Computer Science and Information Systems 12(2):729–742 DOI: 10.2298/CSIS141012023L

VANET Middleware for Service Sharing based on OSGi

Juan Luo, Qiu Pan and Zanyi He

School of Information Science and Engineering, Hunan University
Changsha, Hunan, 410082, China
{juanluo,s1210w133,s1010g127}@hnu.edu.cn

Abstract. VANET is one of the most important parts in smart cities. However, heterogeneous network devices used by different vehicles make data exchange among vehicles become particularly difficult. To solve the problem, a distributed service sharing middleware for VANET called VssOSGi is proposed in the paper. In this middleware, various kinds of functions provided by sensing devices in vehicles are virtually abstracted as services in the network, which decouples applications from devices and provides a unified platform for data exchange among vehicles. As the movements of cars lead to dynamic changes of the network topology, the service discoveries for these vehicles are easily to be invalidated. The paper proposes a distributed service directory selection algorithm SSWA (Stability and Sharing-oriented Weight-based Algorithm) and introduces a service discovery mechanism to make service discoveries for vehicles more effectively and timely. The simulation results show that the proposed middleware could realize widely service discoveries in VANET. Moreover, it requires shorter service response time and less communication traffic with the growing amount of vehicles.

Keywords: VANET, middleware, OSGi, service discovery.

1. Introduction

VANET (Vehicular Ad hoc Network) is a special kind of MANET (Mobile Ad hoc Network), in which vehicles are equipped with wireless communication modules and communicate with each other without help of any infrastructure.

This emerging technology has many potential applications [10], such as vehicle collision alert applications, traffic information services, and other safety or infotainment applications. Among those applications, service sharing among vehicles is an appealing application through which vehicles can share local data (e.g., sensor data and multimedia data) or resources (e.g., storage, access to the Internet) with other vehicles in the network. It is well known that heterogeneous network devices make data exchange among vehicles become particularly difficult. The emergence of middleware enables heterogeneous devices to exchange information through a uniform common interface. Service-Oriented Architecture [3] (SOA) aims to unify common interface, and links the various functional units in a loosely coupled way. Different functional units can manage various mission requirements independently or cooperatively. In a SOA, the services are different functional units in an application system. Service-Oriented Architecture middleware model has attracted a lot of attention due to its low coupling, high scalability and flexibility.

OSGi (Open Service Gateway Initiative), which is an open standard alliance established in 1999, aims to establish open service standards. The service component model in OSGi reflects a good idea of SOA. OSGi, as a basic framework for the introduction of SOA middleware for embedded development, can be a good use of the service prototype development and improvement. It provides services to devices via the network to establish open standards. Meanwhile, it also provides a common software operating platform for a variety of embedded equipments in order to shield the difference between the operating systems and hardware devices. Through Service-Oriented middleware, various kinds of functions provided by sensing devices of vehicles are virtually abstracted as services in networks. Middleware meets the needs of high-level applications by managing services, achieving data exchange and cooperating among vehicles through a common interface. However, most of the existing works have only focused on the problem of service sharing based on middleware in traditional networks, there is no ready-made solution to the problem of service sharing in VANET. To solve this problem, in this paper we propose a distributed service sharing middleware for VANET, which provides a unified platform for data exchange among vehicles. Through it, vehicles can share a variety of services provided by others in the VANET.

2. Related Work

In order to realize service sharing, local resources must be discovered first before being used. We refer to locating distributed resources and services in the network as service discovery. Whether the service directories exist or not, the service discovery architectures can be divided into two categories [16]: directory-based architectures and directory-less architectures. Service directories store the service description for those nodes with shared resources (called service providers). Nodes which request services in the network (called service requesters) send queries to service directories, and then service directories answer them by searching their local service description information.

In the directory-less architectures, the service providers broadcast the service description to other nodes or the service requesters broadcast the requests to other nodes. Nidd *et al.* [11] and Hermann *et al.* [5] periodically broadcast service advertisements to their one-hop neighbors. In [11], service providers not only send local services information to others, but also forward services information provided by others in its cache. However, excessive broadcast redundancy as a result of broadcast storm leads to severe contention at the link layer, packet collisions, inefficient use of bandwidth and processing power. Furthermore, it may disrupt the service due to high contention.

According to the distribution of service directory, the directory-based architecture can be divided into two categories: the centralized directory approaches and the distributed directory approaches. In the centralized directory approaches, one or a few service directory node(s) is(are) selected to store the service descriptions of all the available services in the whole network, which is suitable for both wired networks and wireless local area networks, (e.g., UDDI [1]). However, we cannot find a node to reach all the other nodes in VANET due to the instability of wireless communications. Some nodes may not be able to communicate to the service directory node, and the service discovery will be failure. In the distributed directory approaches [15, 13], the directories are distributed to cope with the availability and scalability problem. In [15], the network topology is divided into several geographical regions, and each region is responsible for storing a group of services. Similarly to the hash table, each key is mapped to a region. Some or all of nodes in the region store the key of the region. In [13], directories are deployed so as to ensure at least one directory is reachable within a fixed number of hops.

Due to the complexity of the underlying implementation of vehicle systems, high coupling, poor reusability and various sensor interfaces, every new vehicle application spends a lot of manpower and time to complete some repetitive tasks, which makes vehicle cost too much. Vehicle middleware shields hardware platforms, operating systems and the diversity of communication protocols for the vehicle applications, eliminates the differences between the various hardware and software. Moreover, Component-based development approach improves the reusability and maintainability of systems and reduces the cost of development for vehicle applications. Service-Oriented VANET middleware [6] is a research hotspot in the field of vehicle middleware research, which greatly reduces the degree of coupling between applications and the underlying platform. It also improves the portability and expansibility of applications, while various kinds of functions provided by sensing devices in vehicles are virtually abstracted as services to be invoked by the upper application. Nieves *et al.* [12] mapped services in the vehicle gateway based on OSGi to UPnP services, in order to share services among different remote devices. Eichhorn *et al.* [2] proposed a Service-Oriented middleware for service discovery and integration of vehicle devices, which makes all the isolated equipment easier to manage.

OSGi framework, defined as the dynamic module system for Java, is a kind of micro kernel middleware platform, which implements an elegant, full and dynamic component model [7]. The application program (bundle) can be dynamically installed, started, terminated, and uninstalled in the form of a plug-in at run time. Each bundle can provide one or more services (exposed interfaces), and other applications are just required to access to services according to the interface description without considering the service providers and the service internal running mechanism. OSGi is developed using Java language, which makes good use of the portability of Java and adapts to different operating systems.

Distributed OSGi has realized interoperability among multiple OSGi frameworks. R-OSGi [14] is a distributed communication component of the OSGi framework. It allows the local OSGi framework to invoke the service from the outer OSGi framework. Wu *et al.* [17] realized service sharing among household appliances by adding service discovery mechanism based on broadcast request to R-OSGi. However, network overhead and time delay of the service discovery based on broadcast request is relatively huge for VANET. DssOSGi [9] utilizes a service discovery mechanism based on centralized directory, and sets up a common service directory on the gateway node in the sensing layer (or on sink node). The embedded sensing devices encapsulate the basic services that they provide in the corresponding bundles and register to the local OSGi framework. Meanwhile, all sensors in the network can get access to services provided by other sensors and realize service binding and information interaction dynamically through a common shared service registry. However, due to the instability of wireless communications, we cannot find a common service directory to reach all the other vehicles in VANET. Some vehicles may not be able to connect to the common service directory, leading to the service discovery failure.

3. The overall architecture of service sharing middleware

The architecture of the distributed service sharing middleware for VANET (VssOSGi) is shown in Fig. 1.

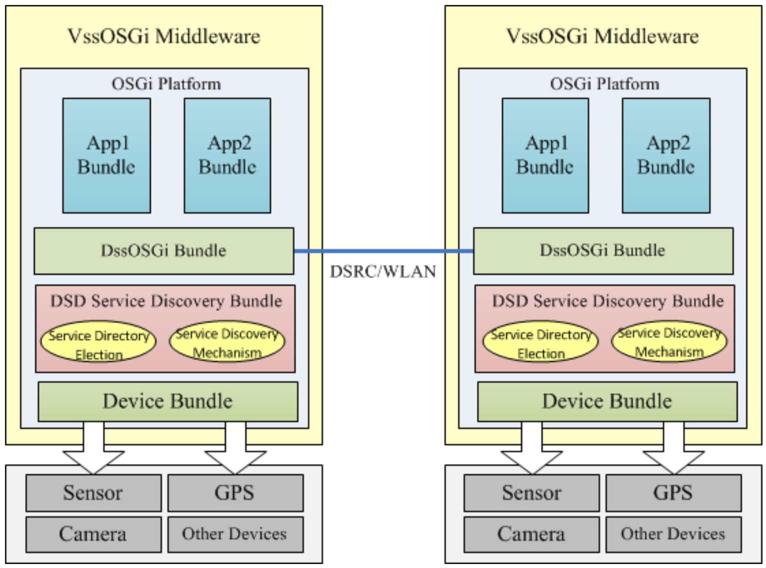

Fig. 1. The overall architecture of VssOSGi.

VssOSGi middleware platform based on OSGi framework extends four layers from bottom to upper, these are Device Bundle, DSD Service Discovery Bundle, DssOSGi Bundle and App Bundle.

The vehicle embedded sensing devices will encapsulate the basic services in the Device Bundle, and middleware platform can directly control the corresponding devices via these bundles. Business logic functions of vehicle applications which are encapsulated in App Bundle, can be published as remote services through DssOSGi and registered to a local service directory to share with other vehicles.

DSD (Distributed Service Directory) Service Discovery Bundle can elect distributed service directory vehicles through Service Directory Election module. Service Discovery Mechanism module is responsible for service discovery among service directory vehicles. Vehicles can register the local device-aware services to a local service directory vehicle through DssOSGi. Vehicle applications can get sharing services registered in the other service directories through DSD Service Discovery Bundle in the local service directory.

Vehicles connect with each other wirelessly through DssOSGi. DssOSGi already has the function of invoking remote services and searching services in the local service directory. According to election results generated from DSD Service Discovery Bundle, DssOSGi decides whether to set itself to act as a local service directory vehicle. According to election results generated from DSD Service Discovery Bundle, DssOSGi decides whether to act as a local service directory vehicle.

4. Service publishing and discovery mechanism

As vehicles are in dynamic environments, the network topology is changed by the movements of vehicles, so the service discovery for these cars is easily to become invalid, which would reduce the efficiency of service discovery. In order to adapt to the dynamic network environment, and achieve a wide range of service discovery and a purpose of reducing communication overhead, DSD Service Discovery Bundle is based on distributed service directory to cope with the dynamic environment of VANET. A service discovery mechanism based on distributed service directory is shown in Fig. 2.

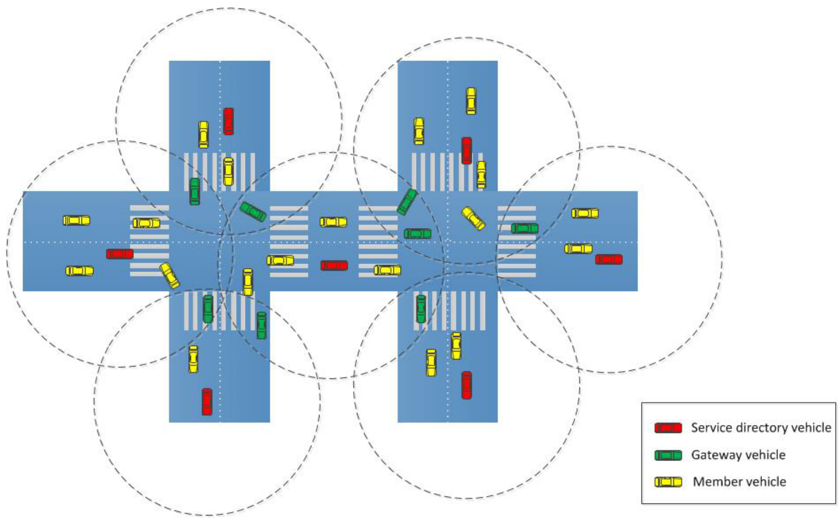

Fig. 2. Service discovery mechanism based on distributed service directory.

Vehicles elect one vehicle as a service directory from their one-hop neighborhoods. The neighborhoods of the service directory will be added to its list of member vehicles. Vehicles which receive messages from two service directories are selected as gateway vehicles. Service directory vehicles are responsible for caching available service index information. Meanwhile, gateway vehicles are responsible for communication between service directory vehicles. Therefore, other neighboring vehicles do not need to cache the service information or flood service request messages to network.

Service publishing and discovery includes three processes: service registration, service request and service response.

First, the service provider P registers the information of service S to the local service directory vehicle D, and service registration is in the form of a lease. P needs to send leases to D periodically, ensuring the service to be updated timely. If P does not continue to send leases, all service information of P will be deleted from the memory of D. Service registration information can not only be obtained through the registration from service providers in one hop, but also be recorded from feedback through searching service among service directories. D saves service information and the information of service provider

P. The storage fields are as follows: $<$*service provider ID, timestamp, valid time, service attributes, local service directory ID*$>$.

Second, we set the request message fields as follows: $<$*service requester ID, timestamp, survival time, service attributes*$>$. If the service requester Q is to request a service, it sends the required service description R to the local service directory D, and D queries its registration information list and matches registration information. If D has stored corresponding unexpired registration information, it replies this registration information to Q. Otherwise, D multicasts R through a multicast tree which takes service directory vehicles as roots and gateway vehicles as leaves of the multicast tree. Multicast message fields are as follows: $<$*source directory ID, timestamp, survival time, multicast hops, service attributes*$>$.

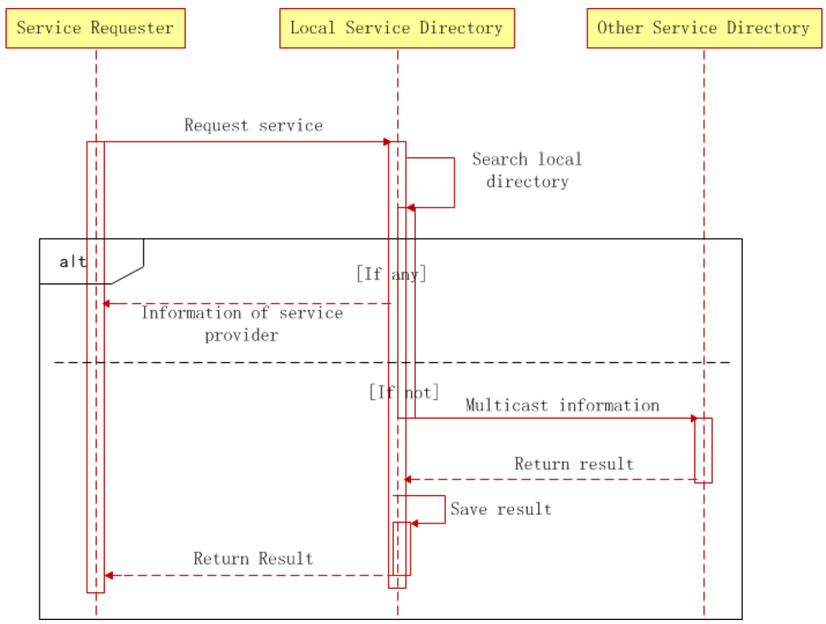

Fig. 3. Process of service discovery.

Finally, the service directory vehicles which receive multicast messages will match R. If achieved, the vehicle which has stored corresponding unexpired registration information replies the matching information to D. D stores this information and replies a message to Q. Otherwise, when a multicast message reaches the end of its lifecycle D would not response any service message. Service directories use tree structure to multicast, which avoids flooding and saves network bandwidth. The working process of service discovery is shown in Fig. 3.

5. The establishment and maintenance of service directory

Previous distributed service directory selection algorithms are proposed for mobile adhoc networks without fully utilizing the advantages and characteristics of VANET. For the moving characteristics of vehicles in VANET, this paper proposes a distributed service directory selection algorithm, named Stability and Sharing-oriented Weight-based Algorithm (SSWA). It takes the vehicle mobility relative to its neighbors and the connectivity with other service directories as important indicators of service directory election. It also uses the tolerance policy based on Monte Carlo to optimize the competition of service directories and improve the stability of the algorithm.

5.1. Calculating the weight

In order to ensure the communication capability of service directory vehicles, the weight takes the difference between the degree d_n (*i.e.* the neighbors' number of vehicle n) and the ideal degree M as a component, described as Eq. (1):

$$D_n = |d_n - M|. \tag{1}$$

In order to ensure the effectiveness of the MAC protocol, the maximum number of members of a service directory vehicle is limited, called the ideal degree M. Since the traffic rules restrict vehicle speed in a certain scope, the vehicles with relatively low mobility are elected to act as services directories, which is beneficial to the stability of the algorithm. Thus, we introduce the relative mobility, defined as the sum of the module of velocity vector difference between the vehicle n and its neighbors:

$$M_n = \sum\nolimits_i |V_i - V_n|, \tag{2}$$

where vehicle i is the neighbor of vehicle n. As gateway vehicles are responsible for communication among service directory vehicles, the more gateway vehicles are the stronger the connectivity is, which can help to improve the ability of service discovery. Thus, the weight takes the difference between the connectivity with other service directories l_n (*i.e.*, the number of the vehicles covered by service directories among neighbors of vehicle n) and the ideal degree M as a component, described as Eq. (3):

$$L_n = |l_n - M|. \tag{3}$$

We assume that the vehicles are equipped with the positioning and navigation system and the position and velocity vector can be obtained directly.

Considering communication coverage, the mobility relative to neighbors and the connectivity with other service directories, the weight of vehicle n is defined as follows:

$$I_n = c_1 \times D_n + c_2 \times c_4 \times M_n + c_3 \times L_n. \tag{4}$$

M_n represents the relative mobility of the vehicle n, $M_n = \sum_i |V_i - V_n|$, and i is the neighbor of vehicle n. D_n represents the communication capability of the vehicle n, $D_n = |d_n - M|$. L_n represents the connectivity with other service directories, $L_n = |l_n - M|$. Vehicle n with smaller weight is more suitable to act as a service directory. c_1, c_2

and c_3 are weighting factors, $c_1 + c_2 + c_3 = 1$, which indicates the relative importance of the various indicators. These weighting factors depend on the state of VANET influenced by vehicle communication radius and vehicle speed. c_4 is a correction factor to adjust magnitude with other indicators .

5.2. The formative phase of service directory vehicles

The election of service directories is a main work in the formative stage of service directory. First, vehicles broadcast their state information (e.g., ID, speed, position) to their neighbors. According to the information obtained from neighbors, each vehicle calculates its weight and informs its neighbors. The election of service directories is divided into several steps as follows:

(1) Find all the neighbors of vehicle n, and calculate d_n.
(2) Calculate degree difference $D_n = |d_n - M|$ of vehicle n. M is the ideal degree.
(3) Calculate the relative mobility of vehicle n with all its neighbors: $M_n = \sum_i |V_i - V_n|$, and vehicle i is its neighbor.
(4) Calculate the difference between the number of the vehicles covered by service directories among neighbors and the ideal degree M: $L_n = |l_n - M|$.
(5) Calculate the weight of vehicle n: $I_n = c_1 \times D_n + c_2 \times c_4 \times M_n + c_3 \times L_n$.
(6) If the weight of vehicle n is smallest among its direct neighbors, n is elected as a service directory vehicle. The neighbors of vehicle n are added to the list of member vehicles of the service directoryand the vehicles which receive information from two or more service directories are selected as gateway vehicles.
(7) Repeat steps (1) to (6) until the election of service directory vehicles is completed. There are three types of vehicles: service directories, the member vehicles of a service directory and the gateway vehicles.

5.3. The maintenance phase of the service directory vehicles

The movement of vehicles leads to communication state change, therefore service directory vehicles are required to be updated timely. Member vehicles monitor the signal strength of their service directories. When the distance between a vehicle and its service directory increases to exceed the effective communication range, the vehicle will join a neighbor service directory vehicle and the service directory updates the list of member vehicles. The election of service directory is triggered if there is a vehicle that cannot be covered by any service directory vehicles.

The competition for a service directory occurs when a vehicle moves into the communication range of a service directory and the weight of the vehicle is less than the service directory. In VANET communication environment, the radio channel is unstable, e.g., multi-channel fading and interference, etc., this might lead to the weight shaking. When the weight difference between two vehicles is not significant, through Monte Carlo optimization, the service directory with a bigger weight will be replaced by a small probability to avoid frequent replacement of service directory caused by the wireless network channel, which enhances the stability of the service directory.

The basic idea of Monte Carlo is to use a certain probability relatively to accept worse states. During the maintenance phase of service directory vehicles, when the weight of a vehicle is less than that of its service directory $(I_n > I_v)$, the service directory will be replaced with the probability of $(1 - p)$. The acceptance probability is defined as $p = e^{(I_v - I_n)}$, and the service directory with a bigger weight continues to be a service directory vehicle with probability p. When the argument x of $y = e^x$ is less than 0, y will increase. The acceptance probability is defined as the exponential function. If the weight difference between n and v is larger, the probability of being replaced increases. Maintenance steps of service directory vehicles are as follows:

(1) Calculate the distance d between vehicle v and its service directory vehicle n. If d is greater than the radius of effective communication, go to (2). If d is smaller than the radius of effective communication and the weight of vehicle v is less than the weight of its service directory , go to (3).
(2) Vehicle n searches a new service directory. If n can find a new one, join it. Otherwise, the election of service directory is triggered. The maintenance phase of service directory vehicles is end.
(3) Calculate the acceptance probability of replacing service directory, and generate a random number r between 0 and 1.
(4) If r is greater than p, vehicle v replaces original service directory n to be a new service directory. Accordingly, n becomes one of its member vehicles.

6. Simulation

Flooding is the easiest solution to share services proactively. Flooding acts as a basic scheme to evaluate performance in much work [8]. So we also choose flooding as basic scheme. The service discovery of R-OSGi (SLP version) is SLP (Service location protocol) [4] which is using flooding mode to disseminate requests. DssOSGi utilizes a service discovery mechanism based on centralized directory, and sets up a common service directory to store available service information in the network. When we publish or get services, we have to communicate with the common service directory. In order to verify the performance of VssOSGi in service discovery and communication overhead, we carry out simulations and compare VssOSGi with R-OSGi(SLP) and DssOSGi using Java in terms of service response time, the number of service request messages, and the success rate of service response.

Table 1. Parameters of experiment

Parameters	value
Vehicle's speed limit(m/s)	5 - 15
Communication range(m)	100
Simulation range(m*m)	600*600
Maximum hops of broadcast packets	4
Interval time of service broadcast(s)	2
Radio propagation type	Wireless Channel

For simplicity, we abstract an urban road model into a combination of intersections and roads. At each intersection a vehicular can run in three different ways: running straight, turning left and turning right, and the probabilities is 0.5, 0.25 and 0.25 respectively. The maximum hops of broadcast packets is 4 and the interval of service broadcast is 2 seconds. The average speed of vehicles is between 5 m/s and 15 m/s, and the effective communication range is 100 m. The ideal degree M is 20. We set 24 regions which generate vehicles randomly in the map, and the number of vehicles we test in the experiment is 24, 48, 72, 96, 120 and 144. Depending on the selected scenario, the weight factor and the correction factors are $c_1 = 0.25$, $c_2 = 0.4$, $c_3 = 0.35$, $c_4 = 0.1$. The experimental parameters are set as shown in Table 1.

An instance of service sharing on VssOSGi middleware platform is shown in Fig. 4. We can describe the process of service sharing as follows. First, vehicle 9 requests a service S, and then it sends the request to the local service directory vehicle 1. Vehicle 1 queries its local memory to determine whether S exists or not. After finding S is not in its local memory, vehicle 1 forwards the service request message to other service directory vehicles 2, 3, 4, 5, 6, 7, 8 through green gateway vehicles. Since service directory vehicle 4 has saved the service information S that vehicle 10 registered. Therefore, vehicle 4 finds matching service information and sends this information back to vehicle 1. Vehicle 1 saves this information and sends it to vehicle 9. Finally, Vehicle 9 binds and invokes the service from vehicle 10. The process of service sharing is completed.

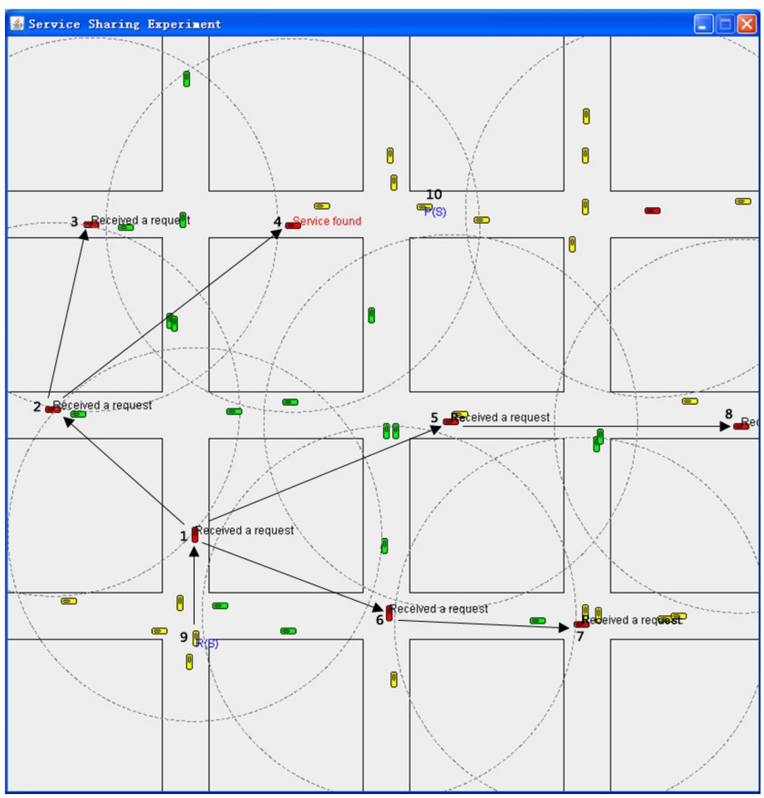

Fig. 4. Instance of service sharing.

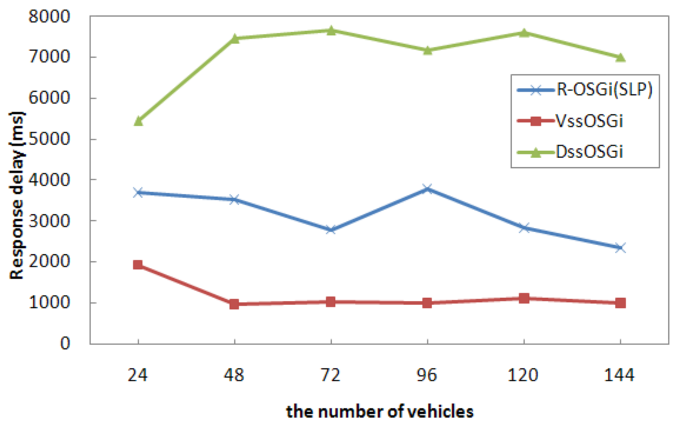

Fig. 5. Service response time in different number of vehicles.

Fig. 5 shows the service response time of the three kinds of service discoveries when the number of vehicles varies. The abscissa is the number of vehicles, and the vertical axis is the service response delay. When the number of vehicles is smaller, VssOSGi has lower network connectivity. The lack of gateway vehicles between service directories leads communication become hardly between service directories, and the service response time is higher than that of service discovery based on flooding. When the number of vehicles is becoming larger, service directory vehicles can override the whole map and share service information with each other through gateway vehicles, thus service response time decreases. Due to the limitation of broadcast performance, the service response delay of R-OSGi(SLP) is always long and flooding may lead to broadcast storm with the growing number of vehicles. The service response delay of DssOSGi is longest because it has only one common service directory which cannot cover all vehicles.

Fig. 6 shows how the number of service request messages changes with the number of vehicles increases. R-OSGi(SLP) broadcasts service requests in its surrounding vehicles periodically. So the number of service requests is increasing dramatically with the growing number of vehicles, which may lead to broadcast storm. VssOSGi only forwards messages between the service directories through gateway vehicles. The number of service request messages keeps low with the growing number of vehicles, thereby greatly reducing the service discovery traffic. DssOSGi only sends messages to the common service directory, therefore, the number of service request messages is least. However, some vehicles may not be able to connect to the common service directory, leading to the service request failure.

The success rate of service response in different number of vehicles is show in Fig. 7. When the number of vehicles is 24, VssOSGi has lower network connectivity. The lack of gateway vehicles between service directories leads to communicating hardly between service directories. So the ability of service discovery is weaker than the number of vehicles is more. And the success rate of service response is less than 90%. However, as shown in Fig. 7, when the number of vehicles is more than 48, the service directory vehicles are enough to override the whole map and share service information with each other through

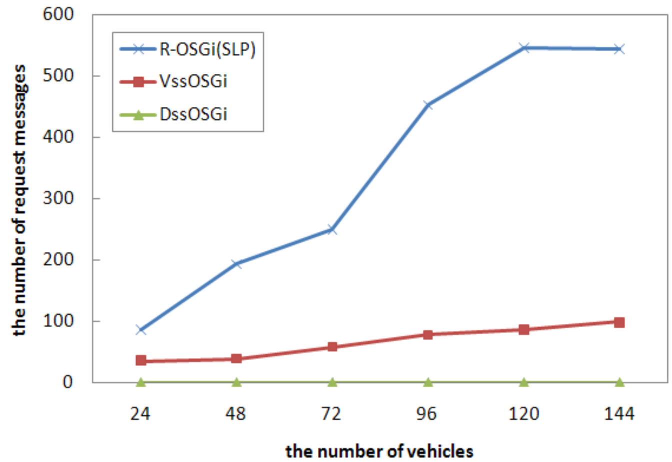

Fig. 6. The number of service request messages in different number of vehicles.

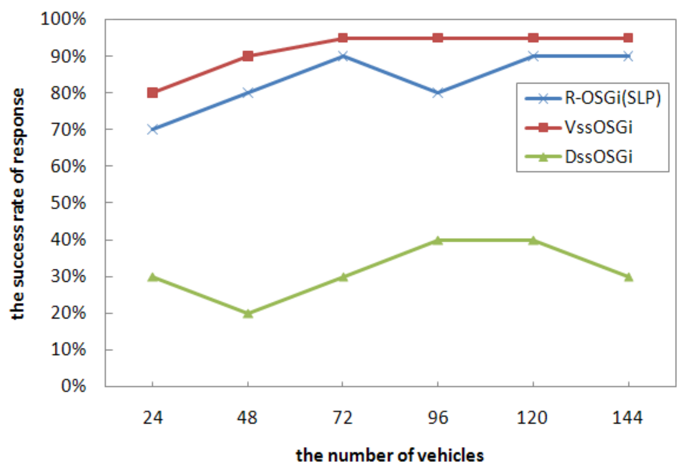

Fig. 7. The success rate of service response in different number of vehicles.

gateway vehicles. Therefore the success rate of service response is higher than 90%. Due to the limitation of broadcast performance, the performance of R-OSGi(SLP) is worse when the number of vehicles is 24 and the success rate of service response also achieves 90% when the number of vehicles is larger. However, DssOSGi has only one common service directory which cannot cover all vehicles, thus the success rate of service response is worst.

The above simulation results show that the proposed middleware framework can discover services well in VANET, and perform high success rate of service response. Furthermore, with the increasing number of vehicles, it can not only reduce communication cost, but also maintain a lower service response time.

7. Conclusion

VANET, as a special mobile ad hoc network, has characteristics of self-organization, no-center and dynamic. In the proposed middleware VssOSGi based on SOA, various kinds of functions provided by sensing devices in vehicles are virtually abstracted as services in VANETs. Through it, vehicles can share a variety of services provided by others in VANETs. This distributed service sharing mechanism overcomes flaws of DssOSGi which cannot discover services in VANETs.

Our future work will focus on studying the middleware platform for scheduling service requests in the large scale service request environment, and designing an appropriate service scheduling algorithm to meet the different requirements of real-time services.

Acknowledgments. This work was supported by National Key Technology R&D Program (2012-BAD35B06); Program for New Century Excellent Talents in University (NCET-12-0164); National Natural Science Foundation of China (61370094); Natural Science Foundation of Hunan (13JJ1014). Grateful thanks are due to the participants of the research for their invaluable help in this study.

References

1. Coalition, U.: The uddi technical white paper (2000)
2. Eichhorn, M., Pfannenstein, M., Muhra, D., Steinbach, E.: A soa-based middleware concept for in-vehicle service discovery and device integration. In: Intelligent Vehicles Symposium (IV), 2010 IEEE. pp. 663–669. IEEE (2010)
3. Guinard, D., Trifa, V., Karnouskos, S., Spiess, P., Savio, D.: Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services. Services Computing, IEEE Transactions on 3(3), 223–235 (2010)
4. Guttman, E., Veizades, J.: Service location protocol, version 2 (1999)
5. Hermann, R., Husemann, D., Moser, M., Nidd, M., Rohner, C., Schade, A.: Deapspace–transient ad hoc networking of pervasive devices. Computer Networks 35(4), 411–428 (2001)
6. Hirschheim, R., Welke, R., Schwarz, A.: Service-oriented architecture: Myths, realities, and a maturity model. MIS Quarterly Executive 9(1), 37–48 (2010)
7. Lin, H., Zeng, X.: Principle of osgi and best practice. Electronic Industry Press (2009)
8. Liu, J., Li, B., Zhang, Q., Zhu, W.: Service locating for large-scale mobile ad-hoc network. International Journal of Wireless Information Networks 10(1), 33–40 (2003)
9. Luo, J., Wu, F., Zhou, F., Li, R.: Researches on wireless embedded middleware for service sharing. In: Advances in Wireless Sensor Networks, pp. 212–220. Springer (2014)
10. Maxemchuk, N.F., Tientrakool, P., Willke, T.L.: The role of communications in cyber-physical vehicle applications. Automotive Informatics and Communicative Systems: Principles in Vehicular Networks and Data Exchange p. 139 (2009)
11. Nidd, M.: Service discovery in deapspace. Personal Communications, IEEE 8(4), 39–45 (2001)
12. Nieves, A., Gómez-Escalonilla, J.S., Madrid, N.M., Seepold, R.: Upnp into a car-gateway middleware with osgi: Interoperability and security. In: Intelligent solutions in Embedded Systems, 2009 Seventh Workshop on. pp. 119–126. IEEE (2009)
13. Raychoudhury, V., Cao, J., Wu, W., Lai, Y., Chen, C., Ma, J.: K-directory community: reliable service discovery in manet. Pervasive and mobile computing 7(1), 140–158 (2011)
14. Rellermeyer, J.S., Alonso, G., Roscoe, T.: Building, deploying, and monitoring distributed applications with eclipse and r-osgi. In: Proceedings of the 2007 OOPSLA workshop on eclipse technology eXchange. pp. 50–54. ACM (2007)

15. Seada, K., Helmy, A.: Rendezvous regions: A scalable architecture for service location and data-centric storage in large-scale wireless networks. In: Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International. p. 218. IEEE (2004)
16. Ververidis, C.N., Polyzos, G.C.: Service discovery for mobile ad hoc networks: a survey of issues and techniques. Communications Surveys & Tutorials, IEEE 10(3), 30–45 (2008)
17. Wu, J., Huang, L., Wang, D., Shen, F.: R-osgi-based architecture of distributed smart home system. Consumer Electronics, IEEE Transactions on 54(3), 1166–1172 (2008)

Juan Luo received the Bachelor's degree in electronic engineering from the National University of Defense Technology, Changsha, China, in 1997, and the Master's and Ph.D. degrees in communication and information systems from Wuhan University, Wuhan, China, in 2000 and 2005, respectively. Currently, she is a Professor and Doctoral Supervisor with the College of Computer Science and Electronic Engineering, Hunan University, Changsha. She is also the Director with the Department of Network and Information Security, Hunan University. From 2008 to 2009, she was a Visiting Scholar with the University of California at Irvine, Irvine, CA, USA. She has authored/coauthored more than 40 papers. Her research interests include wireless networks, cloud computing, and wireless sensor networks.

Qiu Pan received the bachelors degree in communication engineering in 2012, from Guangxi University, Nanning. He is currently a postgraduate, College of Information Science and Engineering, Hunan University, Hunan, China. His research is focused on middleware and vehicular ad hoc network.

Zanyi He received the bachelors degree in communication engineering in 2010, from Hunan University, Hunan. She is currently a postgraduate, College of Information Science and Engineering, Hunan University, Hunan, China. Her research is focused on wireless sensor network.

Received: October 12, 2014; Accepted: April 7, 2015.