An Efficient Crowdsourcing Search Scheme and Aggregation Techniques in Vehicular Ad hoc Networks

Chyi-Ren Dow, Duc-Binh Nguyen, Ming-Fong Tsai, Zi-How Lin, Shiow-Fen Hwang

Department of Information Engineering and Computer Science, Feng Chia University , Taiwan
crdow@fcu.edu.tw, ndbinh.ictu@gmail.com, {tsaimf, m012313, sfhwang}@fcu.edu.tw

Abstract

In recent years, Vehicular Ad-hoc Networks (VANets) have become a popular research field. In addition, conditions happened surrounding our life often have the demand to be supported by other people, and these conditions often occur on the road. This paper proposes an efficient crowdsourcing search scheme and aggregation techniques in the VANets to search objects and collect data. Our proposed mechanism is based on the virtual backbone construction. The regional data exchange can be limited, and the packet transmissions can be reduced. VANets nodes move with high speed and thus may cause dramatic topology changes, and it is not conducive to the frequent swap of information among nodes under limited network bandwidth. We have established coordinator and header mechanisms to manage the information of region nodes, reduce the amount of packet transmission and improve searching efficiency. On the other hand, aggregation and filtering mechanisms are designed and developed for the integrity of the information. Experimental results show that our schemes can effectively provide assistance in terms of search efficiency and satisfactory ratio.

Keywords: VANETs, Crowdsourcing, Aggregation, Search scheme

1 Introduction

Vehicular Ad-hoc Networks are emerging network architecture using wireless network technologies on the intelligent transportation systems (ITS) for mobile information communications. There are many VANets applications, such as cooperative secure services [19], prevention and warning of vehicle collision, video streaming [9, 11], and location-based services (LBS) [13] combined with the nearby area information, etc.

In the past, when people got off a taxi and forgot something in the car, it was not only hard to find the lost property but also wasting time, especially when they did not know the taxi fleet and license plate number. Although taxi fleets can provide services to help people finding their missing property online, it is not immediate when the messages are broadcasted until the driver finds the lost property and returns it. In recent years, with the development of the Internet and the popularity of smart phones, people have begun to help each other through exchanging messages over the network such as looking for an accident escape, animal abusing, bullying and other cases. These Internet users can assist in finding suspects as long as they have enough information.

In fact, the concept of the crowdsourcing already exists in our daily life. Crowdsourcing mechanisms have been widely applied in many fields since created [8, 18] and many applications have been developed with sensors on smart phones and tablets [3, 21]. However, in VANets with crowdsourcing implementation, how to choose the crowd to assist the task and communicate effectively based on the demands, and how to make a proper filtering and screening to ensure the quality and accuracy of the returned data by the crowd, are important issues. In addition, the concept of demand-based distribution and recovery data is similar to the broadcast or multicast schemes which generate a large amount of streaming data. The influence is perhaps not obvious in the wired network architecture with enough bandwidth, but it is a serious problem in VANets environments with limited bandwidth.

Moreover, in order to provide applications with effective crowdsourcing, we need to reduce the time of request spread and data recovery. Both are closely related to the amount of packet transmission. Anycast [15] is a method of information spreading on the network. Its concept involves the case if someone can provide the required services, the person will be accepted after the demand spread. This method not only can reduce the amount of packet transmission, but also can choose appropriate service providers. Thus, anycast is more suitable for applications in wireless network infrastructure with limited bandwidth.

Considering these issues, we propose an efficient crowdsourcing search scheme and aggregation techniques in VANets. This is a distributed searching system based on virtual backbone and geography information in VANets environments. In the proposed scheme, vehicles on the backbone play the role of

*Corresponding Author: Chyi-Ren Dow; E-mail: crdow@fcu.edu.tw
DOI: 10.3966/160792642018031902003

"crowd", and apportion the searching tasks to the crowd through crowdsourcing. The information of vehicles is integrated and exchanged with headers via coordinators in a geographic grid [13]. When a searching task starts, our system will ask the coordinators first, and then the coordinators will forward the searching packet to headers and request vehicles to join the searching task. The number of packets and network load can be reduced through the hierarchical structure. The coordinators and headers will record the information and results from returned packets and the consistency of data. Then we can reduce the searching time by these records in packets. In some situations, our method can also find and track the mobile target, not confined to the target fixed locations.

The remainder of this paper is organized as follows. Section 2 briefly describes the current relevant research and technology. Section 3 discusses in detail about our proposed searching method. Section 4 shows the results of the comparison by computer simulation. Finally, Section 5 draws conclusions and future work.

2 Related Work

In this section, we will introduce some related work. First, the related work to data searching and the data dissemination in VANets are introduced. Second, we present related studies which involve the data aggregation in VANets. Then, the crowdsourcing technologies are described.

2.1 Data Searching in VANets

The topic of data searching has been engaged in network research. Lakas et al. [12] proposed a hybrid cooperation through cooperating vehicles using the store-and-forward technique to share collected information. Some research schemes [7, 10] improved AODV to propose methods to optimize the route of discovery. Lo et al. [17] enabled vehicles to cooperatively aid the source node to discover the location of the destination node without the support of location services. The geographic load balancing routing, namely GLRV is deployed to provide a virtual backbone. GLRV can increase delivery ratio and reduce the transmission latency in hybrid VANets architecture.

2.2 Data Dissemination in VANets

A virtual backbone approach can be used to transmit messages quickly in VANets, e.g., to popularize the multicast routing protocols in VANets. Dow et al. [12] used geo-aware technique to store and share data by P2P in VANets. Choi et al. [5] discussed the stability and efficient broadcast in VANets. Dua et al. [6] proposed a new context-aware congestion resolution protocol, called as-minimum calculated desired time (MCDT) for Intelligent Data Dissemination in VANets to reduce the complexity of transmissions, in which only context-aware data is required for successful transfer of alerts to the passengers. Gnutella [25] proposed a flooding method without centralized mechanism to store resource information. However, this method may cause duplicated resources and redundant transmissions. Yu et al. [22] used the ant colony algorithm to improve the success rate of anycast and shorten the delay.

2.3 Data Aggregation in VANets

How to process the distribution of a vast number of information from different locations is an important issue. Data aggregation technique aims to solve redundant or distributed data problem and improve the communication efficiency. Zhang et al. [24] presented a hierarchical data aggregation scheme which can be used to reduce the transmission of the redundant data resulted from multi-source data collection and multi-path data transmission. Traditional data aggregation methods usually rely on a fixed routing structure to ensure data to be aggregated. However, they cannot be used in highly mobile vehicle environments. The FRIEND system [20] can be used to aggregate traffic data collected by vehicles on roads into a comprehensive, near real-time synopsis of traffic flow conditions. DA2RF [23] is an infrastructure-free data aggregation scheme by restricting forwarders to limit the number of forwarders in VANets. In this way, transmission collisions can be avoided as much as possible.

2.4 Crowdsourcing Technologies

With the increasing popularity of smart phones in recent years, there are many studies using crowdsourcing techniques, including the sensing data collection, transportation issues, data accuracy, and other interesting applications. The CrowdITS system [1] used the smart phones to collect sensing data without additional sensors and communication equipments and make improvements to traffic by collection of traffic information. The CrowdOut system [2] is based on contributions made by mobile users equipped with smart phones. It allows users to report traffic offense in real time and map them on a city plan. There is a systems, named CROSS that supports crowdsourcing disaster surveillance data collection by volunteers exploring threatened areas [4]. On the other hand, Lee et al. [14] searched the event on the road. Liu et al. [16] combined people and mobile sensing devices into a live wireless sensor. It uses this combination to remedy the traditional sensor blind, and also conforms to the concept of crowdsourcing.

3 The Proposed Searching Method

In this section, we introduce our proposed searching mechanisms, including virtual backbone establishment of the tree searching architecture and the design of the crowdsourcing anycast query spreading mechanism. Then, we formulate crowdsourcing data filtering and aggregation mechanisms.

3.1 The Virtual Backbone of the Tree Searching Architecture

Traffic could change very often with complex road structure in urban environments as time goes by. Different distances, directions and speed of the vehicle will cause the network topology changes and affect the stability of the chain. If the city roads are divided by geographic grids in order to establish a backbone tree, we can limit the area of data exchange of information and reduce unnecessary traffic packets. By this way, we can achieve faster and more stable data transmission. Exchanging and maintaining the data table between important nodes in the backbone can simplify data storage and management. The data can be found more quickly, and the searching time can also be reduced with precise management.

When a grid performs the first election or re-election for its grid header, the vehicle stays in each geographic grid with the longest stay period will be elected as a header to manage the information of other vehicles in the grid to reduce the number of switched packets. For example, the bus with the longest stay period will be elected as the grid header to maintain the stability of the network. By doing this way, we can minimize the data hand-off time in a grid. If we cannot obtain the stay duration information of vehicles in the grid, we will select a vehicle which is closest to the center of the grid as the grid header.

Grid headers regularly gather the information of vehicles in the grids and the coordinator within the grid gathers information from adjacent grids in order to accelerate the speed of searching. However, it will generate a lot of queries and return packets when there are too many coordinators. Therefore, the development of an appropriate number of coordinators plays an important role.

As shown in Figure 1, considering the degree of branching of the tree and beginning with the smallest number of grids. If the degree of the header within the grid is greater than 2, it will be elected as the coordinator. Therefore, this step may lead to a coordinator having too many nearby coordinators. For example, coordinators within grids 22 and 54 are superfluous, because there are too many coordinators.

In order to avoid collecting unnecessary information from other paths, a coordinator is created every three hops. However, the coordinator cannot be established if it is located in the leaves of the tree (grids 20 and 58) to avoid excessive control overhead and reduce excess coordinators. Because grids 55 and 58 collect information on the same area, they will result in a large amount of duplicated information and redundant transmission.

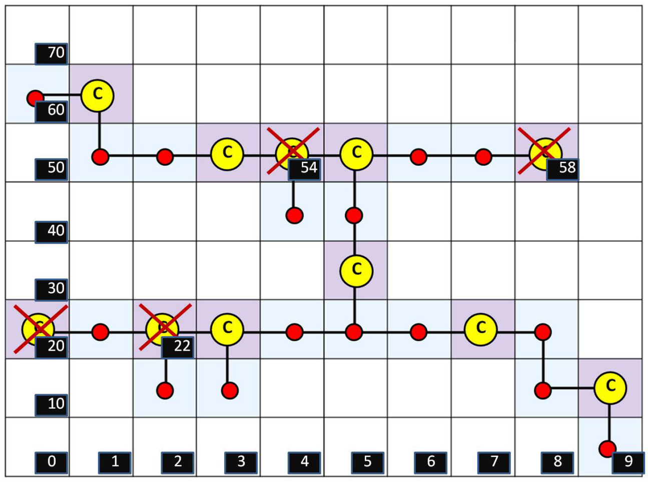

Figure 1. Coordinator establishment

3.2 The Crowdsourcing Anycast Query Spreading Mechanism

The header or the coordinator within each grid maintains a reply table as shown in Figure 2, which contains detailed data fields of each informed object. Vehicles will periodically reply to the information of board objects (such as lost and sensor data, etc.) to headers within grids, and forward the data to coordinators for management. When there is a task (object searching or data collection, etc.), our system will generate query packets as shown in Figure 3 to send to the nearby coordinators. These query packets contain the purpose of tasks (lost, target searching, data collection, etc.), and content (including target characteristics, time, location, etc.). The coordinator will compare the received query packets with the reply table, and then return the result to the source node. If the task needs to cooperate for finding the target, the header within the grid near the target area asks the vehicle within the grid for joining the task after receiving the query packet. When any vehicle responses content, the task starts. When the task is completed, the data will be returned by the header and coordinator. If there is no matched information, the coordinator will pass the query packets to other coordinators through the adjacent headers within the grids. In order to maintain the effectiveness of the searching mechanism, we have developed the threshold of searching time to avoid searching too long or endless searching. We will terminate searching and return results if time is over.

No.	Lost	Sensor Data	Object Searching	$Time_{life}$ (hr)	Describes	Credit Value
1	V			4	Wallet	0
2		V		1	CO_2 Content	0
3			V	5	Dog (White)	0
4		V		3	O_2 Content	0

Figure 2. The Reply Table

Task Type	Describes	Time	Location	$Time_{search}$ (hr)
Lost	Wallet	2014/03/15 AM 09:00	Wenhua Rd., Taichung City	4

Figure 3. The Query Packet

The following assumptions were defined to explain the crowdsourcing search mechanism:

(1) $V=\{v1, v2, \ldots\}$ indicates a set of *Vehicles*.

(2) $G=\{g1, g2, \ldots\}$ indicates a set of Grid Headers, where $G \in V$.

(3) $C=\{c1, c2, \ldots\}$ indicates a set of Coordinators, where $C \in G$.

(4) $R=\{r1, r2, \ldots\}$ indicates a set of Reply table

(5) O={lost, Sensor Data, Object Searching, Timelife, Describes} indicates the properties of board objects

The following algorithm used for crowdsourcing search mechanism, operates as follows:

Crowdsourcing Search Mechanism Algorithm

For each g_j **do**
If (Gotupdate(v_i,o_m)==true) **do**
  UpdateObjects(o_m) to reply table of g_j
  ForwardaUpdate(v_i,o_m) to neighbor c_k
  UpdateObjects(o_m) to reply table of c_k
End For
If (v_i StartQuery(*purpose, content*) ==true) **then**
  While (T_{search} >0) **do**
    ForwardQuery(*purpose, content*) to nearby c_k
    do CompareData(*purpose, content*, R_{Ck})
    If (c_i has *purpose, content*) **then**
      do Response to v_i
    Else if (c_i does not has *purpose, content*) **then**
      BroadcastQuery(*purpose, content*) to neighbor of c_i via adjacent headers
      For each adjacent header g_j **do**
        Broadcast_CooperateQuery(*purpose, content*)
      End For
      For each neighbor c_l of c_k **do**
        CompareData(*purpose, content*, R_{Cl})
      End For
      If (any adjacent header g_j UpdateObjects(o'_m) OR CompareData(*purpose, content*, R_{Cl})== true) **then**
      do Response to v_i
    End if
  End while
End if

3.3 The Data Aggregation and Filtering Mechanism

We have designed the query packet format and reply mechanism to provide the format of returned data. However, getting the number of requirements from the return data is also a problem. Hence we need to formulate the filtering mechanism to deal with these situations.

In addition, the task of data collection needs to be finished by people, the collected data usually needs to be integrated. Each returned data of the same work is combined and based on the tag. In some situations, we require a specific amount of data. All the information will be forwarded to the coordinator after the header within the grid collects complete data. Nodes in our virtual backbone are divided into three levels. Nodes within grids named vehicle are responsible for collecting and providing information. Headers within grids are responsible for collecting information provided by vehicles to the coordinator. As shown in Figure 4, when the source node needs to search target or collect some data (they can be the same or fragments of data for composing complete information). The coordinator will inquire nearby headers to collect information from vehicles. The information collected from vehicles within the grid will be sent to the header, and the header records the number of target objects and sends it to the coordinator. The coordinator will store the information into the reply table after receiving it and combine and send the information back to the source node. The source node selects the desired data from the received data. If the task needs any K (e.g., K=5 in Figure 4) data items, the coordinator will continue to collect data until it meets K and sends them back to the source node.

When the transmitted data has been collected together, it may cause a considerable burden for the network. Thus, we must filter redundant information through hierarchical steps to reduce traffic and leave useful information. We define the details of data fields for the searched target to improve the stringency of data comparison. If the task does not specifically require the same data, the header would not receive the same information after receiving the first information, and the coordinators do the same action, too. We achieve the data filtering through headers and coordinators. If the descriptions of data fields are not detailed enough, the effect of data filtering provided by the headers and coordinators will be limited. It can only let the source node to select accord information.

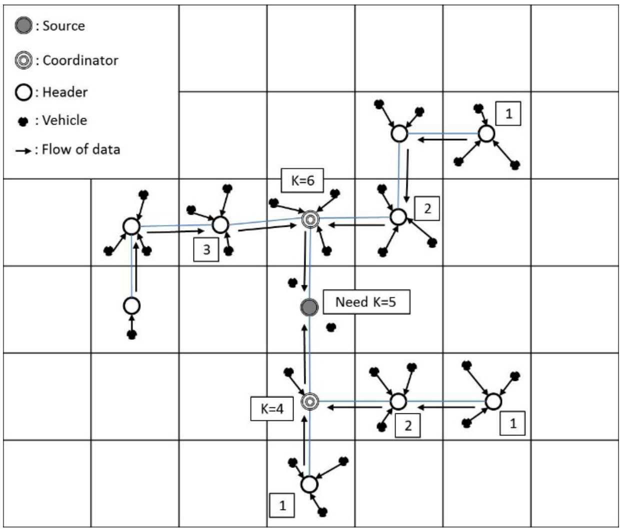

Figure 4. Example of data aggregation

We divide data filtering into three types, as shown in Figure 5(a). For the first type, the searching condition given from the source node to the coordinator is precision and the searching condition given from the coordinator to the header is fuzzy. By this way, we can reduce the difficulty of searching by vehicles and increase the number of data items. The coordinator can filter the data returned by headers before passing to the source node. The second type is shown in Figure 5(b) where conditions given from the source node and the coordinator are both precise. It increases the difficulty of searching by vehicles but reduces the amount of data and reduces the load of networks. The third type shown in Figure 5(c), represents the conditions given from the source node and the coordinators are both fuzzy. The data returned by vehicles will not be filtered by the coordinator, and the source node can select appropriate information.

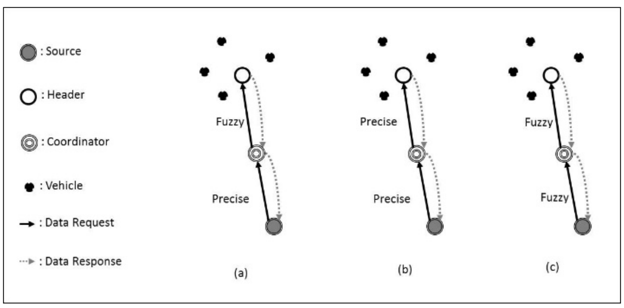

Figure 5. Three types of data filtering

When the new beginning data is received, our mechanisms allow the same information received within the time threshold T for maintaining accuracy and reliability of the information. The number of repetitions will be recorded into the reply table and form a credit value. The higher credit value represents the higher reliability of the data. Each cumulative data will get a survival time T_{life} after time T, and it is not allowed to accept the same data. When the time T_{life} ends, the extra data will be discarded or replaced by new data.

4 Experimental Results

We will show the experimental results in this section. First, the environment of experiment is introduced. Then, the simulation results are presented.

This section aims to evaluate our proposed crowdsourcing search scheme. We use the version 2.35 of Network Simulator 2 (NS-2) as our simulator. The traffic flow is generated by Simulation of Urban Mobility (SUMO) based on the real condition of Taichung City, especially at Feng Chia night market area.

The simulation runs 100 times; each simulation trial takes 200 seconds. The simulation area is 2650m × 2650m. Our method will be compared to other methods, including a traditional method based on AODV and our previous work Geogrid. Assuming the percentage of audience who is willing to help is 80%, the nodes, which possess information and request for assistance, will be chosen randomly.

Table 1. Parameters of simulation environment

Parameters	Value
Map Size	Google Map 2560m*2560m
Mobility Model	Random
Simulation Time	200s
Vehicles Speed	30~60±5Km/h
Number of Vehicles	200~300
Number of Requests	1~10
Mac Protocol	802.11 MAC
Transmission Range	250m

In a comparison with the traditional scheme based on AODV and GeoGrid, our scheme outperforms these schemes, especially for communications with our proposed virtual backbone and geography information hierarchical structure in which each vehicle on the backbone plays a role of crowd to anycast apportion searching tasks. In other hand, our scheme guarantees the integrity of returned search results with proposed aggregation and filtering mechanisms. Our scheme can reduce the amount of packet transmission by reducing the time of request spread and data recovery as well as able to be used in a wireless network environment with limited bandwidth to provide effective assistance in terms of search efficiency and satisfactory ratio.

In the sequence of experiments, our proposed scheme will be compared with the AODV based scheme and GeoGrid. The evaluation based on five performance metrics includes control overhead with the time change, success rate with the distance change, the relations of average end-to-end delay with the number of services, the relations of average end-to-end delay with the number of vehicles, and the relations of average hop count with the number of vehicles.

As shown in Figure 6, we compare the control overhead with the time change. At the beginning, since the purpose of our method is to establish the backbone, the amount of control packets required will be greater than the AODV based scheme and Geogrid. However, once the backbone structure and information forms' contents are completed, the value of control overhead will gradually stabilize.

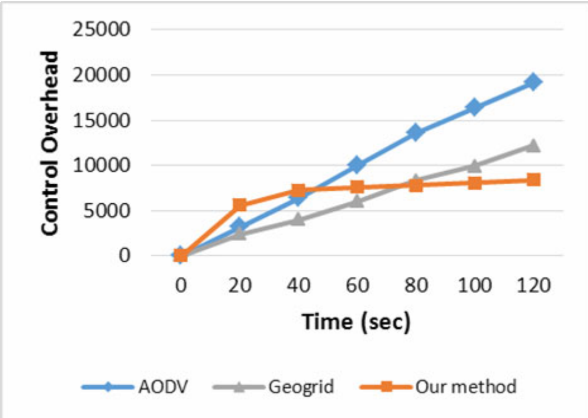

Figure 6. Control overhead vs. time

As shown in Figure 7, the success rate is compared to the change in distance. The range for distance is 500 to 2500. The success rate of the AODV based scheme decreases as the distance increases, because AODV based scheme always starts a new search. Therefore, the success rate will decrease when the distance increases. On the other hand, when Geogrid performs a search, each point is looking for the nearest coordinator for inquiries. Thus, the success rate obtained will display a stable status. In our method, each coordinator works together to maintain and exchange the reply tables. This reduces the search time, which also raises the level of efficiency. In comparison, Geogrid can find the target more effectively.

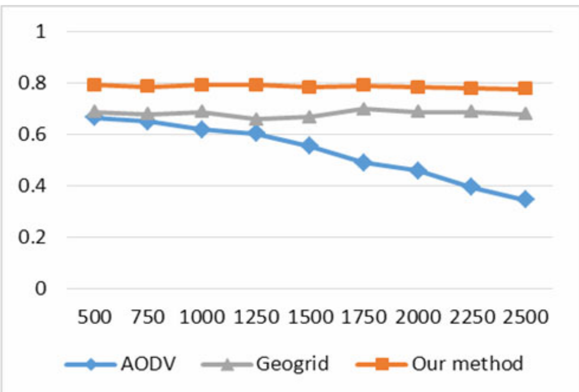

Figure 7. Successful rate vs. distance

Comparisons of average end-to-end delay versus the number of vehicles and services are shown in Figure 8 and Figure 9. The average end-to-end delay will follow the rise with the increase of the number of vehicles because packet collisions increases with the rise of the number of vehicles in Figure 8. Our method has been optimized to minimize hand-off time by transmitting data via the virtual backbone hierarchical through headers and coordinators. By this way, we can maintain the stability of the network and lower down the delay compared to the AODV based scheme and Geogrid.

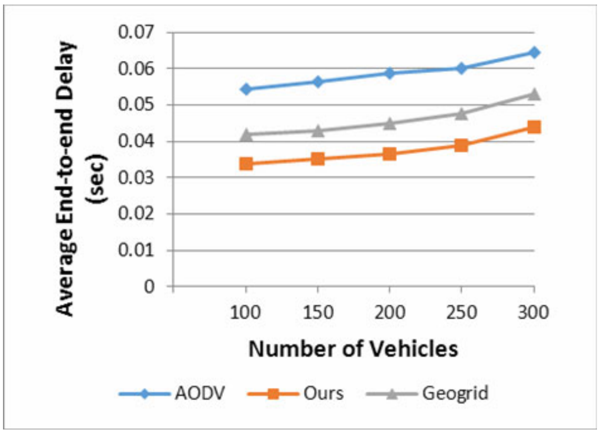

Figure 8. Average end-to-end delay vs. number of vehicles

As shown in Figure 9, we are able to quickly find the information that we need with the increasing of the number of services. In comparison with the AODV based scheme and Geogrid, our method uses the anycast query spreading mechanism to choose an appropriate service to lower down the end-to-end delay.

The value of average hop count would be affected by the number of vehicles. As shown in Figure 10, because the number of vehicles increases from 100 to 300, the average hop count proportionally increase with the number of nodes. Our method uses grids to establish the backbone, and we reduce data transmission via headers and coordinators. Therefore, the average hop count of our scheme is even less than that of the AODV based scheme and Geogrid.

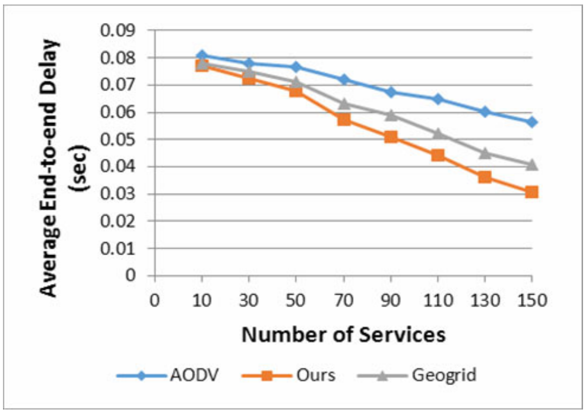

Figure 9. Average end-to-end delay vs. number of services

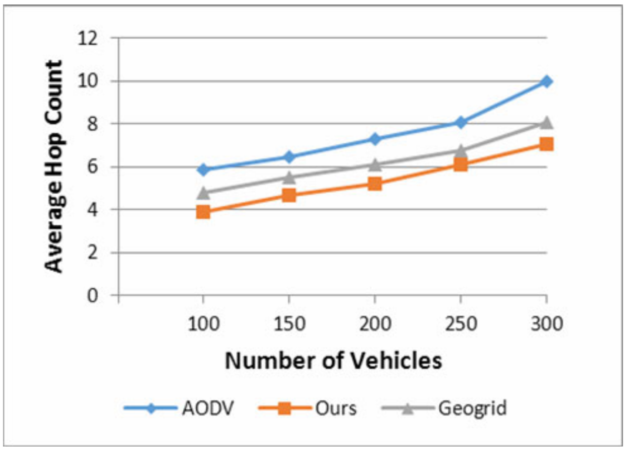

Figure 10. Average hop count vs. number of vehicles

Throughout the evaluation based on five above-mentioned performance metrics, our scheme outperforms the AODV based scheme. Especially, our scheme got better results in comparison with our previous work, Geogrid, which outperforms Gnutella [25] in terms of relations of end-to-end delay with the number of vehicles and relations of average hop count with the number of vehicles.

5 Conclusions

In this paper, we propose an effective crowdsourcing search scheme and aggregation techniques in VANets. It can reduce the number of transmission packets through the establishment of coordinator and header, and provide crowdsourcing search of the query instead of blind searching. The proposed scheme not only improves the results of discovery, but also reduces dissemination time and reply time. The design of the table information can help to exchange all reply discoveries.

We are currently working on the implementation of the lost and found service in Feng Chia night market, which is the largest night market in Taiwan. There are many lost and found demands there. The implementation is about real applications for crowdsourcing search scheme to resolve above-mentioned demands. Furthermore, we will consider more methods for both data aggregation and data filtering in order to achieve more perfect results.

Acknowledgement

The authors would like to thank the Ministry of Science and Technology of the Republic of China for financially supporting this research under Contract MOST104-2221-E-035-020 and MOST104-2221-E-035-021.

References

[1] K. Ali, D. A. Yaseen, A. Ejaz, T. Javed, H. S. Hassanein, CrowdITS: Crowdsourcing in Intelligent Transportation Systems, *Wireless Communications and Networking Conference*, Paris, France, 2012, pp. 3307-3311.

[2] E. Aubry, T. Silverston, A. Lahmadi, O. Festor, CrowdOut: A Mobile Crowdsourcing Service for Road Safety in Digital Cities, *Pervasive Computing and Communications Workshops*, Budapest, Hungary, 2014, pp. 86-91.

[3] N. Cao, S. Brahma, P. K. Varshney, Target Tracking via Crowdsourcing: A Mechanism Design Approach, *IEEE Transactions on Signal Processing*, Vol. 63, No. 6, pp. 1464-1476, March, 2015.

[4] E. T. Chu, Y. L. Chen, J. Y. Lin, J. W. S. Liu, Crowdsourcing Support System for Disaster Surveillance and Response, *Symposium on Wireless Personal Multimedia Communications*, Taipei, Taiwan, 2012, pp. 21-25.

[5] O. Choi, S. Kim, J. Jeong, H. W. Lee, S. Chong, Delay-optimal Data Forwarding in Vehicular Sensor Networks, *IEEE Transactions on Vehicular Technology*, Vol. PP, No. 99, pp. 1-14, September, 2015.

[6] A. Dua, N. Kumar, S. Bawa, J. J. P. C. Rodrigues, An Intelligent Context-aware Congestion Resolution Protocol for Data Dissemination in Vehicular Ad Hoc Networks, *Mobile Networks and Applications*, Vol. 20, No. 2, pp. 181-200, April, 2015.

[7] F. Iqbal, M. I. K. Babar, M. H. Zafar, M. F. Zuhairi, I-AODV: Infrastructure Based Ad Hoc On-demand Distance Vector Routing Protocol for Vehicular Ad Hoc Networks, *IEEE International Conference on Smart Instrumentation, Measurement and Applications*, Kuala Lumpur, Malaysia, 2013, pp. 1-5.

[8] M. Kim, S. O. Park, W. K. Joo, K. Choi, Y. S. Jeong, Y. K. Kim, Tag Based Collaborative Knowledge Management System with Crowdsourcing, *Journal of Internet Technology*, Vol. 14, No. 5, pp. 859-865, September, 2013.

[9] N. Kumar, J. H. Lee, J. J. P. C. Rodrigues, Intelligent Mobile Video Surveillance System as a Bayesian Coalition Game in Vehicular Sensor Networks: Learning Automata Approach, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 16, No. 3, pp. 1148-1161, June, 2015.

[10] N. Kumar, S. Misra, M. S. Obaidat, Collaborative Learning Automata-Based Routing for Rescue Operations in Dense Urban Regions Using Vehicular Sensor Networks, *IEEE Systems Journal*, Vol. 9, No. 3, pp. 1081-1090, June, 2015.

[11] N. Kumar, S. Zeadally, J. J. P. C. Rodrigues, QoS-Aware Hierarchical Web Caching Scheme for Online Video Streaming Applications in Internet-Based Vehicular Ad Hoc Networks, *IEEE Transactions on Industrial Electronics*, Vol. 62, No. 12, pp. 7892-7900, November, 2015.

[12] A. Lakas, M. A. Serhani, M. Boulmalf, A Hybrid Cooperative Service Discovery Scheme for Mobile Services in VANET, *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, Wuhan, China, 2011, pp. 25-31.

[13] C. R. Dow, P. J. Lin, Y. H. Lee, Y. C. Lin, S. F. Hwang, An Efficient Geo-aware Peer-to-Peer *Resource* Discovery and Sharing Scheme in Vehicular Ad-hoc Networks, *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 14, No. 4, pp. 267-278, December, 2013.

[14] K. C. K. Lee, W. C. Lee, B. Zheng, Y. Tian, ROAD: A New Spatial Object Search Framework for Road Networks, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 24, No. 3, pp. 547-560, May, 2012.

[15] P. J. Lin, C. R. Dow, S. C. Chen, C. J. Li, S. F. Hwang, An Efficient Anycast Scheme for Discovering K Services in Mobile Ad-hoc Networks, *5th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, Vancouver, Canada, 2008, pp. 33-37.

[16] J. W. S. Liu, E. T. H. Chu, P. H. Tsai, Fusing Human Sensor and Physical Sensor Data, *IEEE International Conference on Service-Oriented Computing and Applications*, Taipei, Taiwan, 2012, pp. 1-5.

[17] C. C. Lo, J. W. Lee, C. H. Lin, M. F. Horng, Y. H. Kuo, A Cooperative Destination Discovery Scheme to Support Adaptive Routing in VANETs, *IEEE Vehicular Networking Conference*, Jersey, NJ, 2010, pp. 202-208.

[18] Z. Peng, L. Ruoyu, On Operating Mechanism of Crowdsourcing Service Innovation, *8th International Conference on Service Systems and Service Management (ICSSSM)*, Tianjin, China, 2011, pp. 1-8.

[19] C. Sur, Y. Park, K. Sakurai, K. H. Rhee, Providing Secure Location-Aware Services for Cooperative Vehicular Ad Hoc Networks, *Journal of Internet Technology*, Vol. 13, No. 4, pp. 631-643, July, 2012.

[20] S. E. Tawab, S. Olariu, Communication Protocols in FRIEND: A Cyber-Physical System for Traffic Flow Related Information Aggregation and Dissemination, *Pervasive Computing and Communications Workshops*, San Diego, CA, 2013, pp. 447-452.

[21] D. Yang, G. Xue, G. Fang, J. Tang, Incentive Mechanisms for Crowdsensing: Crowdsourcing With Smartphones, *IEEE/ACM Transactions on Networking*, Vol. PP, No. 99, pp. 1-13, May, 2015.

[22] J. Yu, Y. Lin, L. Zhang, X. Zhou, Ant-Based Multi-constrained Anycast Algorithm for Ad Hoc Networks, *International Conference on Communications and Mobile Computing*, Shenzhen, China, 2010, pp. 249-253.

[23] Y. Yuan, J. Luo, W. Yan, T. Zhao, S. Lu, DA2RF: A Data Aggregation Algorithm by Restricting Forwarders for VANETs, *Computing, Networking and Communications*, Honolulu, Hawaii, 2014, pp. 393-397.

[24] L. Zhang, B. Jin, QoS-Oriented Data Dissemination in VANETs, *Parallel and Distributed Processing Symposium Workshops & PhD Forum*, Shanghai, China, 2012, pp. 2581-2521.

[25] Gnutella, *Gnutella Protocol Development*, http://rfc-gnutella. sourceforge. net.src/rfc-0_6-draft.html

Biographies

Chyi-Ren Dow received the Ph.D. degree in computer science from the University of Pittsburgh, PA in 1994. Currently, he is a Professor in the Department of Information Engineering and Computer Science, Feng Chia University, Taiwan. His research interests include mobile computing, ad-hoc wireless networks, telematics, and Internet of vehicles.

Duc-Binh Nguyen received the Master degree in Information Technology from Manuel S. Enverga University Foundation, Philippines in 2010. He is currently pursuing Ph.D. degree with Department of Information Engineering and Computer Science, Feng Chia University, Taiwan. His current research interests include mobile computing, vehicular ad-hoc networks, and Internet of Things.

Ming-Fong Tsai received the Ph.D. degree from the Department of Electrical Engineering, Institute of Computer and Communication Engineering, National Cheng Kung University, Taiwan. He is currently an Assistant Professor with the Department of Information Engineering and Computer Science, Feng Chia University, Taiwan. His current research interests include error-control coding, multimedia communications, and vehicular communications.

Zi-How Lin received the M.S. degree in Information Engineering and Computer Science from Feng Chia University, Taiwan, in 2014. His research interests include mobile computing, vehicle ad-hoc network, crowdsourcing, and data aggregation.

Shiow-Fen Hwang received the Ph.D. degree in Applied Mathematics from National Chiao Tung University, Taiwan in 1991. Currently, she is a Professor in the Department of Information Engineering and Computer Science, Feng Chia University, Taiwan. Her research interests include interconnection networks, mobile computing, and computer algorithms.