SNAVI: Smart Navigation System Based On Traffic Intelligence

K.Gunalan
Final year student,
Dept of CSE
EGS Pillay Engineering College,Nagapattinam
gunarandy@gmail.com

S. Anbananthan
Final year student,
Dept of CSE
EGS Pillay Engineering College,Nagapattinam
anbananthan24@yahoo.in

J.Jayaseelan
Final year student,
Dept of CSE
EGS Pillay Engineering College,Nagapattinam
jayaseelancse@gmail.com

K.Balasubramanian
Assistant Professor,
Dept of CSE
EGS Pillay Engineering College,Nagapattinam
bala_1102@yahoo.com

Abstract*— With the ever-growing popularity of online map applications and their wide deployment in mobile devices and car-navigation systems, an increasing number of users search for point-to-point fastest paths and the corresponding travel-times. On static road networks where edge costs are constant, this problem has been extensively studied and many efficient speed-up techniques have been developed to compute the fastest path in a matter of milliseconds. The static fastest path approaches make the simplifying assumption that the travel-time for each edge of the road network is constant. However, in real-world the actual travel-time on a road segment heavily depends on the traffic congestion and, therefore, is a function of time i.e., time-dependent. The proposed LTI construction to learn the time-variant distributions of the travel times between any two landmarks. At the same time, besides the traffic flow, this method also implicitly incorporates additional factors, such as direction changes and traffic signals. Moreover, this method can find the fastest route in a future time and needs less online communication for data transition. Thus, the solution and the real-time-based approach can complement each other

I. INTRODUCTION

The online shortest path problem is considered under various models of partial monitoring. Given a weighted directed acyclic graph whose edge weights can change in an arbitrary way, a decision maker has to choose in each round of a game a path between two distinguished vertices such that the loss of the chosen path as small as possible. In a setting generalizing the multi-armed bandit problem, after choosing a path, the decision maker learns only the weights of those edges that belong to the chosen path. The algorithm can be implemented with complexity that is linear in the number of rounds and in the number of edges. An extension label efficient setting is also given, in which the decision maker is informed about the weights of the edges corresponding to the chosen path at a total of time instances. Another extension is shown where the decision maker competes against a time-varying path, a generalization of the problem of tracking the best expert.

A version of the multi-armed bandit setting for shortest path is also discussed where the decision maker learns only the total weight of the chosen path but not the weights of the individual edges on the path. The problem of point-to-point fastest path computation in static spatial networks is extensively studied with many pre computation techniques proposed to speed-up the computation. Most of the existing approaches make the simplifying assumption that travel-times of the network edges are constant. However, with real-world spatial networks the edge travel-times are time-dependent, where the arrival-time to an edge determines the actual travel-time on the edge. In this paper, the study the online computation of fastest path in time-dependent spatial networks and present a technique which speeds-up the path computation. It showed that fastest path computation based on a bidirectional time-dependent is significantly improves the computation time and storage complexity. With extensive experiments using real data-sets. It will demonstrate the efficacy of the proposed techniques for online fastest path computation.

II. EXISTING WORK

In existing system the interaction between users' and search was limited. And as search engines were not personalized so privacy is not maintained. It does not differentiate content and location concept so it affect on performance of system. And user profiles are not properly maintained. All processing tasks are done on the client side so it may decrease the performance. A search engine does not think personally. Approximate string search could be necessary when users have a fuzzy search condition or simply a spelling error when submitting the query, or the strings in the database contain some degree of uncertainty or error. In the context of spatial databases approximate string search could be combined with any type of spatial queries, including range and nearest neighbour queries. As a user can select any place as a source or destination, there would be no taxi trajectory exactly passing the query points. That is, cannot answer user queries by directly mining trajectory patterns from the data. Therefore, how to model taxi drivers' intelligence that can answer a variety of queries is a challenge. It cannot be guarantee there are sufficient taxis traversing on each road segment even if the large number of taxis. That is, the system cannot accurately estimate the speed pattern of each road segment. To save energy and communication loads, taxis usually report on their locations in a very low frequency, like 2-5 minutes per point. This increases the uncertainty of the routes traversed by a taxi. Since the road network is dynamic, the system can use neither the same nor a predefined time partition method for all the landmark edges.

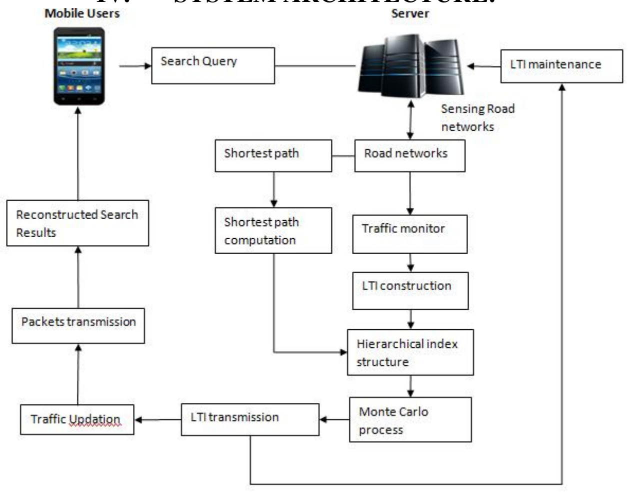
III. PROPOSED SYSTEM

Typical client-server architecture can be used to answer shortest path queries on live traffic data. In this case, the navigation system typically sends the shortest path query to the service provider and waits the result back from the provider (called result transmission model). However, given the rapid growth of mobile devices and services, this model is facing scalability limitations in terms of network bandwidth and server loading. Furthermore, live traffic are updated frequently as these data can be collected by using crowd sourcing techniques (e.g., anonymous traffic data from Google map users on certain mobile devices). As such, huge communication cost will be spent on sending result paths on this model. Obviously, the client-server architecture will soon become impractical in dealing with massive live traffic in near future. The same concern in their work which processes spatial queries in wireless broadcast environments based on Euclidean distance metric.

For spatial queries, the baseline spatial solution is based on the shortest path algorithm. Given a query point q, the query range radius r, and a string predicate, the system expand from q on the road network using the shortest path algorithm until the user reach the points distance r away from q and verify the string predicate either in a post-processing step or on the intermediate results of the expansion. This approach is denoted as the solution. Its performance degrades quickly when the query range enlarges and/or the data on the network increases.

This motivates us to find a novel method to avoid the unnecessary road network expansions, by combining the pruning from both the spatial and the string predicates simultaneously

IV. SYSTEM ARCHITECTURE:



4.1 ARCHITECTURE DESCRIPTION:

In this architecture, the explanation of the bidirectional time-dependent fastest path approach that generalize bidirectional algorithm proposed for spatial networks to time-dependent road networks. this proposed solution involves two phases. At the precomputation phase, partition of the road network into non-overlapping partitions and precompute lower-bound distance labels within and across the partitions with respect to G (V,E). Successively, at the online phase, and use the precomputed distance labels as a heuristic function in this bidirectional time-dependent search that performs simultaneous searches from source and destination.

The precomputation phase of this proposed algorithm includes two main steps in which partition the road network into non-overlapping partitions and precompute lower bound border-to-border, node-to-border, and border-to-node distance labels. With this approach, Then assume that the class of each edge class(e) is predefined and denote the class of a node class(v) by the lowest class number of any incoming or outgoing edge to/from v. For instance, a node at the intersection of two freeway segments and an arterial road (i.e., the entry node to the

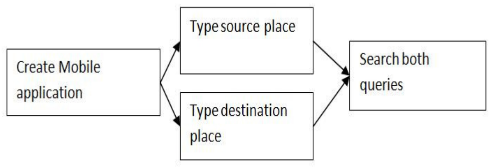
class of the arterial road. The input to this hierarchical partitioning method is the road network and the level of partitioning l. To broadcast a hierarchical index using the (1,m) interleaving scheme, The first partition the index into two components: the index structure and the weight of edges. The former stores the index structure (e.g., graph vertices, graph edges, and shortcut edges) and the latter stores the weight of edges. In order to keep the freshness of LTI, this system is required to broadcast the latest weight of edges periodically.

V. MODULES

- Search Query
- Shortest path computation
- Traffic circumstancess
- LTI construction
- LTI Transmission
- LTI maintenance
- Reconstructed results
- Evaluation criteria

5.1 SEARCH QUERY:

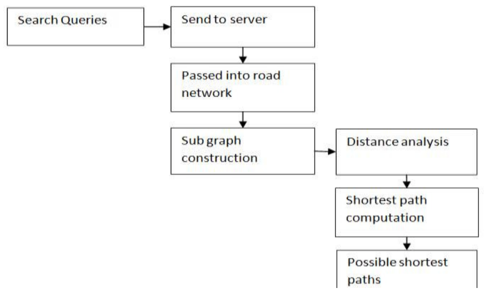
A spatial query is a special type of database query supported by geo databases and spatial databases. The queries differ from non-spatial SQL queries in several important ways. Two of the most important are that they allow for the use of geometry data types such as points, lines and polygons and that these queries consider the spatial relationship between these geometries. In this module, and get the query from users. Queries are related to the location such as places or cities. And create client – server architecture for online shortest path computation.



5.2 SHORTEST PATH COMPUTATION:

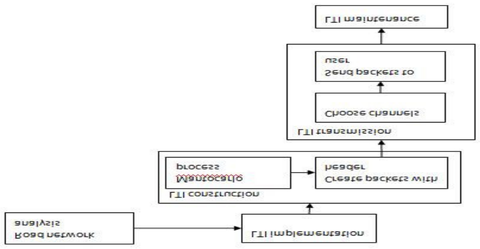
Shortest path computation is one of the most common queries in location-based services (LBSs). Although particularly useful, such queries raise serious privacy concerns.

Exposing to (potentially untrusted) LBS the client's position and her destination may reveal personal information, such as social habits, health condition, shopping preferences, lifestyle choices, etc. In this module, create graph based analysis to find paths and prune the sub paths. Provide the possible paths with time and distances. And highlight the shortest path.



5.3 TRAFFIC CIRCUMSTANCES:

In this module update traffic details based on index transmission model. Create framework called live traffic index (LTI) which enables to collect the live traffic information on the broadcasting channel.



5.3.1 LTI CONSTRUCTION:

This sub module analysis of Hierarchical Index Structures. Monte Carlo process is to execute a set of randomly generated shortest path queries on a temporal index.

5.3.2 LTI TRANSMISSION:

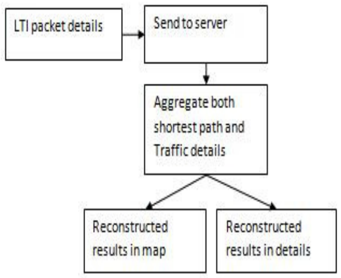
In this sub module, first gets broadcasting scheme to listening to the header of data packets. And analyzed format of data packets with id and checksum values.

5.3.3 LTI MAINTENANCE:

Implement an incremental update approach that can efficiently maintain the live traffic index. To keep the freshness of LTI, every sub graph is required to maintain its corresponding shortcut edges

5.4 RECONSTRUCTED RESULTS:

This module is used to provide online shortest path until the client reaches to the destination. And keeps listening until it discovers possible segment. Finally provide online update shortest path to clients. This paper can provide efficient search results with updated traffic details. The traffic updated properly.


LTI packet details
Send to server
Aggregate both shortest path and Traffic details
Reconstructed results in map
Reconstructed results in details

5.5 EVALUATION CRITERIA:

This module evaluates the performance of the system in size and searchspaces.

VI. RESULTS:

1. The server periodically updates the travel times on these paths based on the latest traffic, and reports the current best path to the corresponding user.

2. Efficiently maintains the index for live traffic circumstances.

3. To the best of the knowledge, this is the first work to give a thorough cost analysis on the hierarchical index techniques and apply stochastic process to optimize the index hierarchical structure.

4. LTI efficiently maintains the index for live traffic circumstances by incorporating Dynamic Shortest Path Tree (DSPT) into hierarchical index techniques. In addition, a bounded version of DSPT is proposed to further reduce the broadcast overhead.

5. LTI reduces the tune-in cost up to an order of magnitude as compared to the state-of-the-art competitors; while it still provides competitive query response time, broadcast size, and maintenance time.

6.The user can obtain the better results based on these parameters such as tune in cost (at client side), (ii) broadcast size (at server side), and (iii) maintenance time (at server side), and (iv) query response time (at client side).

VII. CONCLUSIONS

In this paper, the proposed technique is online shortest path algorithm based on bidirectional graph. Unlike the most path planning studies, Assume the edge weights of the road network are time varying rather than constant. Therefore, this approach yield a much more realistic scenario, and hence, applicable to the real-world road networks. Then compared the approaches with that handful of time-dependent fastest path studies.

This experiments with real-world road network and traffic data showed that proposed approaches outperform the competitors in storage and response time significantly. The system intend to pursue this study in two different directions. First, Then plan to investigate new data models for effective representation of spatiotemporal road networks. This is critical in supporting development of efficient and accurate time-dependent algorithms, while minimizing the storage and computation costs. Second, to support rapid changes of the traffic patterns (that may happen in case of accidents/events; for example), it will intend to study incremental update algorithms for both of approaches.

VIII. FUTURE ENHANCEMENT:

In future work, GPS-Android is employed as mobile sensors probing the traffic rhythm of a city and traffic intelligence in choosing driving directions in the physical world. The proposed system is a time-dependent landmark graph to model the dynamic traffic pattern as well as the intelligence of experienced drivers so as to provide a user with the practically fastest route to a given destination at a given departure time. Then, a Variance-Entropy-Based Clustering approach is devised to estimate the distribution of travel time between two landmarks in different time slots. Based on this graph, Twostage routing algorithm to compute the practically fastest and customized route for end users with weather forecasting has been designed

IX. REFERENCE& BIBLIOGRAPHY:

.

[1] H. Bast, S. Funke, D. Matijevic, P. Sanders, and D. Schultes, “In transit to constant time shortest-path queries in road networks,” in ALENEX, 2007.

[2] G. Dantzig, Linear programming and extensions, ser. Rand Corporation Research Study. Princeton, NJ: Princeton Univ. Press, 1963

[3] R. J. Gutman, “Reach-based routing: A new approach to shortest path algorithms optimized for road networks,” in ALENEX/ANALC, 2004, pp. 100–111.

.

[4] B. Jiang, “I/o-efficiency of shortest path algorithms: An analysis,” in ICDE, 1992, pp. 12–19.

[5] P. Sanders and D. Schultes, “Highway hierarchies hasten exact shortest path queries,” in ESA, 2005, pp. 568–579.

[6] D. Schultes and P. Sanders, “Dynamic highway-node routing,” in WEA, 2007, pp. 66–79.

[7] D. Stewart, “Economics of wireless means data prices bound to rise,” The Global and Mail, 2011.

[8] F. Zhan and C. Noon, “Shortest path algorithms: an evaluation using real road networks,” Transportation Science, vol. 32, no. 1, pp. 65– 73, 1998. 1024

[9] W.-S. Ku, R. Zimmermann, and H. Wang, “Location-based spatial query processing in wireless broadcast environments,” IEEE Trans. Mob. Comput., vol. 7, no. 6, pp. 778–791, 2008.

[10] “Cisco visual networking index: Global mobile data traffic forecast update, 2010-2015,” 2011.

[11] “Google Maps,” http://maps.google.com.

[12]“INRIX inc. Traffic Information Provider,” http://www.inrix.com.

[13] “NAVTEQ Maps and Traffic,” http://www.navteq.com.

[14] “TomTom NV,” http://www.tomtom.com.