© 2017 IJSRST | Volume 3 | Issue 6 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
Themed Section: Science and Technology

Quick Nearest Neighbour Track with Keywords

Javairia Begum[1], Dr. Shaik Shavali[2]
[1]CSE, LORDS Institute of Engineering and Technology, Hyderabad, Telangana, India
jwr.786@gmail.com[1]
[2]CSE, LORDS Institute of Engineering and Technology, Hyderabad, Telangana, India
shaikshavali@lords.ac.in[2]

ABSTRACT

Conventional spatial queries such as vary search and nearest neighbor retrieval, involve solely conditions on objects' geometric properties. Today, several trendy applications call for novel varieties of queries that aim to seek out objects satisfying both a spatial predicate, and a predicate on their associated texts. As an example, rather than considering all the restaurants, a nearest neighbor query would instead invite the restaurant that is the nearest among those whose menus contain "steak, spaghetti, brandy" all at identical time. Presently the simplest solution to such queries relies on the IR2-tree, which, as shown in this paper, includes a few deficiencies that seriously impact its efficiency. Impelled by this, we have a tendency to develop a brand new access methodology called the spatial inverted index that extends the standard inverted index to address multidimensional information, and comes with algorithms which will answer nearest neighbour queries with keywords in real time. As verified by experiments, the projected techniques outperform the IR2- tree in query reaction time significantly, typically by an element of orders of magnitude. Spatial queries, such as range search and nearest neighbour retrieval, involve only conditions on objects geometric properties. A spatial database manages multidimensional objects (such as points, rectangles, etc.), and provides fast access to those objects based on different selection criteria. Now-a-days many applications call a new form of queries to find the objects that satisfying both a spatial predicate, and a predicate on their associated texts. For example, instead of considering all the restaurants, a nearest neighbour query would instead ask for the restaurant that is the closest among those whose menus contain the specified keywords all at the sametime.IR2-tree is used in the existing system for providing best solution for finding nearest neighbour. This method has few deficiencies. So we implement the new method called spatial inverted index to improve the space and query efficiency. And enhanced search is used to search the required objects based on the user priority level. Thus the proposed algorithm is scalable to find the required objects.

Keywords: R-tree, UML, diagrams, brandy, index, R-tree

I. INTRODUCTION

A spatial database manages multidimensional objects (such as points, rectangles, etc.), and provides fast access to those objects based on different selection criteria. The importance of spatial databases is reflected by the convenience of modelling entities of reality in a geometric manner. For example, locations of restaurants, hotels, hospitals and so on are often represented as points in a map, while larger extents such as parks, lakes, and landscapes often as a combination of rectangles. Many functionalities of a spatial database are useful in various ways in specific contexts. For instance, in a geography information system, range search can be deployed to find all restaurants in a certain area, while nearest neighbour retrieval can discover the restaurant closest to a given address. Today, the widespread use of search engines has made it realistic to write spatial queries in a brand new way. Conventionally, queries focus on objects' geometric properties only, such as whether a point is in a rectangle, or how close two points are from each other. We have seen some modern applications that call for the ability to select objects based on both of their geometric coordinates and their associated texts. For example, it would be fairly useful if a search engine can be used to find the nearest restaurant that offers “steak, spaghetti, and brandy” all at the same time. Note that this is not the “globally” nearest restaurant (which would have been returned by a traditional nearest neighbour query), but the nearest restaurant among only those providing all the demanded foods and drinks. There are easy ways to support queries that combine spatial and text features. For example, for the above query, we could first fetch all the restaurants whose menus contain the set of keywords {steak, spaghetti, brandy}, and then from the retrieved restaurants, find the nearest one. Similarly, one could also do it reversely by targeting first the spatial conditions—browse all the restaurants in ascending order of their distances to the query point until encountering one whose menu has all the keywords. The major drawback of these straightforward approaches is that they will fail to provide real time answers on difficult inputs. A typical example is that the real nearest neighbour lies quite far away from the query point, while all the closer neighbours are missing at least one of the query keywords. Spatial queries with keywords have not been extensively explored. In the past years, the community has sparked enthusiasm in studying keyword search in relational databases. It is until recently that attention was diverted to multidimensional data .The best method to date for nearest neighbour search with keywords is due to Felipe et al. They nicely integrate two well-known concepts: R-tree, a popular spatial index, and signature file [11], an effective method for keyword-based document retrieval. By doing so they develop a structure called the IR2-tree, which has the strengths of both R-trees and signature files. Like R-trees, the IR2-tree preserves objects’ spatial proximity, which is the key to solving spatial queries efficiently. On the other hand, like signature files, the IR2-tree is able to filter a considerable portion of the objects that do not contain all the query keywords, thus significantly reducing the number of objects to be examined. The IR2-tree, however, also inherits a drawback of signature files: false hits. That is, a signature file, due to its conservative nature, may still direct the search to some objects, even though they do not have all the keywords. The penalty thus caused is the need to verify an object whose satisfying a query or not cannot be resolved using only its signature, but requires loading its full text description, which is expensive due to the resulting random accesses. It is noteworthy that the false hit problem is not specific only to signature files, but also exists in other methods for approximate set membership tests with compact storage (see and the references therein). Therefore, the problem cannot be remedied by simply replacing signature file with any of those methods. In this paper, we design a variant of inverted index that is optimized for multidimensional points, and is thus named the spatial inverted index (SI-index). This access method successfully incorporates point coordinates into a conventional inverted index with small extra space, owing to a delicate compact storage scheme. Meanwhile, an SI-index preserves the spatial locality of data points, and comes with an R-tree built on every inverted list at little space overhead. As a result, it offers two competing ways for query processing. We can (sequentially) merge multiple lists very much like merging traditional inverted lists by ids. Alternatively, we can also leverage the R-trees to browse the points of all relevant lists in ascending order of their distances to the query point.

II. EXISTING SYSTEM

Spatial queries with keywords have not been extensively explored. In the past years, the community has sparked enthusiasm in studying keyword search in relational databases.

It is until recently that attention was diverted to multidimensional data. The best method to date for nearest neighbour search with keywords is due to Felipe et al. They nicely integrate two well-known concepts: R-tree, a popular spatial index, and signature file, an effective method for keyword-based document retrieval. By doing so they develop a structure called the IR2 - tree, which has the strengths of both R-trees and signature files.

Like R-trees, the IR2 - tree preserves objects’ spatial proximity, which is the key to solving spatial queries efficiently. On the other hand, like signature files, the IR2 -tree is able to filter a considerable portion of the objects that do not contain all the query keywords, thus significantly reducing the number of objects to be examined.

DISADVANTAGES OF EXISTING SYSTEM:

- Fail to provide real time answers on difficult inputs.
- The real nearest neighbour lies quite far away from the query point, while all the closer neighbours are

missing at least one of the query keywords.

III.PROPOSED SYSTEM:

In this paper, we design a variant of inverted index that is optimized for multidimensional points, and is thus named the spatial inverted index (SI-index). This access method successfully incorporates point coordinates into a conventional inverted index with small extra space, owing to a delicate compact storage scheme.

Meanwhile, an SI-index preserves the spatial locality of data points, and comes with an R-tree built on every inverted list at little space overhead. As a result, it offers two competing ways for query processing.

We can (sequentially) merge multiple lists very much like merging traditional inverted lists by ids. Alternatively, we can also leverage the R-trees to browse the points of all relevant lists in ascending order of their distances to the query point. As demonstrated by experiments, the SI-index significantly outperforms the IR2 -tree in query efficiency, often by a factor of orders of magnitude.

ADVANTAGES OF PROPOSED SYSTEM:

- Distance browsing is easy with R-trees. In fact, the best-first algorithm is exactly designed to output data points in ascending order of their distances
- It is straight forward to extend our compression scheme to any dimensional space.

IV.MODULE DESCRIPTION:

A. *System Model:-*

- In this module a User have to register first, then only he/she has to access the data base.
- In this module, any of the above mentioned person have to login, they should login by giving their email id and password.
- In this module Admin registers the location along with its famous place. Also he measures the distance of the corresponding place from the corresponding source place by using spatial distance of Google map.
- It means that the user can give the key in which place that the city/location is famous for. This results in the list of menu items displayed.

B. *Hotel Registration:*

- In this module Admin registers the hotel along with its famous dish. Also he measures the distance of the corresponding hotel from the corresponding source place by using spatial distance of Google map

V.SEARCH TECHNIQUES:

Here we are using two techniques for searching the document

1) Restaurant Search,

2) Key Search.

<u>Key Search</u>: It means that the user can give the key in which dish that the restaurant is famous for .This results in the list of menu items displayed.

<u>Restaurant Search</u>: It means that the user can have the list of restaurants which are located very near. List came from the database.

- Map View
- Distance Search
- Neighbour Search

A. *Map View:-*

- The User can see the view of their locality by Google Map (such as map view, satellite view).
- As our goal is to combine keyword search with the existing location-finding services on facilities such as hospitals, restaurants, hotels, etc., we will focus on dimensionality 2, but our technique can be extended to arbitrary dimensionalities with no technical obstacle.
- Note that the list of each word maintains a sorted order of point ids, which provides considerable convenience in query processing by allowing an efficient merge step. For example, assume that we want to find the points that have words c and d. This is essentially to compute the intersection of the two words' inverted lists.

B. Distance Search

- The User can measure the distance and calculate time that takes them to reach the destination by giving speed. Chart will be prepared by using these values. These are done by the use of Google Maps.
- Traditional nearest neighbour search returns the data point closest to a query point.

- We consider that the data set does not fit in memory, and needs to be indexed by efficient access methods in order to minimize the number of I/Os in answering a query

C. Neighbour Search:

- In this module we implement our neighbour Search. The other problem with this search algorithm is that the indexing information has to be replicated in the broadcast cycle to enable twice scanning.
- The first scan is for deciding the search range, and the second scan is for retrieving k objects based on the search range.
- Therefore, we propose the Nearest Neighbour query approach to improve the preceding on-air query algorithm.
- The system attempts to verify the validity of k objects by processing results obtained from several peers.

VI.UML DIAGRAMS:

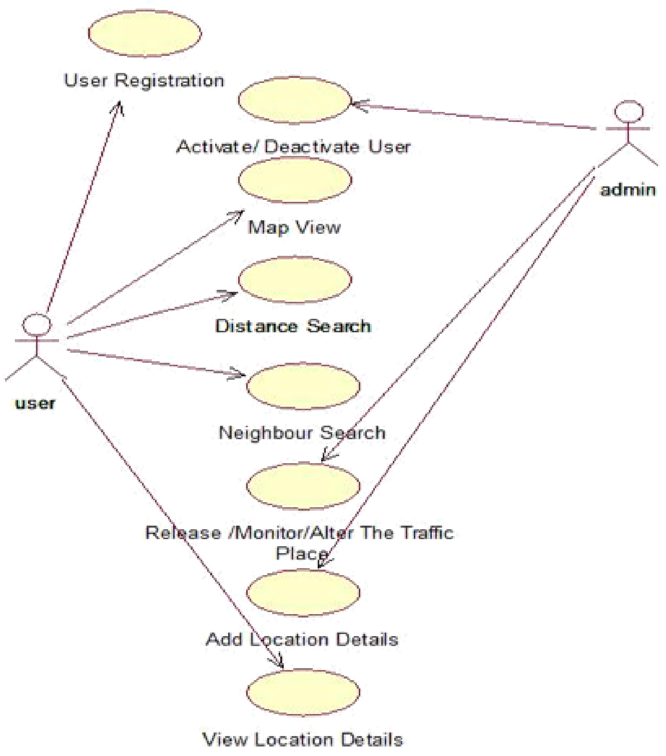


Fig-1: Use-case Diagram

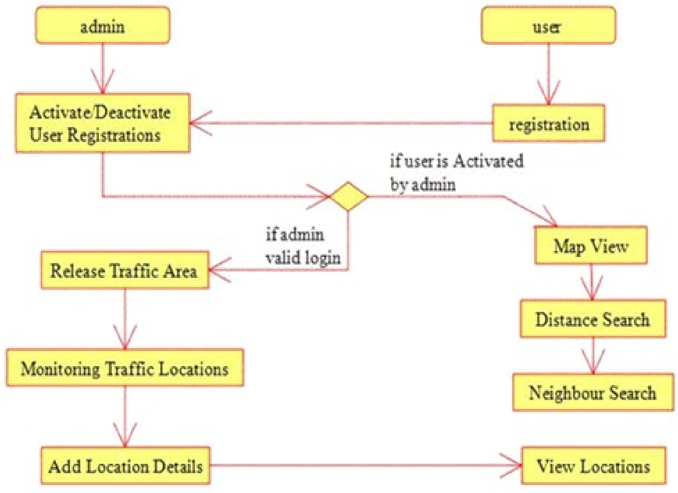
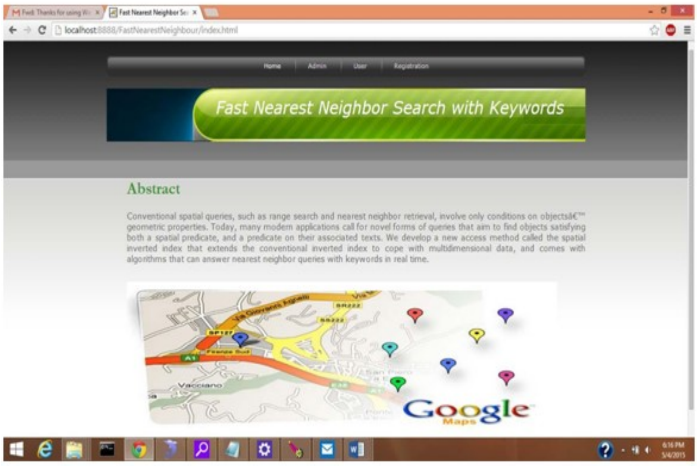


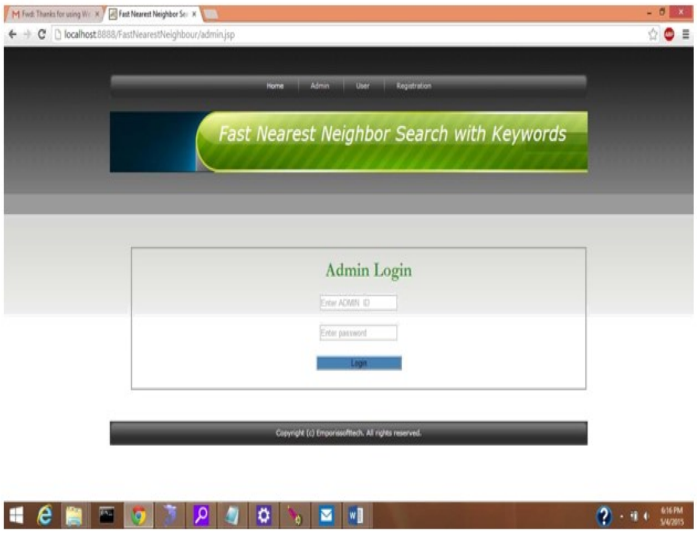
Fig-2: Data Flow Diagram

VII.SCREEN SHOT:

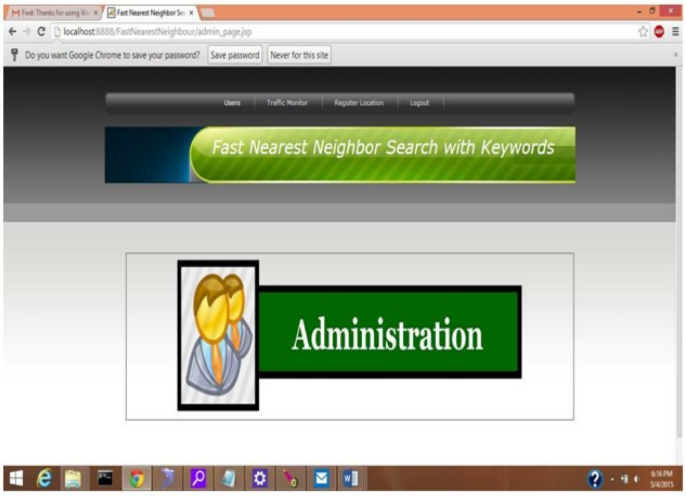
A. Home Page:



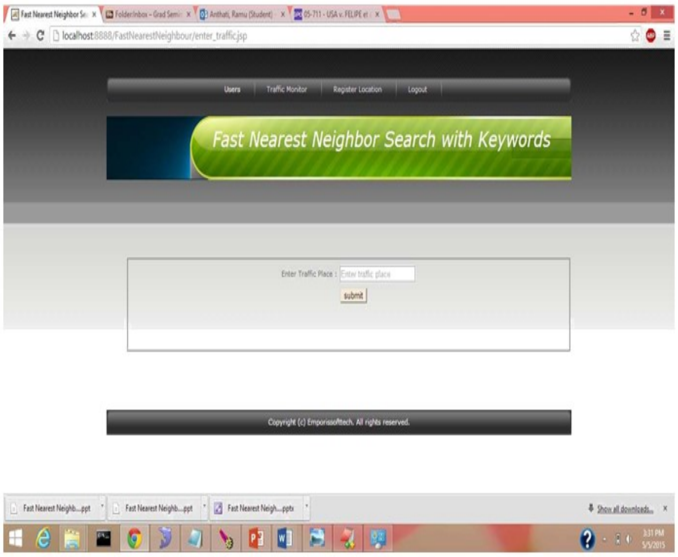
B. Admin Login:

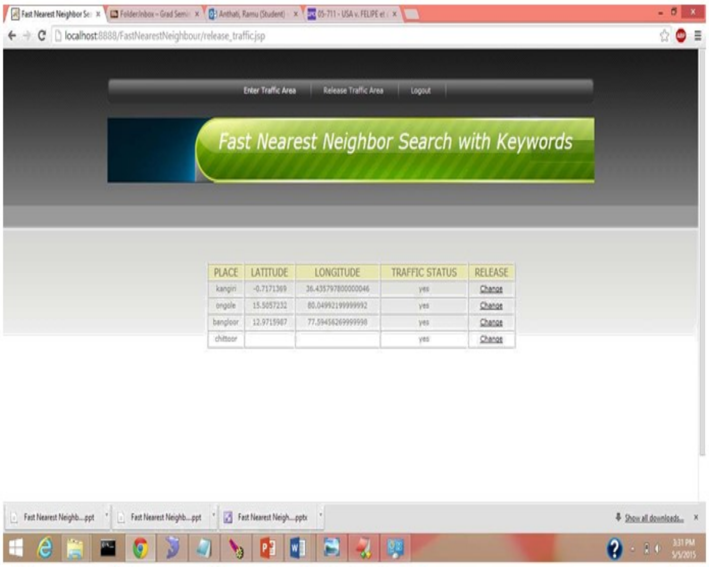


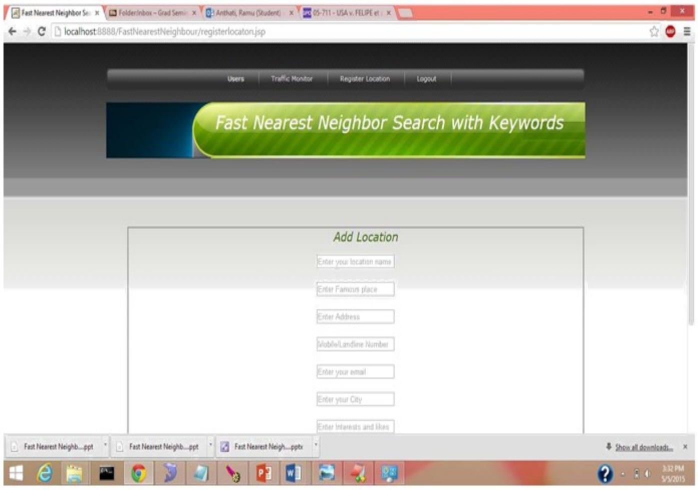
C. Admin Page:



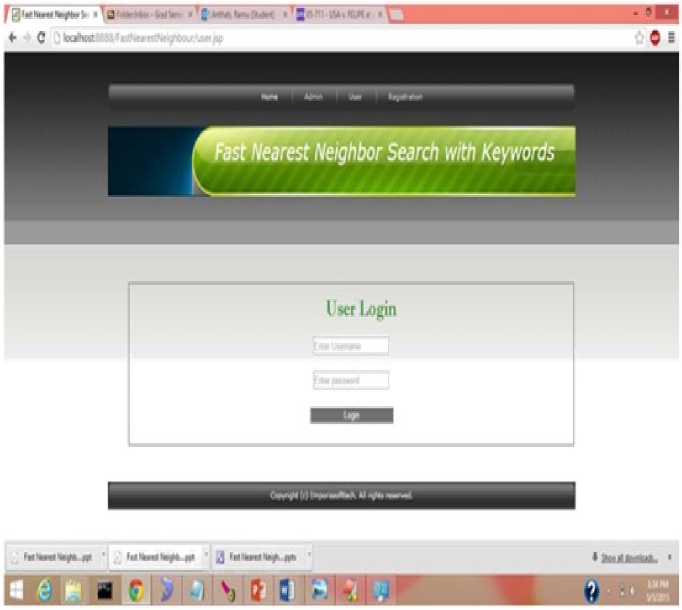
D. Admin Function:



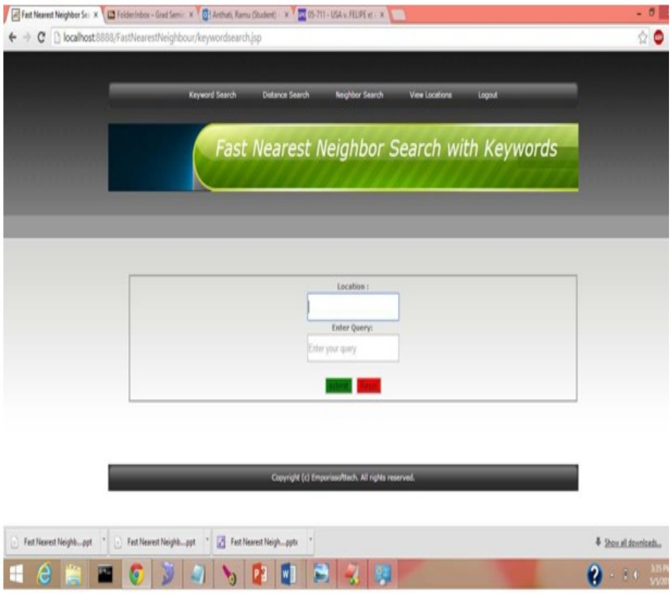


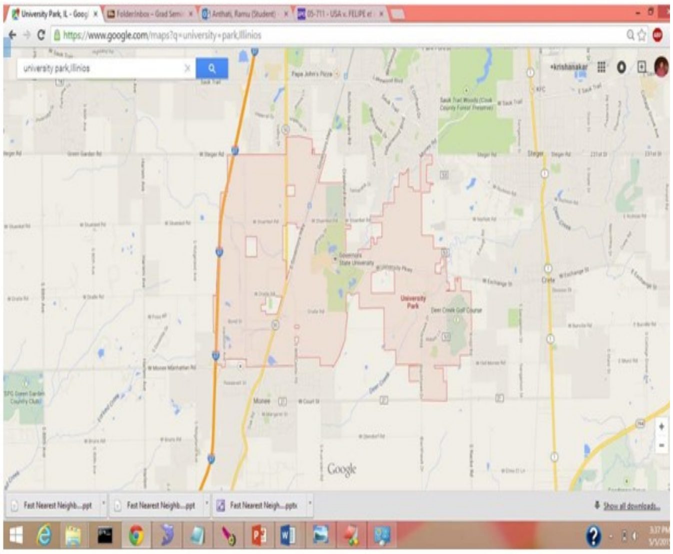


E. User Login:



G. Keyword Search:





VIII.CONCLUSION

We have seen plenty of applications calling for a search engine that is able to efficiently support novel forms of spatial queries that are integrated with keyword search. The existing solutions to such queries either incur prohibitive space consumption or are unable to give real time answers. In this paper, we have remedied the situation by developing an access method called the spatial inverted index (SI-index). Not only that the SI-index is fairly space economical, but also it has the ability to perform keyword-augmented nearest neighbour search in time that is at the order of dozens of milli-seconds. Furthermore, as the SI-index is based on the conventional technology of inverted index, it is readily incorporable in a commercial search engine that applies massive parallelism, implying its immediate industrial merits.

III. REFERENCES

[1] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, “Maxprop: Routing for vehicle-based disruption tolerant networks,” in Proc. IEEE INFOCOM, 2006, pp. 1–11.

[2] M. Chuah and P. Yang, “Node density-based adaptive routing scheme for disruption tolerant networks,” in Proc. IEEE MILCOM, 2006, pp. 1–6.

[3] M. M. B. Tariq, M. Ammar, and E. Zequra, “Message ferry route design for sparse ad hoc networks with mobile nodes,” in Proc. ACM MobiHoc, 2006, pp. 37–48.

[4] S. Roy and M. Chuah, “Secure data retrieval based on cipher text policy attribute-based encryption (CP-ABE) system for the DTNs,” Lehigh CSE Tech. Rep., 2009.

[5] M. Chuah and P. Yang, “Performance evaluation of content-based information retrieval schemes for DTNs,” in Proc. IEEE MILCOM, 2007, pp. 1–7.

[6] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, “Plutus: Scalable secure file sharing on untrusted storage,” in Proc. Conf. File Storage Technol., 2003, pp. 29–42.

[7] L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, “Mediated cipher text-policy attribute-based encryption and its application,” in Proc. WISA, 2009, LNCS 5932, pp. 309–323.