International Journal of Innovative
Computing, Information and Control
Volume 8, Number 8, August 2012

ICIC International ©2012 ISSN 1349-4198
pp. 5573–5590

SERVICE DISCOVERY PROTOCOL FOR NETWORK MOBILITY ENVIRONMENT

Min-Xiou Chen[1], Fu-Hsing Sung[1] and Bing-Yang Lin[2]

[1]Department of Computer Science and Information Engineering
National Dong Hwa University
No. 1, Sec. 2, Da Hsueh Rd., Shoufeng, Hualien 97401, Taiwan
{mxchen; m9821008}@mail.ndhu.edu.tw

[2]Department of Computer Science and Information Engineering
Chung Hua University
No. 707, Sec. 2, WuFu Rd., Hsinchu 30012, Taiwan
m09602008@chu.edu.tw

Received May 2011; revised September 2011

Abstract. *Discovery and management of desired network services is a great challenge in vehicular networks. In this paper, we propose two service discovery architectures based on Service Location Protocol (SLP) for heterogeneous vehicle networks in a Network Mobility (NEMO) environment. We introduce a substitute query technique into SLP to discover the service information in either vehicle directory agents or roadside directory agents, to improve the server hit ratio. A cache mechanism is then introduced to reduce the message overhead of the substitute query. Numerical results indicate that the proposed architecture can provide a high average data hit ratio and a very low message overhead.*

Keywords: Service discovery, Service location protocol, Network mobility, Vehicle network

1. **Introduction.** In recent years, with the rapid expansion of communication technology, Vehicular Networks have received increasing attention from the research community. The improvement in wireless communication and embedded technologies has greatly enhanced the performance of electronic devices, such as smart phones, computers, sensors, and hands-free devices. Integration of this technology into various sub-systems of future vehicles will enhance performance and safety. By integrating a wireless communication interface into vehicle sub-systems, future cars could communicate with other vehicles and with a fixed network; numerous Internet protocols, applications or multimedia services could be provided by vehicular networks.

A vehicular network [1,2], as shown in Figure 1, is composed of two types of components: Roadside Gateways (RGs) and Road Vehicle Nodes (VNs). RGs are connected to routers which in turn are connected to the infrastructure network. RGs also provide a wireless interface so that VNs can access the infrastructure network. VNs are the cars circulating along the roads that are equipped with some sub-systems, such as vehicle navigation system or position system, and a wireless interface. VNs can use the wireless interface to communicate with each other via an ad hoc network or via RGs. Several suggestions have been proposed to solve the routing problem in vehicular networks [3-5].

VNs provide vehicles with wireless access to the network. Thus, users traveling in vehicles equipped with VNs can access network services provided by these sub-systems or the infrastructure network through the in-vehicle wireless access network. These network services can be grouped into two types: security services and convenience services. The security services enhance vehicle safety on roads, and the convenience services provide

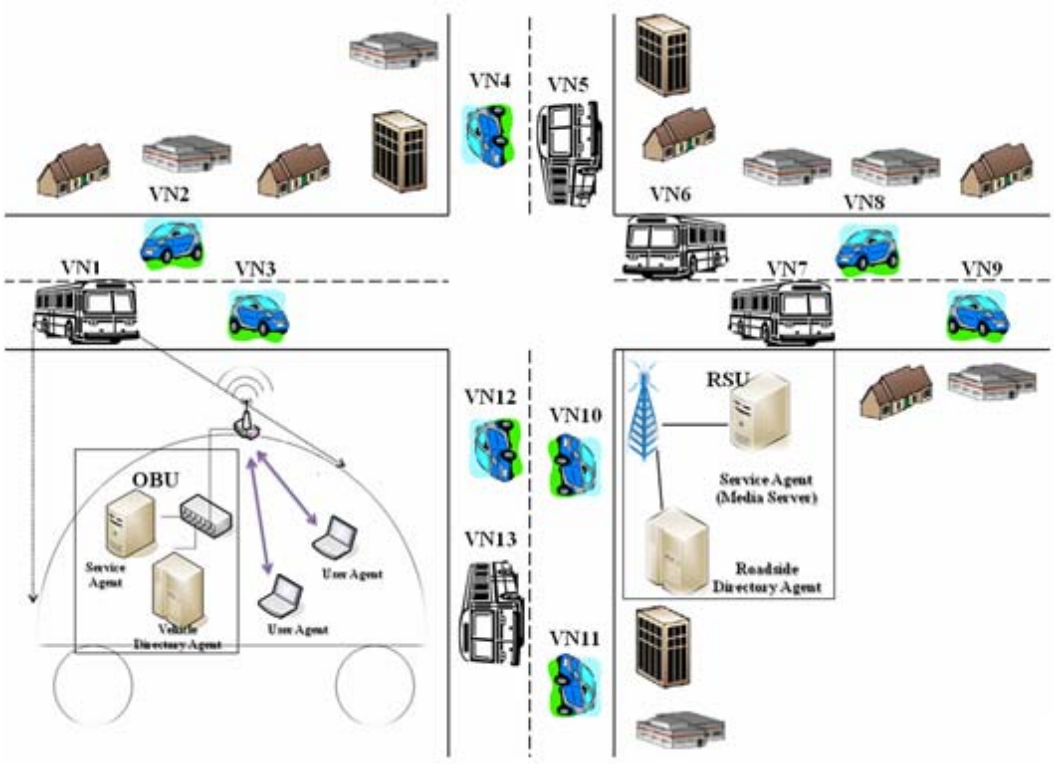

FIGURE 1. Vehicular network

services for passengers in the vehicles. However, how to discover and manage the desired network services becomes a great challenge in such vehicle networks.

Service discovery protocols allow devices to advertise service, detect other devices, and find specific services on the network [6-10]. There have been many research efforts directed at the service discovery problem. Some of the proposed service discovery protocols are designed on the network layer, and others are designed on the application layer. Moreover, according to the proposed architecture, these research results can be divided into a centralized directory-based service discovery architecture, a distributed directory-based service discovery architecture, and directory-less service discovery architecture.

It is very important to design a service discovery mechanism with low network overhead, fast service response time, high service hit ratio, and good network scalability. A directory-based service discovery protocol can provide lower network overhead and faster service response time. Service Location Protocol (SLP) [11,12] is a directory-based service discovery protocol that allows computers and other devices to find services in a local area network without prior configuration. Using SLP, the devices announce services or discover services with faster service response time and high network scalability on an unmanaged network.

However, with SLP the users can only discover the desired service if it exists in a local discovery agent that maintains limited service information, so the service hit ratio may be very poor. To the best of our knowledge, most current service discovery protocols are designed for fixed networks or for Mobile Ad-Hoc Network (MANET)/ Vehicular Ad-Hoc Network (VANET) and are not designed for the network mobility (NEMO) environment. As shown in Figure 1, in the VN, an OBU (Onboard Unit) can contain the function of a directory server, and the service provided by OBU can be registered by the directory server. Let us consider a scenario in which a user sits on a bus, which is equipped with an OBU. The user has a smart phone or a laptop computer attached to an in-vehicle

wireless access network. When the user sits on the bus, he can discover the service from the directory server.

Suppose that a roadside directory server is connected via an RG, the user sitting on a bus may not know the roadside directory server, and cannot discover the server from the roadside directory server. On the other hand, suppose the users sitting on a bus know the roadside directory server and can discover the server from the roadside directory server. When dozens of users sit on a bus or more than one hundred users sit on a train, each trying to discover the server from the roadside directory server, hundreds of service discovery messages will be sent to the roadside directory server, and the roadside directory server may be overwhelmed by these service discovery messages.

In this paper, we propose a Substitute Request Message and a cache mechanism in the SLP directory agent in order to provide a better service discovery mechanism in vehicular networks and in the network mobility environment. The SLP directory agent can use the Substitute Request Message to discover services from another SLP directory agent (such as the roadside directory server in Figure 1) and cache these search results in its database. Then, individual users sitting on the bus do not need to send hundreds of service discovery messages to the roadside directory server to discover the service, and only need to send the discover messages to the directory server equipped OBU to get the service registered in the roadside directory server (improving the service hit ratio). Thus, with the Substitute Request Message and cache mechanism, the service hit ratio and message overhead of SLP can be improved.

The remainder of this paper is organized as follows. The related literature, standards and protocols are discussed in Section 2. The proposed architecture, Substitute Request Message, and cache mechanism for the SLP directory agent are proposed in Section 3. Section 4 presents a performance analysis of the proposed architecture. We provide an example to show the utility of our architecture in Section 5. In Section 6, a performance evaluation for the proposed system is provided. Our conclusions and plans for future work are given in Section 7.

2. **Related Work.** Previous research results can be divided into a centralized directory-based service discovery architecture, distributed directory-based service discovery architecture, and directory-less service discovery architecture.

In the centralized directory-based service discovery architecture, the service discovery process works as a server to store service information and to respond to a service discovery. The server may be called the directory agent, lookup service, or some other names in different protocols. Java Intelligent Network Interface (JINI) [13,14] is a protocol based on JAVA. In JINI, the central directory is called Lookup Service. The Lookup Service allows a device to register its service when connecting to the network for first time. The client gets specific service information by connecting to the Lookup Service and then querying it. Compared with a directory-less architecture, a Lookup Service can reduce unnecessary message exchanges between the service provider and the service requester because the directory can aggregate service information on the network and prevent nodes from searching blindly. Nodes can get abundant service information by making just one query process to the directory. The drawback of the centralized architecture is that the directory may have a high load if the network becomes large or the service discovery gets more frequent. And the centralized architecture will face high risk if the single directory has a problem.

The second strategy is the directory-less architecture. In this strategy, the service discovery protocol has no central node for the service discovery process. Universal Plug and Play (UPnP) [15] is a directory-less service discovery developed by Microsoft. It

constructs a p2p network and uses the Simple Service Discovery Protocol (SSDP) as the service discovery of UPnP. There are three components in the network: device, service, and control points. The control point can discover and control the device in the UPnP network.

The authors in [16] proposed using the Simple Service Discovery Protocol (SSDP) to provide service discovery. In the SSDP, the device advertises its service by multicast when it joins the network, and then the control point can receive the service information. DEAPspace proposed in [17] uses proactive single-hop broadcasts to recognize all devices. It has high overhead due to the necessity of placing beacons at regular intervals and thus is not scalable. Konark's proposal in [18,19] uses multicasts for service advertisement and service discovery. Konark supports the push and pull methods, and all the devices have cache to maintain service information.

In [20], the authors proposed a distributed service discovery, Group Service Discovery (GSD). GSD classifies services to the different groups according to service type. It uses group information to reduce service message volume. Each node advertises its identification and group information that were provided to it by the corresponding service provider. Other nodes wanting to attach the specific service do so by querying the nodes that contain group information about the service. In order to reduce message overhead under the loop situation, the Candidate Node Pruning Enhanced Group-based Service Discovery Protocol (CNGPSDP) proposed in [21] has two schemes, candidate node pruning and broadcast simulated unicast, to enhance GSD protocol. In a directory-less architecture, these approaches usually use multicast or broadcast to perform service discovery. This may cause a large overhead between service provider and service requester especially when the network becomes large.

The other strategy is distributed directory-based architecture. More than one node works as a directory in the network and the nodes cooperate with each other. Chord proposed in [22] that the directory be distributed among many nodes by constructing a ring network, but this may cause high maintenance overhead between nodes if node mobility increases. In [23], the authors proposed a distributed directory service discovery and showed that the directories can afford the load of service discovery. The service provider registers a service to the nearest directory. Then the directory will register the service to all the directories on the network. The relevant research is summarized in Table 1.

TABLE 1. Summary of various discovery strategies

Strategy	Architecture	Advantage	Disadvantage
Directory	Centralized	Easy to manage	Operation involves high risk
	Distributed	Adjusts dynamically to different situations	High management overhead
Directory-less		No management overhead	Broadcast storm problem, non-scalable

An extension of SLP for peering using a Directory Agent (DA) has been proposed in the Mesh-enhanced Service Location Protocol (mSLP) [24]. In the proposed architecture, each DA should establish a peer connection with some DAs with shared scope, and periodical exchanges of the DA Advertisement (DAAdvert) message maintain the peer relationship. Peer DAs also exchange new service registrations in shared scopes via message forwarding. Thus, this protocol improves the consistency among peer DAs by automatically sharing registrations, and provides registration information recovery when the DA is rebooted.

However, the NEMO environment was not considered in the proposed architecture. The vehicle DAs must send many messages to establish and maintain the connections, and exchange each registration message. A heavy maintenance overhead is the major drawback when mSLP is implemented in the NEMO environment. Moreover, a DA cannot share registrations with another DA in a different scope.

Bonjour, also known as zero-configuration networking, enables automatic discovery of computers, devices, and services on IP networks, and was proposed based on the DNS-Based service discovery architecture [25]. The proposed protocol allows clients to discover a list of desired services using standard DNS queries. The mechanism of Multicast DNS [26] should be used to provide service discovery for mobility environments. However, the limitation of this protocol is that it is intended for small routerless networks. Thus, the proposed solution cannot properly support service discovery for NEMO.

3. System Architecture.

3.1. **Service location protocols.** The Service Location Protocol (SLP) supports the user in finding the existence, location, and configuration of available network services. SLP provides service registration, service announcement, and service discovery on an unmanaged network. The Uniform Resources Locator (URL) is used to locate a service. To efficiently manage services, three network components are proposed in the SLP architecture. They are Service Agents (SAs), Directory Agents (DAs), and User Agents (UAs).

The SLP UA is a software entity attached to a specific device, and can invoke a service discovery for desired service. The users use their SLP UA to find the available services using only a description of the desired service. Their SLP UAs send the request with the description of the desired service to the SLP SA or the SLP DA, and the SLP SA or the SLP DA returns the response with the URL for the desired service. After receiving the response message, the SLP UA will show the service information and the user can establish a connection with a service provider using the URL.

The SLP SA is a device that provides one or more services. The SLP SA can actively register services to the SLP DA or return the service information to the SLP UA when the desired service matches. The SLP DA is a software entity that works as a directory server to manage the services registered from the SLP SAs. The SLP DA also returns the service information when the SLP UA sends its discovery request.

SLP has the ability to manage networks on various scales, from small unmanaged networks to large enterprise networks. In a large network, the network devices and services can be grouped into several scopes and each device or service would be in one or more scope. According to the definition of SLP, the scope is a set of network devices or services that can be described as a simple string, and is denoted as the domain name in a large network. In a vehicular network, a scope can be a vehicle, a line of vehicles, or a small area of a city.

SLP provides two service discovery processes. In the first process, an SLP UA sends a service request packet to some SLP SAs by multicasting or broadcasting. The SAs that have the desired service will send back a service reply packet to the SLP UA. This kind of approach is suitable for small, unmanaged networks. In the other process, an SA first issues a registered message containing all of the services that they provide to the SLP DAs of its scope, and the SLP DA sends backs an ACK packet when the register process is successful. Then, an SLP UA can send a unicast service request packet to the SLP DA of its scope, and the DA sends back a reply packet with the service information of the desired service. This approach is suitable for larger networks.

3.2. **Basic system overview.** The system overview is shown in Figure 1. Some Roadside Units (RSUs) are deployed along the side of the road, and each RSU includes an RG, and a Roadside DA. RGs are connected with the infrastructure network and have wireless interfaces that allow VNs to access the infrastructure network through RGs. The RSU also can provide service and work as an SLP SA. In a VN, an OBU (Onboard Unit) is equipped for controlling the actions of the embedded sub-systems and network devices. The OBU can work as both a Vehicle DA and an SLP SA. The OBU also provides wireless interfaces that allow users to access the services provided by the OBU and the infrastructure network through these wireless interfaces.

Let us consider a scenario in which a user sits on a bus, which is equipped with an OBU. The user has a smart phone or a laptop computer attached to an in-vehicle wireless access network. The user could use the SLP UA to discover the services provided by the OBU. The SLP UA sends a discovery message to the Vehicle DA, and the Vehicle DA returns the response with the URL for the desired service to SLP UA. However, the SLP UA may not find the desired service from the Vehicle DA. According to the definition of SLP, because the SLP UA does not directly connect with the RG, the SLP UA cannot send a discovery message to the Roadside DA. Thus, the SLP UA cannot find any network service from the Roadside DA.

In the paper, we introduce a Substitute Request Message and cache policy into the SLP. Thus, the user can send a Substitute Request Message to a Vehicle DA to discover services from another Vehicle DA or a Roadside DA, and the Vehicle DA will cache the search results and provide these results for the next search request. The question arises, how can the SLP UA or the Vehicle DA know the address of the Roadside DA? Let us consider a scenario in which a vehicle with a Vehicle DA enters the area and connects with the RSU. The Vehicle DA can send a multicast server request to discover the address of the Roadside DA. The Roadside DA will then return the response message to the Vehicle DA, and the Vehicle DA could store the address of the Roadside DA in its database. The Vehicle DA piggybacks the current Roadside DA address inside empty service replies such that the UA can become aware of the Roadside DA's address. Finally, the SLP UA can get the address of the Roadside DA from the Vehicle DA.

3.3. **Substitute request message.** The Substitute Request Message format is a derived form of the Service Request message. The Function ID in the header of SLP message is used to define the message type, and the ID of Substitute Request Message is 13. The major differences between the Substitute Request Message and Service Request message are the <DA add> string length field and the <DA add> string. The <DA add> string length field indicates the number of bytes in the <DA add> string, and the <DA add> string field contains the target DA address.

Both the SLP UA and DA should add the Substitute Request function. Thus, when an SLP UA in a vehicle gets the address of the Roadside DA from its Vehicle DA, the SLP UA can send a Substitute Request Message to the Vehicle DA. This Substitute Request Message contains the address of the Roadside DA and a discover request. When the Vehicle DA receives this Substitute Request Message, it uses the <DA add> string to retrieve the IP address of the Roadside DA and forwards the Substitute Request message to the Roadside DA. The Roadside DA returns its reply to the Vehicle DA. A caching mechanism could be implemented into the Vehicle DA so that this DA can cache the search results for use in subsequent service discovery. The detailed procedures are shown in Figure 2.

As shown in Figure 2, the Vehicle DA has two IP addresses, and the SLP UA is attached to the in-vehicle network. A media server registers its service with the Roadside

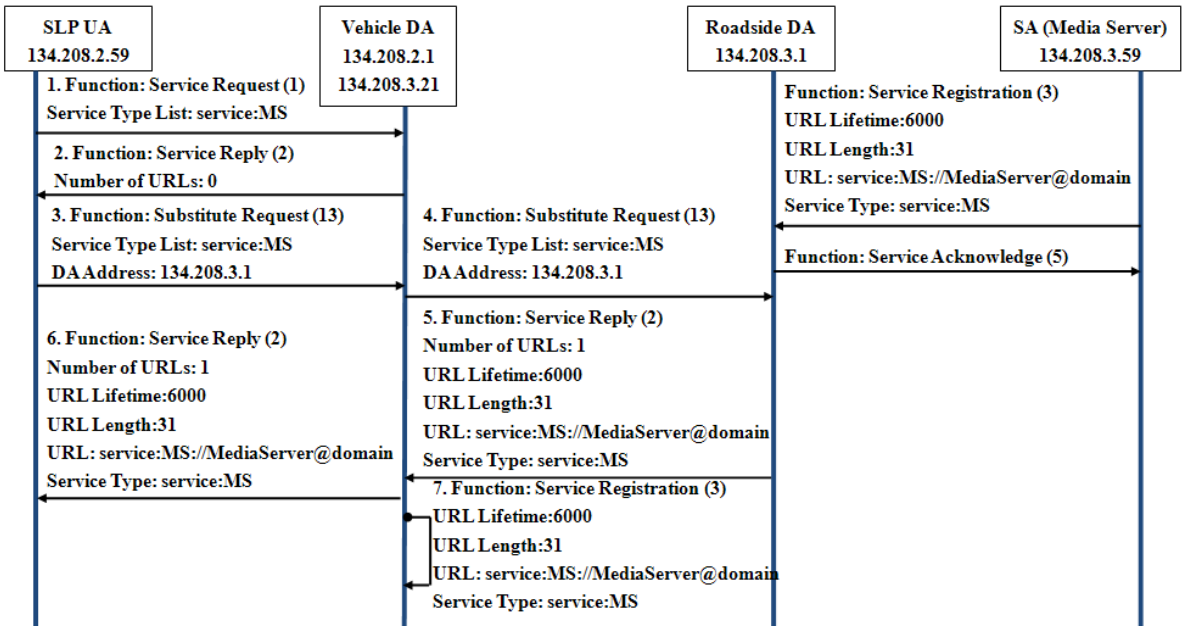

FIGURE 2. Substitute request flows

DA. When the SLP UA wants to discover a media server in the in-vehicle domain and it sends a Service Request message to the Vehicle DA, the Service Reply message will not contain any service information because there is no local media server in the in-vehicle network. The SLP UA then sends a Substitute Request Message with the Roadside DA address to the Vehicle DA. The Vehicle DA will resolve the Roadside DA's IP address and forward the Substitute Request message to the Roadside DA.

Because the Roadside DA contains the service information registered by the media server, a Service Reply message with the service information is sent from the Roadside DA. When the Vehicle DA receives the Service Reply message, the service information will be extracted from the Service Reply message and registered by the vehicle DA, and a copy of the Service Reply message will be sent to the SLP UA. Later, when a new Service Request is sent to the Vehicle DA, the Service Reply message will contain the media server information.

3.4. **Extended service discovery architecture on vehicular networks.** In the basic system module, the Vehicle DA can send a multicast server request to get the address of the Roadside DA, and only the SLP UA can send a Substitute Request Message with the Roadside DA address. Suppose the Vehicle DA could send a Service Request Message to a Roadside DA, the Vehicle DA would have more service information in its local database. Moreover, the Vehicle DA can manually add SLP DA addresses to its local database. For example, some public transit company may provide special network services for their transportation. The Vehicle DA can send a Substitute Request message with the company's DA address to the Roadside DA. The Roadside DA will resolve the company's DA address and send a Service Request message to the company's DA. When the Roadside DA receives the Service Reply message sent from the company's DA, the service information will be extracted from the Service Reply message and registered by the Roadside DA, and a copy of the Service Reply message will be sent to the Vehicle DA. Later, when a new Service Request is sent to the Roadside DA, the Service Reply message will contain the desired server information.

Thus, based on the Substitute Request Message and cache mechanism, a hierarchical directory agent architecture or Peer-to-Peer directory agent architecture can be proposed. For example, the hierarchical directory agent architecture can be constructed as the DNS system. When a lower level DA cannot find the desired service, an SLP UA can send the Substitute Request Message with the higher level DA address to the lower level DA. The lower level DA will send a Service Request message to the higher level DA, and store the search results in its database.

A two-layer directory agent architecture can be proposed based on the hierarchical directory agent architecture, as shown in Figure 1. The Vehicle DAs work as lower layer DAs, and the Roadside DAs work as higher layer DAs. The SLP UA issues a local service discovery to the lower layer DA. If the lower layer DA cannot find the desired service in the local database, the lower layer DA will issue a global service discovery procedure. In the global service discovery procedure, the lower layer DA will send a Substitute Request Message to its parent DA, and the <DA add> will contain the parent DA address or another Roadside DA address.

When the higher layer DA or Roadside DA receives the Substitute Request Message, the DA address will be retrieved. If the DA address is its address, the DA will send a Service Reply message to the lower layer DA. Otherwise, the Roadside DA will forward the Substitute Request Message to the destination DA. When the intermediate DA receives the Service Reply message sent from the higher layer DA, the service information will be extracted from the Service Reply message and registered in its database, and a copy of the Service Reply message will be sent to the lower layer DA. Thus, the SLP UAs can discover the desired services from the lower layer DAs, and not send the service request messages to the higher layer DAs or Roadside DAs. Therefore, the message overhead can be greatly reduced and the information hit ratio can be greatly improved in the two-layer directory agent architecture.

Moreover, we also propose a Remote Monitor System based on the two-layer directory agent architecture for Vehicle Network in [27]. The OBU works as a gateway and the function of Vehicle DA was integrated into the gateway. Some webcams can be installed in the vehicle, and managed by the security manager, which works as SLP SA and registers these services into the Vehicle DA. Some SLP DAs can be deployed on the roadside, and these SLP DAs are referred to as Roadside DAs. In the traffic information center (TIA), a TIA DA can be deployed, and the officer can use the Substitute Request Message to the SLP DA to discover the webcam service with the Vehicle DA's IP address. Then, the TIA DA will forward the Substitute Request Message to the Vehicle DA, and get the webcam service information. Finally, the officer can access the Vehicle's webcam service.

4. **Complexities Computation.** In the following we provide the message complexities computation for the proposed system architecture. The message overhead can be defined as the volume of messages when the SLP UAs send query messages to the SLP DAs and receive reply messages sent from the SLP DAs. Because that all the DA in the following architectures should send the DAAdvert message periodically, the overhead of DAAdvert message was not considered. Let R_i denote the Roadside DAi, V_j denote the Vehicle DAj and S_k denote the SLP SAk. The hop count between the R_i and V_j is $count(R_i, V_j)$, and the hop count between the R_i and R_j is $count(R_i, R_j)$. H_i denotes the cache hit ratio of the DAi. Let U_k denote the SLP UA_k, $VDA(U_k)$ denote the Vehicle DA of U_k, and $RDA(U_k)$ denote the desired Roadside DA of the Substitute Request Message sent from U_k. Let $DA(S_k)$ denote the SLP DA of S_k. The hop count between S_k and $DA(S_k)$ is $count(DA(S_k), S_k)$.

Theorem 4.1. *In the original SLP architecture, only the SLP UA can send the service request to the SLP DA, and SLP SA sends the registration to the SLP DA. Thus, the message complexity of the original system architecture is*

$$T_{orig} = \sum_{i=1}^{UA} 2 * count(VDA(U_i), U_i) + \sum_{i=1}^{SA} 2 * count(DA(S_i), S_i) \tag{1}$$

Proof: In the original SLP architecture, the SLP DA can only return the service response to the SLP UA. The SLP SA sends the registration to the SLP DA and receives the Ack message. The SLP DA cannot send any service request to any other SLP DA. Thus, these messages will be transmitted between the SLP UA and SLP DA. The hop count between the SLP UA and its Vehicle DA is $count(VDA(U_i), U_i)$, and the hop count between the SLP SA and its DA is $count(DA(S_i), S_i)$. When the SLP DA receives a service request, it will return a service response to the SLP UA. Thus, when an SLP UA sends a service request, its message overhead is $2 * count(VDA(U_i), U_i)$, and the registration overhead is $2 * count(DA(S_i), S_i)$. Thus, the total message complexity is as Equation (1).

Two service discovery architectures on vehicular networks were proposed in the previous section. The message complexity of the basic system architecture and two-layer directory agent architecture are provided in Theorems 4.2 and 4.3, respectively.

Theorem 4.2. *The message complexity of the basic system architecture is*

$$\begin{aligned} T_{basic} = & \sum_{i=1}^{UA} \left\{ \begin{array}{l} 2 * H_{VDA(Ui)} * count(VDA(U_i), U_i) + \\ 2 * (1 - H_{VDA(Ui)}) * (count(RDA(U_i), VDA(U_i)) + count(VDA(U_i), U_i)) \end{array} \right\} \\ & + 2 * \sum_{i}^{SA} count(DA(S_i), S_i) \end{aligned} \tag{2}$$

Proof: In the basic system architecture, the Vehicle DA can return the service response to the SLP UA if the desired services are found, and will send the Substitute Request Message to the Roadside DA if the desired services are not found. Thus, these messages will be transmitted between the SLP UA and Vehicle DA when the desired services are found in the Vehicle DA, and these messages will be transmitted between the SLP UA and Roadside DA when the desired services are not found in the Vehicle DA. The hop count between the SLP UA and its Vehicle DA is $count(VDA(U_i), U_i)$, and the hop count between the Vehicle DA and its Roadside DA is $count(RDA(U_i), VDA(U_i))$. The SLP SA only sends the registration to the SLP DA, and the registration overhead is $2 * count(DA(S_i), S_i)$. Therefore, the total message complexity of the basic system architecture is as Equation (2).

Theorem 4.3. *Let $TDA(V_j)$ denote the desired DA of the Substitute Request Message sent from V_j. The message complexity of the two-layer directory agent architecture is*

$$\begin{aligned} T_{2layer} = & \sum_{i=1}^{UA} \left\{ \begin{array}{l} 2 * H_{VDA(Ui)} * count(VDA(U_i), U_i) + \\ 2 * (1 - H_{VDA(Ui)}) * \\ (count(TDA(VDA(U_i)), VDA(U_i)) + count(VDA(U_i), U_i)) \end{array} \right\} \\ & + 2 * \sum_{i}^{SA} count(DA(S_i), S_i) \end{aligned} \tag{3}$$

Proof: In the two-layer system architecture, a Vehicle DA will issue a global service discovery procedure when the Vehicle DA cannot find the desired service in its local

database. In the global service discovery procedure, the Vehicle DA sends a Substitute Request Message to its parent DA or another Roadside DA address. The $TDA(VDA(U_i))$ denotes the desired DA of the Vehicle DA, and the hop count between the Vehicle DA and its desired DA is $count(TDA(VDA(U_i))), VDA(Ui))$. Thus, when the Vehicle DA issues a global service discovery procedure, the message overhead of the global service discovery procedure is $2*count(TDA(VDA(U_i))), VDA(U_i))$. The SLP SA only sends the registration to the SLP DA, and the registration overhead is $2*count(DA(S_i), S_i)$. Thus, the total message complexity is as Equation (3).

Theorem 4.4. *In the Mesh-enhanced SLP architecture, the SLP UAs only send the service request to the SLP DAs, and SLP SAs send the registration to the SLP DAs. But the SLP DA will share its registrations with some DAs in the same scope. Suppose all the SLP DAs are in the same scope and the amount of DA is M, and the number of hand off of each Roadside DA_i is HF_i, the message complexity of the Mesh-enhanced SLP architecture is*

$$T_{mSLP} = \sum_{i=1}^{UA} 2 * count(VDA(U_i), U_i) + \sum_{i=1}^{SA} 2 * (count(DA(S_i), S_i) + M) + \sum_{i=1}^{RDA} 2 * HF_i \quad (4)$$

Proof: In the Mesh-enhanced SLP architecture, the SLP UA can only send the request message to its SLP DA, and receives the service response from the SLP DA. Thus, the message overhead is $2*count(VDA(U_i), U_i)$. In the Mesh-enhanced SLP architecture, the SLP SAs only need to connect to one DA, and register its service. Then, the registration will then be propagated automatically from the SLP DA to other SLP DAs in the same registration scope. Thus, when an SLP SA registers its service to one DA, the propagated registration is M, and the registration overhead is $2 * (count(DA(S_i), S_i) + M)$. According to the definition of mSLP, each Vehicle DA should establish a peer connection with Roadside DAs, and must send some messages to establish and release the connections. Thus, the connection maintenance overhead at least is $2 * HF_i$. Therefore, in the worst case, the total message complexity is as Equation (4).

In the original SLP architecture, if the SLP UAs do not send service requests to Roadside DAs, the message overhead is lower than the basic system architecture and two-layer directory agent architecture. However, the service discovery hit ratio of the original SLP architecture would be much lower than that of the basic system architecture and two-layer directory agent architecture. In order to improve the service discovery hit ratio, the SLP UAs should send the service requests to Roadside DAs to discover the service, and the message overhead of the original SLP architecture would be higher than that of the basic system architecture and two-layer directory agent architecture. Moreover, the message overhead of the basic system architecture and two-layer directory agent architecture can be greatly reduced by the cache mechanism, which the original SLP architecture does not have.

In the Mesh-enhanced SLP architecture, although the SLP UA can get the desired service from any SLP DA, the SLP DAs need to propagate the registrations to other SLP DAs in the same registration scope, and each vehicle DAs must send many messages to establish and maintain the connections. Thus, in the worst case, the registration overhead and maintenance overhead of Mesh-enhanced SLP architecture can be huge. The registration overhead can be reduced when the mesh is used to split the scope into several scopes, but the SLP DA cannot share the registrations with other SLP DAs in a different scope. This means that the service discovery hit ratio will be decreased.

Moreover, suppose the average $HF_i = 0$, the number of SAs is N. From the Equations (3) and (4), the difference of T_{2layer} and T_{mSLP} is

$$T_{2layer} - T_{mSLP} = \sum_{i=1}^{UA} \left\{ 2 * (1 - H_{VDA(Ui)}) * count(TDA(VDA(U_i)), VDA(U_i)) \right\} - 2NM \tag{5}$$

Let the number of UAs is X and the average cache hit ratio of the DA is 0.5. When the Roadside DAs are deployed as a grid network, the average $count(TDA(VDA(Ui)), VDA(Ui))$ will be $\sqrt{M}$. Equation (5) can be simplified as the following equation:

$$T_{2layer} - T_{mSLP} = X\sqrt{M} - NM \tag{6}$$

It means that when $X \geq N\sqrt{M}$, the message overhead of two-layer directory agent architecture would be larger than that of Mesh-enhanced SLP architecture. However, from the simulation results, the average hit ratio of two-layer directory agent architecture is very high, and the handoff overhead of Vehicle DA was not considered in Equation (6). Thus, we think that the proposed two-layer directory agent architecture with cache mechanism can provide a higher service discovery hit ratio with a lower message overhead.

5. **System Implementation.** For our implementation, we integrated the SLP UA functions into the Service Integrated User Agent (SIUA) which is implemented based on the SIP communicator project [28] and the ideas proposed in [29,30]. We needed to extend the SLP implementation to add a Substitute Request Message and a cache mechanism to the SLP DA. We implemented the proposed mechanism based on the open-source code for Maven jSLP [31]. We also integrated the proposed mechanism into the residential gateway proposed in [32], which was implemented based on the open-source project of the OW2 Forge Oscar [33]. In our implementation, jSLP is installed in the RG and Vehicle gateway. We added an SLP bridge bundle, as we described in [32], referred to as "SACP", as shown in Figure 3. SACP works as a SLP SA to discover devices and services from the Service Registry of the OSGi platform and register these services and devices with the SLP DA. Using the SACP, the residential gateway provides automated heterogeneous device discovery, registry, and management.

As shown in Figure 3, in the in-vehicle network, the media server includes an SLP SA (Advertiser), which announces the media service to the SLP DA. The SIUA contains the SLP UA functions, and we implemented the SIUA based on the SIP communicator. The

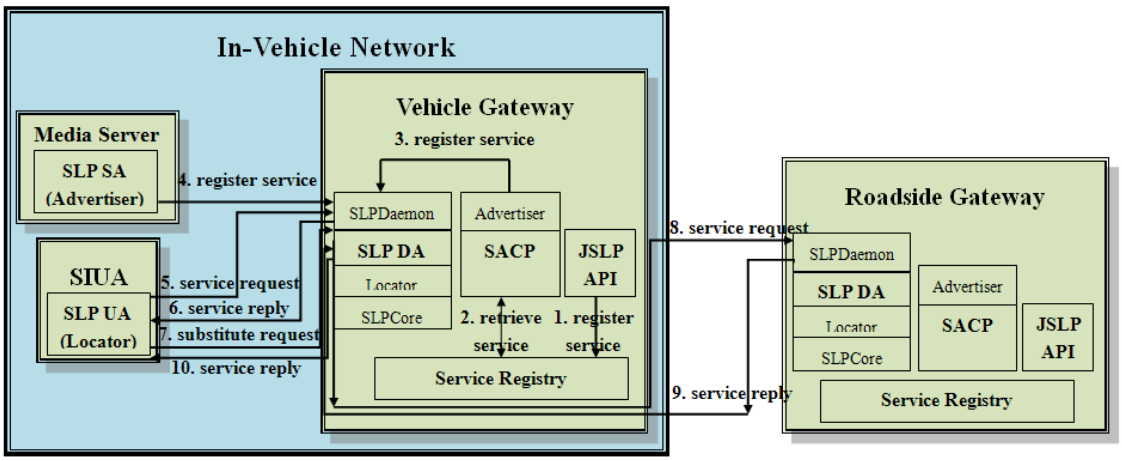

FIGURE 3. The vehicle service discovery network architecture

SLP DA is installed in the vehicle gateway and RG. The SLP DA contains three major modules: SLPDaemon, SLPCore, and Locator. The SLPDaemon is used to process the registration messages sent from SLP SAs and to cache the service information retrieved from the Substitute Request results. The SLPCore is used to process the Service Request messages sent from the SLP UA, and to send the reply message to an SLP UA. The Locator will send a Service Request to another SLP DA after receiving the Substitute Request message. We added a procedure for the Substitute Request Message to the SLPCore.

As shown in Figure 3, when the SLP DA booted, the JSLP API will register its service with the service registry, which is a bundle provided by Oscar. We implement the SACP bundle to discover the services registered in the service registry, and register these services in the SLP DA. Next, the SLP SA will send the register message to SLP DA. When the SLP DA receives the service request sent from an SLP UA, it will reply with the service reply message and search results to the SLP UA. Suppose the service reply message contains no search result, the SLP UA can send a Substitute Request Message to the SLP DA. When SLPCore receives a Substitute Request Message sent from an SLP UA, it calls the Locator to send a Service Request message to another SLP DA according to the information in the Substitute Request Message. When the Locator receives a Service Reply, it sends the registration message to the SLPDaemon and passes the reply message to the SLPCore. The SLPCore then forwards the Service Reply to the SLP UA.

Initially, the media server registers the streaming services with the roadside SLP DA. The SIUA1 then sends a Service Request to the vehicle SLP DA to discover a streaming service. However, the vehicle SLP DA has no streaming service information and will send a Service Reply with an empty entry. Thus, the SIUA1 sends the Substituted Request to the vehicle SLP DA with the roadside SLP DA's address. When the vehicle SLP DA receives the Substitute Request, it sends the Service Request to the roadside SLP DA to discover the streaming service. When the vehicle SLP DA receives the Service Reply with the streaming service information sent from the roadside SLP DA, the vehicle SLP DA will register the service information, and forward the Service Reply to the SIUA1. When the SIUA1 receives the Service Reply, the streaming service information will be shown on the GUI.

6. **Performance Evaluations.**

We performed a simulation to evaluate the performance of our architecture. We conducted simulations to compare the original SLP architecture, the basic system architecture without cache mechanism, and the basic system architecture with cache mechanism. The Roadside DAs are deployed as a grid network. Each SA provides only one service and is randomly deployed in the grid network. In the original SLP architecture, the SLP UAs are randomly deployed in the grid network. In the basic system architecture, the Vehicle DAs are randomly deployed in the grid network, and each Vehicle DA has some SLP UAs. The SLP UAs are also randomly deployed and each SLP UA will initiate a service discovery request with the nearest Roadside DA or Vehicle DA.

In each simulation, each SLP SA will register its service with the nearest Roadside DA. The mobility profile of Vehicle DA is generated based on the city mobility model. The object made 200,000 moves in each mobility profile to ensure that the resulting mobility profile was statistically significant. To ensure stable results, 100 mobility profiles were run of each simulation. The parameters are described in Table 2.

Four performance metrics are considered in the paper. There are Hit Ratio, Message Overhead, Capability of Service Information Collection, and Substitute Query ratio. The

TABLE 2. Simulation parameters

Number of Roadside Directory Agents	400
Number of User Agents	12000
Number of Service Agents	4000
Number of Vehicle Directory Agents	1200
Service Discovery tasks per run	100000
Total services on the network	20
Global Discovery attempt frequency	1∼9
Total runs	20

Hit Ratio is the ratio of successful service request to the total service request. The Hit Ratio refers to the chance of success when a service first enters the architecture. The Message Overhead is the hop count of a service discovery process. It includes the service request packets and the service reply packets, the necessary message of local service discovery and global service discovery. The Capability of Service Information Collection refers to how much service information a DA can cache for the run time. The Substitute Query ratio displays the utility rate of substitute query to normal query in each environment.

In Figures 4 to 7, the red lines denote the results of the original SLP architecture, the green line denotes the results of the basic system architecture without cache mechanism and the blue line denotes the results of the basic system architecture with cache mechanism. Figure 4 shows the average hit ratio of different SLP architectures at different service discovery loads. As shown in Figure 4, the average hit ratio of the original SLP architecture is very low and always the same no matter what the load of Substitute query rate or discovery tasks is. This is because the SLP UAs only discover the desired service in its local SLP DA. When the load of Substitute query rate or discovery tasks increased, the average hit ratio of the basic system architectures increases. This is because when the SLP UAs cannot find the desired service from the local SLP DA, the SLP UAs can send a Substitute Request Message to a Vehicle DA to discover services from another SLP DA.

Figure 5 shows the average message overhead of different SLP architectures at different service discovery loads. As shown in Figure 5, the average message overhead of the original SLP architecture is very low and always the same no matter what the load of Substitute query rate or discovery tasks is. This is because the SLP UAs only discover the desired service in its local SLP DA. When the number of Substitute query rate is 1 or when discovery tasks increase, the average message overhead of the basic system architecture with cache mechanism is slightly increased. Moreover, when the number of Substitute query rate increases, the average message overhead of the basic system architecture with cache mechanism is very stable. This is because most of the service information will be cached in each Vehicle DA, and the SLP UAs do not need to send the Substitute Request Message. It is very obvious that the average message overhead becomes very heavy when the cache mechanism is not implemented in the SLP DA.

Figure 6 shows the cache ability of different SLP architectures at different service discovery loads. It is very obvious that the number of cached service increases when the number of Substitute query rate or discovery tasks increases. Thus, as shown in Figure 6, the SLP UAs can find the desired service from the local SLP DA, and do not need to send the Substitute Request Message.

Figure 7 shows the substitute query ratio of different SLP architectures at different service discovery loads. The original SLP architecture has no ability to provide the substitute query, and its substitute query ratio is 0. When the number of Substitute query rate is 1 or when the discovery tasks increase, the substitute query ratio of the basic

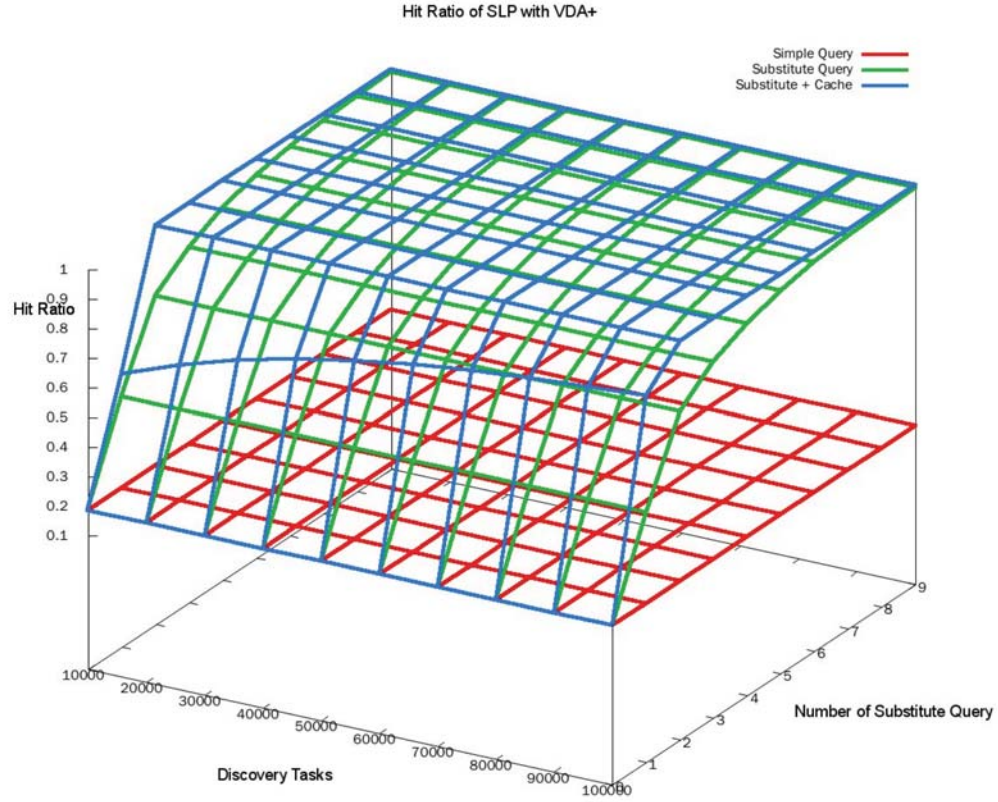

FIGURE 4. Average hit ratio

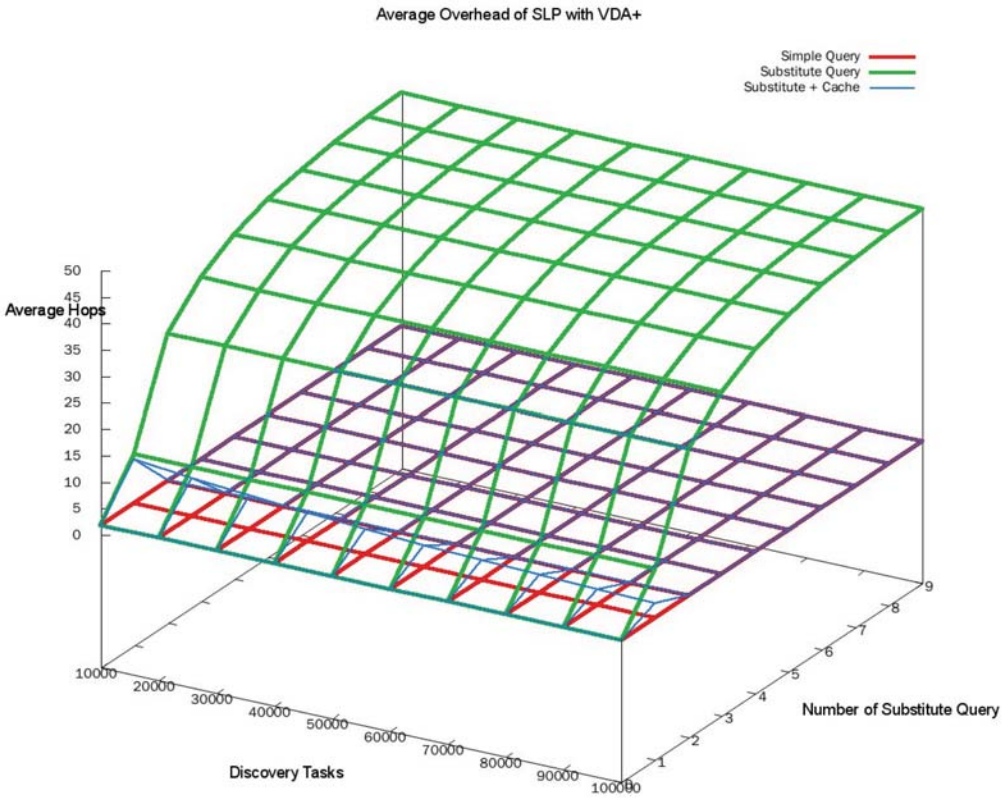

FIGURE 5. Average message overhead

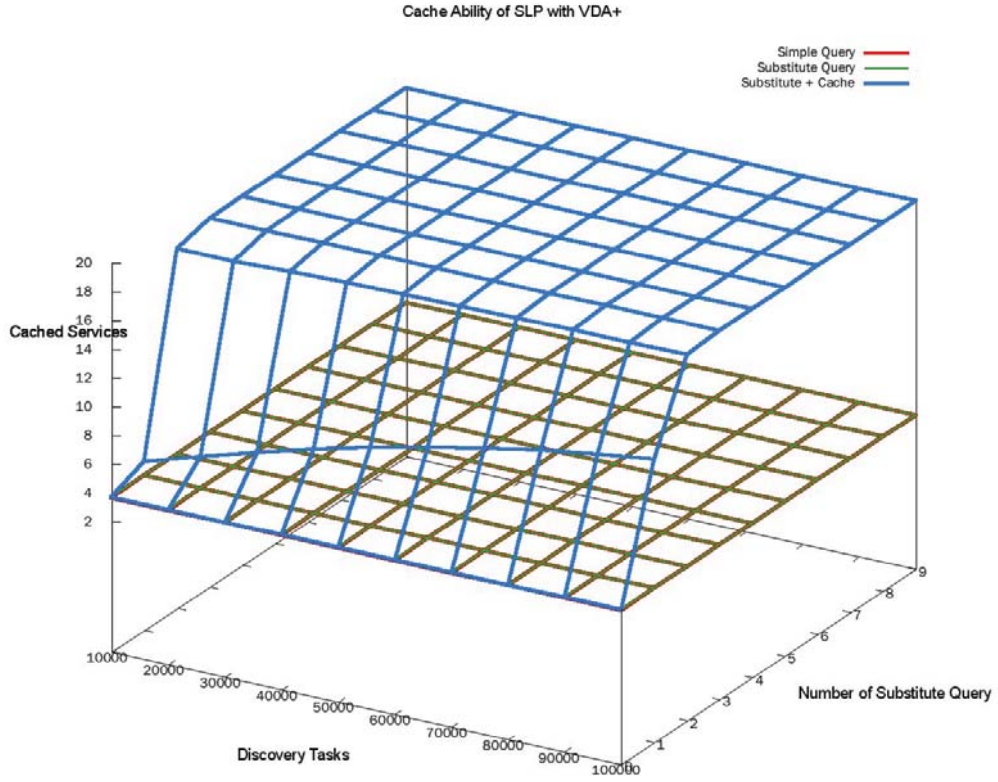

FIGURE 6. Cache ability

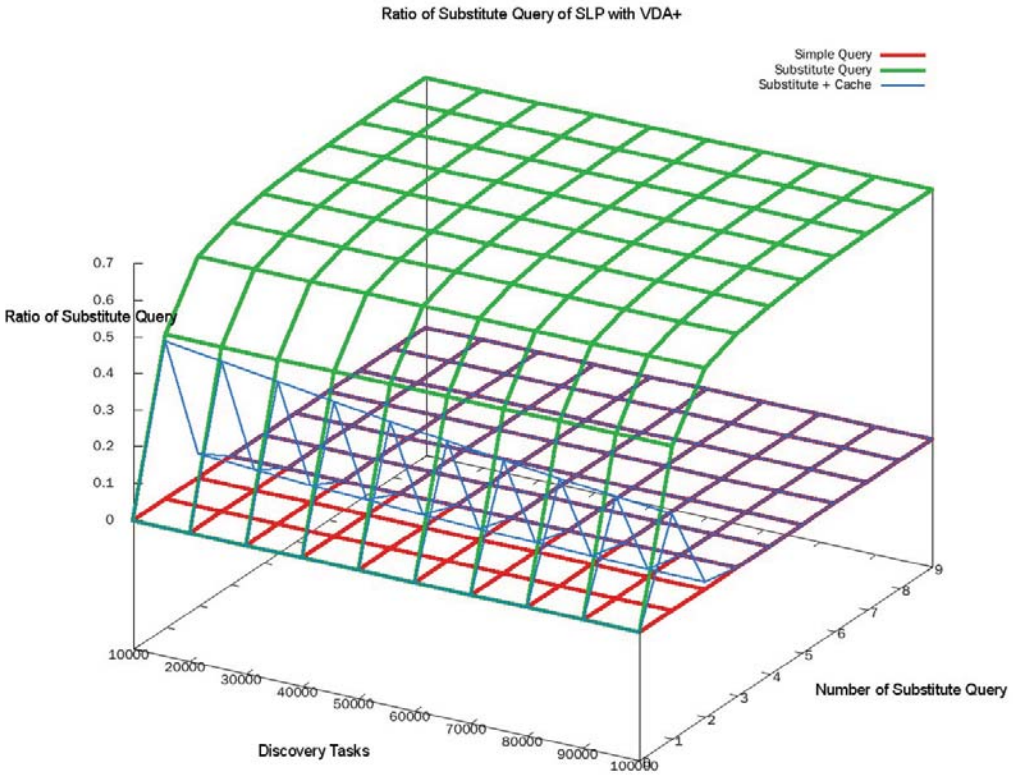

FIGURE 7. Substitute query ratio

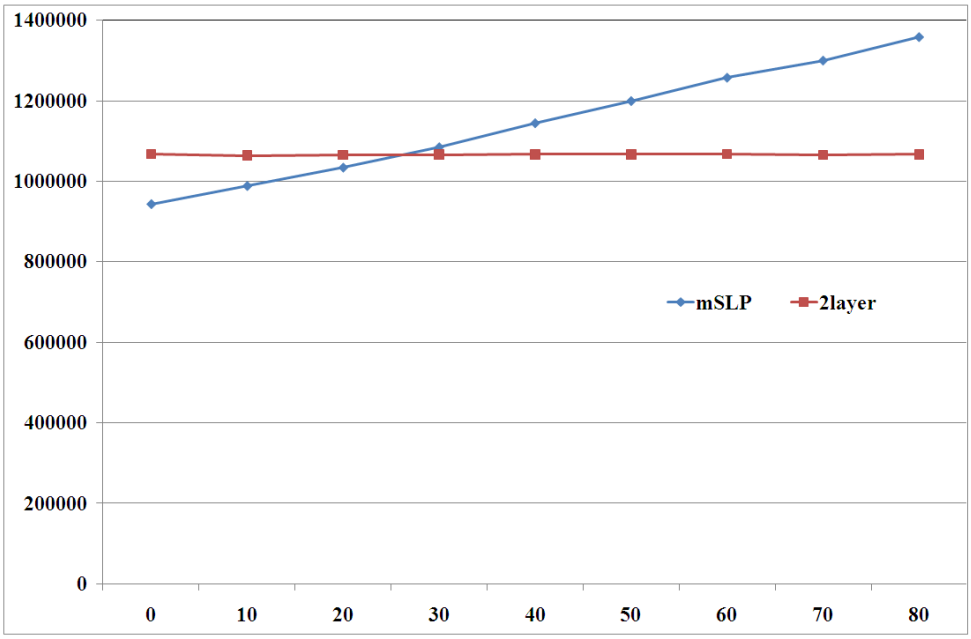

FIGURE 8. Message overhead

system architecture with cache mechanism is about 40% to 50%. When the number of substitute queries increases, the substitute query ratio of the basic system architecture with cache mechanism is reduced. This is because most of the service information will be cached in each Vehicle DA, and the SLP UAs can find most of the service information from its local SLP DA, and do not need to send the Substitute Request Message. It is very obvious that the substitute query ratio becomes very heavy when the cache mechanism is not implemented in the SLP DA. These simulation results show that the proposed substitute query mechanism can improve the average hit ratio, and the cache mechanism can reduce the message overhead of Substitute Request Messages.

Figure 8 shows the message overhead of mSLP architecture and two-layer SLP architecture at different vehicle speed. In the mSLP architecture, each Vehicle DA should maintain a peer connection with Roadside DAs. Each Vehicle DA must send a release message to the old Roadside DA when the Vehicle DA leaves a grid, and send a request message to establish a connection with the new Roadside DA when the Vehicle DA enters a grid. According to the Equation (4), when the vehicle speed is increased, the message overhead is increased. In contrast to the mSLP architecture, the message overhead of two-layer SLP architecture is very stable. In fact, as shown in Equation (3) and Figure 5, the message overhead of two-layer SLP architecture will be affected by the hit ratio and the number of Substitute Request Message. The results of Figure 4 show that the hit ratio of two-layer SLP architecture is very high when the number of Substitute Request Message is 3. Therefore, when the vehicle speed is increased, the message overhead of two-layer SLP architecture could be better than that of mSLP architecture.

7. **Conclusions.** A great challenge in recent years has been the question of how to discover and manage the desired network services in vehicle networks. Most current service discovery protocols are designed for fixed networks or only for MANET/VANET, and are unsuitable in the network mobility (NEMO) environment. In this paper, we proposed a service discovery based on SLP for heterogeneous vehicle networks within a NEMO environment. We introduced the Substitute Request Message to provide a substitute query technique between the SLP DAs. The cache mechanism is also implemented into the SLP

DA to reduce the message overhead of Substitute Request Messages. Numerical results indicate that the proposed architecture can provide high average data hit ratio, and very low message overhead.

Acknowledgment. This work was supported by the National Science Council of Taiwan under Grants NSC97-2221-E-259-036, NSC98-2221-E-259-014 and NSC99-2221-E-259-006.

REFERENCES

[1] R. Baroody, A. Rashid, N. Al-Holou and S. Hariri, Next generation vehicle network (NGVN): Internet access utilizing dynamic discovery protocols, *Proc. of the IEEE/ACS International Conference on Pervasive Services*, pp.81-88, 2004.

[2] A. Boukerche and K. Abrougui, Performance evaluation of a hybrid adaptive service discovery protocol for next generation heterogeneous vehicular networks (HVNs), *Proc. of the 3rd ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*, 2008.

[3] X. Hu and M. Huang, An intelligent solution system for a vehicle routing problem in urban distribution, *International Journal of Innovative Computing, Information and Control*, vol.3, no.1, pp.189-198, 2007.

[4] Y. Wu, P. Ji and T. Wang, An empirical study of a pure genetic algorithm to solve the capacitated vehicle routing problem, *ICIC Express Letters*, vol.2, no.1, pp.41-45, 2008.

[5] X. Hu, Y. Li, J. Guo, L. Sun and A. Z. Zeng, A simulation optimization algorithm with heuristic transformation and its application to vehicle routing problems, *International Journal of Innovative Computing, Information and Control*, vol.4, no.5, pp.1169-1181, 2008.

[6] R. Friedman and G. Kliot, Location services in wireless ad hoc and hybrid networks: A survey, *TR CS-2006-10*, Department of Computer Science, The Technion, 2006.

[7] C. N. Ververidis and G. C. Polyzos, Service discovery for mobile ad hoc networks: A survey of issues and techniques, *IEEE Communications Surveys & Tutorials*, vol.10, no.3, pp.30-45, 2008.

[8] J. Su and W. Guo, A survey of service discovery protocols for mobile ad hoc networks, *Proc. of the International Conference on Communications, Circuits and Systems*, pp.398-404, 2008.

[9] A. N. Mian, R. Baldoni and R. Beraldi, A survey of service discovery protocols in multihop mobile ad hoc networks, *IEEE Pervasive Computing*, vol.8, no.1, pp.66-74, 2009.

[10] S. Helal, Standards for service discovery and delivery, *IEEE Pervasive Computing*, vol.1, no.3, pp.95-100, 2002.

[11] E. Guttman, C. Perkins, J. Veizades and M. Day, Service location protocol, Version 2, *RFC 2608*, IETF, 1999.

[12] J. Veizades, E. Guttman, C. Perkins and S. Kaplan, Service location protocol, *RFC 2165*, IETF, 1997.

[13] K. Arnold, R. Scheifler, J. Waldo, B. O'Sullivan and A. Wollrath, *Jini Specification*, 1st Edition, Addison-Wesley Longman Publishing Co., Inc, 1999.

[14] Jini.org, *Jini Architecture Specification*, http://www.jini.org/wiki/Jini_Architecture_Specification.

[15] *Universal Plug and Play*, http://www.upnp.org/.

[16] Y. Y. Goland et al., *Internet-Draft: Simple Service Discovery Protocol/1.0*, http://quimby.gnus.org/internet-drafts/draft-cai-ssdp-v1-03.txt, 1998.

[17] R. Hermann, D. Husemann, M. Moser, M. Nidd, C. Rohner and A. Schade, DEAPspace: Transient ad-hoc networking of pervasive devices, *Proc. of the 1st ACM International Symposium on Mobile Ad Hoc Networking & Computing*, pp.133-134, 2000.

[18] S. Helal, N. Desai, V. Verma and C. Lee, Konark – A service discovery and delivery protocol for ad-hoc networks, *Proc. of the 3rd IEEE Conference on Wireless Communication Networks*, pp.2107-2133, 2003.

[19] C. Lee, A. Helal, N. Desai and S. Helal, Konark: A system and protocols for device independent, peer-to-peer discovery and delivery of mobile services, *IEEE Trans. Systems, Man and Cybernetics, Part A*, pp.682-696, 2003.

[20] D. Chakraborty, A. Joshi, Y. Yesha and T. Finin, Toward distributed service discovery in pervasive computing environments, *IEEE Trans. Mobile Computing*, vol.5, no.2, pp.97-112, 2006.

[21] Z. Gao, L. Wang, M. Yang and X. Yang, CNPGSDP: An efficient group-based service discovery protocol for MANETs, *Computer Networks*, vol.50, no.16, pp.3165-3182, 2006.

[22] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. F. Kaashoek, F. Dabek and H. Balakrishnan, Chord: A scalable peer-to-peer lookup protocol for Internet applications, *IEEE/ACM Trans. Networking*, vol.11, no.1, pp.17-32, 2003.
[23] H. Artail, K. M. Mershad and H. Hamze, DSDM: A distributed service discovery model for manets, *IEEE Trans. Parallel and Distributed Systems*, vol.19, no.9, pp.1224-1236, 2008.
[24] W. Zhao, H. Schulzrinne and E. Guttman, Mesh-enhanced service location protocol (mSLP), *RFC 3528*, IETF, 2003.
[25] S. Cheshire and M. Krochmal, *DNS-Based Service Discovery*, Internet-Draft, IETF, 2011.
[26] S. Cheshire and M. Krochmal, *Multicast DNS*, Internet-Draft, IETF, 2011.
[27] M. X. Chen and B. L. Lin, Design remote monitor system based on OSGi platform in vehicle gateway for vehicle network, *Proc. of the 2nd International Conference on Next Generation Information Technology (ICNIT)*, pp.12-17, 2011.
[28] *SIP Communicator*, http://sip-communicator.org/.
[29] M. X. Chen, C. J. Peng and R. H. Hwang, SSIP: Split a SIP session over multiple devices, *Computer Standards and Interfaces*, vol.29, no.5, pp.531-545, 2007.
[30] M. X. Chen and F. J. Wang, Session integration service over multiple devices, *International Journal of Communication Systems*, vol.23, no.5, pp.673-690, 2010.
[31] *Maven – jSLP*, http://jslp.sourceforge.net/.
[32] M. X. Chen and T. C. Tzeng, Integrating service discovery technologies in OSGi platform, *Computer Standards and Interfaces*, vol.33, no.3, pp.271-279, 2010.
[33] *Oscar OSGi Framework*, http://oscar.ow2.org/.