



# Design and field evaluation of geographical location-aware service discovery on IPv6 GeoNetworking for VANET

Satoru Noguchi, Manabu Tsukada, Thierry Ernst, Astuo Inomata, Kazutoshi Fujikawa

## ► To cite this version:

Satoru Noguchi, Manabu Tsukada, Thierry Ernst, Astuo Inomata, Kazutoshi Fujikawa. Design and field evaluation of geographical location-aware service discovery on IPv6 GeoNetworking for VANET. EURASIP Journal on Wireless Communications and Networking, SpringerOpen, 2012, 2012 (1), pp.29. <hal-00784409>

HAL Id: hal-00784409

<https://hal.inria.fr/hal-00784409>

Submitted on 4 Feb 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

RESEARCH

Open Access

# Design and field evaluation of geographical location-aware service discovery on IPv6 GeoNetworking for VANET

Satoru Noguchi<sup>1\*</sup>, Manabu Tsukada<sup>2</sup>, Thierry Ernst<sup>2</sup>, Astuo Inomata<sup>1</sup> and Kazutoshi Fujikawa<sup>1</sup>

## Abstract

Service discovery is an essential component for distributed mobile applications in vehicular communication systems. While there have been numerous service discovery protocols, applications for vehicular communication systems pose additional requirements: discover services according to geographical position inside dynamic mobile environments. In this article, we propose a geographical location aware service discovery mechanism for vehicular ad-hoc networks (VANETs). The proposed mechanism exploits an IPv6 multicast service discovery protocol on IPv6 GeoNetworking specified by the GeoNet project. Thanks to the GeoBroadcast mechanism, it efficiently propagates service discovery messages to a subset of nodes inside a relevant geographical area with encapsulating IPv6 multicast packets. We implemented the proposed mechanism to the ns-3 network simulator, furthermore we integrated the prototype system using CarGeo6, an open source implementation of IPv6 GeoNetworking, with openSLP. Our simulation and real field evaluation results show the system can discover services with low latency and low bandwidth usage in VANETs even via multi-hop.

**Keywords:** service discovery, VANET, IPv6, GeoNetworking, multicast, ITS applications, ns-3 network simulator, field evaluation

## 1 Introduction

Applications for intelligent transportation system (ITS) aim at providing road users with improved traffic safety, traffic efficiency, and additional values in vehicular communication systems [1]. Recently various ITS stakeholders have been working on specifying ITS applications [2,3]. In general, ITS applications are distributed mobile applications composed of a number of distinct *services*; software components integrated into wide variety of nodes in vehicular ad-hoc networks (VANETs), in which most participants are mobile nodes equipped with vehicles. For instance, services can (i) provide characteristics of vehicles and the roadside, e.g., mechanical condition, colors of traffic light, etc. (ii) process consumers' request, e.g., manipulate electronic gates, perform payment, notify drivers with road traffic information, and (iii) aggregate road traffic information

from other vehicles and the roadside. Each application may consume multiple services, therefore services should be self-contained, modular, and application independent entities so that service consumers can share and reuse existing services.

Service discovery protocol (SDP), which dynamically discovers communication endpoints of available services, is essential for distributed applications in order to orchestrate necessary services remotely in mobile networks. To communicate with necessary services, applications may directly send data to a particular group of hosts that *may* operate necessary services, otherwise they selectively send data to a number of hosts that *certainly* operate necessary services. In the former case, applications simply broadcast data to the considered network regardless of existence of appropriate services. On the other hand, in the latter case, at first applications resolve the communication endpoint of necessary services, and then they exclusively deliver data to the discovered services. In general, broadcasting is suitable for critical use cases within single-hop distance, which

\* Correspondence: satoru-n@is.naist.jp

<sup>1</sup>Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan

Full list of author information is available at the end of the article

have tight latency requirements (e.g., detect and notify a risk of crash to driver) because it can quickly propagate messages without any communication in advance. However, if applications need to communicate with services located in multi-hop distance, the broadcast-based communication may waste bandwidth, especially in a dense network. In the worst case, it leads applications to send data to all nodes inside the network. Furthermore, even if there are no appropriate services in the network, the broadcast based communication cannot prevent applications to propagate data. Although applications may use a static configuration or a centralized directory server to discover available services instead, such solutions are not applicable because of the characteristics of ITS services: services are mostly nomadic in VANETs, thus available services in a VANET are time-varying, and VANETs are not capable of introducing a static centralized entity. On the other hand, SDPs enable applications to dynamically discover the existence, characteristics, and communication endpoints of services. Even if the physical and topological locations of services change frequently, SDPs can discover actually available services. Thanks to SDPs, applications only need to specify a type and attributes of the service. SDPs then return a list of appropriate services that contains communication endpoints of them.

Although a number of SDPs have been proposed to discover nomadic services within mobile ad hoc networks (MANETs), discovering ITS services in VANETs raises further requirements:

1. Applications need to discover services according to geographical location, because ITS-related services are highly dependent on geographical location, such as cameras embedded in a particular intersection, vehicles within a certain distance of a corner, etc.
2. Services should be discovered as quickly as possible due to tight latency requirements of ITS applications (i.e. from 10 to 1000 ms [1,2]).
3. ITS applications need to avoid consuming unnecessary bandwidth.

A potential solution is to use IP multicast in cooperation with ad-hoc routing protocols, so that it can efficiently react to the change of the network topology due to the mobility of the vehicles and also avoid broadcasting to all vehicles in the network. IPv6 multicast-based SDPs, specifically service location protocol version 2 (SLPv2) [4,5] and multicast DNS (mDNS) with DNS-based service discovery (DNS-SD) [6,7], over existing MANET routing protocols may therefore be possible foundation for service discovery in VANETs.

However, such a solution does not satisfy the requirement 1, 'geographical service discovery', mentioned

above. Despite they can manage geographical location as an attribute of service within the SDP mechanism in the application layer, in this case service discovery messages must be delivered to all nodes in a certain multicast group, which is assigned to discover services. Consequently, the requirement 3, 'efficient bandwidth consumption' mentioned above, is not met either.

In this article, we propose a service discovery mechanism that locates services inside a particular geographical area in VANETs. Our main contributions in this research are:

- A low latency, low-cost geographical service discovery using a legacy IPv6 multicast based SDP over IPv6 GeoNetworking.
- Dynamic geographical destination management from SDP.
- Field evaluation of the SDP over IPv6 GeoNetworking with actual implementation on Linux.

The proposed mechanism is composed of IPv6 multicast-based service discovery protocol in combination with geographical addressing and routing; SLPv2 with [8] by IETF, and IPv6 GeoNetworking defined by the GeoNet project [9]. Modifications of SLP for IPv6 specified in [8] enable SLP to use multiple IPv6 multicast groups which allows one-multicast-address-for-one-service usage. IPv6 GeoNetworking furthermore enables to deliver regular IPv6 packets according to geographical location so that upper layer entities can transparently use the geographical routing functionality as legacy IPv6.

The rest of this article is organized as follows. Section 2 shows existing VANET routing mechanisms and SDPs, then Section 3 describes assumptions of SDPs for ITS applications. Section 4 proposes our geographical location aware service discovery mechanism using SLPv2 and IPv6 GeoNetworking. We then present evaluation results in Section 5. Section 6 finally concludes the article.

## 2 Related work

A service discovery mechanism is basically identified as a combination of an application layer SDP and a particular network layer protocol; in other words, a SDP works on an underlying routing protocol. In contrast to the separate integration of application layer SDPs on routing protocols, cross-layer solutions have also been studied. They directly inject SDP capabilities into underlying routing protocols, e.g., [10]. Although the cross-layer solution efficiently discovers services thanks to the direct interaction with the SDP and routing protocol, it loses the modularity of each protocol since SDPs are in turn tightly connected to a particular routing protocol. Regarding the service discovery for ITS, the separated

solution is more feasible because VANETs for ITS will be developed and operated by several different stakeholders, which may use a number of different communication technologies [1,2].

The selection of SDPs and underlying routing protocols is highly dependent on use cases. For instance, a UDP-based SDP with link-local scope IP multicast on traditional routing protocols can be used for small-scale static network [4,6,11]. For wide-area service discovery (i.e., in the Internet), SDP with overlay routing mechanisms on several intra/inter-domain IP routing protocols can be used [12]. On the other hand, for mobile service discovery within a VANET, SDPs need to be integrated over ad-hoc routing protocols, e.g., [13-15]. This section therefore explores existing SDPs and ad-hoc routing protocols that can be used for VANET.

### 2.1 Service discovery protocols

SDPs provide a set of mechanisms such as service description language, discovery function, registration function, and application layer messaging mechanism [16]. In this paper, we focus on these two functions and the messaging mechanism.

SDPs are basically composed of service consumers (SCs), service providers (SPs), and service directories (SDs). SPs manage the description of services (e.g., service type, service specific attributes, and communication endpoint of services), and possibly register their available service descriptions to SDs, while SCs try to discover services by asking SPs and SDs. SDs are centralized cache entities that act as services' directory. SDs may not be introduced to a small and/or mobile network, in which such a centralized entity does not fit the characteristics of the network, i.e., infrastructure-less mobile ad-hoc network. Depending on SDPs, each component can communicate with either unicast or multicast with/without SD: unicast is appropriate for wide-area service discovery, in which SCs need to discover a topologically distant service in the Internet. In this case, SCs and SPs normally use SDs to avoid broadcasting messages. On the contrary, multicast can be used for small, dynamic mobile network. In such a case SCs and SPs do not need to rely on SDs; they can directly propagate messages within a considered network. Consequently, IP multicast based SDPs fit the characteristics of VANET, in which most participating nodes are mobile and there is no centralized entity.

One of SDPs based on IP multicast is SLPv2 standardized by IETF [4,5]. It introduces three system components: user agent (UA), service agent (SA), and optional directory agent (DA), which correspond to the above-mentioned SC, SP, and SD, respectively, in the context of this article. A UA issues a service request (*SrvRqst*), which contains a type and attributes of the requested

service, to SPs or DAs. If a service managed by a SP and/or DA satisfies the request, the SP/DA returns a service reply (*SrvRply*), which contains a list of URL representation of all available services in the considered network, to the UA. UAs can either directly send *SrvRqst* to SAs via IP multicast if DAs are not installed. If DAs are available, UAs must send the request to DAs via unicast. SAs always return *SrvRply* to UAs via unicast.

While SLPv2 has originally used only one IPv4 multicast address to communicate with SAs, the modification specified by [8] has enabled SLPv2 to discover services over IPv6. It allows using multiple IPv6 multicast addresses assigned for each service (available address range is FF0x::1:1000/118 [17]). SAs join the multicast groups that correspond to their type of service. The multicast address is determined according to a hash algorithm, which generates a numerical value (0-1023: corresponds to the range of the allocated multicast address) from a service type's string representation. From the communication point of view, the benefit of this modification is to send *SrvRqst* to a specific subset of nodes that certainly manages a particular service using the IPv6 multicast group assigned for each service. If there are a large number of SAs that operate several different services, this modification can significantly reduce bandwidth usage.

Multicast DNS (mDNS) with DNS-based service discovery (DNS-SD) is also a well-known service discovery mechanism using IP multicast [6,7], which supports both IPv4 and IPv6. It discovers services using the regular DNS message via IP multicast with introducing a special DNS domain '.local.' Available services are described in a list of DNS resource records, e.g., SRV, TXT, etc. Contrary to regular DNS, DNS servers (corresponds to SDs) are not necessarily used because nodes having mDNS/DNS-SD capability (corresponds to SPs) work as distributed DNS servers, thus DNS clients (corresponds to SCs) are able to directly communicate with SPs.

Unlike SLPv2 with the IPv6 modification, in mDNS/DNS-SD, only one IP multicast group is used for both request and reply (FF0x::FB in IPv6 [17]). All SCs inside a considered network are therefore able to *listen* SP's reply for a particular discovery request. It may help to discover services quickly as a kind of cache entry.

Thanks to the link-local scope IP multicast, each SDP can efficiently work in static and/or local networks. However, regarding the service discovery for ITS in VANET, they may consume significant bandwidth because they do not have any dedicated mechanism to handle geographical position but rely on regular IP multicast. The geographical position of SP is just one of attributes of services in the application layer SDP.

Although they use IP multicast, SCs need to ask all SPs joining a multicast group within a considered network to discover a service, which is inside a specific geographical area.

## 2.2 Routing protocols for VANET

We classify routing protocols for VANET as two categories: traditional MANET routing protocols and the geographical routing protocol designed for VANET.

AODV [13] and OLSR [14] by IETF are representative routing protocols of the former category. These protocols dynamically construct network topology using flooding reactively or proactively, respectively. They enable to deliver packets with limited number of forwarding instead of broadcasting, so that a sender can deliver packets rapidly without wasting bandwidth. Each protocol has capability to take care of link failure due to nodes' mobility; their routing algorithm can automatically reconstruct the routing topology. However, although they can be applied to dynamic mobile networks, they do not satisfy the previously mentioned requirements of ITS applications: discover services according to geographical position. As we mentioned above, it may be possible to leave the geographical position handling to SDP, but in such a case sender nodes need to transmit service discovery packets to all nodes in the worst case.

On the other hand, IPv6 GeoNetworking is a reference specification of IPv6 operated over GeoNetworking developed by the GeoNet project, which conforms the C2CNet specification [2,15]. C2CNet, specified by the CAR 2 CAR communication consortium [18], is a communication layer dedicated to car-to-car communications and is located between the network layer and the link layer. It supports geographical addressing and routing by means of encapsulation of IPv6 packet with a new C2C header containing geographical locations. Although the C2CNet layer exchanges packets without IP, the GeoNet project has defined how to transmit IPv6 packet over C2CNet ('IPv6 over C2CNet'). This is performed transparently to upper layers; In IPv6 GeoNetworking-enabled VANETs, each node is assigned a C2CNet identifier. When a node sends out an IPv6 packet, the C2CNet layer encapsulates the packet with the C2C header, which includes the C2CNet identifier of the IP next hop node. The C2CNet layer thereby makes the routing decision with the C2CNet identifier and nodes' geographical location.

IPv6 GeoNetworking has four types of geographical routing mechanisms: GeoUnicast, GeoBroadcast, TopoBroadCast, and GeoAnycast. Depending on the mechanisms, several types of the geographical destinations (GeoDestination) can be specified with geographical coordination and descriptions of a shape, such as a

circle area with a particular radius of a geographical position. These geographical routing mechanisms are mapped to the IPv6 unicast, multicast, and anycast so that upper layer entities can transparently use these mechanisms without direct interaction between the C2CNet layer. Users therefore only need to support the legacy IPv6 stack.

In the GeoNet project, IPv6 GeoNetworking has been integrated with TUN virtual interface onto regular Linux, which enables to communicate with IPv6 and C2CNet without modifying the routing mechanisms in the kernel. IPv6 GeoNetworking-enabled routers at first receive regular IPv6 packets on their ingress (egress) interface, and pass the packets to the C2CNet module inside userland via the virtual interface. Then subsequent communication is performed in the C2CNet layer [19,20].

From the point of view of service discovery, GeoBroadcast is the most appropriate mechanism to discover services inside a certain geographical area, because the GeoBroadcast mechanism delivers a packet to all nodes inside a specific GeoDestination via IPv6 multicast.

## 3 Assumptions

### 3.1 Network architecture

In our study, ITS equipment deployed in vehicles and the roadside comply with the *ITS station reference architecture* from ISO/ETSI [1,21,22]. Each ITS station is assumed to be equipped with at least a router, i.e., mobile router (MR for the vehicle ITS station) or access router (AR for the roadside ITS station). Other nodes (e.g., hosts running applications, cameras, gateways to the CAN bus, ...) are possibly connected to the MRs/ARs through an ITS station internal network. MRs/ARs are equipped with at least (i) one wireless egress interface to communicate with other MRs/ARs and (ii) one wired/wireless ingress interface to connect to the ITS station internal network. IPv6 works as a mandatory network layer communication protocol and MRs/ARs provide certain network prefixes to their attached hosts. Each attached host therefore has a global IPv6 address configured from a network prefix assigned by its station-internal MR/AR. ARs provide Internet access to MRs.

Although the *ITS station reference architecture* allows integrating the MRs/ARs with host functionality into an identical node, in this article, we focus on the separated integration: hosts are separately installed as nodes attached to MRs/ARs.

We assume that all MRs/ARs participating within a VANET support IPv6 GeoNetworking. They communicate with each other as one-hop neighbors from the IPv6 point of view, because IPv6 GeoNetworking takes care the multi-hop routing in the C2CNet layer. Each

MR/AR can obtain its current geographical position through embedded GPS.

In this article, we focus on the service discovery inside a VANET without connecting to the Internet, in which vehicle stations and roadside stations have same functions; only difference is that a vehicle station is a pair of a MR and attached hosts, whereas a roadside station is a pair of an AR and stationary attached hosts.

### 3.2 Communication scenarios

In VANET, two types of typical use case scenarios of service discovery are identified: vehicle-to-vehicle (V2V) and vehicle-to-roadside/roadside-to-vehicle (V2R/R2V) discovery. It is assumed that service developers publish their service type and supported attributes to application developers beforehand, so that applications can use available services.

In the V2V scenario, an application in a vehicle communicates with several services in other vehicles. Destination hosts are determined with vehicles' geographical position and service specific attributes, e.g., type of station, trajectory, equipment, etc. For instance, when a vehicle is going to merge into a lane, an application in the vehicle notifies relevant vehicles of its presence. In this case, the application needs to discover notification services in vehicles running inside a relevant area, or that in parking vehicles planning to go into the relevant area. For the latter type of vehicle, the application additionally needs to discover route planning services in the parking vehicles and check if they plan to go inside the area.

In the V2R/R2V scenario, an application in a vehicle (or roadside) host communicates with other roadside (vehicle) hosts according to geographical position and service specific attributes. For example, if a driver wants to park the vehicle near a specific destination (e.g., municipal office, supermarket, etc.), an application in the vehicle host needs to discover services that manage public and/or pay parking lots near the destination. While determining a parking lot, the application also needs to discover some sort of road congestion monitoring service located in the roadside along the path to the parking lot.

A service is identified with a service type and optional service specific attributes. Basically, there are multiple services with an identical service type.

## 4 Geographical location aware service discovery on IPv6 GeoNetworking

In this section, we propose a geographical location aware service discovery mechanism for VANET. Our design principles are as follows:

- Discover services according to geographical position: when a SC tries to discover services, in addition

to the service type and its attributes, the SC specifies a GeoDestination in which requested services should be discovered. The size of the GeoDestination does not depend on the wireless communication range of MRs/ARs but just depends on applications' requirements.

- Specify GeoDestination for each service: each SC should be able to specify application specific GeoDestinations separately. GeoDestinations should dynamically be configured by SCs.
- Avoid transmitting service discovery messages unnecessary distant nodes: the service discovery mechanism should not propagate discovery messages to nodes outside the requested GeoDestination.

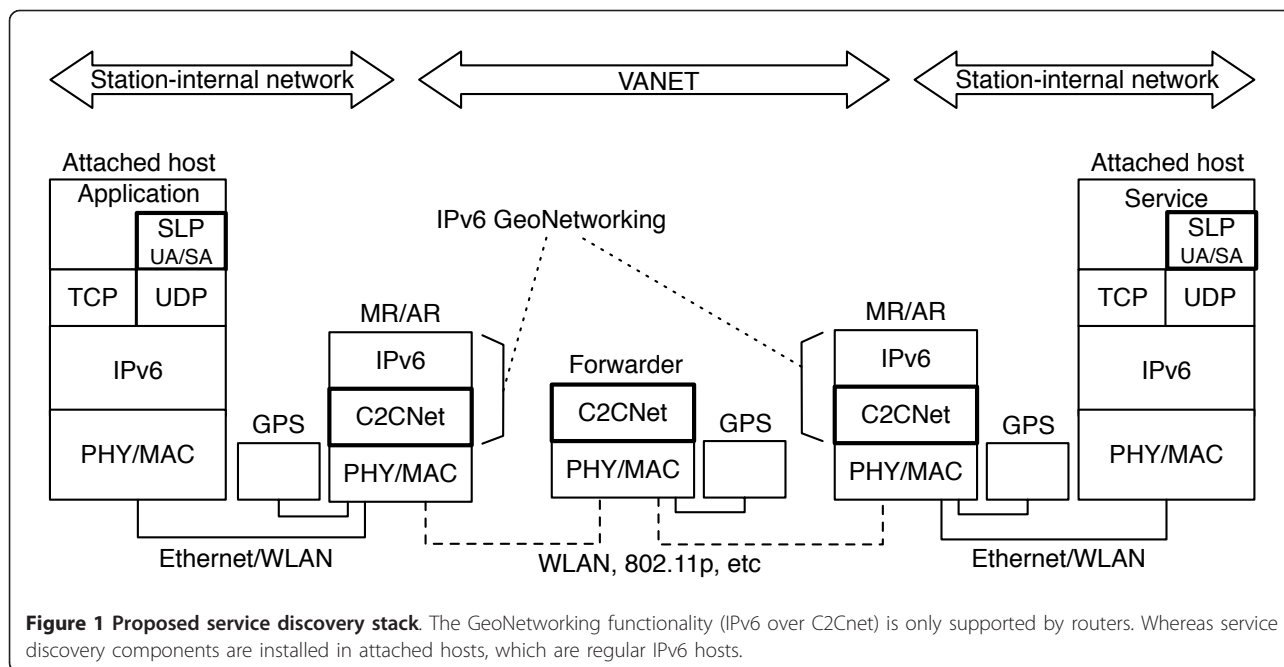
Additionally, the solution should keep the modularity of protocol layers so that each protocol could be integrated separately for future improvement. Consequently, we propose a solution consists of an application layer SDP over IPv6 GeoNetworking. In contrast to the solution using application layer SDPs on traditional network layer routing protocols, in our solution, IPv6 GeoNetworking supports geographical routing through legacy IPv6 so that the SDP discovers services according to geographic position transparently using geographic routing with avoiding propagating packets to entire network.

One of the challenges of harmonizing SDPs with IPv6 GeoNetworking is how SDPs can assign application specific GeoDestinations to IPv6 GeoNetworking without directly merging SDPs and IPv6 GeoNetworking functions. We thus introduce an interface in IPv6 GeoNetworking for SDP, which enables to configure GeoDestinations from external components without losing the modularity of protocols.

We use SLPv2 as a foundation of the SDP in our solution, because the IPv6 modification of SLPv2, which enables to handle multiple IPv6 multicast groups mentioned in Section 2, meets the above-mentioned design principles. In the following sections we use the SLPv2's terms UA and SA as the SC and SP, respectively.

### 4.1 SLP-based service discovery over IPv6 GeoNetworking

SLPv2 and IPv6 GeoNetworking are integrated separately: SLPv2 components are integrated into attached hosts, on the other hand IPv6 GeoNetworking is only integrated into MRs/ARs. Therefore, attached hosts can be conventional PCs that support regular TCP/IP protocols, while MRs/ARs do not necessarily support application layer entities, shown in the proposed protocol stack in Figure 1. Routers that only support GeoNetworking may also be installed as forwarders because of the non-IP multi-hop support by GeoNetworking. Applications and services are integrated into attached hosts.



On behalf of applications, SLPv2 components discover services. We only use UAs and SAs without DAs. When a service is activated, it registers the service type and attributes to the SA running in the local host. SAs do not advertise and forward registered information to any other hosts but passively reply to discovery requests from UAs. As mentioned in Section 2, in SLPv2, SAs that work with a particular service join a specific IPv6 multicast group, which is uniquely determined from the service type using SLPv2's hash function (the address range is FF0x::1:1000/118).

In our system, the service discovery is performed with following three mechanisms:

- **IPv6 multicast *SrvRqst* over GeoBroadcast:** UAs try to discover services using IPv6 multicast *SrvRqst* over GeoBroadcast. The multicast *SrvRqst* is delivered as a GeoBroadcast packet among MRs/ARs, thus it is received only subset of nodes joining the corresponding IPv6 multicast group inside a particular geographical area [23]. From the SLP components' point of view, *SrvRqst* is a legacy IPv6 multicast packet transmitted to a multicast group corresponding to the requested service. However, as described in Section 2, the IPv6 GeoNetworking mechanism encapsulates the *SrvRqst* into a GeoBroadcast packet, which is disseminated to all nodes located inside a specific GeoDestination described with coordinates and the size of the requested area [23]. MRs/ARs inside the

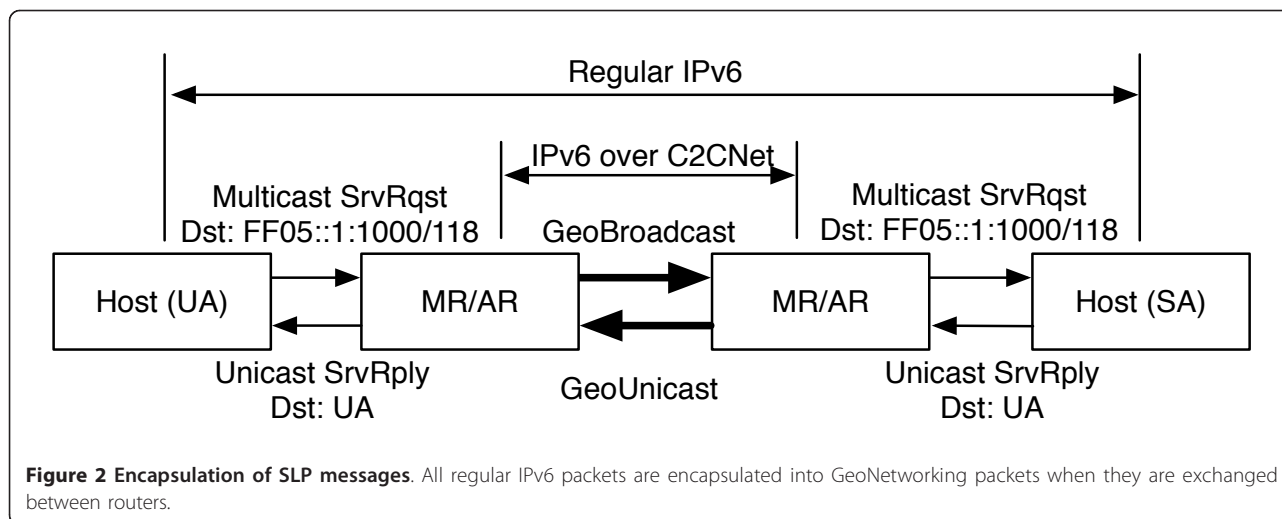
GeoDestination decapsulate the packets to IPv6 multicast *SrvRqst* and deliver to SAs.

The decision of encapsulating/decapsulating packets from/to IPv6 multicast and GeoBroadcast is made with a list of mapping information stored in IPv6 GeoNetworking. The mapping entry is a pair of IPv6 address and GeoDestination, e.g., < FF0E::1234, GeoDestination [latitude, longitude, radius]>

- **IPv6 unicast *SrvRply* over GeoUnicast:** SAs reply to UAs' *SrvRqst* using IPv6 unicast *SrvRply* over GeoUnicast. The unicast *SrvRply* is delivered as a GeoUnicast packet among MRs/ARs using geographical routing.

Like the *SrvRqst* over GeoBroadcast described above, the IPv6 unicast *SrvRply* is encapsulated to a GeoUnicast packet. The GeoUnicast mechanism delivers packets to a node based on its geographical position. Basically, UAs' geographical positions are recorded from the header of received GeoBroadcast (i.e., multicast *SrvRqst*) packets. If the position is not available (e.g., expired), IPv6 GeoNetworking resolves the position using the location service mechanism specified in [23]. Received GeoUnicast packets are decapsulated in a MR/AR to an IPv6 unicast *SrvRply* packet and delivered to the UA. The packet encapsulation mechanisms of the *SrvRqst* and *SrvRply* are depicted in Figure 2.

- **GeoDestination management through TCP unicast:** The GeoNet specification have employed one of the potential solutions to determine GeoDestination



**Figure 2 Encapsulation of SLP messages.** All regular IPv6 packets are encapsulated into GeoNetworking packets when they are exchanged between routers.

by using the destination IPv6 address: for instance, regarding the GeoBroadcast, an IPv6 multicast address is statically mapped to a corresponding geographical area using a configuration file that assigns a GeoDestination with a radius around the centre of the area where the packet shall be propagated (i.e., FF0E::1 corresponds to a circle of 500 m radius of the sender). In this solution, users of IPv6 GeoNetworking (i.e., application developers) need to specify all possible pairs of IPv6 multicast address and GeoDestination beforehand.

However, the solution relying on the static configuration cannot be applied to the proposed SDP because the GeoDestination for each trial of service discovery cannot statically be configured. The necessary GeoDestination is different in each application/discovery scenario.

To overcome this issue, we introduce a mechanism into IPv6 GeoNet-working that enables external entities to dynamically configure the Geo-Destination mapping information for each IPv6 multicast address. UAs are extended to send the mapping information to their station-internal MRs/ARs via TCP unicast: when they issue a multicast *SrvRqst*, they additionally send a unicast packet that indicates the GeoDestination corresponds to the requested IPv6 multicast address. This TCP unicast packet is only delivered from attached hosts to station-internal MRs/ARs, thus it does not require any additional overhead to VANET.

Regarding the multi-hop IPv6 multicast routing from an attached host to the other hosts in *SrvRqst*, MRs/ARs simply forward the multicast packets between their ingress and egress interface. They do not use any dedicated multicast routing mechanism because MRs/

ARs can communicate with each other as one-hop IP neighbor in the considered VANET thanks to IPv6 GeoNetworking, shown in Section 2. The proposed mechanism therefore does not require additional overhead to build and maintain the multicast routing topology. Note that we use the site-local scope IPv6 multicast (i.e., FF05::1:1000/118) instead of the link-local scope since UAs and SAs are out of link-local scope; they are located in attached hosts behind different MRs/ARs.

#### 4.2 Operation sequence

The proposed mechanism is identified with three phases: service activation, service discovery, and service operation. Overall operations are described as follows:

##### Service activation

1. When a service is installed and activated, it registers its characteristics (i.e., service type and attributes) to a locally-running SA.
2. The SA joins an IPv6 multicast group determined from the service type using the SLP's hash function. A MLD report is sent to the station-internal MR/AR.

##### Service discovery

1. An application requests a UA in the local host to discover a service. The application sends the description of the discovery request (i.e., requested service's characteristics and relevant geographical area) to the UA.
2. From the application's request, the UA calculates the corresponding IPv6 multicast address like service activation phase, and then sends a pair of  $\langle$  IPv6 multicast address, GeoDestination  $\rangle$ , to its station-internal MR/AR via TCP unicast (**GeoDestination management**).



3. IPv6 GeoNetworking in the MR/AR creates or updates the list of mapping entry of GeoDestination with the received pair of IPv6 multicast address and GeoDestination.
4. The UA then performs the **IPv6 multicast SrvRqst over GeoBroadcast**: it issues the multicast SrvRqst designated to the corresponding IPv6 multicast address to the station-internal MR/AR.
5. In the MR/AR, the multicast packet is forwarded from the ingress interface to the IPv6 GeoNetworking virtual interface. Then IPv6 GeoNetworking determines the GeoDestination by looking up the previously created mapping entry in Step 3.
6. The IPv6 multicast packets are encapsulated into the GeoBroadcast packets and sent out on the egress interface. Note that the header of the GeoBroadcast packet contains the sender MR/AR's geographical position, C2CNet identifier of GeoNetworking, and the requested GeoDestination. The C2CNet identifier is obtained from sender's IPv6 unicast address.
7. MRs/ARs located inside the GeoDestination receive the GeoBroadcast packets on their egress interface. They check if there are attached hosts belonging to the corresponding multicast group (i.e., an SA that operates the requested service) on their ingress interface. If there are corresponding SAs, MRs/ARs decapsulate the GeoBroadcast packets into

the regular IPv6 multicast packets and send it to SAs via their ingress interface. At the same time, MRs/ARs record the sender's geographical position and node ID of GeoNetworking included in the header of GeoBroadcast packets.

8. If a SA knows a service that satisfies the requested characteristics, it performs **IPv6 unicast SrvRply over GeoUnicast**: the SA issues the uni-cast SrvRply to the UA's unicast address via its station-internal MR/AR.

9. The MR/AR at first resolves the IPv6 address of the destination MR/AR for the UA using legacy IP unicast routing. Then IPv6 GeoNetworking in the MR/AR determines the geographical position of the destination MR/AR from the recorded information.

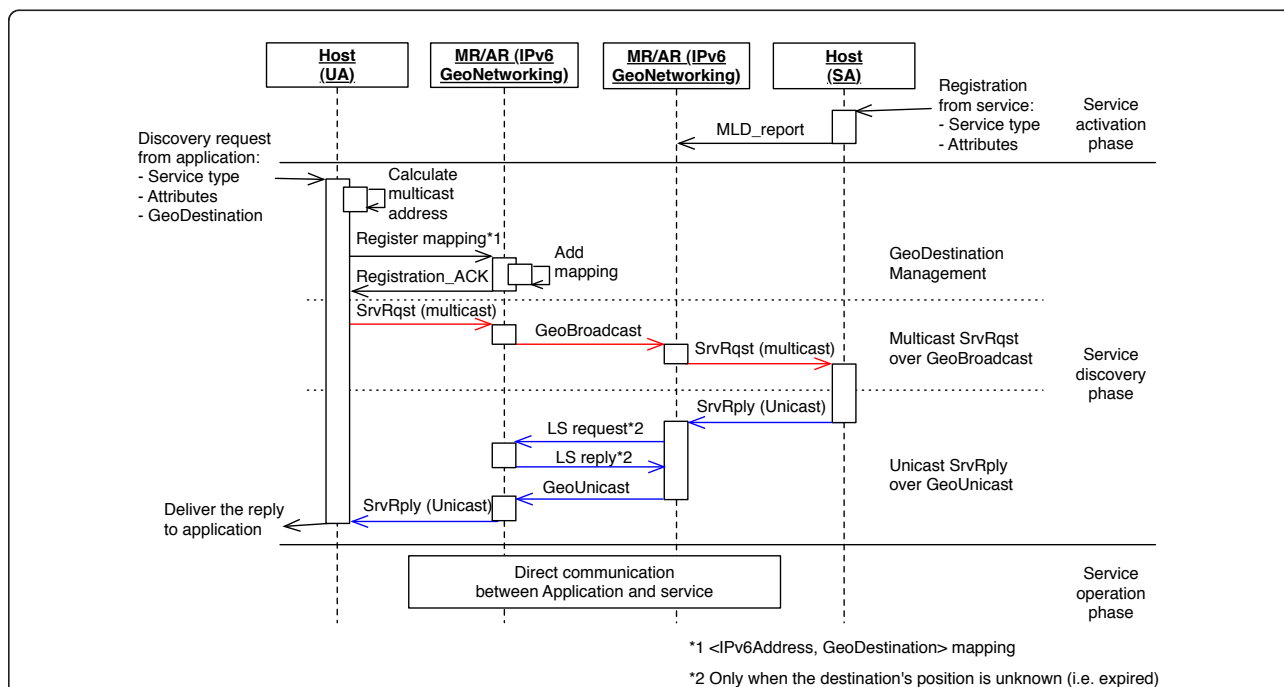
10. The IPv6 unicast packets are encapsulated into the GeoUnicast packets and sent out on the egress interface.

11. The destination MR/AR corresponding to the GeoUnicast receives the GeoUnicast packets on its egress interface, and decapsulates it into IPv6 unicast packets. Finally the UA receives the unicast SrvRply.

**Service operation**

Finally, the application and the service start to communicate with each other.

Figure 3 shows the overall messaging sequence.



**Figure 3 Messaging sequence.** Overall messaging sequence is composed of three phases and the service discovery is mainly performed in the second phase. Note that LS is the abbreviation of Location Service, control messages of IPv6 GeoNetworking, which determines the geographical location of a node.

Suppose a set of all nodes in the considered network  $N$ , a group of nodes that join the corresponding IPv6 multicast group  $G_{mc}$ , a group of nodes being inside a GeoDestination  $G_{gd}$ . Only if a node  $N_i \in N$  satisfies  $(N_i \in G_{mc}) \cap (N_i \in G_{gd})$  can receive *SrvRqst* packets, shown in the selective propagation of multicast *SrvRqst* packets in Figure 4. Thanks to this mechanism with the SLP's per-service IPv6 multicast address assignment, the proposed mechanism can avoid propagating service discovery messages unnecessarily large size of geographical area. In addition, UAs and SAs do not necessarily take care of their current location to discover services since geographical position is managed by IPv6 GeoNetworking in MRs/ARs.

### 5 Experiments

In order to observe the cost and performance of the proposed mechanism, we implemented a prototype system. At first, we performed scalability evaluations using the network simulator ns-3 [24], we then integrated the system into our real field testbed and conducted field tests.

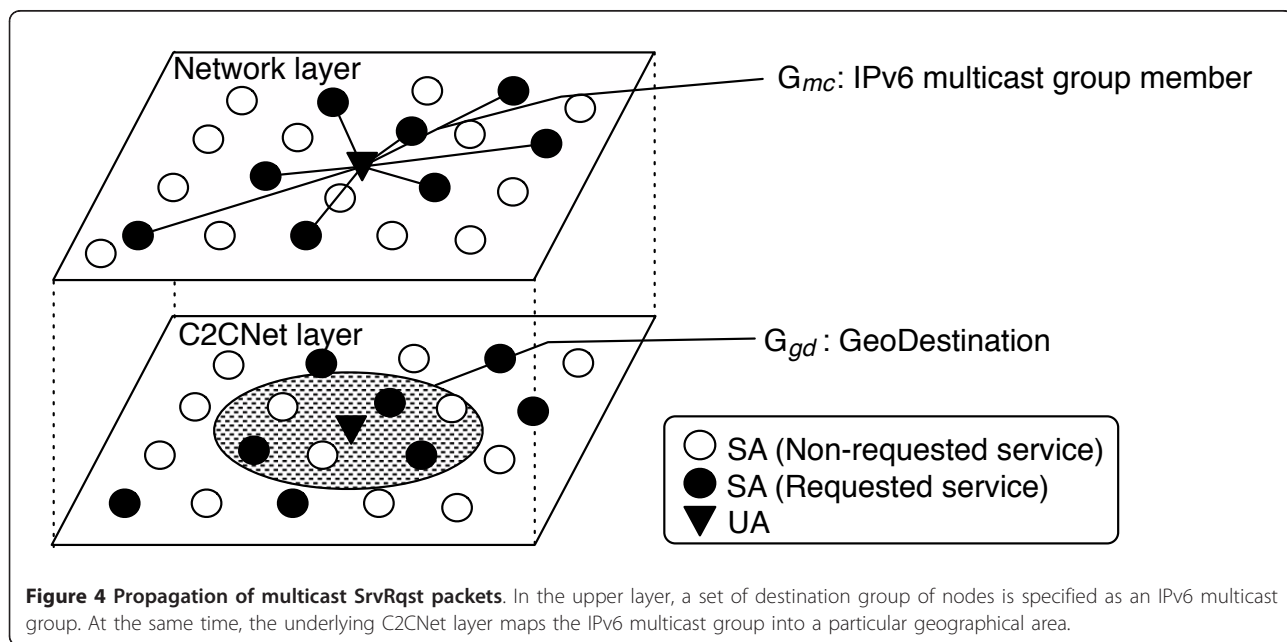
In each evaluation, an UA in a vehicle station (a pair of router and host in a vehicle) periodically tried to discover services within a VANET composed of several roadside stations, in which attached hosts worked as SAs with 100 services. The UA sequentially issued *SrvRqst* until it transmitted 100 requests; it issued a *SrvRqst* 1s after receiving the first *SrvRply* for the previously sent *SrvRqst*. In each request, the UA randomly determined a service and GeoDestination.

The vehicle moved within the VANET. The MR in the vehicle station was always inside the radio

communication range of at least one other station. Each router was equipped with a wireless egress interface with the standard IEEE 802.11b MAC layer. Its radio communication range was configured to 130 m. IPv6 GeoNetworking was configured to the default settings specified in [23] (e.g., 500 ms for beacon sending interval). SLP was also configured to the default settings specified in [4,8].

We measured the following metrics in each evaluation:

- Discovery success rate: the rate of successfully replied *SrvRqst*.
- End-to-end latency: the delay in successfully replied service discovery phases between the UA and the SAs.
- Amount of control messages transmitted per router: the number of bytes transmitted by the egress interface in each router. It represents the total amount of IPv6 GeoNetworking messages including Beacon, GeoUnicast, GeoBroadcast, and Location Service. Note that we did not take into account the station-internal communication between attached hosts and routers, because it is performed via Ethernet that ensures enough bandwidth and stability.
- Amount of control messages transmitted per router per received *SrvRply*: the number of bytes transmitted by the egress interface in each router for each received *SrvRply*. It shows the ratio of control messages transmitted to delivered one *SrvRply* to the UA, which represents the efficiency of control messages.



### 5.1 Simulation setup

We integrated the proposed system into ns-3 version 3-12.1. We implemented one new application and one new ns-3 model: SLP and IPv6 GeoNetworking. Our SLP application works as a part of ns-3 application model, while IPv6 GeoNetworking is an independent ns-3 model that works with the Internet model and Net-device model. The SLP application only supports limited functions required to evaluate our system, therefore most features were not implemented, e.g., authentication, multiple scope handling, DA, non-networking related functions, etc. On the other hand, the IPv6 GeoNetworking model fully complies to [23] except the geographic position management. The ns-3 mobility model manages node's position on behalf of GPS, thus we implemented a function that obtains current position from the mobility model and passes to the IPv6 GeoNetworking model. Note that in the simulation, we assumed that the position information is always accurate. In addition, we modified the UDP models to support IPv6 since the model in ns-3.12-1 only supports IPv4.

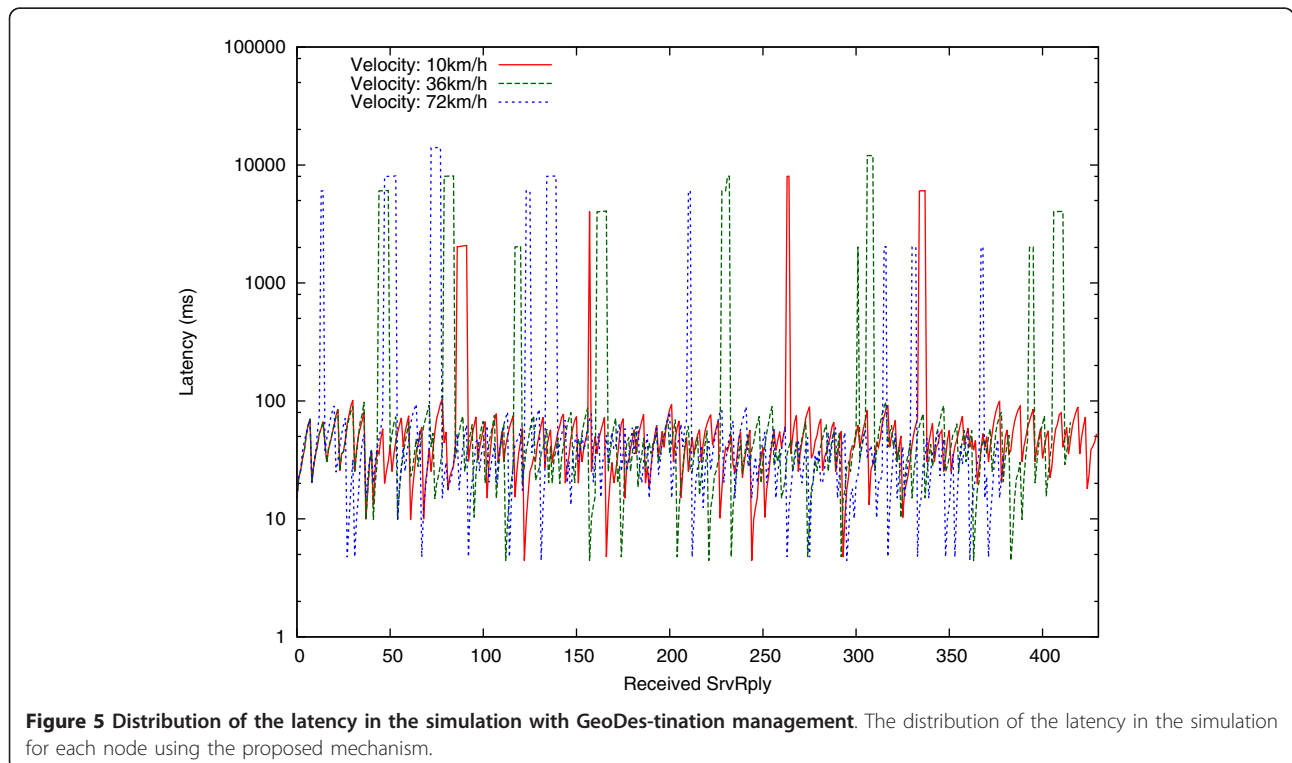
The basic settings of the simulation complied with the one described above. The simulated VANET was composed of 100 stations in a 1000 m × 1000 m rectangular field. The stations were located in a 2D-grid, in which the distance between each node was 100 m. The vehicle followed the random waypoint mobility model with

following three scenarios of velocity: low mobility (10 km/h), medium mobility (36 km/h), and high mobility (72 km/h). While the shape of GeoDestination was always circle, its center position and radius were randomly determined (the range of radius was: 75-150 m).

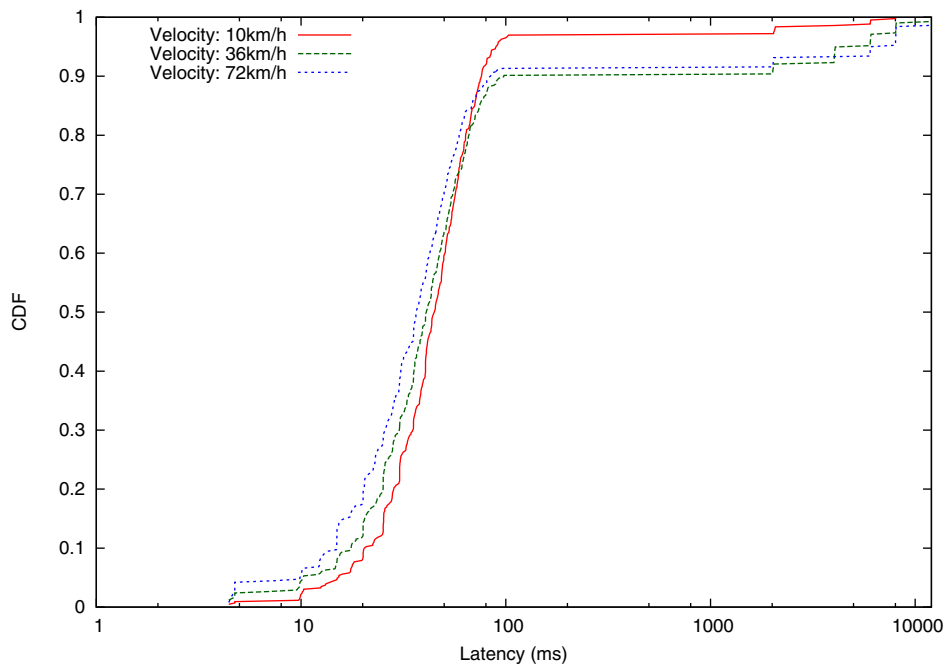
We also performed the same tests without using the GeoDestination management mechanism. In these tests the GeoDestination was fixed to cover all nodes in the VANET. This evaluation similarly corresponds to the solution in which the application layer manages the geographical position with relying on geographic-agnostic, all-node flooding.

### 5.2 Simulation results

We obtained the following simulation results from averaging 10 different-seed runs for each setting. The discovery success was 100%, 96%, and 94% in the scenario with velocity of 10 km/h, 36 km/h, and 72 km/h, respectively. The end-to-end latency was proportional to the velocity: 4/8022/175, 4/12034/592, 4/14068/678 ms (minimum/maximum/average) in each scenario. Figure 5 shows the distribution of end-to-end latency for each received *SrvRply*, and Figure 6 shows its CDF. The amount of control messages transmitted per router was 204, 174, and 167 Bytes/s for each scenario, respectively, while the amount of control messages transmitted per router per received *SrvRply* was 0.47, 0.41, and 0.43 Bytes/s.



**Figure 5** Distribution of the latency in the simulation with GeoDestination management. The distribution of the latency in the simulation for each node using the proposed mechanism.



**Figure 6** CDF of the latency in the simulation with GeoDestination management. The CDF of the latency in the simulation for each node using the proposed mechanism.

In the simulation without the GeoDestination management mechanism, the end-to-end latency was 4/557/273, 4/593/286, and 4/10408/2807 ms (minimum/maximum/average) in each scenario, respectively. Figure 7 shows the CDF of the end-to-end latency in each scenario. The amount of control messages transmitted per router was 1727, 1719, and 1729 Bytes/s for each scenario, respectively, while the amount of control messages transmitted per router per received *SrvRply* was 1.7, 2.1, 1.2 Bytes/s. Table 1 shows the description of the simulation results.

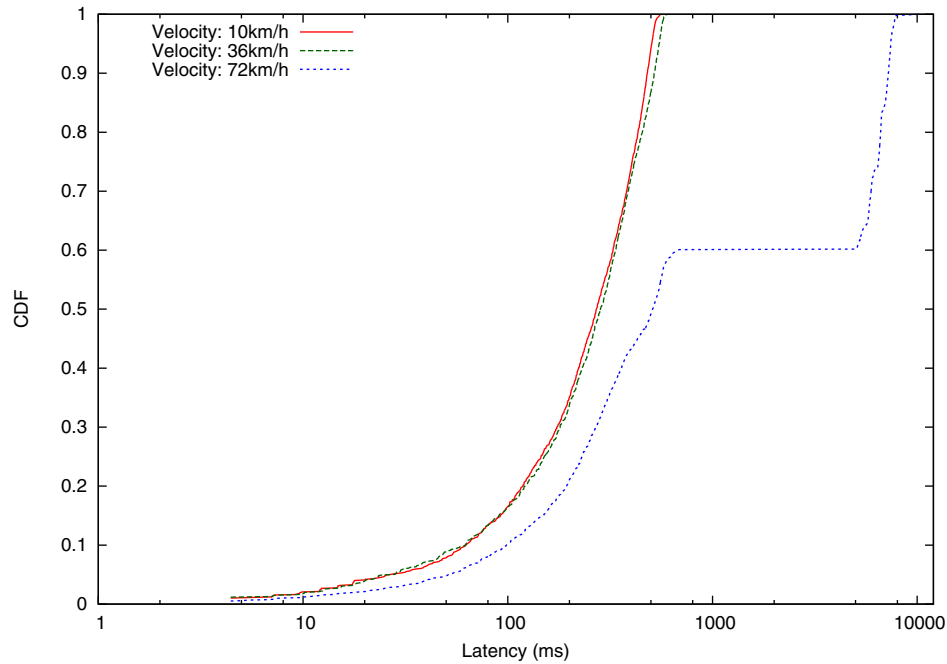
### 5.3 Field evaluation setup

We actually implemented the proposed system by extending OpenSLP 2.0 Beta 1 [25] and CarGeo6 [26] on Linux. OpenSLP is an open-source implementation of SLP including the modification for IPv6, and CarGeo6 is an open-source implementation of IPv6 GeoNetworking in compliance with the reference specification of the GeoNet project. In order to forward IPv6 multicast packets between ingress and egress interfaces in each router, we also implemented a multicast forwarding daemon.

We integrated the system into our field testbed at NAIST campus in Japan, which is composed of three sets of routers and attached hosts. Routers were equipped with one Ethernet port as an ingress interface, and one wireless 802.11 b/g card on the 2.4GHz frequency band as an egress interface. The data rate of the

egress interface was configured to 6Mbps. We used Ubuntu 10.10 (kernel 2.6.35.11) for all nodes. Each router was able to get current geographical position through a GPS receiver via its USB-serial connection. In order to get coordinates, gpsd-2.96 [27] was installed as a local TCP server. While routers ran CarGeo6 and the IPv6 multicast forwarding daemon to operate IPv6 GeoNet-working and IPv6 multicast forwarding, attached hosts were conventional PCs with OpenSLP and traditional IPv6 multicasting functions. Applications and services only manipulated a set of OpenSLP functions without generating application/service specific traffic.

The basic settings of the field evaluation complied with the one described above. As shown in Figure 8, three stations were located along the roadside. Each station had a router and an attached host: Station1 (MR, Host1), Station2 (AR1, Host2), and Station3 (AR2, Host3). The Station1 worked as a vehicle station, thus it moves around the other two stations. An UA worked on the Host1, in which the UA periodically tried to discover services with randomly selected GeoBroadcast radius. On the contrary, the other stations were stationary. The Host2 and Host3 operated SAs (SA1 on the Host2, SA2 on the Host3). The Station1 moved inside the direct communication range of Station2, while it did not enter the Station3's direct communication range. It means that the UA in the Station1 could communicate



**Figure 7 CDF of the latency in the simulation without GeoDestination management.** The CDF of the latency in the simulation for each node without using the GeoDestination management mechanism.

with the SA1 via single-hop whereas the SA2 via multi-hop.

In the field evaluation, the center position of GeoDestination for GeoBroadcast was fixed to the position of the MR, since the CarGeo6 implementation at the moment only supported such *Vehicle-Centred GeoBroadcast*. Therefore the UA only specified the randomly selected radius of GeoDestination (the range of radius was: 50-200 m). The system configurations are shown in Table 2.

#### 5.4 Field evaluation result

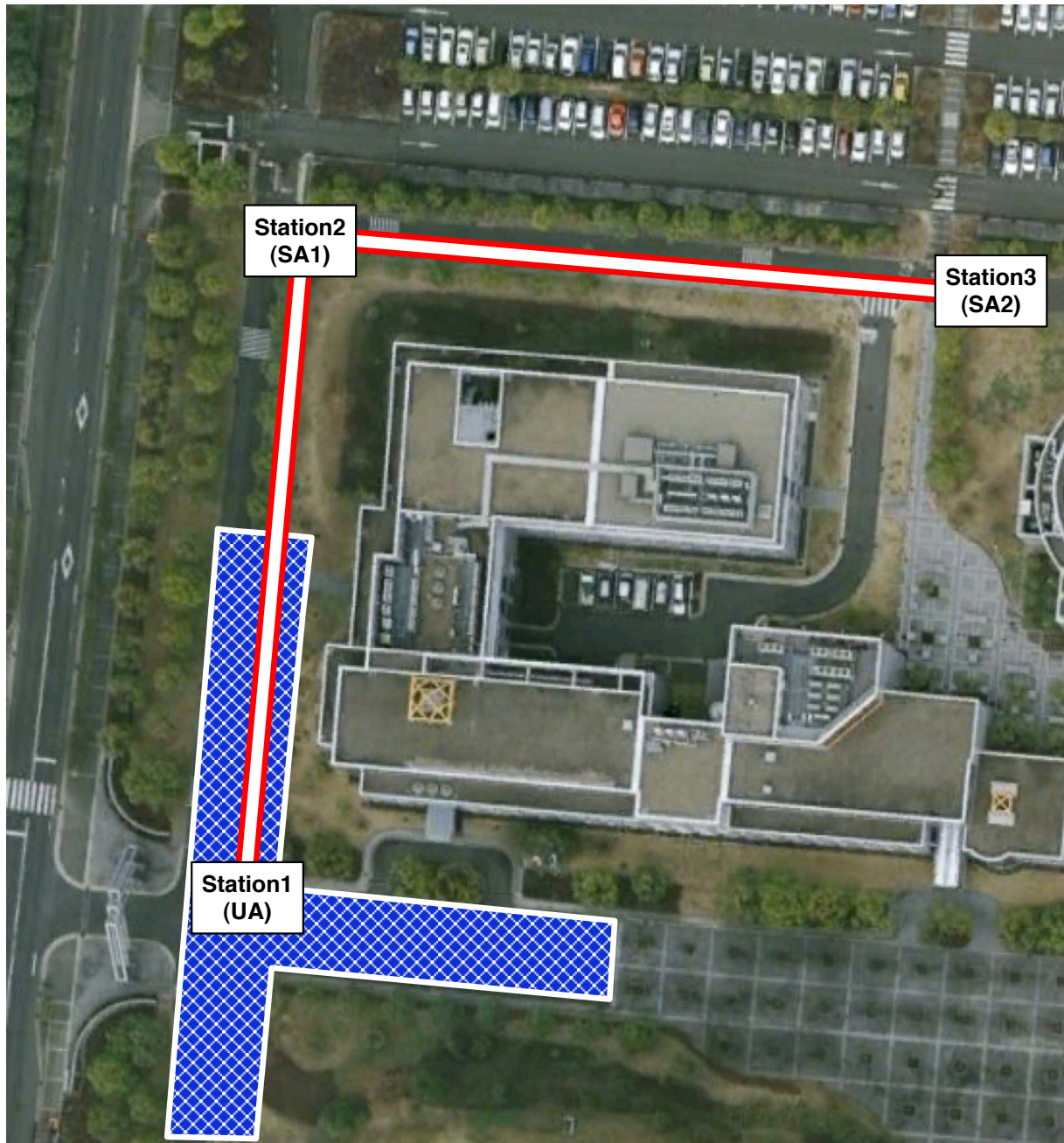
The discovery success rate was 86%: in the single-hop case, it was 96% while 77% in the multi-hop case, and the end-to-end latency was 3.5/170/28.2 ms (minimum/maximum/average). Figure 9 shows the CDF of end-to-end latency of received *SrvRply*. In the field evaluation,

we additionally evaluated the end-to-end latency per hop: the single-hop latency between the UA to the SA1 was 3.5/23.3/7.8 ms, whereas the two-hop latency between the UA and the SA2 was 27.9/170/48.6 ms, shown in Table 3.

The amount of control messages transmitted per router was 243 Bytes/s. The MR in the Station1, in which the Host1 operates the UA, transmitted 300 byte/s while that in the Station2 was 223 Bytes/s, and the Station3 was 204 Bytes/s. Figure 10 shows the proportion of the types of control messages in each router. In the evaluation, the size of each packet was as follows: (i) GeoBroadcast: from 189 to 264 Bytes, (ii) GeoUnicast: 199 Bytes, (iii) Location Sservice: from 86 to 94 Bytes, and (iv) Beacon: 78 Bytes. The size of the GeoBroadcast packet, which contains a *SrvRqst* message, was variable because of the SLP's retransmission algorithm.

**Table 1 Summary of simulation results**

Scenario	UA's velocity (km/h)	Success rate (%)	Latency (ms)			Transmitted packet (Bytes/s/node)
			Min	Max	Avg.	
Yes	10	100	4	8,022	175	204
Yes	36	96	4	12,034	592	174
Yes	70	94	4	14,068	678	167
No	10	100	4	557	273	1,727
No	36	100	4	593	286	1,719
No	70	100	4	10,408	2,807	1,729



**Figure 8 Network topology in the field evaluation.** In the field evaluation, Station0 moves around the meshed area while Station1 and Station2 do not move. Station0 always communicate with Station2 via Station1. The UA works in the attached host in Station0. The SAs work in the attached hosts in the other stations.

### 5.5 Analysis

According to Figure 5, the proposed system stably works in any evaluations. It shows that the trend of the distribution of the end-to-end latency did not change in a particular period of time.

Regarding the end-to-end latency, the CDFs in Figures 6 and 9 show that the 90% of the successful *SrvRqst* were finished within 100 ms in any mobility scenarios. Even in the actual environment, the UA mostly discovered services within 20 ms via single-hop, and 60 ms via

**Table 2 System settings in the field evaluations**

Entity	Parameter	Setting
Hosts2, Host3 (SA)	Number of services	100
Host1(UA)	Service discovery interval (Hz)	1
	Radius of GeoBroadcast (m)	Random (50-200)
All Routers	GPS position update frequency (Hz)	1
Station1	Driving speed (km/h)	0-20
	Distance from SA1(Single hop) (m)	10-100
	Distance from SA2(Multi hop) (m)	120-200

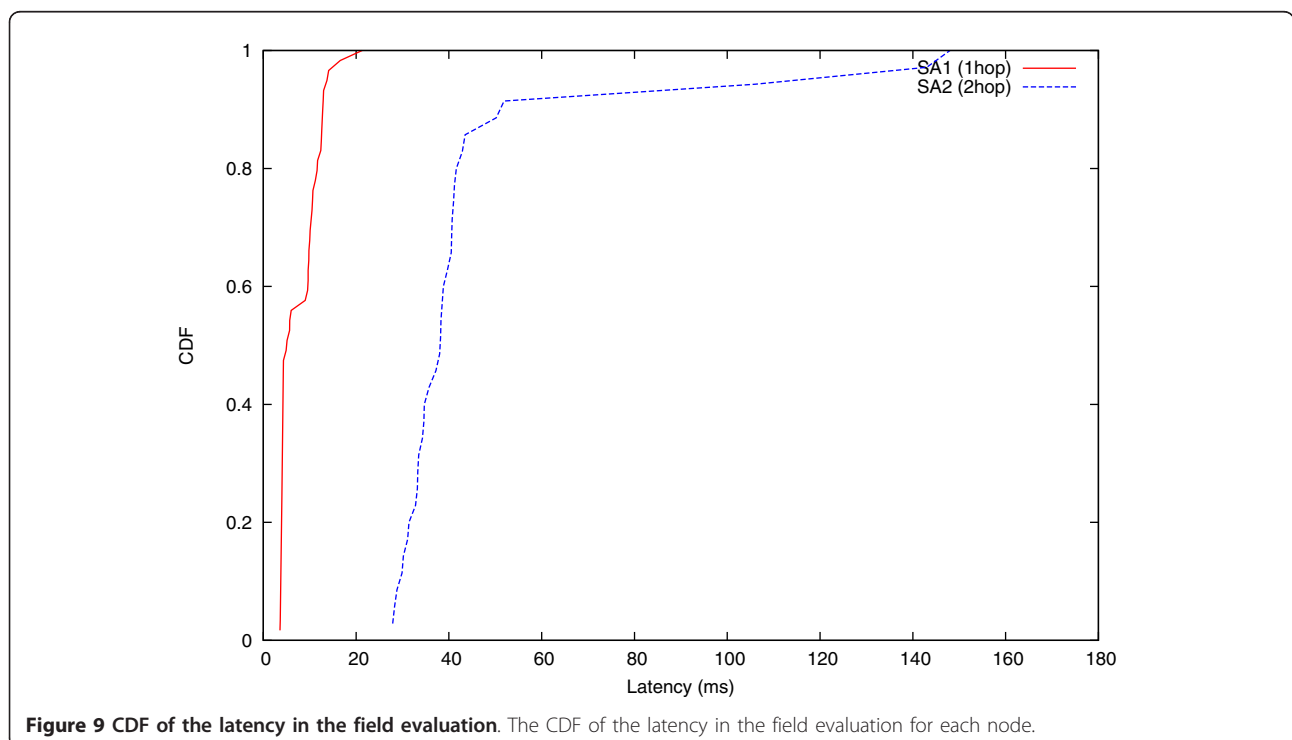
multi-hop, as plotted in Figure 9. The reason for the high latency that appeared in the simulations (i.e., 2000 ms or more) was caused by the SLP's retransmission algorithm, which exponentially increases the wait interval for each discovery trial from 2 to 15 s [4]. Such retransmissions may have occurred in discovering services in distant GeoDes-tinations that requires many hops. Consequently, the evaluation results shows that the proposed mechanism mostly discovers services within the required acceptable latencies specified in ITS related researches and standards (mainly about 100-500 ms) [1,2].

On the other hand, in the application layer solution with all node flooding, more than 80% of discovery trials needed more than 100 ms to discover services, as shown in Figure 7. In these scenarios, flooded packets seriously

congested the VANET. For instance, in the high mobility scenario in which the UA's velocity was 72 km/h, more than 50% of discovery trials took the latency over 900 ms.

In terms of the hop count in the actual environment, the success rate dropped to 77% in the multi-hop case. We consider it was caused by the Location Service mechanisms of CarGeo6, which needs additional communications between routers. The Location Service mechanisms is used to determine the geographical position of the UA to send back *SrvRply*, therefore we consider following two possibilities: (i) the AR2 in the Station3 could not successfully record the UA's position from received *SrvRqst*, or (ii) the recorded position is expired due to the UA's mobility. In the future we need to investigate this issue.

The overhead of control messages in each successful discovery in the simulation is about 0.4 Bytes/s for each mobility scenario. It shows that our system requires consistent costs to deliver a *SrvRply* to a UA even in the high mobility scenario. It efficiently reduces the overhead of control messages compare to the all node flooding based solutions, as depicted in Table 1 thanks to the Geo-Broadcast management mechanism, which enables to deliver *SrvRqst* to only limited number of nodes inside the dynamically assigned GeoDestinations. The field evaluation result also shows that the distant nodes (nodes in the Station 3 located in two-hop distance from the Station 1) successfully avoided to process



**Table 3 End-to-end latency and discovery success rate the field evaluations per hop**

	Success rate (%)	Latency (ms)			Transmitted packet (Bytes/s/node)
		Min	Max	Average	
SA1 (1 hop)	96	3.5	23.3	7.8	223
SA2 (2 hop)	77	27.9	170	48.6	204

unnecessary packets sent to a GeoDestination that does not cover the position of the nodes.

As shown in Figure 10, the large portion of the overhead was caused by Beacon messages, which is periodically sent to single-hop neighbors at 500 ms intervals by each node regardless of our service discovery mechanism. The overhead caused by the service discovery (except the Beacons) in the actual environment was 178 Bytes/s in each node; 124/31/22 Bytes/s in MR/AR1/AR2 (connected to UA/SA1/SA2 respectively). According to our preliminary test that evaluated the available throughput in the field test bed, maximum available bandwidth in the filed test setting was about 5Mbps (625 KBytes/s). Consequently, it shows that the proposed system discovers services with fairly small overhead in terms of bandwidth consumption.

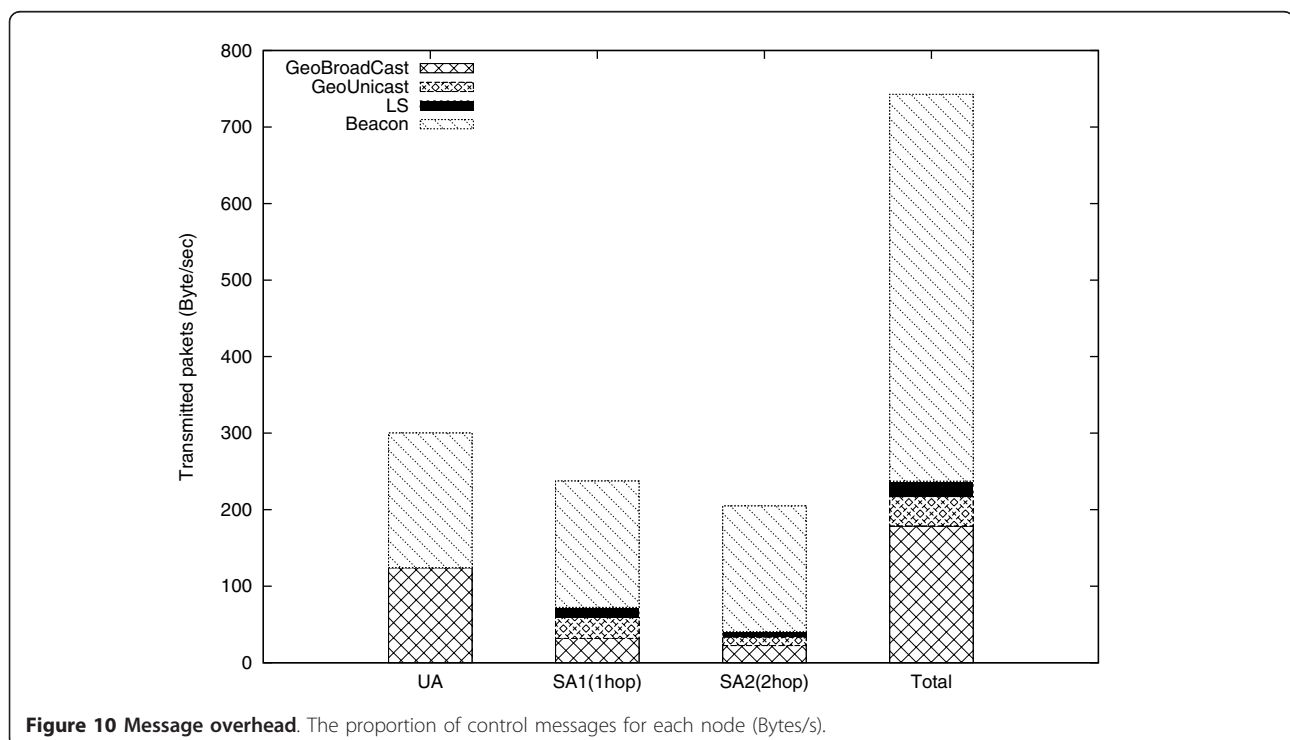
### 6 Conclusion and future work

In this article, we presented a geographical location-aware service discovery mechanism for ITS applications in VANETs. The proposed mechanism is a harmonization of SLPv2 and IPv6 GeoNetworking developed in

the GeoNet project. Furthermore, we integrated a GeoDestination management mechanism that enables IPv6 GeoNetworking users (i.e., applications) to dynamically specify GeoDestination: the representation of geographical area to which the packet should be delivered. Using the ns-3 network simulator, we showed that the IPv6 multicast-based service discovery using GeoBroadcast with the GeoDestination management rapidly discovers services without propagating discovery packets to entire network unnecessarily.

In addition to the simulations, we actually implemented the proposed mechanism into Linux using the OpenSLP and CarGeo6 implementations. Its evaluation was performed in the field testbed in our campus in Japan. The field evaluation results showed the system discovers services with small overhead.

As a next step we are conducting further field evaluations with more realistic use cases. Although the proposed mechanism discovered services via not only single-hop but also multi-hop with fairly low latency and control message overhead, further improvements are necessary since the discovery success rate, specifically in the



**Figure 10 Message overhead.** The proportion of control messages for each node (Bytes/s).



multi-hop scenario in the field measurement, drops sharply. It is also necessary to investigate how to discover services from/to the Internet in combination with IPv6 mobility support protocols.

#### Abbreviations

AR: access router; DA: directory agent; GeoDestination: a set of information that describes a particular geographical area, e.g., < Latitude, Longitude, Radius >; MR: mobile router; SA: service agent; SC: service consumer; SD: service directory; SLP: service location protocol; SP: service provider; SvrRply: service reply; SvrRqst: service request; UA: user agent; VANET: vehicular ad-hoc network

#### Acknowledgements

The authors would like to thank the Tunisian developers from ESPRIT and all the people involved in the CarGeo6 implementation for their effort in the development and support of the GeoNetworking layer and for their collaboration with the IMARA project-team (INRIA).

#### Author details

<sup>1</sup>Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan <sup>2</sup>LaRA (The Automated Road), a Joint Research Unit between INRIA (project-team IMARA) and Mines ParisTech (CAOR lab), France

#### Competing interests

The authors declare that they have no competing interests.

Received: 20 July 2011 Accepted: 1 February 2012

Published: 1 February 2012

#### References

1. P Papadimitratos, A La Fortelle, K Evenssen, R Brignolo, S Cosenza, Vehicular communication systems: enabling technologies, applications, and future outlook on intelligent transportation. *IEEE Commun Mag.* **47**(11), 84–95 (2009)
2. CAR 2 CAR Communication Consortium, CAR 2 CAR Communication Consortium Manifesto, Overview of the C2C-CC System. Version 1.1 (2007)
3. ETSI: Intelligent Transport Systems (ITS), Vehicular Communications (VC); Basic Set of Applications; Definitions, *ETSI TR 102 638*, V1.1.1 (2009)
4. E Guttman, C Perkins, J Veizades, M Day, Service Location Protocol, Version 2. RFC 2608 (Proposed Standard). Internet Engineering Task Force, <http://www.ietf.org/rfc/rfc2608.txt> (1999). Updated by RFC 3224
5. E Guttman, C Perkins, J Kempf, Service Templates and Service: Schemes. RFC 2609 (Proposed Standard). Internet Engineering Task Force, <http://www.ietf.org/rfc/rfc2609.txt> (1999)
6. Multicast DNS, <http://www.multicastdns.org/>
7. DNS Service Discovery, <http://www.dns-sd.org/>
8. E Guttman, Service Location Protocol Modifications for IPv6. RFC 3111 (Proposed Standard). Internet Engineering Task Force, <http://www.ietf.org/rfc/rfc3111.txt> (2001)
9. The FP7 European project GeoNet, <http://geonet-project.eu/>
10. Li L, L Lamont, A lightweight service discovery mechanism for mobile ad hoc pervasive environment using cross-layer design, in *Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW '05)*, Ottawa, Canada, 55–59 (2005)
11. UPnP Forum, <http://www.upnp.org/>
12. I Stoica, D Adkins, S Zhuang, S Shenker, S Surana, Internet indirection infrastructure. *IEEE/ACM Trans Netw.* **12**(2), 205–218 (2004). doi:10.1109/TNET.2004.826279
13. C Perkins, E Belding-Royer, S Das, Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental). Internet Engineering Task Force, <http://www.ietf.org/rfc/rfc3561.txt> (2003)
14. T Clausen, P Jacquet, Optimized Link State Routing Protocol (OLSR). RFC 3626 (Experimental). Internet Engineering Task Force, <http://www.ietf.org/rfc/rfc3626.txt> (2003)
15. GeoNet Project, D1.2 Final GeoNet Architecture Design (2010)
16. AN Mian, R Baldoni, R Beraldi, A survey of service discovery protocols in multihop mobile ad hoc networks. *IEEE Pervasive Comput.* **8**, 66–74 (2009)
17. IPv6 Multicast Address Space Registry, <http://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xml>
18. The CAR 2 CAR Communication Consortium, <http://www.car-to-car.org/>
19. M Tsukada, IB Jemaa, H Menouar, W Zhang, M Goleva, T Ernst, Experimental evaluation for IPv6 over VANET geographic routing, in *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference (IWCMC'10)*, Caen, France, 736–741 (2010)
20. S Noguchi, M Tsukada, IB Jemaa, T Ernst, Real-vehicle integration of driver support application with IPv6 GeoNetworking, in *Proceedings of IEEE 73rd Vehicular Technology Conference (VTC 2011-Spring)*, Budapest, Hungary, 1–5 (2011)
21. ETSI: Intelligent Transport Systems (ITS), Communications Architecture. ETSI EN 302 665 V1.1.1 (2010)
22. ISO/IEC: ISO 21217:2010, Intelligent Transport Systems - Communications access for land mobiles (CALM) - Architecture. International Organization for Standardization (2010)
23. GeoNet Project, D2.2 Final GeoNet Specification (2010)
24. ns-3 discrete-event network simulator, <http://www.nsnam.org/>
25. OpenSLP, <http://www.openslp.org/>
26. CarGeo6 project, <http://www.cargeo6.org/>
27. gpsd - a GPS service daemon, <http://www.catb.org/gpsd/>

doi:10.1186/1687-1499-2012-29

**Cite this article as:** Noguchi et al.: Design and field evaluation of geographical location-aware service discovery on IPv6 GeoNetworking for VANET. *EURASIP Journal on Wireless Communications and Networking* 2012 **2012**:29.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)