Online shortest path computation using Live Traffic index

Dekonda Sindhuja[#], R Vasavi[*] , A Kousar Nikhath[#]

[#] *M.Tech(SE) Student,VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad,India*

[*] *Assistant Professor(CSE),VNR Vignana Jyothi Institute of Engineering and Technology,Hyderabad,India*

[#]*Assistant Professor(CSE),VNR Vignana Jyothi Institute of Engineering and Technology,Hyderabad,India*

Abstract*— Online Shortest path computation using live traffic index in road networks aims at computing the shortest path from source to destination using Live traffic index.In this paper,index transmission model along with Live traffic index technique is used.In this technique,first the traffic provider collects the traffic data and transmits to the traffic server broadcaster.The traffic server broadcaster indexes and optimizes the traffic data and broadcasts to the navigation client.Graph partitioning and stochastic process are the new techniques used to optimize the index.The fast query response time and short tune in cost at client side ,small broadcast size and maintainence time at server side are the extra features achieved in the system.

Keywords —Live traffic index,Stochastic process,Shortest path.

I. INTRODUCTION

Computing the shortest path is important task in the spatial databases.The path computed using the pre-stored data is not accurate.Hence,there is necessity for the live traffic data.There are several online service traffic providers like Navteq[1],Tom tom[2],Google maps[3].But these traffic providers doesnot provide data continuously due to high costs.Client-server architecture is previously used for the shortest path retrievals where the client sends the request and server responds to it.This architecture scales poorly if there are more than two clients.According to telecommunication expert[4],In 2015 the capacity provided by the cellular networks should increase 100 times than in 2011.The communication costs spent on retrieving the shortest route is high. Malviya et al[5] used the client server architecture for shortest routes.In this system,the server stores the estimated paths between source and destination and updates them periodically and returns the shortest path to the required user.

As the system employs client server architecture,it doesnot work for more number of users.In addition,the shortest route is not accurate.Hence there is a need to broadcast the traffic data .In raw transmission model,the traffic data is transmitted over the wireless network.The navigation system receives the live traffic data from the broadcast channel and calculates the shortest path .The broadcast channel transmits the updated packets to the navigation system for each broadcasted cycle.The main disadvantage in this model is client receives more number of traffic updates than required.A new model known as index transmission model [6] is Introduced where the index is transmitted over the broadcast channel.The advantage in this model is client receives only the required portion of the road network.In [6],the time required for the index recomputation is 2 hrs for San Franciso road network.it requires huge costs to update the index.Hence, in our paper a technique Live traffic index is added to the index transmission model.

II.RELATED WORK

In this paper, the important factors of OSP are considered(i)tune-in cost (ii) broadcast size(iii)maintenance time(iv)query response time.The tune in cost should be reduced because the power consumption is based on tune in cost.Due to short tune in cost, by using selective tuning[7] the client receives more services like parking slot,weather related updates.The maintenance time is defined as the time taken to update the index based on the live traffic data.The response time should be very fast in few milliseconds.The broadcast size is similar to the discontinuation of receiving the latest index data.The traffic data received by the client is maintained and shortest path is computed using the uninformed search where the graph nodes are arranged in the ascending order of the distances from the source and the shortest path to the destination is discovered.Bi directional search[8] can be employed where the Dijkstra's algorithm is executed from forward and backward.But the response time is high and client receives more number of updates.

Goal directed methods eliminate the edges that is not required to the shortest path.goal directed methods include ALT[9] and arc Flags[10].In ALT,the distance between the nodes are pre computed using A* search,Landmarks.Delling and Wagner[11] introduced a paradigm for ALT[DALT] so that the edge weights can be updated in dynamic graphs.The maintenance cost is reduced.

A tree structure is maintained for shortest path in Dynamic shortest path tree.In index transmission model,the traffic broadcast server transmits the live index not the complete data various Road map hierarchical methods like reach[12], highwayhierarchies(HH), contraction hierarchies(CH)[13],and Transit node routing(TNR)[15].In HH,CH,TNR the query response time is fast because the shortcuts are calculated and stored in the index.In TNR,the shortest path is computed based on the two transit nodes A(S) from source and A(t) from destination.The maintenance time is high because there is no efficient method to update the pre computed data.

The index structure is maintained in hierarchical way in hierarchical index structures.In Hierarchical multi graph model(HiTi)[16] the index is maintained in hierarchy not the road network.But the high maintenance time and broadcast size are the disadvantages.In Hierarchical encoded path view[HEPV],the nodes and edges are maintained in large graph which is splitted into smaller graphs for shortest path computation.But the index storage is a big problem.our proposed system Live traffic index supports all the factors like short tune in cost,less maintenance time,small broadcast size,quick query response time. To summarize, our work makes the following contributions:

1) The traffic provider collects the traffic data of road networks via road sensors and broadcasts to the traffic broadcast server.
2) The traffic broadcast server collects the traffic data and index structure is constructed for the traffic data .The (1,m) broadcasting scheme is used for transmitting.
3) The index structure is optimized by graph partitioning technique and stochastic process is applied.The index is organized by using Dynamic Shortest Path Tree(DSPT)[17]and broadcast size is also reduced.
4) The navigation client after receiving the required index from the Traffic broadcast server,the shortest path is computed using the dijkstra's algorithm.
5) Live Traffic index reduces the tune in cost and broadcast size and low maintenance time ,fast query response time is acheived by the above features.

III.SYSTEM DESIGN

System Overview:

LTI system consists of navigation clients, traffic provider like google maps. The Traffic provider collects the traffic data from the road sensors and the traffic broadcast server collects the updates regularly and transmits them to the navigation client.The shortest path is computed on the required index.In our paper,the traffic updates are given importance not the graph structure updates of the road network.Every client receives the graph structures periodically In fig 1,LTI system consists of road network and road monitors for analyzing traffic updates.The traffic provider transmits the index to the server broadcaster.The server broadcaster constructs the index containing the header and data.

Constructing the index:

The index is constructed using the hierarchical index structures.The road network is considered as a graph G.Graph G(V,E) consists of nodes and edges connecting the nodes.The graph is splitted into smaller sub graphs (SG1,SG2,SG3……)and the hierarchical index structure is constructed.In fig 2,the road network is splitted into 10 sub graphs and the hierarchical tree structure is shown for the corresponding subgraphs.For example,the subgraph SG2 consists of {e,d} nodes and the edge {(e,d)} connecting the two nodes.There is also connectivity between the subgraphs using the edges.for example,the subgraphs SG6 and SG7 are connected using the edges {(g,h)} and {(f,i)}.The hierarchical index structures store the pre computed edges.so that the shortest path is computed quickly.for example to compute the shortest path using hierarchical structure,bottom-up procedure is used to retrieve the path from index.Based on the graph G, still smaller search graph GQ is constructed.for example,the shortest path to q(b,d) is computed using the sub graphs SG1,SG2,SG3.

Cost analysis:The space required for the hierarchical index is given as

$$|I|=\sum(|Vsg| + |Esg| + |Tsg| + |Psg| + tree \qquad (1)$$

Where Vsg represents the nodes,Esg represents the edges connecting the nodes,Tsg represents the connection between the child entries,Psg represents the precomputed information.Tree space is negligible,Vsg and Esg ,Tsg are directly obtained from the graph structure.Hence the space is optimized as

$$|I|=|G|+\sum|Psg| \qquad (2)$$

Index is minimized by reducing the precomputed information.The precomputed information |Psg| is minimized by splitting the graph into more sub graphs. The search space of query q is defined as

$$S(I,q)=|Es \cup Et \cup \{Tsg \cup Psg : \forall SG \in G\}| \qquad (3)$$

To minimize the S(I,q) there are three objectives.

O1-The number of leaf entries in the Es and Et.
O2-The precomputed information Psg
O3-The number of edges in search graph

All the three objectives are interrelated. Objective O2 is achieved by applying graph partitioning.As the precomputed information Psg is achieved the number of leaf entries In Es and Et are reduced.Hence,O1 is also achieved indirectly.To achieve the objective O3,stochastic process is applied.

Graph Partitioning:

The number of subgraphs created depends on the factor ϒ.Minimising the precomputed information is nothing but minimizing the overhead of index I.

$$Obj(I)=\min \frac{\sum|Psg|}{\min\{|Vsg|\}} \qquad (4)$$

Min{|Vsg|} is the normalized factor. Minimising the index refers to finding the best cheeger cut.A cheeger cut is defined as the number of edges removed from the graph such that the graph is splitted into sub graphs.In [18],the quadratic discrete

problem defined gives the best cheerger cut value.The cheeger cut is defined by the second small eigen value λ and also its eigen vector υ.the eigen vector is further decomposed to minimize the objective function.The border nodes are removed to enhance the quality of the cheeger cut.

Construction of index using the Stochastic process:

The objective O3 is obtained using the stochastic process.The graph partitioning returns only the tree index.The tree index satisfies only the first two objectives.The average size avg(S(I')) is found using stochastic algorithm.The search graph is studied at every partitioning of the graph.The smallest search graph is attached to the index.In this process an algorithm is run using the priority queue and the index.The

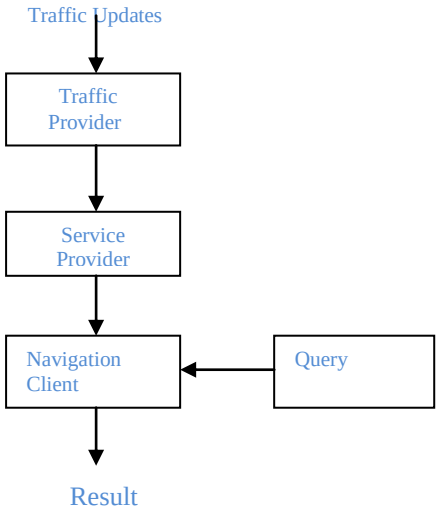

Fig 1:System architecture

cheeger cut ϒ is very important in constructing the index.The small value of ϒ gives the large number of leaf in the graph

Transmission of LTI index:

In the various broadcasting schemes,(1,m) interleaving scheme [23] is used in our approach.For an instance broadcasting cycle with m= 3 packets is shown where the information set involves six data gadgets. First, the server partitions the data set into m equi-sized information segments. Every packet contains a header and a data phase, the place a header describes the broadcasting time table of all packets. On this example, the variables i and n in every header characterize the last broadcasted item and the whole number of items. The server periodically declares a series of packets.

We use a instance to explain how a client receives her data from the broadcasted channel. Feel that a client needs to question for the data object o5.In the broadcast channel,the client tunes and waits until the next header is broadcasted. For example, the client is taking note of the header of the first packet, and finds out that the third packet has o5. With the intention to save power, the client sleeps except the broadcasting time of that packet. Then, the client wake-usp and listens to the header which contains the data packet.

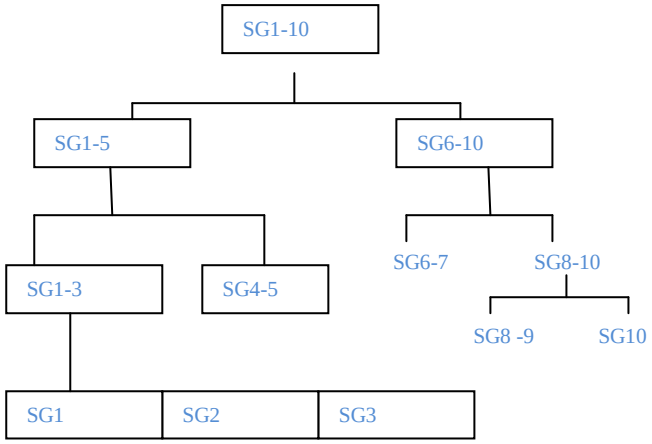

Fig 2:Hierarchical index structure

LTI Transmission on Air:

To broadcast a hierarchical index utilising the (1,m) interleavingscheme, we first partition the index into two add-ons:the index structure and the burden of edges. The former retailers the index structure and the other one stores the weight of edges. To be able to preserve the freshness of LTI,our method is required to broadcast the today's weight of edges periodically.Table 1 suggests the layout of a header/data packet in our model. Id is the offset of the packet in the present broadcast cycle and checksum is designed for error-checking of the header and information. Word that the packet does not preserve any offset knowledge to the following broadcast cycle or broadcast section.The offset will also be matched up with the aid of the corresponding identification due to the fact the constitution of LTI is pre-saved at each client. In our model, the header packet outlets a time stamp set T for checking new updates and information loss healing.

Client receiving index from the air:

The procedure for how the client receives the air index is shown here.For the LTI structure shown in fig 3,the broadcast cycle of the client is shown in the fig 4.The broadcasting scheme used is (1,2) interleaving and the edge weights of subgraphs are stored in the data packet.For example,in the subgraph SG2 the edge weights are stored in the second packet.if the driver is moving from node b to node d and the navigation system first listens to the air index at the third packet of segment 1.

offset	1	2	3	4	5	6	7		9	10	11	12	13	14	15	16
0	Id				checksum											

Table 1: Air index format

.As shown in fig 4,the navigation system of the client sleeps until it receives the segment 3 because the search graphs are located at SG1-SG3 and SG4-SG5.The other segments are collected in the next broadcast cycle.To update the subgraphs,the navigation system checks the time stamp T which is present in the header packet.For example,there are two edge updates ,one graph edge in SG5 and shortcut in SG4-5.So,the navigation system checks the Stamp regularly whether the graph edge and shortcuts are updated or not.

IV.SYSTEM IMPLEMENTATION

All the above mentioned techniques are put together for computing the shortest path.There are two algorithms,one algorithm is run at the client and other algorithm is run at the server. The first step is to construct the index.In service algorithm,the service provider constructs the index.First the live traffic data is put in the graph structure.The graph is partitioned into smaller graphs namely SG1,SG2……and also the graph partitioning algorithm is applied such that the index is optimized.The service provider collects the live traffic update periodically and updates the subgraphs.The client executes the client algorithm.Based on the source and destination,the client generates the search graphs.Subsequently,the client listens to
the header segments.Based on these segments,it updates the search graph and the shortest path is computed on the search graph.

Stochastic partitioning algorithm:

Partition(G:the graph, y:the number of partitions)
1: (z,v):= G and n:=root of I
2: insert (n,G,v,z) into PQ in decreasing order to z
3:while |PQ| < y do
4: (n,G,v,z):=PQ.pop()
5: for k:=2 to y-|PQ|+1 do
6: decompose G into SG1……..
7: form a temporal index I' that attaches SG1
8: if avg(S(I')) is better than best s then
9: update best s and best SG:={ SG1……}
10: attach best SG as n's children
11: for i:=1 to |best SG| do
12: insert (n,SG,v,z) into PQ
13: return

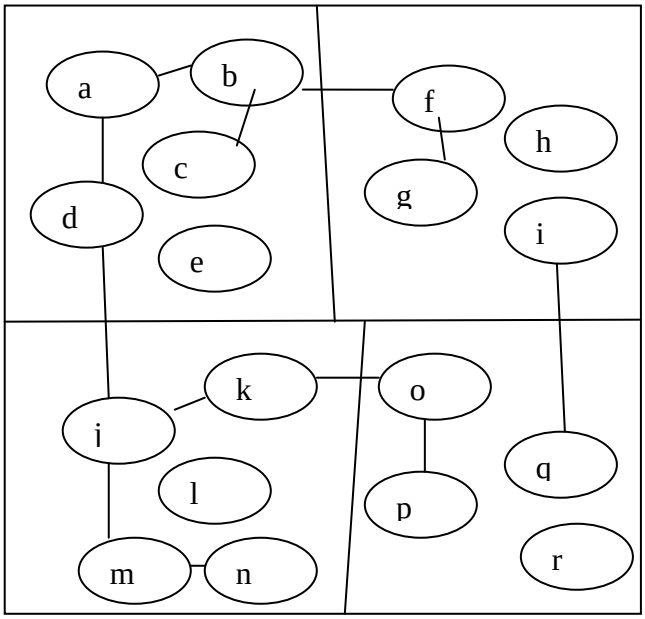

Fig 3:subgraphs

The above stochastic process is done at the service provider,where a priority queue PQ is maintained and the index is inserted into the queue.The average search graph is found out and updated based on it.

Id	1	Id	2	Id	3	Id	4	Id	5	Id	6	Id	7	Id	8	Id	9
i	0 I 0 0 0	SG1 data	SG2 data	SG3 data	SG4 data	SG5 data		0 I 0 0 0	SG6 data	SG7 data	SG7 data	SG8 data	0 I 0 0 0	SG9 data	SG10 data	SG1-3 data	SG4-5 data

Fig 4:Client tune in procedure

Experimental results

The experimental results shows that using Live traffic index the shortest path is computed between nodes.The shortest path is computed very fastly because the traffic data is broadcasted in the index format.

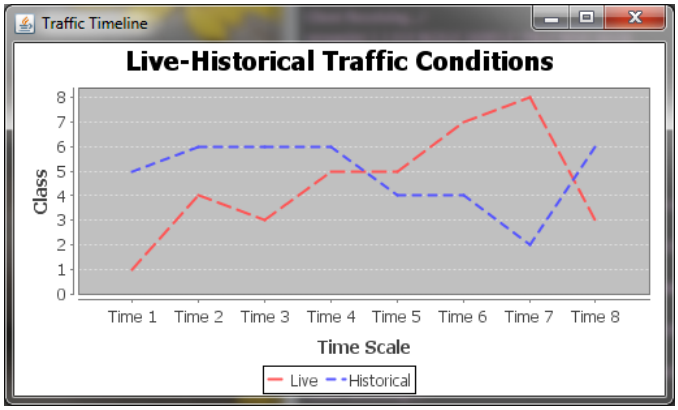

Fig 5 Live and Historical traffic conditions

In fig 5,a graph is plot on the Live traffic data and historical traffic data.By gathering the Live traffic data there are many changes fluctuations in the graph where as by using the historic data there are no fluctuations.

V.CONCLUSION

In online shortest path computation using live traffic index ,the shortest path result is computed/updated based on the live traffic circumstances. The existing work is carefully analysed and their inapplicability to the problem (due to their prohibitive maintenance time and large transmission overhead) is discussed. To address the problem, a promising architecture that broadcasts the index on the air is suggested. An important feature of the hierarchical index structure which enables us to compute shortest path on a small portion of index is identified and the index is minimized using the graph partitioning and stochastic process. Further,the index is broadcasted using the broadcasting scheme and enables the navigation client to find the path from source to destination.

ACKNOWLEDGEMENT

I feel privileged to offer my sincere thanks and deep sense of gratitude to **Dr. C. Kiranmai** (Principal & Professor), **Dr. B. Raveendra Babu** (Professor and Head of the Department, CSE Department),**Mrs.R.Vasavi** (Project Coordinator, CSE department, Assistant Professor), of VNRVJIET for extending their co-operation in doing this project. My special thanks to all the Faculty members of the Department for their valuable advices at every stage.

REFERENCES

[1] " NAVTEQ Maps and Traffic," http://www.navteq.com, 2014
[2] "TomTom NV," http://www.tomtom.com, 2014.
[3] "Google Maps," http://maps.google.com, 2014.
[4] "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2010-2015," 2011.
[5] N. Malviya, S. Madden, and A. Bhattacharya, "A Continuous Query System for Dynamic Route Planning," Proc. IEEE 27th Int'lConf Data Eng. (ICDE), pp. 792-803, 2011.
[6] G. Kellaris and K. Mouratidis, "Shortest Path Computation on Air Indexes," Proc. VLDB Endowment, vol. 3, no. 1, pp. 741-757, 2010.
[7] J.X. Yu and K.-L. Tan, "An Analysis of Selective Tuning Schemes for Nonuniform Broadcast," Data and Knowledge Eng., vol. 22,no. 3, pp. 319-344, 1997.
[8] G.Dantzig, Linear Programming and Extensions, series Rand Corporation Research Study Princeton Univ.Press, 1963.
[9] A.V. Goldberg and R.F.F. Werneck, "Computing Point-to-Point Shortest Paths from External Memory,"Proc. SIAM Workshop Algorithms Eng. and Experimentation and the Workshop Analytic Algorithmics and Combinatorics(ALENEX/ANALCO), pp. 26-40, 2005.
[10] M. Hilger, E. K€ohler, R. M€ohring, and H. Schilling,"Fast Point-to-Point Shortest Path Computations with Arc-Flags," The Shortest Path Problem: Ninth DIMACS Implementation Challenge, vol. 74,pp. 41-72, American Math. Soc., 2009.
[11] D. Delling and D. Wagner, "Landmark-Based Routing in Dynamic Graphs," Proc. Sixth Int'l Workshop Experimental Algorithms (WEA),pp. 52-65, 2007.
[12] R.J. Gutman, "Reach-Based Routing: A New Approach to Shortest Path Algorithms Optimized for Road Networks," Proc. Sixth Workshop Algorithm Eng. and Experiments and the First Workshop Analytic Algorithmics and Combinatorics (ALENEX/ANALC),pp.100-111, 2004.
[13] B. Jiang, "I/O-Efficiency of Shortest Path Algorithms:An Analysis,"Proc. Shortest Path Queries," Proc. 13th Ann.European Conf.
[14] P. Sanders and D. Schultes, "Highway Hierarchies Hasten Exact Shortest Path Queries," Proc. 13th Ann.European Conf. Algorithms(ESA), pp. 568-579, 2005.
[15] R. Geisberger, P. Sanders, D. Schultes, and D. Delling,"Contraction Hierarchies: Faster and Simpler Hierarchical Routing in Road Networks," Proc. Seventh Int'l Workshop Experimental Algorithms (WEA),pp. 319-333,2008.
[16] S. Jung and S. Pramanik, "An Efficient Path Computation Model for Hierarchically Structured Topographical Road Maps," IEEE Trans. Knowledge and Data Eng., vol. 14, no. 5, pp. 1029-1046, Sept.2002.
[17] E.P.F. Chan and Y. Yang, "Shortest Path Tree Computation in Dynamic Graphs," IEEE Trans. Computers, vol. 58, no. 4, pp. 541-557, Apr. 2009.
[18] T. B€uhler and M. Hein, "Spectral Clustering Based on The Graph – Laplacian," Proc. Int'l Conf. Machine Learning (ICML), p. 11, 2009.
[19] T. Imielinski, S. Viswanathan, and B.R. Badrinath, "Data on Air:Organization and Access," IEEE Trans.Knowledge and Data Eng.,vol. 9, no. 3, pp. 353-372.