key indexing point (KIP) and distributed object sensors. The IPs collect data from object sensors within their range and organize the data into an inverted index table. IPs periodically send aggregated information to the KIP. The KIP publishes the aggregated information forwarded by all IPs. The IPs also provide message routing for the traffic between IPs, KIP and users. A user can issue a local or distributed query. A user performs a local query by directing his query to a specific IP within the transmission range of his wireless device. A distributed query is performed when the user queries the KIP. The KIP then returns a list of IPs that may have the data items that the user is interested in. The user can then query the individual IPs. Our approach in this paper differs from theirs in multiple ways: (a) our architecture is distributed such that it can still work without the indexing point, (b) our system uses replicated data or indices to improve query performance, (c) users can query data-generating nodes directly.

III. SYSTEM MODEL

In this work, we assume that there are two types of network entities: (a) regular nodes which are wireless devices e.g. PDAs and cell-phones carried by individuals, (b) indexing and storage points (ISPs) which are special moving nodes which run server modules that support content-based information retrievals. These ISPs may be wireless routers mounted on moving vehicles e.g. campus buses [4]. All the nodes (either ISPs or regular nodes) support communications using disruption tolerant networking technology [1],[2]. Users can subscribe to the information retrieval services via the traditional subscription procedures similar to ordering Internet services. The users' wireless devices will run the client modules that support content-based information retrievals.

We further assume that users can publish their data items e.g. region-based weather, sensor data readings, enemy intelligent information for others to query. The data items can be labeled using some meta-data descriptions. For example, a data item describing enemy intelligent information may take the form: (<category= "Al-Qaeda.GRPlocation">,<clearance, ">", 5 >, encrypted data). This data item basically describes the location of an Al-Qaeda group that only soldiers with a clearance higher than 5 can access. The nodes that publish data items are referred to as the data nodes. The published data items can be stored at regular nodes or at the ISPs. Any publisher can store his data items at some designated ISPs based on the categories the data items belong to or store his data items at any ISPs which he encounters opportunistically. In addition, a publisher can choose to replicate his published data items to increase the chances of letting others find them.

Any authorized user can send a query to retrieve data items of interests to him. A query may take the form: Query(<category="Al-Qaeda.GRPLocation",<clearance,">", 3>, expired:20:00). This query looks for the locations of all Al-Qaeda groups that can be accessed with a clearance level above 3. The query expires at a certain time. Since the nodes in the network scenarios we are considering are frequently partitioned, a user can choose to duplicate his query to increase his chances of getting a reply before his query expires. The nodes that generate the queries are referred to as the querying nodes.

There are four important components in our information retrieval system: (a) the data caching component is related to how and where the published data items are stored, (b) the index advertising component is related to how published data is advertised, and (c) the query dissemination component is related to how queries are disseminated, and (d) the message routing component which determines how a response is routed to a querying node. We will explain the approaches used for different components in Section IV.

IV. DATA/INDEX/QUERY DISSEMINATION SCHEMES

In this work, we assume that each data item has two attributes denoted as (x,y) and each of the attribute has a value that falls within a range [dmin,dmax]. Each query has a query identifier and an expiration time. Each data also has a data identifier, some meta-descriptions, and an expiration time. Each data-generating node maintains two bloom filters: one for attribute 1, and one for attribute 2. In our networks, every node sends beacons periodically for node discovery. When a data node or ISP hears the beacon of a new node, it sends an index advertisement (IA) message that contains its bloom filters information. That way, any node knows if the other node (which sent the IA) it encounters has any data items of interest to itself.

A. Data Caching

A publisher can (a) store published data items only in his own devices but let ISPs advertise the meta-data descriptions (referred to as index) of his published data items, or (b) select other nodes to store replicated copies of his published data items. Our earlier work [5],[17] has indicated that data replication helps to improve the success rate of single-attribute queries with expiration time.

If more than two replicated copies of each data item is generated, then the binary spread [11][17] method can be used to quickly disseminate all the replicated copies. A publisher can choose nodes which move faster to store his replicated data items since such nodes can meet more nodes. A friendliness metric which is a moving average of the number of new nodes encountered by a node within an observation window is used to identify nodes with higher mobility. The nodes that can meet more nodes have a higher FM value. The FM metric can be included in the beacons periodically sent out by a node. When a publisher's wireless device encounters another node, it can tell whether the other node has a FM value that exceeds a certain threshold. If it does, then the publisher will send his replicated data copies to that node. If the FM metric is not supported, then a publisher will repeatedly share half of his replicated copies to any node that his wireless device encounters until he only has one copy left.

B. Index Advertising

A publisher may decide to ask other nodes to advertise the data items that he publishes. There are two ways whereby this can be done: (a) Pre-determined ISP Advertisement (PISA), (b) Opportunistic Regular Node Advertisement (ORNA).

In PISA, when a publisher generates a data item, it will send meta-data descriptions of this data item to some pre-determined ISPs. Each ISP is chosen to advertise indices of the data items with their first or second attribute values falling into a particular range as in the Mercury [8] approach. For example, assume that dmin=1,dmax=8 for both attributes of a data item. *ISP1* advertises indices of data items with values of their first attribute falling within the range [1,4] or values of their second attribute falling within the range [5-8], *ISP2* advertises indices of data items with values of their first attribute falling within the range [5-8], or values of their second attribute falling within the range [5-8], *ISP3* advertises indices of data items with values of their first attribute falling within the range [1-4] or values of their second attribute falling within the range [1-4], and *ISP4* advertises indices of data items with the values of their first attribute within the range [5-8] or values of their second attribute within the range [1-4]. Then, when a publisher generates a data item, it will send notification messages to three ISPs which take care of the range its attribute values fall into. For example, if a data item has attribute values (1,5), then notification messages will be sent to *ISP1, ISP2*, and *ISP3*.

In ORNA, a publisher generates M tokens for each index of a newly generated data items and binary spread them to nodes that the publisher encounters. That means, each node that has an index with more than one token is allowed to pass half of the tokens to another node that it encounters until it only has an index with one token.

C. Query Dissemination

A node, n_1, which is interested in data items with certain attribute values e.g. (3,6) does not know where such items are stored if the ORNA approach is used. There are two ways whereby node n_1 can find such data items: (a) since each data node broadcasts its index advertisement, any node that can hear such an advertisement can determine if the other node has data items of interest to itself, (b) if n_1 encounters another node that advertises indices, it can query that node if its advertised indices match with pending queries which n_1 stores. A response message that contains the data node identifier will be sent to the querying node. Then, the querying node can directly retrieve the data items from the relevant data-generating nodes. In the PISA scheme, a querying node knows which ISP to enquire for the information of data nodes with matching data items. Upon receiving the information from the ISP, the querying node can proceed to retrieve the relevant data items. In addition, a node also can retrieves data items from a data node that it encounters.

To allow more queries to successfully retrieve data items, a node can decide to replicate its query. Our early work [5],[17] has shown that query duplication also helps in improving the query success rate and reduces the query response time. If a query is replicated and stored at multiple nodes, then multiple query responses may be generated. Such redundant responses cause the transmission efficiency of the system to be low. To increase the transmission efficiency of the information retrieval system, each node can cache the identifiers of the query responses it has generated so that it does not relay any redundant query responses.

D. DTN Message Routing Scheme

Once a query response is generated by a node, the query response will be delivered to the querying node using the underlying DTN message routing scheme. In this paper, we use Prophet [3] as our DTN message routing scheme.

Prophet uses the history of encounters and transitivity to route messages for intermittently connected networks. In this scheme, each node broadcasts a beacon periodically. This probabilistic routing scheme establishes a probabilistic metric called delivery predictability at every node A for each known destination B. This metric indicates how likely it is that node A will be able to deliver a message to that destination. The delivery predictability ages with time and also has a transitive property, i.e., a node A that encounters node B which encounters node C allows node A to update its delivery predictability to node C based on its (A's) delivery predictability to node B and node B's delivery predictability to node C. In Prophet, a node will forward a message to another node it encounters if that node has higher delivery predictability to the destination than itself. Such a scheme was shown to produce superior performance than epidemic routing [10]. The three equations used for updating the delivery predictability are as follow:

$$P(a,b) = P(a,b)_{old} + (1 - P(a,b)_{old}) * \alpha \qquad \text{----- Eq(2a)}$$

$$P(a,b) = P(a,b)_{old} \times \gamma^{k} \qquad \text{----- Eq(2b)}$$

$$P(a,c) = P(a,c)_{old} + (1 - P(a,c)_{old}) * P(a,b) * P(b,c) * \beta \qquad \text{--Eq(2c)}$$

In [3], α is set to 0.75, β is set to 0.25 and γ is set to 0.98.

When a node receives a message that needs to be relayed, it will pass it to another node which has the highest delivery predictability to the destination.

E. Three Information Retrieval Schemes

In this work, we specifically consider three schemes: (a) Pre-determined ISP Advertisement (PISA) scheme, (b) Opportunistic Regular Node Advertisement with Index Duplication (ORNA-ID) scheme, and (c) Opportunistic Regular Node Advertisement with Data Duplication (ORNA-DD).

In PISA, only index duplication is used but not data or query replication. In PISA, ISPs are used to advertise indices of all generated data items. In addition, data generating nodes also advertise the indices of all data items they publish. Every time a data generating node generates a data item, it will send meta-descriptions of that data item to three pre-determined ISPs. A data node will advertise its aggregated index to any newly discovered neighbor. When a node n_1 encounters a data-generating node, n_2, n_1 can determine from the index advertisement sent by n_2 whether n_2 carries any data items that match the stored queries n_1 has. If n_2 has matching data items, then n_1 will retrieve such items from n_2. When a node n_1 generates a query, it will send the query to the closest ISP that contains the index information of data items that match this query. Upon receiving the reply from the ISP, node n_1 will send queries to the appropriate data-generating nodes to retrieve the data items of interest.

Index duplication but not query or data replication is used in the ORNA-ID scheme. In this scheme, there is no ISP. Regular nodes are used to store indices of published data items and/or queries that have not expired yet. Every node that generates a data item creates M copies of an index, and binary-spray these M copies of the index to nodes that they encounter. Every node that has more than one copy of the index can give half of the copies it has to any node that it encounters that does not have this index yet. Binary spraying is used because this is the optimal way of spreading extra copies of the same message to as many nodes as possible in sparsely connected networks. When a node encounters another node, they will exchange their aggregated indices with each other. Thus, a node n_1 can determine if the other node (say n_2) it encounters has any data items that match its stored queries. If there is, n_1 will send n_2 queries to retrieve those data items. Every query contains the identifier of the original querying node. If n_2 does not contain matching data items but have index information of the matching data items, then n_2 will send information of the data-node that generate these matching data items to n_1. Next, n_1 can send messages directly to the originating data nodes to retrieve such data items. In addition to index duplication, query duplication can be used in this scheme too.

In the ORNA-DD scheme, data duplication is used instead. Regular nodes are used to store copies of published data items. Every data-node generates W copies of any data item it generates and binary spread these copies to the nodes it encounters. Again, any node that stores data items advertises the aggregated indices of these data items to any new nodes it encounter. If the other node discovers that the advertised list contains data items that match its stored queries, it will send messages to retrieve the data items.

V. PERFORMANCE EVALUATION

A. Simulation Setup

To investigate the usefulness of the three informational retrieval schemes we describe in Section IV.E, we implement these schemes in NS-2 version 2.31 [12]. We assume that the wireless bandwidth is 2 Mb/s and the transmission range is 250 m. We use a network that consists of 40 nodes randomly distributed over (a) 1400x1400m^2, (b) 2000x2000m^2 (default), (c) 3000x3000m^2 , and (d) 4000x4000 m^2.

Node Movement Model: The nodes move either according to the random waypoint (RWP) mobility model [13] or the Zebranet [14] model. For the RWP model, each node selects a random destination and moves towards the destination with a speed chosen randomly between (vmin, vmax) m/s. After the node reaches its destination, it pauses for a period of time and repeats this movement pattern. Unless otherwise stated, vmin is set to 1, and vmax is set to 5 m/s for all nodes in the homogenous model. For the Zebranet model [14], we create a semi-synthentic model as follows: we synthesize node speed and turn angle distributions from the observed data and create other node-movements using the same distribution. We use both distance and time scaling to fit the original data found in the trace into the network environment that we are interested in. The average node speed in the Zebranet model is 6m/s.

Data Item Generation Model: There are 10 data generating nodes. Each data node generates a data item every 50s. The values of the two attributes of each data item are chosen uniformly from the range [1,8]. Therefore, there are 64 types of data items. We keep track of the number of data copies for each type in the system and re-select attribute values of a new data item such that there are about 3 copies of data items of a particular type. Each data item has an expiry time of 1000s. The data size is fixed at 1000 bytes.

Query Model: There are 20 nodes that generate queries. By default, each querying node generates a query every 50s. The query looks for all data items of a particular type (recall that we have 64 types of data items and there are on the average three data items per type). Each query uniformly chooses one of these 64 data types. Each query has an expiration time of 1000s.

The performance metrics that we use to compare different combinations of schemes are

- Query success ratio –there are two metrics related to this. The first one (referred to as query success rate) measures the ratio of the number of queries which successfully retrieve at least 1 data item over the total number of generated queries. The second one (referred to as overall success ratio) measures the ratio of queries that successfully retrieve all data items in that category over the total number of generated queries.
- Query response time – this is measured as the average time it takes for the successful query response to arrive back at a node that issues a query.
- Data efficiency – this is measured as the number of useful data bytes over the number of total transmitted data bytes (does not include control overhead).

Each data point reported in the simulation results section is the average of 10 runs.

B. Simulation Results

1) Impact of Node Density

In our first experiment, we compare the performance of the various schemes using networks of different node density. For the ORNA-ID scheme, we replicate four copies of each index. For the ORNA-DD scheme we replicate four copies of each generated data item. We try both the RWP and the Zebranet mobility models. We will discuss more on the impact of mobility models in the next subsection. Here, we focus our discussion on the performance comparison of the three schemes.

The results for average query success ratio and overall success ratio for the PISA, ORNA-ID, ORNA-DD schemes are plotted in Figures 1(a), 1(b) & 1(c) respectively. The results indicate that as the network becomes sparser, fewer queries can be successful due to the increasing node encounter time. The results also indicate that the ORNA-ID scheme performs slightly better than the PISA scheme in dense environment but similar success rate in very sparse environment irrespective of whether the RWP or Zebranet model is used. This indicates that either architecture can be used when the network is very sparse and the nodes can roam over the whole area. The PISA scheme should be used if nodes only trust specially deployed storage nodes while the ORNA-ID scheme can be used if nodes trust one another. The ORNA-DD gives the best success ratio, the overall success rate, and smaller query response time. The ORNA-DD scheme achieves 91% higher query success rate and 128% higher overall success rate when compared to those achieved using the other two schemes in the sparsest scenario (40 nodes over 4000x4000m^2). This shows that data duplication yields better results than index duplication in sparse network. The only drawback of the ORNA-DD scheme is that more storage space is required in the network. We also have conducted some simulation studies where the query replication feature is added to the ORNA-ID scheme but the results indicate that there is no further improvement in the query performance.

Figures 2(a), and 2(b) plot the successful query response time distribution obtained using the PISA and the ORNA-DD schemes. Figure 3 plots the data efficiency obtained using the PISA and the ORNA-DD schemes. We did not include the response time distribution and the data efficiency of the ORNA-ID scheme because they are very similar to those achieved using the PISA scheme. The average query response time achieved by the ORNA-DD scheme is the smallest because of the data replication. The average query response time for the PISA scheme is slightly better than the ORNA-ID scheme because the querying nodes know which ISP to query for the information of data items that will match its stored queries.

Since we use the same index replication factor, the PISA and ORNA-ID schemes achieve similar data efficiency. The advantage of the ORNA-ID scheme over the PISA scheme is that no special ISP nodes need to be deployed. The disadvantage is that we have to trust all regular nodes. However, the data efficiency for the ORNA-DD scheme is lower than that of the PISA or the ORNA-ID scheme because (a) we include the cost of data duplication in the ORNA-DD scheme in our data efficiency computation, and (b) the query generation rate we use is comparable to the data generation rate. When we increase the query rate to 1 query/10s, then ORNA-DD achieves better data efficiency as the PISA/ORNA-ID scheme.

Another interesting observation is that the data efficiency achieved using the PISA scheme has a concave shape with its lowest data efficiency occurring when the node density corresponds to 40 nodes over 2000x2000m^2 . Such a concave shape is due to the fact that we only consider the data efficiency of successful queries. As the network becomes sparser, it takes longer time for the querying node to retrieve data items from the data-node after it gets a reply from the ISP so the data efficiency drops. Beyond a certain node density level, such queries can no longer be successful. In sparser networks, queries can only be successful if the querying nodes happen to discover data nodes (which results in higher data efficiency because data items are retrieved using single hop transmissions). However, we observe a convex shape for the ORNA-DD scheme. This may be due to the fact that using 4 replicated copies in the 40 nodes over 2000x2000m^2 scenario gives the querying nodes the best chance to encounter data nodes and hence the highest data efficiency. It will be interesting to develop an analytical framework to compute the optimal replication factor for achieving a certain query success rate and data efficiency.

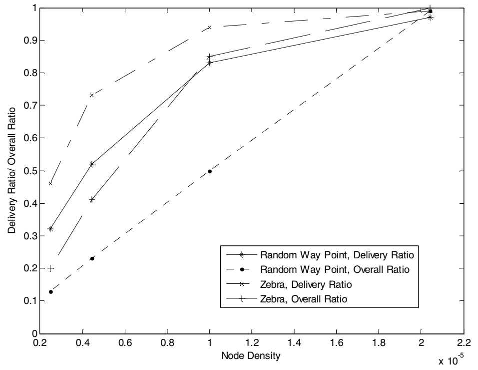

Figure 1(a) Success Ratio/Overall Success Ratio for PISA

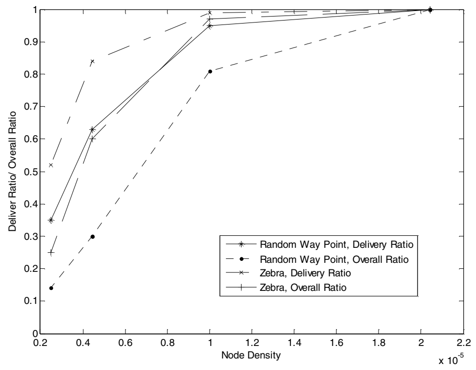

Figure 1(b) Success Ratio/Overall Success Ratio of the ORNA-ID scheme

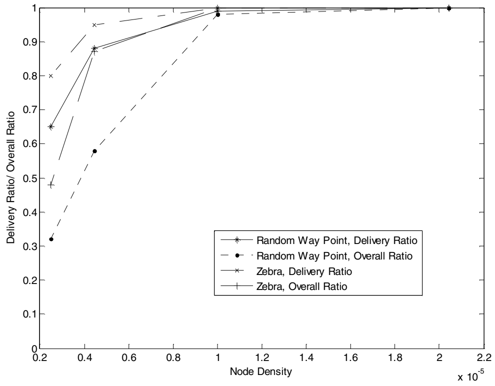

Figure 1(c) Success Ratio/Overall Success Ratio of the ORNA-DD scheme.

2) *Impact of Mobility Model*

From the plots in Figures 1, 2, and 3 we observe that higher query success rate and overall success rate is achieved when the Zebranet model is used as compared to those achieved using the RWP model. This is because the nodes within the Zebranet model move faster than the nodes in the RWP model. Recall that the average node speed for the RWP model is 2.5m/s while the average node speed for the Zebranet is 6 m/s. The mean successful query response time is also slightly smaller with the Zebranet model.

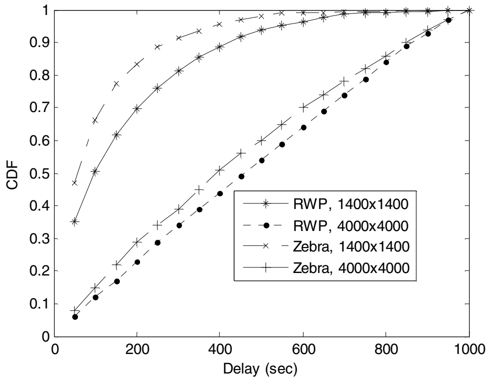

Figure 2a) CDF of the successful query response time for the PISA scheme

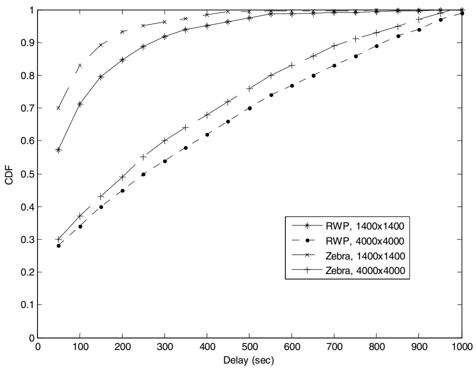

Figure 2b) CDF of the successful query response time for the ORNA-DD scheme.

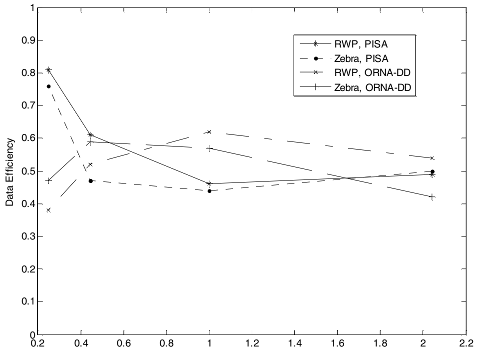

Figure 3 Data Efficiency vs Node Density for the PISA and the ORNA-DD schemes.

VI. CONCLUDING REMARKS

In this paper, we have presented three information retrieval schemes and compare their performances using queries with two attributes. Our results indicate that when the network is sparse, index duplication is not sufficient to achieve high query success rate. Data duplication needs to be used instead. Our result show that with data duplication, the query success rate can improve by 91% and the overall success rate can improve by 128%. This is just a preliminary work. We intend to compare the performance of these three schemes using other realistic mobility models e.g. [6], [18]. In this paper, we only consider queries with values that are uniformly selected among possible ranges. We intend to consider non-uniform queries that follow the Zipf-like distribution [15]. In addition, we intend to compare the query performance of various schemes using range queries. Last but not least, we intend to develop an analytical framework that can be used to find optimal index and data duplication factors which can meet a certain query success rate requirement.

ACKNOWLEDGMENT

This work has been supported by DARPA under Contract W15P7T-06-C-P430. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the sponsor of this work.

REFERENCES

[1] K. Fall, " A delay tolerant network architecture for challengednetworks", Proceedings of ACM Sigcomm, 2003..

[2] V. Cerf et al, "Delay Tolerant Networking Architecture", RFC4838, April, 2007

[3] A. Lingren et al, "Probabilistic Routing in Intermittently Connected Networks", Proceedings of Workshop on Service Assurance with Partial and Intermittent Resources, Aug, 2004.

[4] J. Burgess et al, "MaxProp: Routing for vehicle-based disruption tolerant networks", Proceedings of IEEE Infocom, 2006.

[5] M. Chuah, P. Yang, "Performance evaluations of a content-based information retrieval scheme for DTNs", Proceedings of Milcom, October, 2007.

[6] MIT Media Lab: Reality Mining. http://reality.media.mit.edu.

[7] B. Sheng et al, "Data Storage Placement in Sensor Networks", Proceedings of ACM Mobihoc, May, 2006.

[8] A. R. Bharambe et al, "Mercury: supporting scalable multi-attribute range queries", Proceedings of ACM Sigcomm, Aug/Sept, 2004.

[9] L. Yin, G. Cao, "Supporting Cooperative Caching in Adhoc Networks", Proceedings of IEEE Infocom, 2004. T. Spyropoulos et al, "Efficient routing in intermittently connected mobile networks: single copy case" to appear in IEEE/ACM Transactions on Networking, 2007.

[10] A. Vahdat, D. Becker, "Epidemic Routing for partially connected adhoc networks", Technical Report CS-200006, Duke University, April, 2000

[11] T. Spyropoulos et al, "Efficient routing in intermittently connected mobile networks: multiple copy case" to appear in IEEE/ACM Transactions on Networking, 2007.

[12] "The network simulator ns-2", [Online] at http://www.isi.edu/nsnam/ns/.

[13] J. Broch et al, "A Performance Comparison of Multhop wireless Adhoc Network Routing Protocols", ACM Mobicom, pp 85-97, Oct, 1998.

[14] Y. Wang et al, "Erasure-Coding Based Routing for Opportunistic Networks", Proceedings of ACM workshop on WDTN, 2005.

[15] H. Wang et al, "Snoogle: A search engine for the physical world", to appear in Proceedings of Infocom, April, 2008

[16] T. Hara, "Effective Replica Allocation in AdHoc Networks for Improving Data Accessibility", Proceedings of IEEE Infocom, 2001.

[17] M. Chuah, P. Yang, "Data-Centric information retrieval schemes for Disruption Tolerant Networks", Lehigh CSE Department Technical Report, Dec, 2007.

[18] X. Zhang et al, "Modeling of a Bus-based Disruption Tolerant Network Trace: Mobility Modeling and Impact on Routing", Proceedings of ACM Mobicom, 2007.