IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Optimizing Energy Efficiency in Wireless Sensor Networks (WSN) via Parallel Concentric Circle using Rotated Itinerary Structure Based KNN (PCIKNN) Query Processing Technique.

Anand Nayyar

Department of Computer Applications & IT KCL Institute of Management and Technology, Jalandhar, Punjab, India

anand_nayyar@yahoo.co.in

Abstract

Nowadays, the interest cum research is picking up in area of Wireless Sensor Networks. Wireless sensor networks are being used in various monitoring applications (e.g., environmental monitoring and military surveillance) over a wide geographical region. In these applications, spatial queries that collect data from wireless sensor networks play an important role. One such query is the K-Nearest Neighbor (KNN) query that facilitates collection of sensor data samples based on a given query location and the number of samples specified (i.e., K). Recently, itinerary-based KNN query processing techniques, which propagate queries and collect data along a predetermined itinerary, have been developed. Previous studies show that itinerary-based KNN query processing algorithms are able to achieve better energy efficiency than other existing algorithms developed upon tree-based network infrastructures. However, how to derive itineraries for KNN query based on different performance requirements remains a challenging problem. In this research paper, we propose a Parallel Concentric-circle using rotated Itinerary structure based KNN (PCIKNN with Rotated itinerary) query processing technique that derives different itineraries by optimizing either query latency or energy consumption in Wireless Sensor Networks.

Keywords: Wireless Sensor Networks, Energy Efficiency, K-nearest neighbor query, PCIKNN, Query Optimization in WSN.

Introduction

Recent advances in micro-electro-mechanical systems (MEMS) technology, wireless communications [2] and digital electronics have enabled the development of low-cost, low-power, multifunctional sensor nodes that are small in size and communicate untethered in short distances. These tiny sensor nodes, which consist of sensing, data processing, and communicating components, leverage the idea of sensor networks based on collaborative effort of a large number of nodes. WSN nodes are battery powered, the routing protocol should consume less energy and also it should be ensured that the information transfer delay should be less. Due to the importance of geographical features in WSN applications, spatial queries [6] that aim at extracting sensed data from sensor nodes located in certain proximity of interested areas become an essential function in WSNs. Use of this WSN is implemented in the data mining concept which proved to be effective one in terms of energy consumption and query latency. Existing system deals with the use of GPSR in the routing phase, KNN boundary estimation and spatial irregularity in order to route the query using this GPSR protocol .we optimize the performance in terms of energy consumption and query latency of using GPSR in WSN.

Itinerary based KNN query processing in Parallel Concentric Itinerary based KNN (PCIKNN)

Itinerary-based query processing algorithm [1] based on optimized parallel concentric-circle itineraries, namely PCIKNN. We have three Phases 1) Routing phase 2) The KNN boundary estimation phase 3) The query dissemination phase. It is shown with example in Fig 1. Initially, a KNN query, issued at a source node, is routed to the sensor node nearest to the query point q (referred the home node) at the routing phase. Next, in the KNN boundary estimation phase, the home node estimates an initial KNN boundary (i.e., the solid boundary line circle in Fig. 1a), which is likely to contain K nearest sensor nodes from q. Finally, in the query dissemination phase (as shown in Fig. 1b), the home node propagates the query to each node within the estimated initial KNN boundary. While the KNN query propagates along certain well-designed itineraries, query results are collected at the same time.

GPSR in PCIKNN

GPSR protocol [12] is the earliest geographical routing protocols for adhoc networks which can also be used for WSN environment. The GPSR adapts a greedy forwarding strategy and perimeter forwarding strategy to route messages. It makes uses of a neighborhood beacon that sends a node's identity and its position. GPSR, PCIKNN [1] adopts two modes (i.e., the greedy mode and the perimeter mode) in the KNN query propagation. In the greedy mode, a message is forwarded by selecting the sensor node making the most progress toward the destination. When a void region is encountered, PICKNN switches to the perimeter mode and continues to move forward.

Since void regions may appear in branch-segments, peri-segments, and return segments, we develop methods to bypass void regions, correspondingly. When a KNN query reaches a void region on a branch segment, the KNN query is split into two KNN query threads. We classify the left KNN query thread and the right KNN query thread. The left KNN query thread continues to move forward by using the left-hand rule to select the next Q-node close to the branch-segment. The right KNN query thread acts similarly based on the right-hand rule. If these two KNN query threads reach a concentric-circle, both two KNN query threads fork two additional KNN query threads and move forward along the peri-segments. Note that after bypassing void regions, both the left and the right KNN query threads will merge into one KNN query that keeps propagating along the branch-segment.

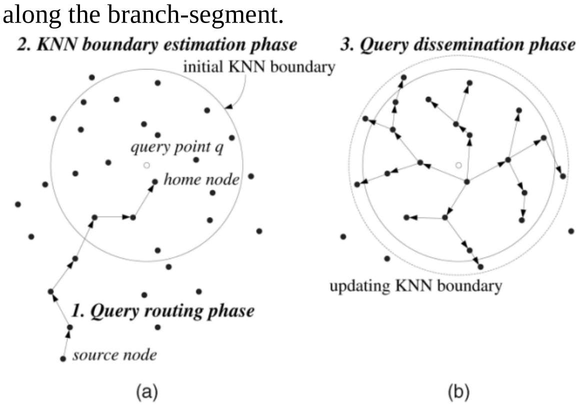

Fig.1. An overview of itinerary-based KNN query processing.

Overview Of Pciknn

The below diagram gives the overall idea of PCIKNN with Customized GPSR see fig 2.In the diagram mentioned below source node is the arbitrary node where we propagate the initial spatial query to it. And Home node which estimates the boundary estimation after the itinerary propagation of query using GPSR in the routing phase.

Routing phase and KNN boundary estimation phase

Routing is decided using the respective head set members in the sensor nodes where source node is called the arbitrary node [1]. The head set is decided on a routine basis with reference to the energy level of the signal received to the source node at the time of reception of spatial query. A query message with its Q is geographically routed from the sink node s to the nearest neighbor (i.e., the home node np, where p denotes the number of hops along the routing path) around the query point q. From the Fig 1 the information of the sensor network is gathered along with the routing procedure without the aid of any infrastructure. Upon receiving Q and the collected information from the previous phase, the home node estimates a searching boundary, named KNN boundary, with radius R by using an efficient (specifically, linear time) KNNB algorithm. The estimated boundary is not fixed and will be dynamically adjusted (by the other nodes) as long as additional information is available in the next phase.

Design of Concentric-Circle Itineraries

Given a query point q and an estimated KNN boundary [1], the area within the boundary can be divided into multiple concentric-circle itineraries. KNN query is first propagated along branch-segments in each sector. Along the branch-segment, a Q-node broadcasts a probe message and collects partial results from D-nodes within the region width of w. For each sector, when the KNN query reaches one of the concentric-circles, two KNN query threads are forked to propagate along the two peri-segments, while the original KNN query continues to move along the branch-segment. To propagate a KNN query in two peri-segments, the Q-node in the branch segment first finds two Q-nodes in peri- mode. After bypassing voids, it changes back to the greedy segments and evenly divides the partial query result collected to these two Q-nodes.

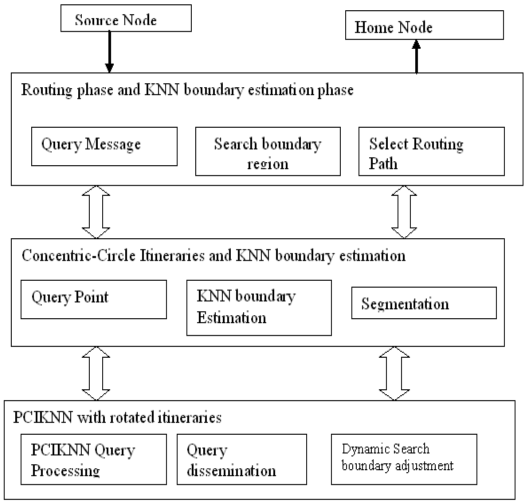

Fig 2: Architectural diagram of PCIKNN.

Rotated Itinerary Structures of PCIKNN

Here a rotated itinerary structure in PCIKNN is dealt in order to known the boundary estimation. Without loss of generality, we consider an example, where the number of concentric-circles is 4 (i.e., C = 4).

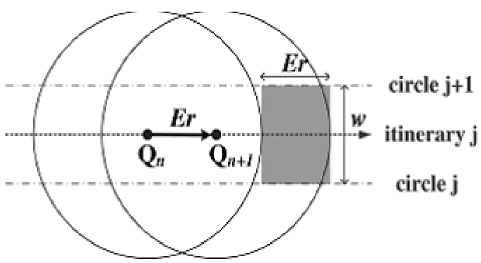

Fig 3: Design of Concentric circle.

As shown in Fig. 3, for each concentric-circle, the branch-segment is rotated by θ degree. Assume that S is the number of sectors and C is the number of concentric-circles. The rotation angle θ is set to 2π/S × C such that the return-segments of all concentric-circles are evenly distributed within a KNN boundary. The dotted straight lines in Fig. are the return- segments. For each concentric-circle, sensor nodes along the return-segments are different, and thus, the energy consumption of sensor nodes along the return-segments is balanced. While the rotated itinerary of PCIKNN deals with the energy exhaustion problem of sensor nodes along the return-segments, the query latency is slightly increased because the total length of branch-segments in one sector is increased.

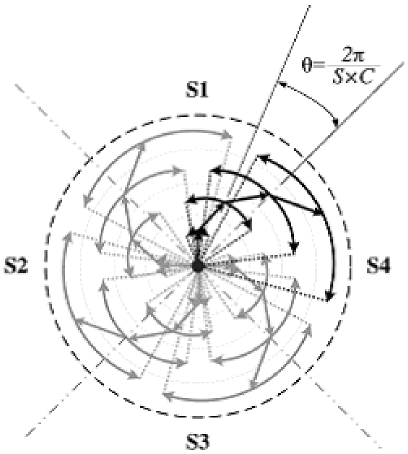

Fig 4: Parallel concentric itineraries in PCIKNN

Dynamic KNN boundary estimation

Network information collected during the routing path of KNN query is used to derive the network density, which, in turn, is used to estimate a KNN boundary. In the fig 5, we describe how PCIKNN updates these two values during the routing phase. Message transmitting from node Ni to node N+1; the gray area is the newly explored area, denoted by EA i. The number of sensor nodes in EA i is denoted by inc i+1. By adding inc i+1 to Num, we have the updated number of nodes encountered so far.

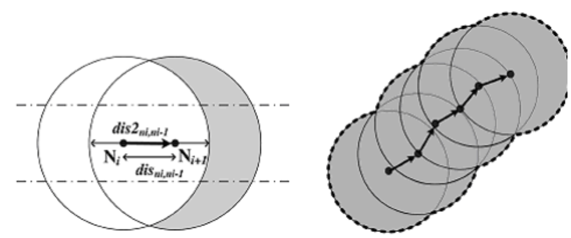

Fig: 5 Coverage areas estimated in the routing phase.

Spatial Irregularity

When a message (with the KNN query or partial results) reaches a void region, the message needs to find a way to bypass the void. Similar to GPSR, PCIKNN adopts two modes (i.e., the greedy mode and the perimeter mode) in the KNN query propagation. In the greedy mode, a message is forwarded by selecting the sensor node making the most progress toward the destination. When a void region is encountered, PICKNN switches to the perimeter mode. After bypassing voids, it changes back to the greedy mode and continues to move forward. Since void regions may appear in branch-segments, peri-segments, and return segments, we develop methods to bypass void regions, correspondingly. Fig. 5a shows an example of bypassing void region along the branch-segments and peri-segments, where gray areas are void regions. In the figure,

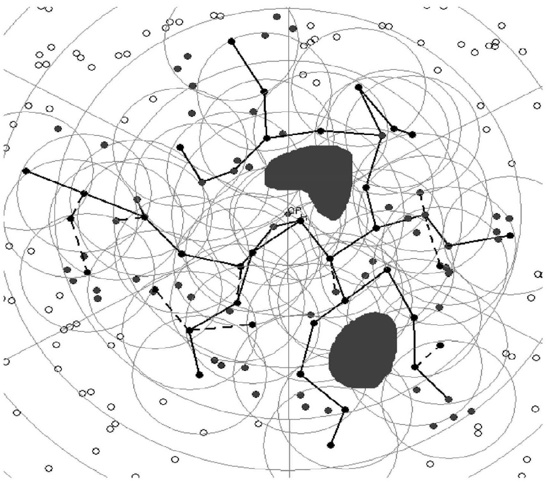

Fig 5a: An example of bypassing void regions in PCIKNN.

the bold and dotted lines refer to the KNN query propagation along the branch-segments and peri-segments, respectively. Also, the black and gray nodes are Q-nodes and D-nodes, respectively. When a KNN query reaches a void region on a branch segment, the KNN query is split into two KNN query threads. We classify the left KNN query thread and the right KNN query thread. The left KNN query thread continues to move forward by using the left-hand rule to select the next Q-node close to the branch-segment. The right KNN query thread acts similarly based on the right-hand rule. If these two KNN query threads reach a concentric-circle, both two KNN query threads fork two additional KNN query threads and move forward along the peri-segments. Note that after bypassing void regions, both the left and the right KNN query threads will merge into one KNN query that keeps propagating along the branch-segment. If a void exists on a peri-segment, the KNN query will decide which rule to use according to the relative position of the peri-segment.

Performance Evolution

Simulation Model

Our simulation is implemented in Network Simulator 2 [3][4].There are 1,000 sensor nodes randomly distributed in a 500 _ 500 m2 region and the transmission range of a node is 40 m. For each sensor node, the average number of transmitting or receiving messages is 30 ms. In default settings, sensor nodes are static. For each query, the location of a query point q is randomly selected. The default value of K for each KNN query is 100. A sensed datum is 4 bytes long and the query result is not aggregated. The broadcasting period of beacon messages is 3 s. A KNN query is considered as answered when the query result is returned to the source node. In each round of experiment, five queries are issued from randomly selected source nodes. Each experimental result is derived by obtaining average results from 10 rounds of experiments. For a fair comparison, we obtain the result of PCIKNN and compare with PCIKNN using Customized GPSR with the minimum latency/energy consumption.

Experimental Results on KNN Boundary Estimation

We evaluated the proposed KNN boundary estimation With GPSR, we set the value of K to be 350. To avoid the effect of network boundary, query points in the middle region (i.e., 100 m _ 100 m) are selected. The linear regression function of PCIKNN is set to $H(dist(N_iN_{i+1})) = (-76.9166* dist(N_iN_{i+1}) + 4999.0903)$ by linear regression technique in [10]. The optimal KNN boundary is the average distance of the Kth distant nodes of all queries derived by the experiments. As shown in Fig. 6a, PCIKNN is very close to the optimal KNN boundary under various network density. However, the boundary estimated in PCIKNN has little decrement in the optimal KNN boundary. This performance graph shows that query latency is decreased by using the PCIKNN and it improve the efficiency in query latency.

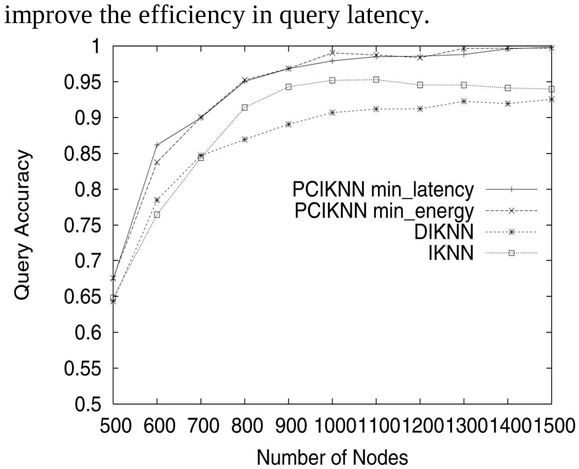

Fig. 6a Query Latency performance graph

Performance with respect to Energy Consumption

In this experiment, we evaluate PCIKNN with GPSR by varying the value of K from 50 to 400. As shown in fig 6 a, the latency of PCIKNN slightly increases because the length of branch segments is increased. It can be seen in Fig.6b that the total energy consumption of PCIKNN

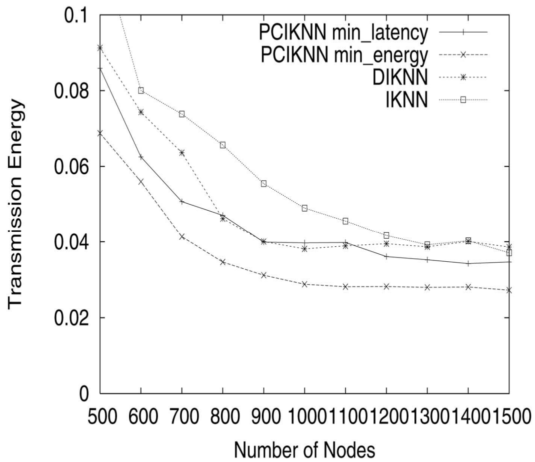

Fig. 6b Energy consumption performance graph.

Conclusion & Future Scope

In this paper, we proposed an efficient itinerary-based KNN algorithm, PCIKNN using rotated itinerary structure, for KNN query processing in the sensor network. PCIKNN disseminates queries and collects data along pre-designed itineraries with high parallelism. We derived the latency and the energy consumption of PCIKNN and then by optimizing the derived formulas, we are able to determine the appropriate number of sectors for PCIKNN. The optimized PCIKNN has some advantages which are listed as under:

- Nonparametric architecture
- Accurately determines the KNN boundary
- By pass void region in non-uniform area.

In the near future, we would like to enhance this concept by taking more simulation parameters to reduce more energy consumption and latency issues.

References

[1] Tao-Yang Fu, Wen-Chih Peng and Wang-Chien Lee,” Parallelizing Itinerary-Based KNN Query Processing in Wireless Sensor Networks.” IEEE Transactions on knowledge and Data engineering, vol. 22, pg 711-729, may 2010.

[2] F.L.Lewis,”Wireless Sensor Networks.”Technologies Protocol and Applications ed.D.J.Cook and S.K.Das , John Wiley,New York ,2004.

[3] Study of Network Simulator-2, S-38.3148 Simulation of data networks / fall-07.

[4] K. Waclena, “Lists and keyed lists.” [Online]. Available: http://www2.lib.uchicago.edu/keith/tcl-course/topics/lists.html

[5] “Ns-2 trace formats.” [Online]. Available: http://nsnam.isi.edu/nsnam/index.php/NS-2 Trace Format

[6] M. Demirbas and H. Ferhatosmanoglu, “Peer-to-Peer Spatial Queries in Sensor Networks,” Proc. Third Int’l Conf. Peer-to-Peer Computing, pp. 32-39, 2003.

[7] D. Estrin, R. Govindan, and J. Heidemann, “Next Century Challenges: Scalable Coordination in Sensor Networks,” Proc.ACM/IEEE MobiCom, pp. 263-270, 1999.

[8] H. Ferhatosmanoglu, E. Tuncel, D. Agrawal, and A.E. Abbadi,“Approximate Nearest Neighbor Searching in Multimedia Databases,”Proc. 17th IEEE Int’l Conf. Data Eng. (ICDE), pp. 503-511,2001.

[9] D. Goldin, M. Song, A. Kutlu, H. Gao, and H. Dave, “Georouting and Delta-Gathering: Efficient Data Propagation Techniques for Geosensor Networks,” Proc. NSF Worshop GeoSensor Networks,2003.

[10] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan,“Energy-Efficient Communication Protocol for Wireless Microsensor Networks,” Proc. 33rd Hawaii Int’l Conf. System Sciences,pp. 8020-8029, 2000.

[11] G.R. Hjaltason and H. Samet, “Distance Browsing in Spatial Databases,” ACM Trans. Database Systems, vol. 24, no. 2, pp. 265-318, 1999.

[12] B. Karp and T.H. Kung, “GPSR: Greedy Perimeter Stateless Routing for Wireless Networks,” Proc. ACM/IEEE MobiCom, pp. 243-254, 2000.

[13] Y.B. Ko and N.H. Vaidya, “Location-Aided Routing (LAR) in Mobile Ad Hoc Networks,” Wireless Networks, vol. 6, no. 4,pp. 307-321, 2000.