ISSN:0975-9646

INCOG QUERY PROCESSING IN NETWORKS

YETCHERLA KUMAR SEKHAR, U. NANAJI
Department of CSE, Saint Theresa Institute of Engg. & Technology, Garividi, Vizayanagaram, (A.P.), India

***Abstract*— Advances in sensing and tracking technology enable location-based applications but they also create significant privacy risks. *Anonymity* can provide a high degree of privacy, save service users from dealing with service providers' privacy policies, and reduce the service providers' requirements for safeguarding private information. However, guaranteeing anonymous usage of location-based services requires that the precise location information transmitted by a user cannot be easily used to re-identify the subject. This paper presents middleware architecture and algorithms that can be used by a centralized location broker service. The adaptive algorithms adjust the resolution of location information along spatial or temporal dimensions to meet specified anonymity constraints based on the entities who *may* be using location services within a given area. Using a model based on automotive traffic counts and cartographic material, we estimate the realistically expected spatial resolution for different anonymity constraints. The median resolution generated by our algorithms is 125 meters. Thus, anonymous location-based requests for urban areas would have the same accuracy currently needed for E-911 services; this would provide sufficient resolution for way finding, automated bus routing services and similar location-dependent services.**

I. INTRODUCTION

The low cost and small size of positioning equipment (e.g., GPS receivers) have allowed their embedding into PDAs and mobile phones. The wide availability of these location-aware portable devices has given rise to a flourishing industry of location-based services (LBS). An LBS makes spatial data available to the users through one or more location servers (LS) that index and answer user queries on them. Examples of spatial queries could be "Where is the closest hospital to my current location?" or "Which pharmacies are open within a 1 km radius?". In order for the LS to be able to answer such questions, it needs to know the position of the querying user. When a user u wishes to pose a query, she sends her location to a trusted server, the anonymizer (AZ), through a secure connection (e.g., SSL). The latter obfuscates her location, replacing it with an anonymizing spatial region (ASR) that encloses u. The ASR is then forwarded to the LS. Ignoring where exactly u is, the LS retrieves (and reports to the AZ) a candidate set (CS) that is guaranteed to contain the query results for any possible user location inside the ASR. The AZ receives the CS and reports to u the subset of candidates that corresponds to her original query. In order for the AZ to produce valid ASRs, the users send location updates whenever they move (through their secure connection). The described model is shown in Figure 1

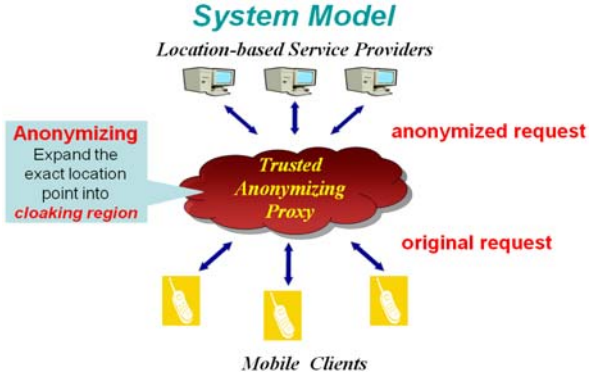


Fig 1. Location based service provider

II. RELATED WORK

Section II.I reviews related work on road network databases and Section II.II surveys the literature on spatial anonymity.

II.I SPATIAL QUERY PROCESSING IN ROAD NETWORKS

In general, a road network can be modeled as a eighted graph G = (N;E). N contains the network nodes, while E is the set of edges. Nodes n in N model road intersections, locations of road turns, or positions where traffic conditions change (e.g., a street gets narrower). On the other hand, every edge e connects two nodes and is associated with a non-negative weight w(e). Weight w(e) may represent, for instance, the traveling time from one node to the other. Figure 2 shows an example of a road network. Edge n1n2 has weight 3, and its endpoints are nodes n1 and n2. Let p be a point on an edge e with weight w(e). The partial weight from p to an end-node of e is proportional to their (Euclidean) distance, while the sum of the two partial weights is equal to w(e). For instance, object o1 (shown as a solid point) lies on edge n3n4 and has partial weights 1 and 3 from nodes n3 and n4, respectively. Similarly, user u (the hollow point) falls on edge n2n3 and both of its partial weights are 2.

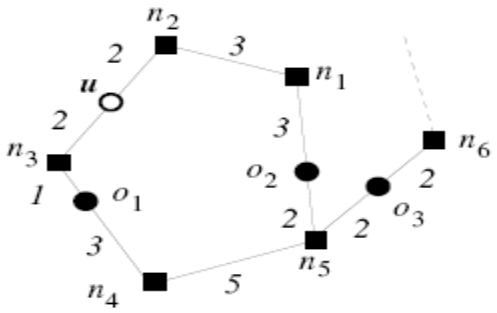


Fig. 2. Road network example

The network distance dN(u; o) between a user u and an object o is defined as the sum of edge weights along the shortest path (in the network) from u to o. In our example, the network distance dN(u; o1) between user u and object o1 equals to 2+1=3. Its derivation is strongly related to shortest path computation. In case of a small network, main memory shortest path algorithms (e.g., Dijkstra's algorithm) can be applied to compute dN(u; o). Otherwise, disk-based data structures are utilized. Query Processing by Network Expansion. Users are often interested in location-based queries such as r-range and Knn queries, in the context of a road network. Given a distance threshold r and a user location u, the r-range query returns all objects within (network) distance r from u. On the other hand, the kNN query retrieves the k objects that are closest to u. In the rest of the paper, the term distance refers to the network distance, and the r-range and kNN queries refer to their network versions (unless otherwise specified). developed efficient indexing and processing methods for the above queries. T hey proposed the following disk-based structures for indexing the road network and the data objects: (i) the adjacency index packs adjacency lists of network nodes into disk blocks, (ii) the edge R-tree

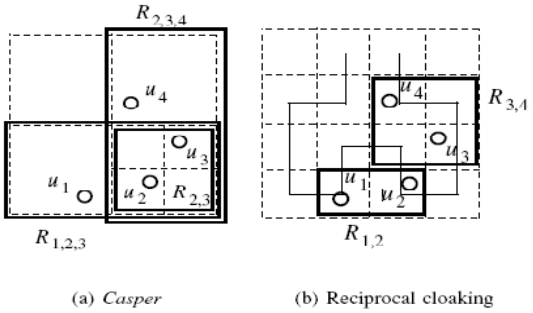
spatially indexes the network edges, and (iii) the object R-tree (ORT) organizes the locations of the data objects Network expansion is a well-known technique for evaluating r-range and kNN queries. Starting from the user location u, it discovers objects on encountered edges while traversing the network like Dijkstra's algorithm, until the query results (i.e., data objects of interest) are found. Suppose that, in Figure 2, user u issues a range query with r = 9. First, we access the adjacency index to identify edges within the query range, following the steps in Table 1. A min-heap H is employed for organizing entries of the form (ni; dN(u; ni))(for encountered nodes ni) in ascending order of distance dN(u; ni). In our example, the edge n2n3 containing u is initially identified, and its end-nodes n2 and n3 (both with distance 2) are inserted into H. In each iteration, the node ni with the minimum distance is de-heaped from H, its incident edges ninj are recorded, and its adjacent unvisited nodes nj (having dN(u; nj) within the range) are inserted into H.Tthe first three steps of Table 1, edges n2n3, n2n1, and n3n4 fall completely within the query range. However, at step 4 the de-heaped node n1 has distance 5 from u and only the partial edge n1n5(4) lies within the range r = 9. The process ontinues until H becomes empty. Having discovered the relevant edges, we probe the ORT to retrieve the result objects, o1 and o2.

Step	De-heaped entry	H contents	Found edges
1	—	$(n_2, 2)$ $(n_3, 2)$	n_2n_3
2	$(n_2, 2)$	$(n_3, 2)$ $(n_1, 5)$	n_2n_1
3	$(n_3, 2)$	$(n_1, 5)$ $(n_4, 6)$	n_3n_4
4	$(n_1, 5)$	$(n_4, 6)$	$n_1n_5(4)$
5	$(n_4, 6)$	—	$n_4n_5(3)$

TABLE 1
Steps for range search, r=9

II.II INCOG LOCATION-BASED QUERIES

Recently, considerable research interest has focused on preventing identity inference in location-based services. Studies in this area typically assume the model described in Section 1, proposing spatial cloaking (i.e., location obfuscation) techniques. In the following, we describe existing techniques for ASR computation (at the AZ) and query processing (at the LS). At the end, we cover alternative location privacy approaches and discuss why they are inappropriate to our problem setting.Spatial Cloaking at the AZ. In general, the AZ applies the concept of K-anonymity to hide the querying user location u. The idea is to compute an anonymizing spatial region (ASR), containing u and at least K - 1 other user locations. This offers privacy protection in the sense that the actual user position u cannot be distinguished from others in the ASR, even when malicious LS is equipped/advanced enough to possess all user locations. This spatial K-anonymity model is most widely used in location privacy research/applications, even though alternative models are emerging. Casper is the first work on efficient and scalable AZ implementation for ASR computation. A quad-tree is utilized for indexing user locations and deriving ASRs. Suppose that the AZ needs to compute a 2-anonymous region (i.e., K=2) for querying user u1 in Figure 3(a). The AZ first locates the leaf quad that contains u1 and traverses the tree pwards until it identifies a region covering at least K users (including u1). In this case, the AZ derives rectangle R1;2;3 (containing three users) as the 2-anonymous region of u1.



(a) *Casper* (b) Reciprocal cloaking

Fig 3: Spatial K-anonymous cloaking, K=2

III. OBFUSCATION

Obfuscation is the process of degrading the quality of information about a person's location, with the aim of protecting that person's location privacy. The term "obfuscation" is introduced, but several closely related concepts have been proposed in previous work. The "need-to-know principle" aims to ensure that individuals release only enough information that a service provider needs to know in

order to provide the required service. The idea of a need-to-know principle is closely related both to obfuscation and the fundamental fair information practice principle of consent and use. Snekennes investigates a privacy policy-based approach to enforcing the need-to-know principle in location aware computing by adjusting precision of location information

The work in develops and tests an algorithmic approach to obfuscating *proximity queries* based on imprecision. A simplified version of the algorithm introduced in is summarized in. The algorithm assumes a graph-based representation of a geographic environment (for example, a road network). An individual protects his or her location privacy by only reporting a set *O* of locations (an *obfuscation set*), one of which is that individual's actual location (figure 3.1a). For an obfuscation set *O*, the location-based service provider must compute the relation d (figure 3.1b), where *o*d *p* means *o*, *p* □ *O* are most proximal to the same point of interest (POI). The algorithm then proceeds according to three possibilities. First, all the locations in the obfuscation set may be most proximal to a single POI (*O* □ *O*/d), in which case that POI can be returned to the Second, the individual may agree to reveal a more precise representation of his or her location, in which case the algorithm can reiterate. Otherwise, the best estimate of the most proximal POI us returned . The analysis in shows that efficient mechanisms for computing the relation d can ensure that the entire algorithm has the same computational (time) complexity as a conventional algorithms for proximity queries, and that the algorithm must terminate in a finite number of iterations. Obfuscation has several important advantages that complement the other privacy protection strategies. Obfuscation and anonymity are similar, in that both strategies attempt to hide data in order to protect privacy. The crucial difference between obfuscation and anonymity is that while anonymity aims to hide a person's identity, obfuscation is an explicitly spatial approach to location privacy that aims to allow a person's identity to be revealed. Potentially, this combats one of the key limitations of anonymity

approaches: the need to authenticate users. At the same time, degrading the quality of location information makes inferring identity from location more difficult. Obfuscation is flexible enough to be tailored to specific user requirements and contexts, unlike regulatory strategies; does not require high levels of complex infrastructure and is less vulnerable to inadvertant disclosure of personal information, unlike privacy policies; and is lightweight enough to be used without the need for trusted privacy brokers, unlike many anonymity approaches. Obfuscation aims to achieve a balance between the level of privacy of personal information and the quality of service of a location-based service. Current research has indicated that there exist many situations where it is possible to expect high quality location-based services based on low quality positional information . Consequently, in situations where the user requires a higher quality of service than can be achieved at a user's minimum acceptable level of privacy, then other privacy protection strategies must be relied upon instead. Further, obfuscation assumes that the individual is able to choose what information about his or her location to reveal to a service provider. While this may be realistic when using client-based or network-assisted positioning systems and when sharing location information with a third party location-based service provider, dealing with the entities that administer network-based positioning systems still requires privacy protection based on regulatory or privacy policy approaches.

Location privacy and location-aware computing 1

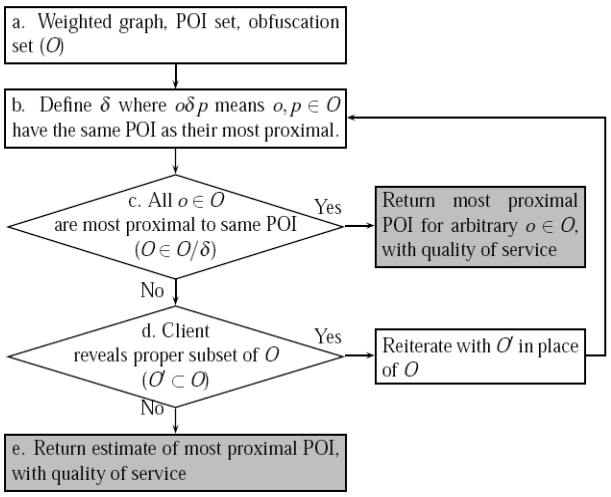


Fig 4 : location privacy and location-aware computing

IV NETWORK-BASED ANONYMIZATION

In this section, we present the cloaking algorithm of our NAP framework. Our primary objective is to guarantee reciprocity based anonymity. In NAP, the AZ nonymizes u with a set of line segments/edges instead of a spatial region (ASR). The crux of our cloaking method is to utilize a global edge ordering; i.e., an ordered sequence that contains all network edges exactly once. The edge ordering is setting-sensitive, i.e., it specifies which end-node of the edge precedes the other. We refer to the position and setting of an edge in the ordering as the edge order and the edge setting, respectively. To avoid confusion, the setting of an edge depends solely on the ordered sequence, and has nothing to do with the direction (in the case of directed networks) of the road segment it models. Figure 5(a) shows a road network, and an ordering of its edges. The number next to each edge indicates its order and the arrow its setting. The edge ordering defines an implicit linear order among the users themselves. In particular, a user u precedes another u0 if the edge of u has smaller order than that of u0. If they fall on the same edge ninj (with setting from ni to nj), u precedes u0 if it is closer to ni. Ties among coinciding users are resolved arbitrarily. This recedence relationship defines the order ordu of each user u. The position of a user in the defined sequence is referred to as the user order. The example in Figure 5(a) contains 10 users whose subscript indicates their order (i.e., user u3 has order 3, etc).Reciprocity in NAP is achieved by conceptually partitioning the user ordering into buckets of K users each, and forwarding to the LS the edges corresponding to the bucket of the querying user u. This set of edges is called the Anonymizing Edge List (AEL) of u. Specifically, let U be the set of users registered with the AZ and assume that a querying user requires anonymity of degree K. Set U is partitioned into B = bjUj=Kc buckets, each containing K users, except the last one which may contain up to 2 _K ☐1; the i-th bucket bi (for i < B) consists of users with order from (i ☐ 1) _ K + 1 to i _ K, and the B-th is assigned the remaining ones. Consider the network in Figure 5(a), where jUj = 10, and assume that u6 poses a query with anonymity requirement K = 3. This results into 3 buckets, b1 = fu1; u2; u3g, b2 = fu4; u5; u6g, and b3 = fu7; u8; u9; u10g. User u6 belongs to b2 and is anonymized together with the other users in it. The boundary users (i.e., first and last) of b2 are u4 and u6, whose edges have orders 5 and 7. The AEL is formed by collecting all edges with orders between 5 and 7; i.e., it comprises edges n6n8, n8n9, and n9n2, shown bold. We follow the above global edge/user ordering approach, because a local one could lead to privacy breach in a way similar to Figure 3(a) for Casper. Specifically, upon interception
of an AEL generated by our method, one cannot infer who among the users in the corresponding bucket was the querying one. In other words, the AEL for any user in the same bucket is identical, and therefore an adversary cannot pinpoint the query originator with a probability higher than 1=K (recall that each bucket contains K or more users). Hence, our cloaking method satisfies reciprocity. Reciprocity, in turn, is a sufficient condition for anonymity [23] and, thus, NAP guarantees K- anonymity to the querying users. In the rest of this section, we describe edge ordering strategies. Then, we present particular techniques for the anonymization procedure. Finally, we analyze the properties of the proposed edge orderings.

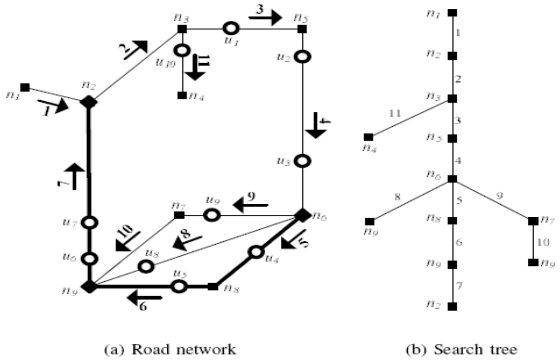


Fig 5: Edge ordering

V ANONYMIZATION PROCEDURE

Given an edge ordering, the next question is how AEL computation can be implemented efficiently at the AZ. Parameter K is not known in advance and varies, since different users have different anonymity requirements, and even queries by the same user may specify different K, depending on the nature of the queried data. As buckets are defined according to K, they cannot be explicitly materialized. Instead, the AZ employs an index that keeps the users sorted on their order and allows efficient AEL computation for arbitrary K. The index is an aggregate B-tree (similar to an aggregate R-tree), whose internal nodes keep for each child the number of users in the corresponding sub-tree. Figure 6 shows this tree in the example of Figure 5(a). For each user (e.g., u6) we store the ID of the edge it falls on (n9n2), the edge's order (7), and its distance from the edge's first end-node (jn9u6j). The latter two values are used (primarily the edge order and secondarily the distance from the first end-node) as the sorting key of the tree.

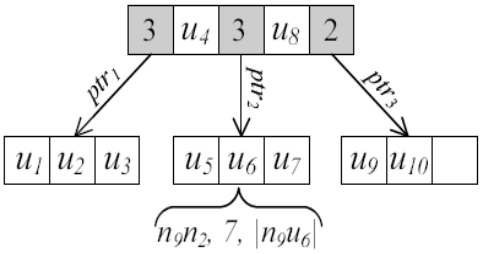


Fig 6: Aggregate B-tree

In Figure 6 the numbers in the shaded boxes correspond to the aggregate information maintained, i.e., the cardinalities of the sub-trees rooted thereof. Note that we use a B-tree instead of a B+-tree (i.e., user information is also stored in internal nodes), because it is faster for in-memory indexing.

V.I ANONYMOUS QUERY PROCESSING

In this section we describe AEL query processing at the LS; we present algorithms for minimal and inclusive CS computation for a single query, we propose additional optimizations for the case where multiple AEL queries are processed in a batch. we demonstrate the generality of NAP with respect to the network storage scheme used at the LS.

a) Single Query Processing

Processing is based on a direct implementation of the theorem uses (network-based) search operations as off the-shelf building blocks. Thus, the NAP query evaluation methodology is readily deployable on existing systems, and can be easily adapted to different network storage schemes, as we discuss in Section 5.3. As a case study, in this section we focus on the storage scheme and the network expansion framework, in order to provide a concrete NAP prototype.

Algorithm 1 Candidate Set Computation in NAP

Search(AEL L)

1. Identify the border nodes of L
2. Query the ORT and collect all objects falling on L edges
3. For every border node n
4. Perform range (or NN) search with parameter r (k) at n
5. Form CS as the union of objects retrieved in steps 2 and 4

b) Batch query processing

The LS processes queries in discrete timestamps, and multiple AEL-based queries may be arriving in the same timestamp. In this case, the queries are evaluated in a batch. Below we propose strategies aiming at maximizing computation sharing among different queries.

VI EXPERIMENTAL EVALUATION

we evaluate the robustness and scalability of our proposed methods on a real road network. Our algorithms were implemented in C++ and experiments were executed on a Pentium D 2.8GHz PC. We measured the average of the following performance values over a query workload of 100 queries: (i) anonymization time and refinement time at the anonymizer AZ, (ii) I/O time and CPU time for query processing at the location server LS, and (iii) the communication cost (in terms of transmitted points) for the anonymizing edge list AEL and the candidate set CS.

Scalability Experiments

In this section, we investigate the scalability of NAP with respect to various factors. To provide an indication of the space requirements, we note that for the largest tested data sizes (i.e., jUj=200000 and jOj=1024000), the AZ uses only 12.5 MBytes of main memory (including the network graph) and the LS needs a total of 23.5 MBytes hard disk storage. End-to-end time. Before a lower level study, we present an experiment on the overall response latency. Specifically, from the user's viewpoint, the end-to-end time captures the elapsed time between issuing a query and obtaining the results. It includes the processing time at AZ, the computation time at LS, and the communication time between AZ and LS.

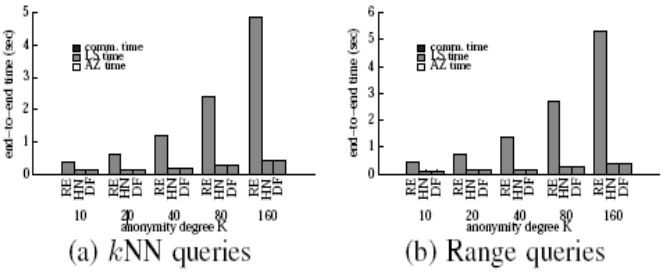


Fig 7: End-to-end time vs. anonymity degree K

Figure 7 shows the end-to-end time as a function of the anonymity degree K, assuming a communication bandwidth of 10Mbps. Clearly, the processing cost at LS dominates the end-to-end time, while the communication (between AZ and LS) and the AZ computations account only for a small percentage of the total time. It is worth mentioning that the processing (including anonymization and refinement) at AZ takes 0.000620 seconds for RE. HN and DF have similar costs. This implies that the AZ is capable of serving 1600 requests per second.

CONCLUSION

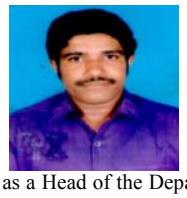
In this paper, we propose the network-based anonymization and processing (NAP) framework, the first system for K-anonymous query processing in road networks. NAP relies on a global user ordering and bucketization that satisfies reciprocity and guarantees K-anonymity. We identify the ordering characteristics that affect subsequent processing, and qualitatively compare alternatives. Then, we propose

query evaluation techniques that exploit these characteristics. In addition to user privacy, NAP achieves low computational and communication costs, and quick responses overall. It is readily deployable, requiring only basic network operations. In the traditional spatial anonymity model, the data owner (e.g., a location-based service) makes its data available using a location server. It may, however, be the case that the owner is outsourcing its database to a third-party (and, thus, untrusted) location server. A challenge here is how to encrypt the owner's data so that they are hidden from the location server, while it can still process anonymous queries. Another interesting question is how (anonymous) users could verify that the location server did not tamper with the original owner data.

REFERENCES

1. http://www.anonymizer.com/.
2. N. Mishra, R. Motwani, U. Srivastava, D. Thomas, J. Widom, and Y. Xu. Vision Paper: Enabling Privacy for the Paranoids. In VLDB, 2004.
3. R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Hippocratic Databases.In VLDB, 2002.
4. A. R. Beresford. Location privacy in ubiquitous computing. PhD thesis,Computer Laboratory, University of Cambridge, 2005.
5. T. Brinkhoff. A Framework for Generating Network-based Moving Objects. GeoInformatica, 6(2):153–180, 2002.
6. A. R. Butz. Alternative Algorithm for Hilbert's Space-Filling Curve.IEEE Trans. Comput., C-20(4):424–426, 1971.
7. C.-Y. Chow and M. F. Mokbel. Enabling private continuous queries for revealed user locations. In SSTD, 2007.
8. C.-Y. Chow, M. F. Mokbel, and X. Liu. A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In GIS, 2006.

Mr. Y. Kumar Sekhar received the B.Tech Degree from JNTU, Hyderabad in 2005 and He is currently pursuing M.Tech in the Department Of Computer Science and Engineering in Saint Theresa Institute Of Engg & Tech Garividi, Vizianagaram, Of JNTUK Affiliation. His research interests include Data Mining and Computer Networks.



Uppe.Nanaji received the B. Tech degree from JNTU, Hyderabad, India and the M. Tech degree in Computer Science Technology from GITAM College Of Engg Of Andhra University Affiliation in Vishakhapatnam in 2003, and he is currently pursuing the Ph. D in Computer Networks from Andhra University Visakhapatnam. He is working as a Head of the Department for CSE in Saint Theresa College Of Engg & Technology Garividi, Vizianagartam (Dist) India. His research interests include Computer Networks & Data Ware Housing