Indexing moving objects on road networks in P2P and broadcasting environments

Hye-Young Kang, Jung-Soo Kim, and Ki-Joune Li

Department of Computer Science and Engineering, Pusan National University,
Pusan 609-735, South Korea
{kykang, jskim}@isel.cs.pusan.ac.kr, lik@pnu.edu

Abstract. Scalability is one of the crucial problems in realizing massively distributed systems such as ubiquitous computing. In this paper, we focus on indexing methods in massively distributed environments. A number of work on indexing in P2P, like CAN and Chord, have been devoted to overcome this problem. The lengths of routing path are $O(dn^{\frac{1}{d}})$ for CAN and $O(\log n)$ for Chord, which are in fact the cost of search, where there are n nodes. In this paper, we propose an alternative indexing scheme not only relying on P2P but also on broadcasting environments. The contributions of this paper include first the reduction of routing path to nearly $O(1)$ for road-oriented query by using broadcasting, and handling the mobility of nodes on road networks.

1 Introduction

Peer-to-Peer (P2P) is a promising approach to overcome the scalability problem of massively distributed systems and ubiquitous computing environments. Several methods have been proposed based on P2P such as Chord [1] and CAN [2]. They provide efficient searching mechanism via distributed lookup tables, which are effectively distributed indices. But they have several weak points in applying it to applications related to GIS or spatial database systems. First, they are not capable of handling mobility of nodes, even though some of them provides geographic routing and searching [3, 4]. Only static locations of nodes are considered by these methods. Second, the length of routing path and hop counts of these methods, which are basically the cost of searching in P2P, are still large.

In this paper, we propose a distributed indexing methods for mobile nodes in P2P environments, where a fraction of index is periodically broadcasted over the mobile nodes. This hybrid approach of indexing with P2P and broadcasting reduces the hop counts nearly to a constant. But due to the limited bandwidth of broadcasting, the index structure to be broadcasted should be designed carefully with minimum size of data as possible. The large size of broadcasting data influences on the performance of search as indicated in [5]. For this reason, we only contain a small fraction of indices in broadcasting messages.

In real applications, most mobile nodes are found on road networks. For example, vehicles are typical example of mobile nodes and they move on road networks. However the previous studies on mobile have focused on euclidian space except [6, 7]. In this paper, we rather focus on mobile objects in road network spaces.

The results of experiments with a real road network data set show that the performance of our method rely on the speed of mobile objects and the frequency of broadcasting and most in-network queries can be processed within four hop counts.

The rest of this paper is organized as follows; in the next section, we present related work and motivations of our research. We present the basic data structures mechanisms of our indexing method in section 3. The algorithms for handling the mobility of nodes and query processing are given in section 4. In section 5, we analyze the performance of our method with a cost model and experiments with a real road network data and simulator. In section 6, we conclude the paper.

2 Related work and motivations

In this paper, we assume that each node has an IP address so that it can directly send a message to others if it knows their IP addresses. Based on this assumption, a number of methods have been proposed to process search queries based on distributed hash tables (DHT) such as Tapestry [8], Chord [1], and CAN [2]. By these methods, each node has a small fraction of distributed lookup tables to index. They were initially intended for processing exact match queries like keyword search. Several extensions have been made to process range query as well. For example, an extension of CAN has been tried to handle range queries [9]. It maps multi-dimensional range to an 1-D key by using Hilbert curve. By similar way, MAAN also extends Chord to process multi-dimensional range queries by introducing a uniform locality preserving hashing function [10]. Even though they are capable of processing spatial query, they have several weak points.

First, they are not intended to process spatial queries, since they do not fully consider spatial properties, such as spatial proximity. For this reason, their processing cost, which is mainly measured by routing hop counts, is far from being optimal. For example, the processing cost of range query for MAAN is $O(\sum_{i=1}^{M}(\log N + s_i N))$, where M is the number of attributes, N is the number of nodes, and s_i is the selectivity for the i-th attribute. $M = 2$, since we are dealing with spatial query [10]. It implies that the cost is approximately a double of the processing cost for 1-dimensional range query. PePeR is also an indexing method for P2P and supports range query [11]. On the other hand, the theoretical lower bound of routing hop counts of PePeR is known as $\frac{4}{d}\sqrt[d]{N}$, where d means the number of dimensions and N is the number of nodes. It grows relatively rapidly when N becomes large. In fact, the ideal cost for processing spatial query should be constant regardless of the number of nodes.

Second, they do not fully support the mobility of nodes, since the update cost of attributes and locations is relatively expensive. Furthermore, the movement of mobile nodes is not taken into account by these methods. For example, it is no longer valid to process spatial query for mobile objects via finger tables of

Chord, since the values of finger table computed from the positions of mobile nodes at a time instance are no longer at the positions.

Third, these methods are intended to process the spatial query in euclidian space, while the mobile object of most applications are on road network space. A number of properties of road network space differ from the euclidian space, such as the definition of distance, the linear reference systems, etc. For example, DisTIN [12], which is a distributed indexing method for P2P environment and supports the mobility, is limited to euclidian space and not applicable to road network space.

For these reasons, our work aims to develop an indexing method for mobile objects on road networks with the following considerations;

- to reduce the routing hop counts to $O(1)$, and
- to support the mobility on road networks.

In order to achieve the above objectives, we apply digital broadcasting environments, which have been already in service such as DvB (Digital Video Broadcasting), or DMB (Digital Multimedia Broadcasting).

3 Distributed Spatial Indexing using P2P and broadcasting: DIMOR-PnB

The performance of spatial indexing in P2P environments is mainly determined by routing hop counts from the node where the query is issued to the node containing the information about the answers. The routing hop counts increase according to the distance between the query and answer nodes. The basic idea of our method will be explained in this section, which provides a routing mechanism with a constant hop count via broadcasting. Our method is based on the following assumptions.

- All nodes move road networks.
- Each node has its IP address.
- Each node can directly communicate with others by IP address.
- Each node has the information on its location, IP address, and optionally the locations and IP addresses of its neighbor nodes
- A broadcasting server collects the information of a set of selected nodes and periodically broadcasts them.

Figure 1 gives a brief sketch on the idea of our method, called DIMOR-PnB(Distributed Indexing Moving Objects on Road network using P2P and Broadcasting). Suppose that a query "*Find the nodes within a distance* d *from a point* Q" is given to a node P_1. If P_1 has only the information of P_2, there is no other choice except forwarding the query message as $P_1 \rightarrow P_2 \rightarrow P_7 \rightarrow P_8 \rightarrow P_9$. On the contrary, if a broadcasting server broadcasts the information of mobile nodes like figure 1, P_1 can choose P_{10} as a short cut of routing the query message. Then the message is forwarded as $P_1 \rightarrow P_{10} \rightarrow P_9 \rightarrow P_8$.

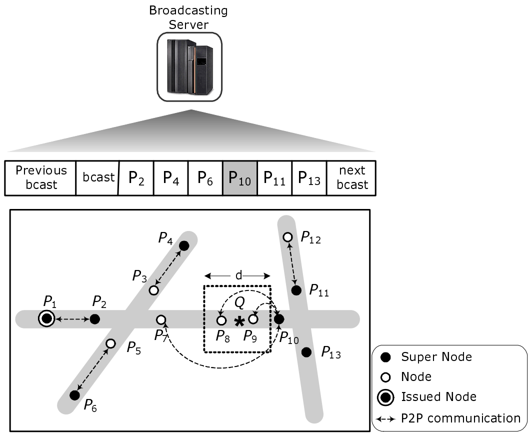


Fig. 1. Example of spatial query processing

The data structures of DIMOR-PnB consist of two parts; firstly broadcasting messages and secondly index stored at each super node. They will be explained in the next subsections.

3.1 Selecting super nodes

The ideal approach would be to let the server broadcast the information of all nodes, so that each node could get the location and IP address of any node and process spatial query by only one hop. In practice, it is impossible since the bandwidth of broadcasting is limited and only a small fraction of band can be allocated to a specific service. Furthermore, the increase of data to broadcast may increase the period of broadcasting and result in an inaccuracy of dynamic information on mobile nodes.

In stead of broadcasting the information of all nodes, we select a certain number of super nodes from the nodes, where the server broadcasts only the information of these super nodes. In this paper, we apply the following rules to select super nodes.

- A super node is to be selected per a road segment.
- If a node enters into a road segment where there is no node, then it becomes the super node.
- If the super node on a road segment moves to other segment, the node that has entered to the node least recently becomes the succeeding super node.

The number of super nodes to be selected is slightly less the number of road segments. We can reduce the number of super nodes by merging unnecessarily segmented roads. Suppose that the data for one super node consists of its IPv6 address and location, then the size becomes 14 bytes per super node. For instance, a metropolitan like Pusan consists of approximately 20,000 road segments. It means that the size of broadcasting message for DIMOR-PnB is about 280 K bytes, which is a reasonable size for terrestrial DMB.

3.2 Management of local topology by neighbor table

Since the broadcasting message does cover only a fraction of index, a complementary index is required. For this reason, each super node keeps a table, called *neighbor table* consisting of the locations and IP addresses of the nodes on its road segment. The data structure of neighbor table is shown in figure 2. Note that the location on a road segment is specified by linear reference systems like $offset$ from the start point rather than (x, y)-coordinate systems. The last two fields, `# of successors` and `(IP, Road ID)*` of the neighbor table will be explained in section 4.

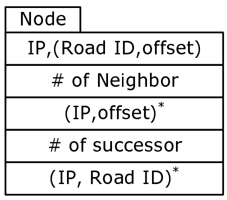


Fig. 2. Data structure of neighbor table

Due to the mobility of nodes, we need to maintain their dynamic membership as well as their location, and the algorithms are presented as follows.

Node Entering: When a node enters to new road segment, the node must listen the next broadcasting message to check existence of super node. In the first case where no super node is found on the new road segment, the node becomes super node and registers to server. Figure 3 shows an example of movement of nodes on road networks. When a node P_2 moves to $S4$ from $S1$ at t_1, P_2 checks existence of super node on $S4$. Since there is no node on S4, P_2 registers itself to the server by sending its location and IP address. Then the broadcasting message will be changed at t_2 as shown by figure 3-b. Second, in case that there is already a super node on the road segment, the node sends its location and IP address to the super node. For example, when P_7 moves to $S3$ from $S4$ at t_2, P_7 checks the broadcasting message and gets the information of the super node P_4. Then, P_7 sends its information to P_4.

Node Leaving: When a node moves to other road segment or disappears, the node should inform its leaving to the super node. In the first case where the leaving node is not the super node, the node should notify its leaving to the super node. For example, when P_2 leaves $S4$ at t_1 in figure 3-a, P_2 is removed

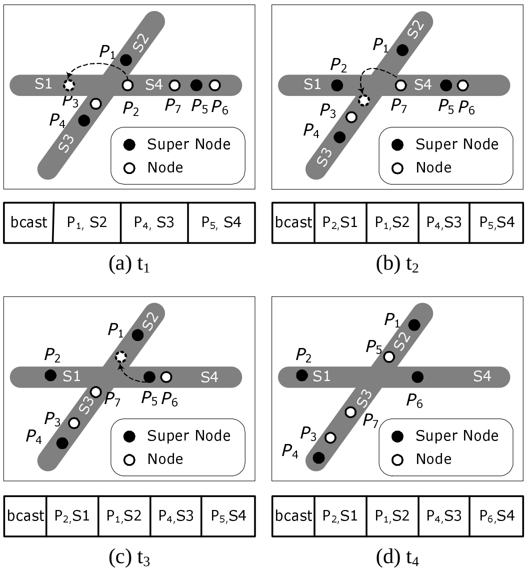


Fig. 3. Example of movement of nodes on the road networks

(a) t_1

P_5
P5,(S4,offset)
of Neighbor :3
P2,offset
P6,offset
P7,offset
of successor : 0

(b) t_2

P_5
P5,(S4,offset)
of Neighbor :2
P6,offset
P7,offset
of successor : 0

(c) t_3

P_5
P5,(S4,offset)
of Neighbor :1
P6,offset
of successor : 0

(d) t_4

P_5
P5,(S2,offset)
of Neighbor :0
of successor : 1
P6,S4

Fig. 4. Change of the neighbor table of P_5

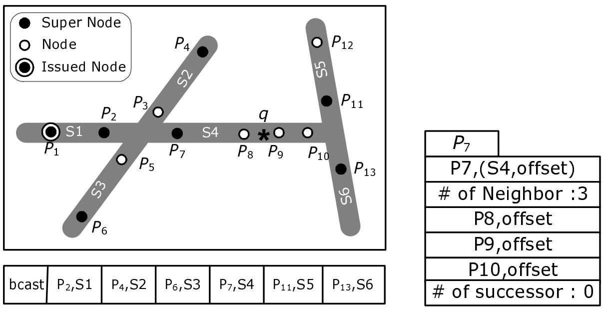
from the neighbor table of super node P_5 as shown by figure 4-a and 4-b. In the second case where the leaving node is a super node, it has to select its successor among the neighbor nodes and transfer the neighbor table to the successor. The change of the finger table for super node P_5 is shown by figure 4-c and figure 4-d. The change of super node must be informed to the broadcasting server as well. If there is no neighbor node on the road segment, the leaving node just asks the server to delete itself from the broadcasting message.

Movement on the same road segment: When a node move on the same road segment, its position of nodes continuously changes, and should be periodically reported to the super node for correct query processing. For this, each node reports own location to super node.

4 Spatial Query Processing based on DIMOR-PnB

In this section, we will present how to apply DIMOR-PnB for processing spatial query by an example of nearest neighbor(NN) query. Other types of spatial queries can be processed by similar way. Suppose that a query "*Find the nearest node to a given point q on road networks.*" is submitted to a node P_1 as shown by figure 5-a. Then this query is to be processed by four steps with DIMOR-PnB as follows.

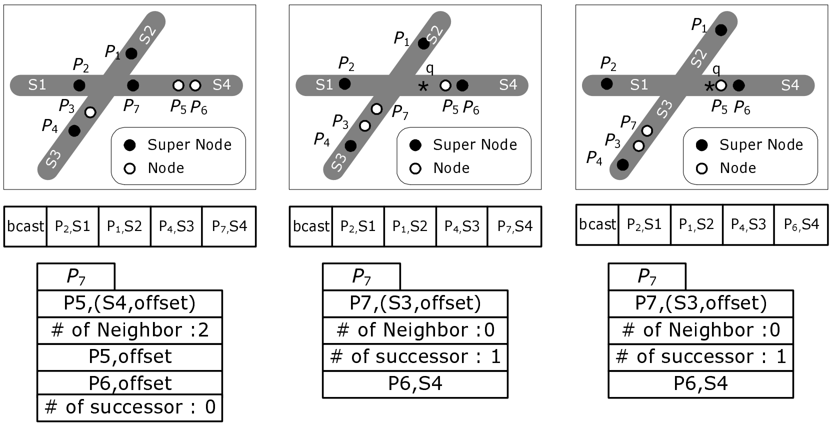
- **step 1**: P_1 searches the super node that is on the same road segment of the query point q from the recently received broadcasting message.
- **step 2**: P_1 sends the query message to the super node P_7 founded by step 1.
- **step 3**: P_7 searches the nearest node to q
- **step 4**: P_7 sends the location and IP address of P_9, the nearest node to q, to the P_1.



(a) Statement of the road networks (b) Structure of P_7

Fig. 5. Example of Nearest Neighbor Query Processing

In some cases where the speed of node is high, there may be a mismatch between the actual position of a super node and its location data in the recent broadcasting message. It may happen when a super node leaves from a road segment after having reported its location to the server. Then location data on the broadcasting message may be no longer valid. In order to avoid this problem, the leaving super node should keep the pointer to its succeeding super node as depicted by figure 2. Note that the successors may be more than one, when the speed is very high. Figure 6 illustrates this case of query processing, where the super node on the road segment has been changed.



(a) t_1 (b) t_2 (c) t_3

Fig. 6. Example of nearest neighbor query processing via successors

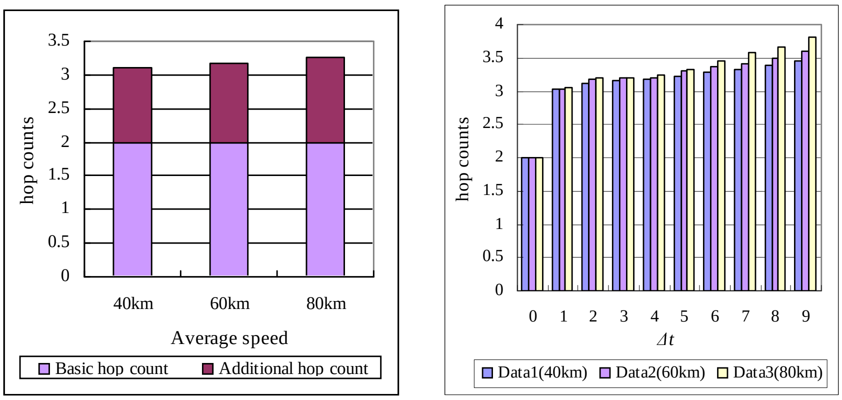
5 Experiments

In order to analyze the performance of our method, we performed several experiments with a real data set of the road networks in Pusan and three synthetic data sets of 100,000 mobile nodes generated with different average speeds, 40Km/h, 60Km/h, and 80Km/h respectively by [13]. We executed 500 nearest neighbor queries to evaluate the performance of our method. We considered routing hop counts as the cost measure of DIMOR-PnB.

Figure 7-a shows the costs of the nearest neighbor query processing. The lower bound of routing hop counts for DIMOR-PnB is 2, since we need one hop for step 2 in section 4 and one more hop for returning answer of query. The additional costs are mainly caused by leaving super nodes. If it is no longer on the same road segment, the actual super node should be retrieved by traversing the successor pointers of neighbor tables. Although the additional cost increases according to the increase of speed, the results show that the increases are very gradual.

Figure 7-b shows the cost according to the differences between the times of query and broadcasting. When there is no difference, it means that the location data in broadcasting message is correct and no additional cost is required regardless of the speed. As the difference increases, the cost increases and the cost differences between different speeds become apparent. However the increase according to the time difference is extremely slow. We observe a big jump of additional cost between $\triangle t = 0$ and $\triangle t = 1$, which is due to leaving super nodes. It means that we will observe another big jump when the leaving super node will change one more road segment.

Most of messages that a super node receives are for updating locations of nodes without changing road segment, as shown by figure 8. It is expected that



(a) NNQ cost according the average speed

(b) NNQ cost according △t

Fig. 7. Query processing costs

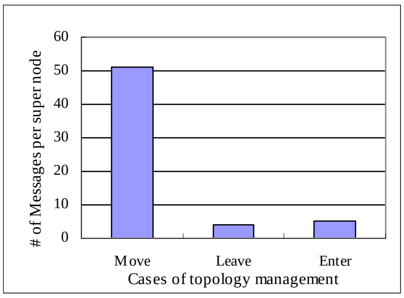


Fig. 8. Number of messages per each super node

we could reduce the number of messages by using more accurate tracking methods such as [14].

6 Conclusion

Although a number of methods have been proposed to process spatial queries, very few attentions have been paid on spatial query process based on road networks in P2P environment.

In this paper, we propose a method for spatial indexing and processing spatial query by a hybrid approach composed of P2P and broadcasting. The contributions of our work are the development of a distributed spatial indexing method by the hybrid approach 1) for mobile nodes, 2) in road network space, and 3) the reduction of routing hop counts to almost constant in comparison with the previous P2P indexing method.

In this paper, we did not deal with the structure of the broadcasting message. As mentioned by [5], an improvement is expected by proper data structures of broadcasting messages.

References

1. Ion Stoica, Robert Morris, David R. Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 149–160, 2001.
2. Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard M. Karp, and Scott Shenker. A scalable content-addressable network. In *Proceedings of International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 161–172, 2001.
3. Brad Karp and H. T. Kung. GPSR: Greedy Perimeter Stateless Routing for wireless networks. In *Proceedings of International Conference on Mobile Computing and Networking*, pages 243–254, 2000.
4. Young-Bae Ko and Nitin H. Vaidya. Location-Aided Routing (LAR) in mobile ad hoc networks. *Wireless Networks*, 6(4):307–321, 2000.
5. Tomasz Imielinski, S. Viswanathan, and B. R. Badrinath. Data on Air: Organization and Access. *IEEE Transactions on Knowledge and Data Engineering*, 9(3):353–372, 1997.
6. Christian S. Jensen, Jan Kolárvr, Torben Bach Pedersen, and Igor Timko. Nearest neighbor queries in road networks. In *Proceedings of ACM International Symposium on Advances in Geographic Information Systems*, pages 1–8, 2003.
7. Christian S. Jensen, Torben Bach Pedersen, Laurynas Speicys, and Igor Timko. Data Modeling for Mobile Services in the Real World. In *Proceedings of International Symposium on Spatial and Temporal Databases*, pages 1–9, 2003.
8. Ben Y. Zhao, John Kubiatowicz, and Anthony D. Joseph. Tapestry: a fault-tolerant wide-area application infrastructure. *Computer Communication Review*, 32(1):81, 2002.
9. Artur Andrzejak and Zhichen Xu. Scalable, Efficient Range Queries for Grid Information Services. In *Peer-to-Peer Computing*, pages 33–40, 2002.
10. Min Cai, Martin R. Frank, Jinbo Chen, and Pedro A. Szekely. MAAN: A Multi-Attribute Addressable Network for Grid Information services. In *Proceedings of International Workshop on Grid Computing*, pages 184–191, 2003.
11. Antonios Daskos, Shahram Ghandeharizadeh, and Xinghua An. PePeR: A Distributed Range Addressing Space for Peer-to-Peer systems. In *Proceedings of International Workshop on Databases, Information Systems, and Peer-to-Peer Computing,*, pages 200–218, 2003.
12. Hye-Young Kang, Bog-Ja Lim, and Ki-Joune Li. P2P Spatial Query Processing by Delaunay Triangulation. In *Proceedings of International Workshop on Web and Wireless Geographical Information Systems*, pages 136–150, 2004.
13. Bo-Ryun Kim. Generation of the Vehicle Trajectories using the Realistic Speed Model. Master's thesis, Pusan National University, 2006.
14. Alminas Civilis, Christian S. Jensen, Jovita Nenortaite, and Stardas Pakalnis. Efficient Tracking of Moving Objects with Precision Guarantees. In *Proceedings of International Conference on Mobile and Ubiquitous Systems*, pages 164–173, 2004.