RFID: Tracking And Scratch Prophecy in Real Time Wireless Environment

M. Thurai pandian[1] R.Sukumar[2]

[1]Asst.Prof, Department of CSE, [2]Professor, Department of CSE

[1]Lord Jegannath College of Engg and Tech., Psn Nager, Ramanathichanputhur, K K dist-629 402

[2]Sethu Institute of Technology, Puloor, Kariapatti, Virudhunagar dirst- 626 106

Abstract— Nowadays the usage of vehicles by the Public is tremendously increasing day by day. In all the countries, lacks and lacks of vehicles running on the road daily. To improve the traffic condition and reduce the vehicle crime we have to launch a project for tracking the vehicles. The challenging of this project is to locate the current position and status of the moving and static vehicles. By using RFID Technology with the help of HINA (Hierarchically Interconnecting Nodes Algorithm) we can tackle this problem. The vehicles enter the local nodes area the information about the vehicles in distributed throughout the city. The main server is control all the information. If suppose the vehicle get accident the server automatically redirect to the ambulance service which in very near that location.

Keywords- RFID, HINA, Traffic, Accident, Vehicle tracking.

I. INTRODUCTION

Intelligent transportation systems are mostly used to track the vehicles [1], [2], [3]. ITS helps to coordinate traffic condition, track the vehicle for improving safety. In this paper we discuss about the shanghai city, Even in 2001, the average number of vehicles running across per minute in daytime was up to 33[4]. According to the accident no one can spend the time to save the peoples who met accident. This project will construct the basic infrastructure and collects the large number of information about the shanghai city. According to this paper the Wireless Access Points (APS) and RFID Reader are deployed throughout the city, To deploying the large number of local nodes actively captioned and logged in a large number of vehicles in the city.

The main goal of the project in first track the vehicles efficiently from the available transportation infrastructure, second prediction of Accidental spots and call the Ambulance automatically. In particularly some situations we have to track the vehicle urgently in such as stolen cars, speedy cars, ambulances, public cars also accidental cars and spots. However the tracking transportation is in metro potion area in very challenging one. We want to track the car just route the query in the local node, the query in processed and routed to the retold by the hopping method. The last node will track the vehicles (where the vehicles to travelled that local node acts as a last node or boundary node) and directly return back to the query node. (Here suppose the RFID tag gets damaged we can't track the vehicles in this time the Distributed data base will give the information about the vehicles and its last location.)The HINA protocol will help to track the vehicle where it is located currently. The exact position of the moving vehicles will track by the HINA. The distance will be calculated and find the exact location.

II. SYSTEM DESCRIPTION

As RFID technology continuously evolves, it has been widely used in tracking various mobile objects such as vehicles [5], [6]. The US government also enacts the TREAD Act which demands RFID tags to be planted in every new tire before September 2007[7].The infrastructure for shanghai city, which is still underway in illustrated in fig 1. RFID readers and wireless APS will be deployed throughout the city typically installed is the cross roads. A local node is responsible for collection data from several RFID Readers which are closely located every local node in act as a server it will accept the query and track the exact vehicles. A moving vehicle attached with an RFID tag can be captured if the emitted signal reaches some reader. The RFID tags can communicated with readers when if cross its coverage area. . A Cisco AirNet 1240AG AP working under IEEE 802.11g has an effective outdoor communication range of about 280mat the transmission rate of 2 Mbps [8]. We connect local nodes to the wide-area ATM network provided by Shanghai Telecom [9] through a dedicated connection or a cheap ADSL connection.

Fig 1. Infrastructure of Shanghai city

III. DESIGN OF HINA:

In this section we give an overview of HINA protocol and the enervative location updating base on the ancestor of a dynamically maintained hierarchy finally discussion the optional configuration of the protocol parameters of HINA.

A. Overview:

HINA managers to solve two critical insure .The first in the code area the HINA is to dynamically update the location information of moving vehicles to all the nodes in the system in a controlled way. which node(Reader) is very close to the vehicles are update more frequently than the other node, Therefore have more accurate information about the current location of the moving vehicles. Upon leveling the query the node have no accurate information then it passes to the other node which on very close to the vehicle. The key to the design of HINA is how malice is the controlled location updating while bounding the maximum number of helps a query is routed. To accomplish this HINA integrates four effective components.

1). Overlay construction

HINA deploy the local nodes in an overlay network that match the real road network. Each node has a link between their adjacent or neighboring nodes. All the local nodes are known their neighbors.

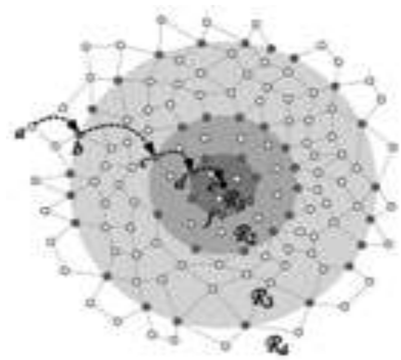

fig 2. Overlay constrction

2). Hierarchy Organization:

HINA divides the local nodes into local regions. In fig 2 illustrated the hierarchical regions of local nodes and the query routing between the nodes [12]. Which constitute the hierarchy of the overlay network? There are 3 regions R1, R_2, R_3. R_1 is the inner region; it has some local nodes. R_2 is the middle region. It has the information about the inner region. It also covers the node which is located in the inner region (R_1). R3 is the outer region of covers all the nodes which are connected in the overlay network; it has the information of the R1 and R_2 regions.

3). Restricted Location Updating:

When the vehicles are moving within R_1 the location updating involves only the small set of nodes in R_1. When the vehicle is moving out of the region R_1 the location updating is extended to move regions, In this case part of the hierarchy needs to be reorganized. This reorganization aims to restrict location updating in R_1 as much as possible. Thereby minimizing network traffic cost for location updating.

B. Query Routing:

The query can injected at any node, all the nodes will know the boundary of the region [fig 2]. The query will forward to the next region node in the hierarchical format. Finally the last node will track the vehicle and formatted to the first query. This method is avoiding the maximum number of hops.

C. Hierarchy Initialization:

The first node that captures a new vehicles trigger an initialization procedure to established the hierarchy for the vehicle. The vehicle can move any direction a region initially designed as a disk in the overlay network. Deployment of local nodes in not necessary to be uniform in the city, They can be move densely deployed for move accuracy is required, the distance measure in terms of helps is the overlay network, Each region has a radius R_1(in hops) .A node which has a distance d from the first node belongs to region R_k if this region is the smallest one that cover the node the radius r_k of R_k in

$$rk = \min_{i=1}^{h} \{ri, vi \le d\}1$$

When h is the maximum number of region in the system. If equals to certain areas the node is the on the boundary of ri . More over for query routing every node maintains a pointer that points to a node which is on the boundary of the immediate inner region. Let next insider denote this pointer.

Local node	Init. Packet	Update packet
boundary	router	router
next inside	Journey	Journey scale

Table 1. Data structure of a node

To establish next insider printer in the nodes the first node initiates initialization packet which contains a route field for setting up these points and a journey field for maintaining the distance that the product has traversed. The first node initializes route and journey to its own IP address and one, respectively. The first node foods initialization packet throughout the network upon receiving the packet a node first sets its next insides to route contained in the packet. Then it checks journey in the packet. It journey equals to the radius of certain region ri the node marks itself as a boundary node of region Ri . It also modifies route in the packet to its own IP. Otherwise it leaves that field uncharged next, it universe journey in the packet by one and rebroadcasts the packets to its neighbors. In addition duplicated initializations packets with larger journey are silently dropped after the initialization procedure terminates the regions are centered at the first node and the hierarchy is established (as illustrated in fig.2). Note that the structure of the hierarchy is distributed in all local nodes (the data structure for a node is shown in table.1)[12]. Therefore the storage overhead for tracking the vehicle at a local node is very small.

D. Restricted Location Updating:

When a vehicle is moving in the city its information is captured by the local nodes. When a node captures the vehicles it performs the location updating and maintain warranty for the vehicle.

Case 1: The capture node (chaser) is an interior node within R1

In this case the hierarchy for the vehicle remains unchanged. The capture node distribute the information about the vehicle location to the other nodes it R1.

Case 2: The capture node is a boundary node of R1.

The vehicle will move out of the region R1 shortly. In fig.3, a node a is the current chaser which is a boundary of R1[12]. When the vehicle moves out of R1 will not cover the vehicle any more. Two problems arise first a future query cannot be routed to the chaser property because the information on the boundary nodes of R1 is out of data. Second to enable the proper routing of a future query the chaser has to flood the location information of the vehicle of R2 every time, which will incur larger network traffic overhead.

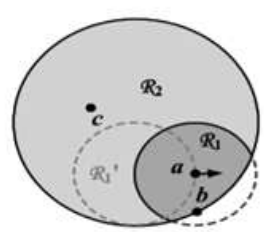

fig 3. Reconstuction of R1

Therefore HINA reorganize the R1. To this end the capture node initials on update packets in which its route and journey is initialized to its own IP address and one, respectively as in an initialization packet. The update packet includes an additional scale field that is used to indicate the area that the update packet should be propagated to in this case the capture node floods the packet within R2 by letting the boundary nodes of R2 stop the flooding. On the one hand, the new R1 is rebuilt from the current capture node within R2. At the same time location information is also updated in the new R1. On the other hand, it updates nodes in R2 about the current position of the R1.

There is a special situation during the reconstruction of R1, where the new R1 is truncated by the boundary of R2.This happen when the capture node is close to the boundary of R2. In this situation, a boundary node of R2 receives an update packet whose journey is less than or equals to r1. As a result this node sets itself as a boundary node of both R1 and R2. We call such that node is the common boundary node of R1 and R2. In this case R1 is no longer a disk because it is destructed in R2. But this does not affect the operation of our petrol.

Case 3: The capture node is a common boundary node of several regions R1, R2….Rj. This is actually a more general situation of case 2. This situation results from constant reconstruction of regions as the vehicle is moving. In this case it is possible for the vehicle to move out of all the regions from R1 to Rj.

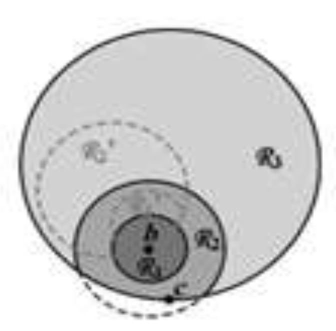

fig 4. Reconstruction of R1 and R2

The system needs to reorganize regions from R1 to Rj . In fig 4.the situation occurs it node b is the current capture node where is also a common boundary node of R1 to R2.

To rebuild regions from R1 to Rj the capture node floods an update packet within Rj+1. As a result all regions from R1 to Rj are reconstructed within Rj+1. In addition the location information of the vehicle within Rj+1 is updated , Similar to case 2 there is also a special situation during the reconstruction of regains from R1 to Rj where several regains say from Ri to Rj might be truncated by some boundary nodes of Rj+1 . Such a boundary node of Rj+1 sell itself as the common boundary node of regions

Ri, Ri+1----Rj+1(

In fig 4 node c is a resulting common boundary node of R2 and R3 [12].

IV. Design Issues:

A. *Scalability:*

HINA can track thousands of vehicles, therefore the system scalability concern in terms of the number of vehicles the number of user and the number of local nodes is initial. HINA only updates a small number of nodes nearby the vehicle; therefore the location updating cost should be small.

B. *Resilience of Unreliable data:*

It is possible that occasionally a vehicle is may not captured by the RFID reader when the vehicle moving too fast, A local node may also fail from time to time. If in critical perform the HINA of a boundary node misses a vehicle passing by this inaccuracy can be easily deflected in this system. If any time a node is region R1 (1 should have received an update packet from a boundary node a Ri-1 before the node itself captures the vehicle otherwise it is aware that the vehicle has escaped from Ri-1 and the corresponding updating process fails. To overcome this problem a boundary node of Ri-1 and triggers updating for the reorganization of regions from R1 to Ri-1 . Under the node it happens to be a boundary node of Ri it performs updating for the reorganization of regions from R1 to Ri instead.

C. *Tracking accuracy:*

While moving the vehicle if may run out of the reading range of RFID readers. This inaccuracy is avoid by enlarge the range of the RFID Readers. Also deploys the large number of RFID Readers is more densely.

D. Node join and maintenances:

In HINA if a single node fail can be severed shortly. A local node will check periodically with its neighbors if any node will receive the unavailable information from any of the node it will report to the system administrator .To join the new node or Reader the old node to track the vehicle in very good accuracy.

E. Data replication

HINA actually has the implicit advantage of protecting important tracking data. Recall that tracking data are replicated in the first region. It implies that the system is still able to track the vehicle even when the captured node becomes unavailable. If some vehicles are particularly important and need additional protection of tracking data, we can make relatively large associated with the vehicle. By this means more data can be replicated in the first region organize for the vehicle.

V. DAMAGE PREDICTION SYSTEM: (DPS)

The entire tracking information's are maintained in a data base. This data base collects all the information about the vehicle while crossing the local node by using the RFID readers. The RFID reader will update the information periodically to the data base. The data base expecting the information from the local nodes and the capture node will send the data within seconds. We can track the vehicle from the data base also. Suppose the RFID tag get damaged by the thief or any other reasons. We can't track the vehicle without RFID tag. In this situation the DPS could identify, there are some problems in that vehicle. In this situation HINA helps to track the vehicle by using the last updated information.

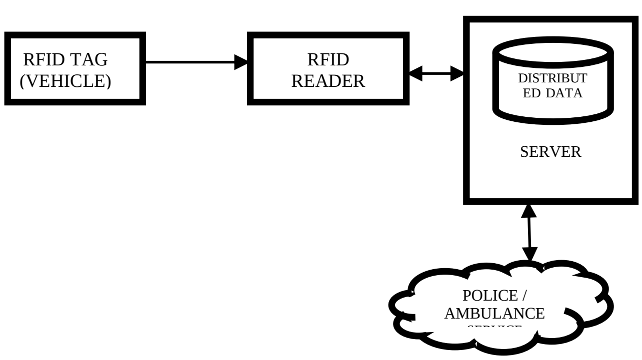

Fig.5. damage prediction system

The damage prediction system will to find the accident spots. The server will collect the information from the reader. The reader gets from the tag. The tag contains the information about the object. When it damage the server will give the information to the national service network (fig.5. illustrated the DPS system).

VI. INTERLINKING WITH OTHER NETWORK:

The server and its data base will link with other network which is related with the vehicle. The other networks may be police control room or Ambulance service. When the data base stop it's updating by the local nodes for the vehicles details. HINA automatically predicts there is some problem in that vehicle. The HINA automatically generates the query and redirected to other networks. Suppose the thief may damage the RFID tag or accident may occur and damage the RFID Tag. The data base will cause the DPS. The DPS will redirect to the related other network. So easily we can capture the vehicle or find the accidental spots automatically.

VII. PROTOTYPE IMPLEMENTATION

To validate the **HINA** design and prove its practical implementation, we have built a prototype system in the campus to track experimental vehicles. This prototype system contains 45 local nodes distributed in our campus. As shown in Fig. 6, local nodes (denoted by red spots) are deployed at crossroads of main roads. Every local node has an IEEE 802.11g wireless network interface connecting the local node to the campus Internet. Furthermore, the overlay network formed by the local nodes is illustrated by the dashed lines in Fig. 9. An overlay connection is established between two nodes if there is a road that immediately connects them.

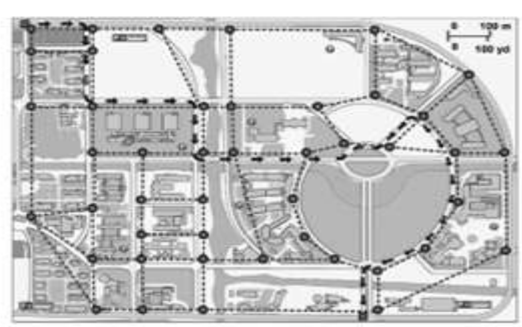

Fig 6. Prototype implementation

In the prototype system, we employ an active RFID system using “Tag Talk First” technology. Fig. 6 shows a typical local node, which is associated with a SP-D300 RFID reader [10] as well as an IEEE 802.11g wireless AP. The inset in Fig. 6 shows an active RFID tag (in highlighted area) attached to a vehicle. The reader’s operating frequency is 2.4 GHz. It connects to the local node via an RS-485 interface and has a data transfer rate of 1 Mbps. The reader has a configurable operation range from 2 to 80m. Each reader can simultaneously detect up to 200 tags in 800 ms. each tag has a unique 64-bit ID. Its battery has a life of 6 to 8 years. Tags send their unique ID signal in random with an average of 300 ms and can be detected at a high speed up to 125 mi/hour.

Besides the RFID system, wireless communication technology is also investigated in our prototype implementation. The **HINA** protocol runs on Red Hat Fedora 5 and uses POSIX.1 socket API to communicate with each other. UDP packets are adopted for location updating and query routing. The size of all packets is 40 bytes, which includes 20 bytes of the IP packet header, 8 bytes of the UDP packet header, and 12 bytes of data.

With this prototype implementation, we conduct a variety of experiments. Since we use the campus Internet as the underlying network, real-time guarantee seems to be nontrivial because the jitter (end-to-end round-trip time) can vary largely. To demonstrate this, we randomly choose two local nodes to measure the round-trip time by ping. It can be seen that the round-trip time increases sharply from 7 p.m. to 11 p.m. Moreover, the peak value can be almost four times larger than that at daytime. Nevertheless, the round-trip time is much more stable during the daytime. Thus, we choose to perform experiments with our prototype system from 10 a.m. to 12 a.m. on 30 June in 2008. We set the real-time constraint to 100 ms and take the maximum transmission delay between two online nodes which is 18.05 ms. Therefore, the resulting r and k are therefore configured to 3 and 1.278, respectively. We let a van carrying an active RFID tag travel at 30 mi/hour along the route as depicted by the dark arrows in Fig. 6. As the van enters an RFID reader’s field and is captured by the reader, the associated node performs location updating accordingly. During the journey which lasts about 4 minutes, we let each node randomly generate a hundred of queries.

Among all the 4.500 queries, the maximum query latency is 90.45 ms, which is strictly shorter than the required real time constraint.

We also notice that the average query latency is about 47.93 ms. the network traffic for location updating among all nodes adds up to 13.2 Kbytes. In contrast, the network traffic for location updating using broadcast on a spanning tree is about 28.8 Kbytes. Since the maximum routing hops of a query is bounded (i.e., 5 hops in this experiment), the network traffic for query routing linearly increases with the number of queries in the system.

The lesson from our prototype implementation is that, with appropriate configuration of the protocol parameters, the query latency can be guaranteed to satisfy the real-time constraint requirement in terms of the number of hops that a message has to traverse. In addition, the overall network traffic overhead, introduced by location updating and query routing, can well accommodate a large number of queries. To further investigate the performance of **HINA** in a large-scale setting, we conduct trace-driven simulations, which are detailed in the following section.

VIII. Performance Evaluations

In this concept was implemented in ns2 [11]. We propose two important metrics to evaluate HINA’s performance.

1. **Maximum query latency:** It is for maximum query response time of a successful one. HINA takes 480ms for a query process.
2. **Network traffic per query:** To answer a query, the system cost should involve two parts of network traffic a) for location updating b) routing query packets.
3. **Effects of protocol parameters:** We can trace 100 vehicles within one hour; we can generate 10^5 queries for different vehicles.

IX. Conclusion

In this paper, we have presented a real time tracking and damage recovering methods of vehicles using HINA protocol. HINA is reducing the congestion and improve the traffic conditions. This article helps to the police for find the vehicle thief. Some of the ongoing researches are there in the RFID security system. This article will helps to the future research on the security using RFID.

REFERENCES

[1] European Commission, The Karen European ITS Framework Architecture, http://www.frame-online.net/, 2004.

[2] The Nat'l ITS Architecture Version 5.1. Dept. Transportation of the US, http://itsarch.iteris.com/itsarch/index.htm, 2005

[3] Ministry of Internal Affairs and Communication National Police Agency, and Ministry of Land, Infrastructure, and Transport of Japan, Vehicle Information and Comm. System, http://www.vics.or.jp/english/index.html, 2006.

[4] Shanghai City Comprehensive Transportation PlanningInst.,http://www.scctpi.gov.cn/chn/chn.asp, 2007

[5] LoJack Corporation, Stolen Vehicle Recovery System, http://www.lojack.com/what/stolen-vehicle-recovery-system.cfm, 2007.

[6] iPico Corporation, Test Report: Single-Lane Vehicle Identification with UHF RFID, http://www.ipico.com/site/iPico_100/pdf/WP_App_HighSpeedVehicleID.pdf, 2007.

[7] "Transportation Recall Enhancement, Accountability, and Documentation (TREAD) Act," The 106th United States Congress, http://www.citizen.org/documents/TREAD%20Ac .pdf, 2000.

[8] Cisco Systems Inc., Cisco Aironet 1240 Series 802.11a/B/G Access Point Data Sheet, guest/products/ps6521/c1650/cdccont_0900aecd8031c844.pdf, 2007.

[9]ShanghaiTelecom, http://www.shanghaitelecom.com.cn/, 2007.

[10] Shanghai Super Electronic Technology, http://www.superrfid.net/english/, 2007.

[11] The Network Simulator, http://www.isi.edu/nsnam/ns/, 2007. ZHU ET AL.: HERO: ONLINE REAL-TIME VEHICLE TRACKING 751 Fig. 19. Maximum query latency versus number of local nodes.Fig. 20.Network traffic per query versus number of nodes.Fig. 21.Network traffic per query versus real-time constraint. Authorized licensed use limited to: Guru AnandanSaminathan. Downloaded on June 19, 2009 at 13:16 from IEEE Xplore. Restrictions apply.

[12]. Hongzi Zhu; Yanmin Zhu and Minglu Li" HERO: Online Real-Time Vehicle Tracking" Parallel and Distributed Systems, IEEE Transactions on Volume: 20, 2009, Page(s): 740 – 752.