NAIST-IS-DD0961205

Doctoral Dissertation

A study of wide-area geographical resource discovery for vehicular communication systems

Satoru Noguchi

February 7, 2013

Department of Information Systems
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Satoru Noguchi

Thesis Committee:

Professor Kazutoshi Fujikawa	(Supervisor)
Professor Suguru Yamaguchi	(Co-supervisor)
Associate Professor Atsuo Inomata	(Co-supervisor)
Professor Hideki Sunahara	(Keio University)
Dr. Thierry Ernst	(Mines ParisTech / INRIA)

A study of wide-area geographical resource discovery for vehicular communication systems*

Satoru Noguchi

Abstract

Discovering resources in numerous mobile networks is one of the essential tasks for mobile applications. In addition to the traditional Internet users, the rapid development of wireless communication technologies has enabled several new types of devices to communicate with each other. One of the most important emerging services using such new devices is Cooperative Intelligent Transport System (C-ITS) applications. C-ITS applications are mainly integrated into vehicles composing Vehicular Communication Systems (VCS) and provide a wide variety of advanced location-based services for road users.

Because VCS are composed of numerous access technologies and communication protocols, an important question in such environments is *how to discover necessary resources*. While there have been a wide variety of resource discovery technologies for a specific type of network including LAN and MANET, C-ITS applications raise further challenges: rapidly discovering resources according to their geographical position and supporting a huge number of mobile nodes. Some of the existing solutions support mobile resource discovery only inside a small network, and others are capable of the wide-area resource discovery. However, they are dedicated to a specific type of network and do not support geographical location- based discovery. The capability of the geographical discovery, appropriate discovery scope selection, low-latency discovery, and mobility support for a huge number of nodes, are essential for future resource discovery systems.

This dissertation presents a geographical location-based resource discovery scheme for VCS. To study the issues and analyze solutions to the goal, this dissertation takes

---

*Doctoral Dissertation, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0961205, February 7, 2013.

the two approaches: the one is a small scope resource discovery mechanism for a single VANET relying on IPv6 multicast in combination with geographical routing technologies, and the other is a wide-area geographical mobile resource discovery mechanism for numerous mobile networks connected to the Internet. The latter approach is built on a hierarchical publish/subscribe architecture and geo-aware VANETs so that users can locate resources according to their geographical position without scalability issue. The former mechanism is evaluated by ns-3 simulations and an actual implementation, while the latter is evaluated via numerical analysis and simulation. Evaluation results show that the proposed mechanism locates mobile resources among a large number of candidates according to their geographical position without overloading both mobile and core network.

Keywords:

Resource discovery protocol, vehicular communication systems, mobile networking, Publish/Subscribe, GeoNetworking

Acknowledgements

To present this dissertation, it takes more than five years of difficult and pleasant time. It has never be done without the help of a number of professors, colleagues, friends and family. I would like to thank all the people.

First and foremost, especially, I would thank my supervisors Professor Kazutoshi Fujikawa and Hideki Sunahara. They were always positive about what I would do, and gave me enough chances to find research topics and necessary facilities to conduct my research. I thank Professor Suguru Yamaguchi for providing a wealth of knowledge and insightful advice, which enabled this dissertation much better. I would also like to thank Associate Professor Atsuo Inomata, who invited me to jump into the academic world.

Half of this research was performed at INRIA in France. The life in INRIA has totally changed my thinking. Working for future Intelligent Transport Systems with many colleagues from a wide variety of countries was stimulate my motivation for research. Especially, supporting the International project GeoNet was very interesting. I thank all the people who gave me a chance to visit to INRIA and join in the project. Especially, I would like to thank Dr. Thierry Ernst, who has been supervising me. His strong belief for IPv6 and Intelligent Transport Systems always encouraged me to proceed my research when I struggled with several difficulties. Also, discussing culture, history, business, and especially bicycle and wine with him was always a pleasure. I also thank the colleagues in INRIA, especially Dr. Jong-Hyouk Lee and Dr. Manabu Tsukada to help me to describe and evaluate the proposed scheme.

I would also like to thank very much all inet-lab staffs and colleagues. I always enjoyed insightful discussions with Associate Professor Satoshi Matsuura, who introduced how to enjoy the college life with tasty foods and drinks. I would thank secretaries Ms. Yuuki Ro, Ms. Rie Kawamoto, and Miss Takako Tanabe for their warm support. Finally, I would like to give a special thanks to my Parents. I think am not a good son since I have always been enjoying unstable positions, such as quitting a job and return to this academic world. Nevertheless, they always supported my decision and gave me a lot of supports. Without their kind support, this dissertation has never been completed.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Resource discovery in Cooperative Intelligent Transport Systems

The era of pervasive computing has now emerged. Beyond personal desktop computers, many devices in everyday life have networking capability and numerous computing resources. Laptops, smartphones, ebook readers, tiny sensors, and even vehicles are now equipped with a wide variety of access technologies and are connected to several different types of networks. Furthermore, many devices are now able to locate themselves using GPS and other functions. In this environment, in recent years, devices not only use services but also provide software and hardware resources to others; in addition to the traditional client-server model, the development of the peer-to-peer (P2P) model has made it possible to create distributed mobile applications that orchestrate several distinct mobile and static resources.

Several location-based services for mobile devices have emerged from this development, such as personal communication, location sharing, and physical/virtual object detection and tracking. Today, people regularly exchange text, voice, and video messages with their family and friends. Furthermore, they share their location and context (where they are and what they are doing). In addition, by detecting neighboring devices, users can utilize personalized services (e.g., localized weather information). Such usage is not limited to physical objects; virtual objects (e.g., point-of-interest information, music, photo, or other kinds of data) can also be detected.

One of the most important realizations of these location-based services in recent years is Cooperative Intelligent Transport Systems (C-ITS) applications. C-ITS applications are communication-based services for ITS mainly integrated into vehicular communication systems (VCS). By enabling vehicles, pedestrians, and the roadside to communicate with each other, road users can be offered significant benefits in road transportation. C-ITS services are therefore designed as mobile applications to provide road users with improved road safety, traffic efficiency, and additional value using networked devices related to road transportation. Such cooperative services are obviously much more efficient in comparison to stand-alone services. Because they save lives and improve road traffic usage, C-ITS applications are some of the most important emerging services in recent years. Some of the typical C-ITS applications studied by researchers and standardization organizations [23, 28, 33, 68, 54, 20] are outlined as follows:

- **Road safety applications:** applications primarily developed in ITS to save human life by avoiding road hazards, such as driving assistance with *pre-crash sensing*, *road hazard warning*, *intersection collision risk warning*, etc.
- **Traffic efficiency applications:** applications to improve road traffic, such as through *co-operative navigation*, *speed management*, *public transportation vehicle tracking*, etc.
- **Value added applications:** applications to provide road users with additional value, including entertainment services and e-commerce, such as *traveler information services*, *point of interest notification*, *Internet (www) access services* etc.

An essential function for these mobile applications is to locate necessary resources. Although numerous software and hardware resources can be used through networks, applications cannot always detect all of them because of the mobility and heterogeneity of modern networks, in which multiple different access technologies and network protocols are combined to connect a wide variety of networks. For instance, vehicles are considered to use ad-hoc networking technologies, while smartphones mainly use cellular or infrastructure based WLAN. In such a situation, even though a device hosting a certain resource is physically in front of other devices, its resource cannot be consumed unless the surrounding devices know of its existence. Relying on static configurations is not sufficient to solve this problem because there is a large number of resources in the Internet. In trivial LANs using Ethernet, local-scope broadcast and multicast easily detect resources in the considered network. However, at present, this is a very complicated task because of the development of several new types of networks, specifically VCS, which rely on a wide variety of advanced networking protocols and access technologies. Traditional point-to-multipoint communication protocols on a single access technology are no longer feasible solutions because physically-neighboring nodes can be connected to various different access channels using different networking protocols.

To locate resources in this complex environment, *Resource Discovery*, a set of techniques to find resources in a certain network, is needed. *Resource Discovery* has been studied from the points of view of several different layers, including node discovery protocols in the network layer, service discovery protocols, and suites of distributed search mechanisms in the application layer. At first, these technologies have been designed for a single static network or a Mobile Ad-Hoc NETwork (MANET), but recent developments in networking technologies, especially VCS, have raised several new challenges. The generic issues in mobile resource discovery are outlined as follows:

- **Heterogeneous connectivity:** most mobile devices are now equipped with multiple access interfaces and protocols that prevent the use of a simple point-to-multipoint communication to discover a certain resource. Smartphones and laptops commonly use more than two network interfaces such as cellular, WLAN, Bluetooth, etc. Further, in VCS, even on a particular interface, multiple different networking protocols can be used simultaneously. Trivial broadcast- and multicast-based resource discovery techniques do not support such a heterogeneous connectivity.

- **Indeterministic and complex user demands:** users' demands in location-based services are much different from those in IP routing, which involve delivering packets to explicitly-determined destinations. In the context of resource discovery, consumers try to locate resources with no concern about *who the resource providers are*; instead, they use much more complicated criteria such as geographical location, organizer, mechanical status, trajectory, etc. For resource consumers, *the identity of the resource provider does not matter*; instead, *whether there is a requested resource that satisfies the consumers' predicates* is more important. At the same time, *accuracy of the information* should also be taken into account: it is necessary to determine whether the information provided by other nodes is valid, accurate, and precise.

- **A huge number of mobile consumers and providers:** networking devices include not only desktop computers, but the huge number of mobile devices. For instance, there are 1 billion actively-used smartphones in the world as of today. Furthermore, regarding ITS, as of 2012, the number of registered vehicles in the world is more than 1 billion (75 million in Japan) [53]. This means 1 billion more mobile nodes will be connected to the Internet when C-ITS get fully deployed.

By tackling these issues, *Resource Discovery* can be used as a common infrastructure for mobile applications for numerous mobile users. However, for the emerging C-ITS applications in the near future, further issues must also be tackled. The following Section outlines these issues and describes the objectives of this dissertation.

1.2 Objectives of this research

This research aims at designing a resource discovery system for C-ITS applications to study the question: *how can a certain resource be discovered based on its geographical position in a wide variety of mobile networks composed of a huge number of mobile nodes?* Because many types of mobile devices (e.g., vehicle, roadside infrastructure, pedestrian) will be connected with each other, the resource discovery system for C-ITS applications needs to support several types of resource consumers and providers. These consumers and providers have several different mobility patterns (movement characteristics of mobile users), for instance, in-vehicle nodes drive in a congested city or on a clear highway, whereas a pedestrian with a smartphone may take a walk in a park, or ride a bicycle along a street. In addition to the above mentioned generic issues, the following C-ITS-specific issues must be tackled by the resource discovery system for C-ITS:

- **Variable scope of discovery:** at the time of discovery, *which node can provide a certain resource* is not known. This means an appropriate discovery scope cannot be determined a priori. Because there are several different types of networks such as an infrastructure-based WLAN, Vehicular-Ad-Hoc NETwork (VANET), and cellular network,a scope that is too large will cause scalability issues. On the other hand, if the scope is too small, consumers may not be able to locate potential resource providers. Numerous resource discovery schemes have been proposed for a specific type of network (e.g., a local area network, MANET, or the Internet) for specific scenarios. Nonetheless, there have been no solutions that could handle all discovery scenarios simultaneously.

- **Geographical discovery:** users try to discover resources using application-specific criteria instead of system- and network-specific criteria. Specifically, geographical position must be supported as a criterion because ITS applications locate resources based on the relative/absolute geographical position. In many cases, the topological location does not matter to resource consumers. Traditional resource discovery techniques may support geographical discovery. However, it is realized only in the application layer. In other words, in these solutions, the geographical position is nothing more than one of the resource attributes. Such solutions relying on conventional point-to-multipoint messaging (e.g, broadcast, IP multicast, etc.) may deliver too many messages to non-relevant geographical areas, which cause scalability issues.

- **Scalable resource management:** there are a huge number of mobile resources (e.g., there are 1 billion vehicles as of 2012), which change their physical/topological location and application-level condition (trajectory, mechanical condition, etc.). Distributed resource discovery solutions cannot deal with this huge number of mobile resources, whereas centralized solutions may face the single point of failure issue and load concentration issue.

- **Rapid resource discovery:** the resource discovery system must locate requested resources within a specific period of time so that applications can start and/or finish their process in time. Because C-ITS applications for road safety or traffic efficiency have tight latency requirements, it is necessary to discover resources as quickly as possible: if appropriate resources cannot be discovered quickly, traffic accidents or road congestion may occur. However, in combination with the scope selection issue, it is difficult to rapidly determine the existence, availability, and location of nodes providing the requested resource. In addition, multiple access technologies and network protocols with several different mobility patterns may prevent to rapid exchange of discovery messages.

To handle these issues, this dissertation describes a method for managing huge numbers of mobile resources (mainly in vehicles) by proposing a wide-area resource discovery system. It relies, not only on a distributed architecture, but also on a partially-centralized architecture, in combination with a publish/subscribe scheme built on a geographic routing protocol and overlay networking technologies.

As described above, the target of the resource discovery system in this research is C-ITS applications integrated into mobile objects such as vehicles, the roadside, and pedestrians. Therefore, the mechanisms proposed in this dissertation will be evaluated in various communication scenarios of C-ITS applications. In VCS, distributed mobile applications are integrated into a wide variety of mobile devices provided by numerous stakeholders to supply road users with improved traffic safety, traffic efficiency, and additional value [33, 50, 68]. The VCS is a group of networks, including VANETs, the Internet, and a wide variety of access networks, e.g., cellular networks, public networks, and private networks. Therefore, C-ITS applications need to efficiently discover necessary resources.

1.3 Contributions

To design a resource discovery system for a wide range of distributed mobile C-ITS applications, this research presents two approaches. The first involves local-area geographical resource discovery for VANETs, and the other involves wide-area publish/-subscribe resource discovery.

To clarify the issues and existing solutions, first, Chapter 2 explores the use cases of resource discovery in C-ITS and state of the art of this research. To analyze the necessity of C-ITS applications, next Chapter then evaluates a practical use case of C-ITS applications by integrating an experimental driver support application into actual vehicles. And then subsequent two chapters describe the above mentioned approaches. The main contributions are outlined in the next clauses.

1.3.1 Use case evaluation: integration of in-vehicle driver support application into actual vehicles

Because *what the C-ITS applications are* is not clear yet as of now, Chapter 3 explores potential use cases of C-ITS applications so that it clarifies their necessity in a realistic scenario. To show the feasibility of C-ITS applications, it presents the actual integration of a driver-support application into vehicles, and evaluates its performance. The application was built on VANET using IPv6 GeoNetworking, a set of mechanisms that enables the delivery of an IPv6 packet to a specific geographical area. Publication [64] is based on this contribution.

1.3.2 Geographical resource discovery to support resource discovery locally

Chapter 4 describes the first approach, which focuses on a small scope of resource discovery: local scope discovery inside a VANET. This approach tackles the problem of *how to locate a subset of resources inside a specific geographical area without incurring massive bandwidth usage* by means of the sub-network layer GeoNetworking protocol and an IPv6 multicast-based service discovery protocol. Publications [66, 65] are based on this contribution.

1.3.3 Wide-area geographical publish/subscribe resource discovery

In Chapter 5, the second approach is presented. It proposes an extended resource discovery mechanism for wider-scope discovery scenarios. To deal with the huge number of resources widely distributed, the proposed mechanism exploits, not only the GeoNetworking protocol, but also a publish/subscribe scheme and overlay routing technologies. Publication [63] is based on this contribution.

Chapter 2

Context and state of the art

This Chapter presents the context of this research and existing solutions of resource discovery. For a better understanding of resource discovery, this chapter is started by exploring the generic definition of resource discovery from several aspects. Cooperative Intelligent Transport System (C-ITS) applications, Vehicular communication systems (VCS), and their potential communication scenarios are then outlined. After discussing the functional and performance requirements for the resource discovery system for C-ITS, a taxonomy of resource discovery solutions is presented by defining an abstraction model. Finally, underlying issues and potential solutions will be shown.

2.1 Resource discovery in mobile networks

2.1.1 High-level definition

The action of discovering something is necessary for any entity to perform a certain task. In everyday life, human beings try to discover something at any given time: for instance, we discover a key to open a door, a novel to spend a holiday, a hospital to take medical treatment, a friend in nearby area to have lunch, a telephone number of a business partner. In other words, most tasks are composed of two actions, discovering and then using something. Historically, only human beings (more precisely, living creatures) have been performing this action. However, these days other entities, machine and software, also need to try to discover something: for instance, warehouse management robots look up goods to deliver them to customers, building security system detects intruders, and web browser locates an IP address from a domain name.

To understand the context of resource discovery, this research defines it as *a behavior of resource consumers to find the location of resource provider using a number of criteria*. Each entity in this context is defined as follows:

- **Resource consumer and provider** is a physical (i.e., living creature and machines) or virtual object (e.g., software, and data).
- **Location** is a position of resource in a certain *space* in which resources are consumed. In the case of physical objects, the location is a physical position of them, while it can be a virtual position for virtual objects: a phone number, socket address, etc.
- **Criteria** is a set of attributes to describe the importance of resource. In other words, the criteria shows a position in a (set of) certain semantical spaces, such as a physical coordinate system, physical structure (e.g, shape, color, size) social graph, computer network, etc.

While human beings have sophisticated ways of resource discovery using the five senses, these days they need to discover resources in a location beyond their five senses: computer networks. Furthermore, as described in the previous clause, machines and software also need to have a way to efficiently discover resources in computer networks. This is one of underlying motivations of this research.

From the point of view of computer network, the action of resource discovery is therefore identified as *discovering communication endpoints of resources using a set of attributes in a network.*

2.1.2 Resource discovery as an infrastructure

One of efficient ways to provide the capability of resource discovery in communication networks is to build it as a common infrastructure, such as the system of DNS, SMTP and POP, etc. Although each node is able to discover resources through its own manner, it is insufficient in terms of interoperability, extensibility and scalability. Consequently, to discover resources in computer networks, resource discovery should be provided as an infrastructure: a set of techniques to locate communication endpoints of a certain software or hardware resource for mobile applications.

In this context, mobile applications are composed of a number of distinct resources divided into an individual, sharable, and reusable entity. Applications dynamically orchestrate resources placed on local and/or remote nodes. This paradigm is well known as *Service Oriented Architecture* (SOA) [67], which enables application developers to avoid developing redundant functions and to cooperate with other developers.

In general, applications may (i) directly send data to a particular group of nodes that may provide necessary resources, or (ii) selectively send data to a subset of nodes that certainly provide the resources. In the former case, one solution is broadcasting data to the considered network regardless of existence of resources. On the other hand, in the latter case, at first applications resolve the communication endpoint of necessary resources, then exclusively deliver data to the discovered nodes. The direct delivery is suitable for latency-demanding use cases within single hop distance because of the quick propagation of messages. However, if applications need to communicate with resources located in more than single-hop distance, the broadcast-based direct delivery may consume too much bandwidth, especially in a dense network. In the worst case, it leads applications to send data to all nodes inside the considered network. Furthermore, even if there is no appropriate resource in the network, the broadcast based communication cannot prevent applications propagating data. The objective of this infrastructure, the resource discovery system, is to enable applications to dynamically discover the existence, characteristics and communication endpoints of resources. Conceptual components and functions of resource discovery will be described in Section 2.8 in detail.

2.1.3 Discovery traffic and data traffic

In this research, the traffic generated in an application's process is identified with two types of sub-traffics: *resource discovery traffic* and *data traffic*. The resource discovery traffic is generated preliminary to the data traffic. It is dedicated to locate resources, similarly to the DNS lookup in the WWW service. On the other hand, the data traffic corresponds to the actual applications' data flow. In other words, at the time of resource discovery, under the transport layer, destination of the data traffic is not determined yet.

2.1.4 Type of resource

The basic definition of resource is *a software and/or hardware entity which has a distinct communication endpoint* (e.g., socket address). They are identified with two categories: application level resource and network level resource. Application level resources provide upper-layer dependent information and services, such as:

- **Information service**: providing (i) hosts' information (e.g., presence of hosts inside a certain area with hardware/software status of them), (ii) environmental information acquired with e.g., sensors.
- **Processing service**: processing hardware/software entities; (i) manipulating embedded actuators (e.g., electronic gate), (ii) providing consumer services (e.g., on-line payment, instant messaging).

On the other hand, network level resources provide networking-dependent services, such as Internet gateway (e.g., Access Router), addressing (e.g., DHCP server), name resolution mechanism (e.g., DNS, LDAP), etc.

2.1.5 Potential services

Mobile applications provide mobile devices with a number of distinct type of services, for instance:

- **C-ITS services:** one of emerging services for road users including drivers of vehicles and pedestrians. C-ITS services aim at providing road safety, traffic efficiency and additional value by exchanging a wide variety of road traffic information, such as the presence of vehicle and pedestrian, mechanical condition, etc. For road safety, single hop broadcast is mainly used, on the other hand, longer-range, two-way, point-to-point communications can also be used for some of road safety and other types of services. These services have been considered as distributed applications that dynamically orchestrate necessary resources [29, 30].
- **Information sharing for autonomous vehicles:** to drive safely without the support of drivers, this service provides autonomous vehicles with exchanging sensing information among nearby vehicles. It is useful in the case that only a few

set of vehicles have an expensive radar sensor, but not the others. In this case, vehicles equipped with the better sensor publish this services so that other vehicles, driving in a platoon for example, could benefit from this service (e.g., exploit their raw data differently for other purposes).

- **Location based services:** location based services in our everyday life, such as text messaging with nearby friends and family, personalized Web browsing performed in every mobile devices. The difference between the location based services and traditional services is to use the location of each node.

This research focuses on services for C-ITS and autonomous vehicles.

2.1.6 Focusing on vehicular communication systems

Resource discovery is a mature paradigm; there have been a large number of popular solutions from the point of view of *service discovery*, *neighbor discovery*, and *distributed search* techniques. Existing solutions have been designed on a specific protocol for a specific usage, such as service discovery protocols for a local area network, for a MANET, or wide-area discovery protocols supporting the Internet. Historically, these solutions have been studied at first for single and small-scale static network, then the concept of resource discovery has been extended to MANET in cooperation with ad-hoc routing protocols. At the same time, Internet-based solutions for the wide-area discovery have been designed using overlay networking technologies.

While these existing solutions may be able to used in VCS for C-ITS, however, a VCS-dedicated resource discovery should also be considered rather than re-using existing solutions. C-ITS applications are one of most important and emerging services in recent years, which have a number of characteristics, as described in Chapter 1. A resource discovery system in C-ITS needs to support functions in several types of resource discovery solutions: in some cases it should perform a local-scope discovery, on the other hand, wide-area discovery may also be performed. Furthermore, each vehicle may move through several mobility patterns, such as a vehicle in congested intersections, on clear highway, in grid-shaped streets, etc. It is therefore necessary to focus on VCS; a VCS-dedicated resource discovery system should be investigated.

2.2 Basic requirements

To support the needs of C-ITS applications, the resource discovery system should support the following functional and performance requirements:

- **Universal visibility:** the resource discovery system shall enable resource consumers to discover resource providers in C-ITS regardless of underlying network topology and access technologies; as long as resource providers are reachable from resource consumers, the system shall enable the consumers to discover a certain provider.

- **Geographical location based discovery:** the resource discovery system shall allow resource consumers to discover resource providers according to geographical position. Since they are mainly equipped with mobile devices, geographical location shall be a primal criterion for resource discovery.

- **Incremental and extensible deployment:** the resource discovery system shall be incrementally deployed and improved on the existing Internet architecture. It means components of the system shall be able to be dynamically added or deleted to existing networks without re-designing the Internet. Furthermore configurations of the system and the representation of resources shall also be able to be dynamically modified.

- **Scalability:** the resource discovery system shall support a huge number of participants in C-ITS. Specifically, it shall support all the main and potential users in C-ITS: 1 billion vehicles, which is the number of registered vehicles is in the world, while 74 million in Japan [53].

- **Latency:** the resource discovery system shall discover resources within a certain limited period of time to support critical use cases of C-ITS applications, such as 100ms for road safety, 200 to 500ms for traffic efficiency and 500ms for other applications, mentioned in 2.4. Note that the definition of latency in this dissertation is the acceptable length of time it takes for a resource discovery request to be sent plus the length of time it takes for a resource discovery result of the request to be received.

2.3 Use cases of mobile applications

Abstracted use cases of mobile applications are defined with the type of actors, such as vehicles, the roadside and pedestrians, and ITS centers. Note that this section explores the abstracted use cases from the point of view of application without assumptions about resource discovery mechanisms, underlying network topologies, routing protocols. It means the following use cases are not relevant from the network-layer communication mode, such as V2V, V2R, and V2I etc. Discussions on the resource discovery mechanism with underlying networking technologies will be described in the next Section.

- **Mobile-from/to-Mobile use case:** a mobile node communicates with other mobile nodes shown in Figure 2.1. In the case of a traffic efficiency application, when a vehicle is going to merge into a lane, relevant vehicles (vehicles running on the same lane) recognize its presence and notify drivers that the vehicle is approaching. In this use case, destination nodes are determined according to geographical position and application specific attributes (e.g., user type, device type, vehicle type, mobility pattern). The destination may be only one mobile node, a set of mobile nodes, or all mobile nodes inside a certain area. For shorter range communication, applications may use the direct broadcast-based communication inside a network, whereas static nodes may help their messaging as forwarders in case of the range is longer than direct communication range.
- **Mobile-from/to-Static use case:** a mobile node communicates with static nodes connected to a mobile network or vice versa. For instance, as shown in Figure 2.2, when a cyclist using a smartphone is going through a highly congested area, the application communicates with an infrastructure that monitors the road traffic condition in the area. Destination nodes are determined with geographical position and specific attributes (e.g., type of infrastructure, organization). The destination may be only one node or all nodes which satisfy a certain criteria. Applications basically try to communicate with other nodes via direct one-to-one or one-to-multiple connections inside a mobile network, but they may also communicate through other nodes in the Internet if shorter range communication does not satisfy their requirements.

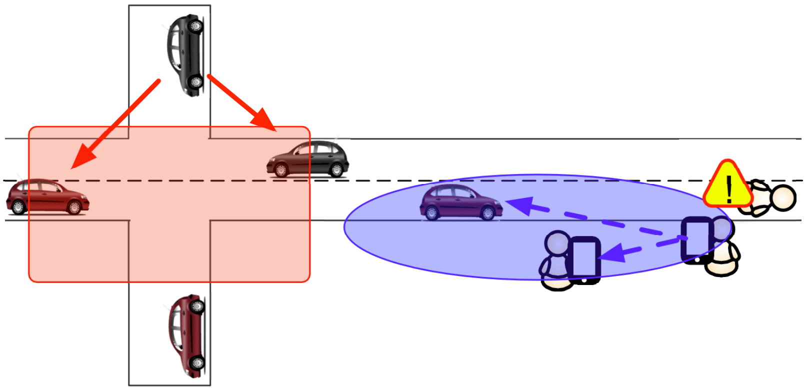


Figure 2.1: Mobile-from/to-Mobile communication scenario.

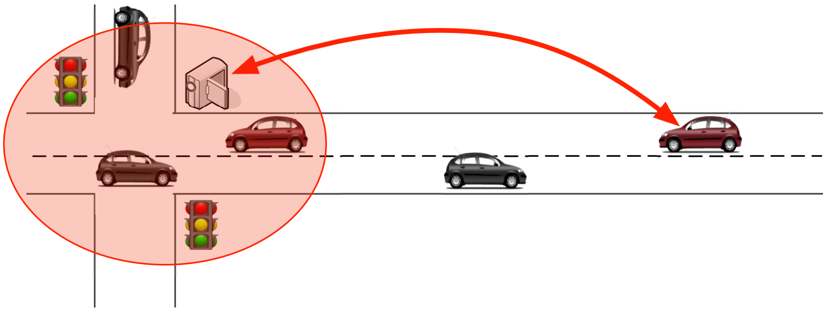


Figure 2.2: Mobile-from/to-Static communication scenario.

- **Mobile-from/to-Internet use case:** mobile nodes and Internet nodes communicate with each other. For example, in the case of a traffic efficiency application shown in Figure 2.3, when a vehicle detects a certain road traffic event (e.g., accident, congestion, road work), it reports descriptions of the event to an ITS center in the Internet. The ITS center informs the event to a set of vehicles inside a relevant geographical area. The destinations is determined with specific attributes (e.g., type of vehicle, trajectory, supporting services). In the Internet-to-Vehicle case, vehicles' geographical location is also included to the selection criteria. The destination may be only one host or all hosts satisfying a certain criteria.

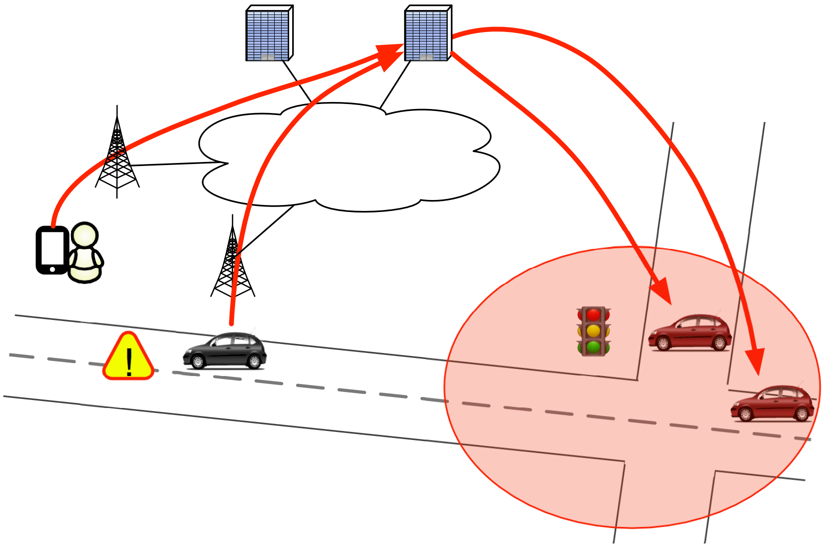


Figure 2.3: Mobile-from/to-Internet communication scenario.

2.4 Cooperative ITS applications

Cooperative ITS (C-ITS) applications are distributed applications integrated into mobile and static nodes. As outlined in Section 1, three main categories of C-ITS applications have been identified: road safety, traffic efficiency, and others (i.e., infotainment and value-added services). The road safety application aims at reducing the number of road traffic accidents, while the traffic efficiency application aims at improving the efficiency of road traffic. On the other hand, the other types of applications provide road users with additional values including commercial services [28, 33, 68, 54]. These applications are characterized with their performance requirements and the type of communication. The descriptions of each applications are listed below:

- **Road safety application:** to save human life in critical situations, road safety applications have stringent performance requirements (e.g., the latency requirement is less than 100ms [33, 68, 54]). For this reason, they mostly communicate inside a VANET via one-way, short-range communication with Vehicle-to-Vehicle (V2V), Vehicle-to-Roadside (V2R), or Roadside-to-Vehicle (R2V) communication modes. Internet hosts may enhance them through Vehicle-to-Internet (V2I) and Internet-to-Vehicle (I2V) communication.

- **Traffic efficiency application:** communication requirements of traffic efficiency applications are relatively relaxed comparing to the road safety applications (e.g., from 200ms to 500ms latency) [68, 54]). These applications may perform any types of communications.

- **Others:** the other kinds of applications basically do not have tight performance requirements (e.g., 500ms latency) [68, 54]). Internet-based, two-way unicast communication may be used to support infotainment services, but other communication modes are also possible.

2.5 Cooperative intelligent transport systems

The target of this research is C-ITS applications mainly integrated into VCS, parts of Intelligent Transport System (ITS). This Section clarifies *what is C-ITS*.

2.5.1 Network architecture

C-ITS consist of interconnected mobile and static networks including VCS (VANETs and ITS-dedicated access networks), public access networks, and the Internet. In VCS, participating nodes are hosts/routers either in vehicles or in the roadside, while pedestrians may also join in C-ITS. Static hosts, such as servers in ITS centers connected to the Internet, can also be involved in C-ITS. Figure 2.4 shows the simplified network topology considered in this research.

Several projects and standards organizations for ITS have been specifying a communication architecture, protocols and functions integrated into vehicles and the roadside. One of well-known definition is *ITS station*, a pair of host and router supporting the *ITS station reference architecture* standardized by ISO and ETSI [34, 50, 68, 51]. The ITS station reference architecture is an ITS-dedicated protocol stack basically following the ISO/OSI reference model [49]. It defines four horizontal protocol layer (Applications, Facilities, Networking & Transport, and Access layer) and two vertical planes (Management and Security planes) shown in Figure 2.5. The Facility layer is a new protocol layer which provides applications with a collection of common functions, such as information about neighboring area aggregated from different sources.

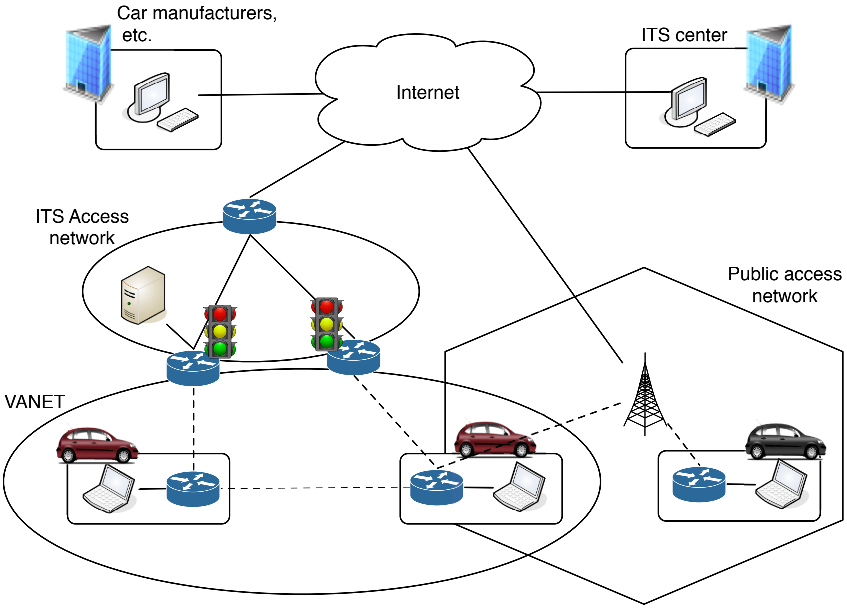


Figure 2.4: C-ITS network.

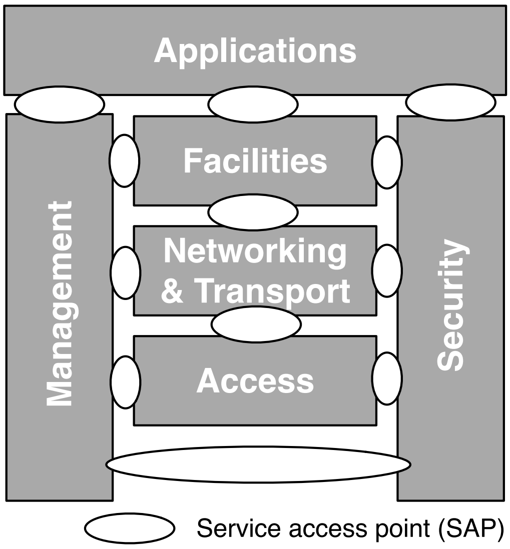


Figure 2.5: ITS station reference architecture.

In VCS, vehicles are equipped with at least one mobile router (MR), whereas access routers (ARs) are installed along the roadside. These routers compose VANET: they construct multi-hop capable wireless ad-hoc network topologies. ARs and possibly MRs provide Internet access to other MRs. Both MRs and ARs are capable of connecting application hosts which operate C-ITS applications. To connect to the Internet, IPv6 is considered as a mandatory communication protocol in ITS network because of its capabilities of mobility support and huge address space [32]. In addition to IPv6, IP mobility support protocols (e.g., NEMO [31]) are considered to manage the change of position of the vehicles within the network. MRs and ARs are equipped with multiple access technologies, such as WLAN, IEEE802.11p, cellular, etc. In the definition of ISO, these routers and host are called *ITS-S router* and ITS-S host, respectively. Note that the capabilities of the routes and application host may be integrated either into single device or separated devices.

2.5.2 IPv6 GeoNetworking

To deliver packets to a certain geographical area, the CAR 2 CAR Communication Consortium [1] has initiated *GeoNetworking*: a geographic addressing and routing protocol. They have developed the C2CNet, a communication layer supporting GeoNetworking and is located over the link layer [23].

While GeoNetworking is essential for C-ITS, IPv6 is also a mandatory communication protocol thanks to its huge address space, security mechanism, and mobility support extensions [32]. Therefore the GeoNet project, [6] an European project, has specified *IPv6 GeoNetworking*, which enables IPv6 operating over GeoNetworking. It conforms the C2CNet specification [41]. The GeoNet project has defined how to transmit IPv6 packets over C2CNet (namely *IPv6 over C2CNet*) using the encapsulation.

In accordance with the specification and evaluation results of the GeoNet project, the European Telecommunications Standards Institute (ETSI) has proposed a GeoNetworking protocol and its adaptation mechanism to IPv6 (*GN6ASL: GeoNetworking to IPv6 Adaptation Sub-Layer*) [39], which enables IPv6 operated over the network-layer GeoNetworking protocol [38]. ETSI's GeoNetworking is based on the GeoNet's results, therefore it is a full set of non-IP geographical communication technologies, including a geographical addressing scheme, geographical routing algorithms, and geographical location resolution mechanism (namely *Location Service*).

GeoNetworking is located between the network layer and the link layer, and GN6ASL provides interfaces to IPv6 so that upper layer entities can transparently use GeoNetworking through the conventional IPv6. GeoNetworking makes the routing decision according to the link-layer address (i.e., MAC address), GeoNetworking address (basically generated from the link-layer address), and geographical position. Although GeoNetworking exchanges packets according to the link-layer address and the GeoNetworking address without IPv6 address, GN6ASL enables the network layer to use the GeoNetworking capabilities; delivering IPv6 packets to a certain geographical area by encapsulating the packed with a GeoNetworking header. In IPv6 GeoNetworking-enabled VANETs, each node is assigned an address. When a node constructs an IPv6 packet, GN6ASL encapsulates the packet with a GeoNetworking header, which includes the IP next hop's GeoNetworking address and/or a specific geographical position. The transmission mode (e.g., GeoUnicast, GeoBroadcast) is determined from the destination IPv6 address; for instance, if the destination is IPv6 unicast address, packets are delivered as GeoUnicast packets, on the other hand, in the case of IPv6 multicast, packets are sent as GeoBroadcast packets. IPv6 itself is thus used in a conventional manner; when upper layers issue IPv6 packets, the IPv6 GeoNetworking mechanism transparently encapsulates and delivers them as Geocast packets respectively. In particular, GeoBroadcast is useful to deliver packets to all nodes inside an certain geographical area, described as a circle, rectangle, or ellipse.

2.5.3 Participating nodes

C-ITS are composed of a wide variety of mobile nodes supporting several wireless access technologies and protocols. According to the type of mobility, three types of nodes can be classified: vehicle, roadside, and pedestrian. Each user is classified according to the type of device and mobility patterns as follows:

- **Vehicle** a set of mobile nodes equipped with a car, bus, or train, etc. It can be a smartphone, tablet PC, laptop, and small handheld computers dedicated to a specific usage. This type of user linearly moves on the road at relatively high velocity (40 km/h on average). Historically, this type of node has been called as On Board Unit (OBU) from a generic point of view. From the point of view of networking, it is called *vehicle ITS station* in ISO and related ITS project [51].

Table 2.1: Participants in C-ITS

Type	Device	Access technology	Mobility	
			Pattern	AVG. Speed
Vehicle	Smartphone, tablet PC, laptop, dedicated device	Cellular, WLAN, 802.11p	Along the road	40 km/h
Roadside	dedicated device	Cellular, WLAN 802.11p	(Static)	-
Pedestrian	Smartphone, tablet PC	Cellular, WLAN	Random	5 km/h

- **Roadside** a set of static nodes equipped with the roadside, or some static objects (e.g., a signboard of temporary road work). It can be a dedicated networking device. Historically, this type of node has been called as Road Side Unit (RSU)from a generic point of view. From the point of view of networking, it is called *roadside ITS station* in ISO and related ITS project [51].

- **Pedestrian** a mobile node equipped with a person on foot. It can be a smartphone, tablet PC, laptop, and small handheld computers dedicated to a specific usage. This type of user freely moves around any place at a relatively low velocity (5 km/h on average). As of now ITS stakeholders are concentrating on the development for vehicles, thus applications for this type of node are not addressed so much yet, however it is included as an actor of C-ITS, such as *Personal ITS station* specified in [50].

Table 2.1 shows the characteristics of these nodes.

2.6 Local scope resource discovery

This section presents the state of the art of resource discovery techniques designed for small scale network. To explore existing solutions from broader point of view, a wide variety of discovery strategies are listed, such as not only service discovery protocols but also other mechanisms used to locate certain resources.

2.6.1 Neighbor Discovery Protocol (NDP)

The Neighbor Discovery Protocol (NDP) [62] by IETF is a networking protocol designed for IPv6 as a part of the TCP/IP protocol suite. It is used with IPv6 to discover neighboring IPv6 hosts and routers on the same link, to automatically configure IPv6 address, and to locate link-layer address of on-link destinations. A typical function of NDP is Router Solicitation (RS) and Advertisement (RA), which enables IPv6 hosts to locate routers on the same link, as a part of StateLess Address Auto Configuration (SLAAC). It is triggered from either RSs by IPv6 hosts or unsolicited RAs by routers. On the other hand, Neighbor Solicitation (NS) and Advertisement (NA) are used to locate link-layer address of a neighbor. NSs and NAs are also used to verify the neighbors' presence and to validate the duplication of IPv6 Address.

NDP is mainly used over multicast-capable links; it relies on the link-local scope IPv6 multicast. In summary, this protocol supports link-local scope resource discovery on IPv6 networks to locate IPv6 and link-layer addresses. It is a purely decentralized protocol in which no administrative entities are needed.

2.6.2 Service Location Protocol (SLP)

Service Location Protocol (SLP) [46] by IETF is an application layer protocol dedicated for service discovery as a part of the TCP/IP protocol suite. Basically it works on UDP over IPv4/IPv6 in a single local area network. It is used to locate networking devices such as network printers. In SLP, services are described with Service URL: `service:<srvtype>://<addrspec>` and optional attributes. SLP also relies on link-local scope IP multicast: a User Agent (UA) directly sends **multicast Service Request** to Service Agents (SAs) to locate services. Although [46] specifies a pull-based protocol so that UAs get information about service only when they generate request messages, [55] describes a notification mechanism that enables UAs to be informed when a new service appears or disappears without periodic polling.

SLP has primally been designed for IPv4, then extended for IPv6 by [44], in which each service is assigned a particular IPv6 multicast address calculated from the Service URL so that UAs can locate a certain service by selectively delivering service request messages. To describe services, [45] has also been proposed.

2.6.3 Universal Plug and Play (UPnP)

Universal Plug and Play (UPnP) [77] specified by the UPnP Forum [14] is a group of protocols to manage plug-and-play devices basically on the same link based on the TCP/IP protocol suite. It has six internal steps: addressing, discovery, description, eventing, control, and presentation. Any device can use UPnP as long as it supports IP, UDP, HTTP and XML. To discover network address of devices and their services, UPnP uses Simple Service Discovery Protocol (SSDP) that discovers devices using a multicast variant of HTTP over UDP.

UPnP is a fully decentralized mechanism composed of control points acting as service consumers and devices as service providers. They mainly multicast SSDP messages (including the type of device, universally unique identifier, optional parameters etc.) to advertise and search devices and their services. SSDP messages are described by partly complying to the HTTP header format.

2.6.4 multicast DNS and DNS based Service Discovery

The combination of Multicast DNS (mDNS) [10] and DNS-based Service Discovery (DNS-SD) [4, 21] is a service discovery technique using standard DNS components and messages, as an IP-based replacement of AppleTalk. They are parts of Zero Configuration Networking (Zeroconf) protocols [15]. The Apple's implementation of these protocols are announced at first *Rendezvous*, later *Bonjour*, included in Mac OS X.

mDNS is a multicast-enabled variant of unicast DNS. Each node acts as a DNS server, which maintains information of its services using DNS resource records. All nodes supporting mDNS join an IP multicast group allocated to mDNS, to which service consumers, acting as DNS clients, deliver DNS messages.

DNS-SD is used to describe services using DNS SRV and TXT records without changing the existing DNS messages. Using the SRV record, a service is described as a name of the form `<Instance>.<Service>.<Domain>`, while optional attributes are described with the TXT record as pairs of key-value data units.

The pair of mDNS and DNS-SD is a decentralized service discovery protocols, in which centralized entity is not needed: a mDNS/DNS-SD-enabled service consumer directory multicasts discovery requests to service providers, and then service providers also multicast reply messages. It is mostly used for locating services in the same network.

2.6.5 Konark

Konark [57] is a middleware designed to discover device-independent services mainly on small handheld mobile devices composing ad hoc networks, based on SOAP. It describes services using XML and discovers through HTTP; each Konark-enabled node act as a micro-HTTP server.

Similar to the other local-scope resource discovery mechanisms, Konark performs fully distributed, link-local scope multicast-based service discovery; a client multicasts discovery messages to a certain IP multicast group, and service providers reply to it. In addition to this mechanism, it uses *Konark service gossip protocol*: each device has its own service repository including information about services offered by **other** nodes. Each node listens to service advertisement, request and response messages sent by other nodes, and multicasts messages containing a list of services which are not advertised by other nodes yet. In other words, this protocol lets nodes exchange their cache without duplication of information (*repeated gossip*) so that Konark is able to reduce network traffic for service discovery.

2.6.6 Cooperative Awareness Message (CAM)

The Cooperative Awareness Basic Service specified by ETSI [37] is one of mandatory facilities in the ITS station reference architecture. It broadcasts Cooperative Awareness Messages (CAMs) to provide information of presence, positions and status of nodes (mostly, in-vehicle nodes) to neighboring nodes. CAMs are delivered to single-hop neighbors basically within an IEEE 802.11p network. A number of data elements have been specified, such as the ITS station's unique identifier, type of station, position, speed, heading, vehicle's profile (e.g., size, capacity), and current status (e.g., about doors, exterior lights). While CAMs are generated periodically, the event-based generation rules are also applied depending on the difference of heading, speed, position so that each node is able to detect the presence and descriptions of neighboring nodes rapidly.

2.6.7 Group-based Service Discovery (GSD)

Group-based Service Discovery (GSD) [25, 26] is a service discovery protocol designed for MANETs. It is based on the peer-to-peer caching of service advertisements and group-based selective forwarding of service requests. Each node periodically advertises information about its own services and cached ones (i.e., group identifier of services advertised by other nodes) to a pre-configured hop neighbors. The caches contribute to discover services efficiently; service requests are not broadcasted but selectively forwarded to nodes hosting requested services so that it helps to reduce network bandwidth usage.

In GSD, a service belongs to a hierarchy of groups. Each node supplements a group information of service in advertisement and discovery messages, thus the selective forwarding mechanism makes use the group information to selectively deliver discovery messages. In other words, the group information is used as a application-layer routing metric.

2.6.8 Distributed Directory Service (DDS)

The Distributed Directory Service (DDS) [29, 30] is a yellow pages service designed in the CVIS project [3]. The DDS provides two discovery mechanisms, the one is an active *search* (request-reply style discovery) and the other is a passive *publish/subscribe* style discovery. The communication endpoint of a service is described using URI. Both discovery mechanisms rely on broadcast and multicast, in particular, the *search* mechanism uses mDNS and DNS-SD.

2.7 Wide-area resource discovery

This section presents existing wide-area resource discovery techniques. Not only service discovery protocols but also distributed search mechanisms will be listed to study existing solutions from broader point of view.

2.7.1 SLP using Directory Agent

SLP supports not only distributed, link-local scope service discovery but also centralized, wide-area discovery. Using Directory Agents (DAs), SLP is able to be operated in larger networks. DA is a centralized directory for services: if a DA is present in a considered network, UAs unicast **Service Request** only to the DA instead of multicasting the request to SAs. In this case SAs also unicast **Service Registration** to the DA to store the descriptions of their available services.

2.7.2 Mesh-enhanced SLP (mSLP)

Mesh-enhanced Service Location Protocol (mSLP) [82] is an extension of SLP that enables SLP to support a scope-based fully-meshed peering Directory Agents (namely MDA: Mesh-enhanced DA) to improve the reliability and consistency. mSLP specifies that multiple DAs can share service registrations of a particular scope. MDAs serving the same scope construct fully-meshed relationship.

mSLP specification improves SLP's consistency; each SA just needs to register its service to one of DAs supporting the corresponding scope, then MDAs propagate the information. SAs do not need to re-discover and re-register their service to newly-added DAs.

2.7.3 Wide area Bonjour

Wide Area Bonjour [4] is an extended usage of DNS-SD to support wide-area service discovery in combination with DNS update [79] and traditional, centralized unicast DNS. Thanks to the DNS update protocol, a service provider is able to register its services as DNS resource record to a DNS server in its domain. Even in other domains, service consumers (DNS clients) can try to locate the service using the traditional unicast DNS mechanism.

In combination with mDNS, it is able to support not only local-scope point-to-multipoint discovery but also wide-area point-to-point service discovery, for instance, at first service consumers try to discover a certain service in the same link with mDNS, then, as a next step, they can request a DNS server in the Interned for the service.

2.7.4 Universal Description Discovery and Integration (UDDI)

Universal Description Discovery and Integration (UDDI) [13] specified by OASIS is a XML-based distributed registry to discover Web services. A UDDI registry is composed of one or multiple UDDI nodes, which manage a particular set of UDDI data: XML data elements describing Web services. UDDI uses SOAP to publish, discover, and manage UDDI data.

Web services provide their descriptions to a UDDI registry, while clients look up the registry to use the service. These communications are performed using SOAP. UDDI registries are considered to be in well-known locations.

2.7.5 Lightweight Directory Access Protocol (LDAP)

The Lightweight Directory Access Protocol (LDAP) [81, 72] is an Internet protocol for accessing distributed directory services that act in accordance with X.500 data and service models [48]. LDAP is a lightweight variant of the Directory Access Protocol in X.500. To discover a certain directory entry, LDAP clients communicate with LDAP servers using TCP. It supports TLS and also SSL. LDAP data are accessed using LDAP URL, a URI for a server implementing LDAP (e.g., `ldap://` or `ldaps://`) [73].

2.7.6 Content sharing systems based on structured overlay network

The structured overlay network is a kind of peer-to-peer (P2P) system that enables a number of nodes to manage arbitrary resources according to a certain globally consistent algorithm. Even if there are huge number of nodes, many P2P systems built on top of the structured overlay network can discover a certain resource within $O(logN)$ hops (where N means the number of nodes in the system) without relying on broadcasting.

One prominent example is Distributed Hash Table (DHT). In DHT algorithms (e.g., CAN [70], Chord [74], Kademlia [60], Pastry [71]), each resource is stored by one of nodes as a set of key-value pairs, in which the key is a hashed identifier of the value, and the value is actual data (e.g., a file). Regardless of the number of nodes, DHT can locate a node that has a certain key-value pair using its key within $O(logN)$ hops. Clients can determine the key from the identifier of value (e.g., filename) by hash function, such as MD5, SHA-1. DHT has been used as a building block of distributed resource sharing systems.

2.7.7 INS/Twine

INS/Twine [19] is a DHT-based naming and discovery system using Chord. It is an enhancement of Intentional Naming System (INS) [16]. INS is a service discovery protocol for dynamic and mobile networks, in which Intentional Name Resolvers (INRs) compose an application layer overlay network relying on traditional IP unicast. Clients and service providers attach to one of INRs, and service information advertised by the providers are distributed among INRs without specific configurations (e.g., assignment of data to INRs). INR leaves the underlying networking protocol stack (IP unicast), thus it can be integrated any types of TCP/IP-based networks.

In addition to the traditional address-based routing (namely *early-binding* in [19]), INS performs the content-based routing in the application layer (*late binding*): messages are routed according to the name instead of network address on the INR overlay. Two types of message delivery mechanism are supported, *intentional anycast* and *intentional multicast*. The former is an application-layer anycast to deliver messages to a node hosting an *optimal* service (e.g., a least-loaded printer). On the other hand, the latter is used to send messages to all nodes hosting a certain type of service that satisfies clients' criteria. From the network layer point of view, unicast is used to perform these routing.

While INS composes an RTT-based spanning-tree INR overlay, INS/Twine integrates a DHT (Chord) to it to improve INS's performance and scalability.

2.7.8 Splendor

Splendor [83] is a location aware mobile service discovery model taking into account security and privacy. The main targets of Splendor are handheld and wearable computers with several access technologies, such as 3G, 802.11, etc. Splendor considers untrusted environment: participants (clients, services, and network infrastructure) are not trusted by each other. Splendor defines the following four types of components: clients, services, directories, and proxies. The proxy is used to help authentication, to protect privacy for service providers and to offload mobile services' computational work: proxies perform service lookup on behalf of clients, register descriptions of services on behalf of service providers so that services' and clients' mobility is abstracted from other components.

To initiate communications among clients, services, and directories, Splendor relies on multicast. Each component knows multicast addresses a priori and multicast to discover a point of attachment in Splendor. After connecting one of nodes, each component communicates with each other using unicast. If a mobile service changed its location, (e.g., changes its IP address), it can lookup a new directory to re-register its services.

Although Splendor adopts a centralized architecture, the proxy enables to support mobility of clients and services while preserving security and privacy.

2.8 Classification of resource discovery solutions

This section classifies the previously mentioned resource discovery techniques into a number of frameworks by introducing an abstraction model, and then discusses issues and potential solutions. Although a number of studies of resource discovery have already classified existing solutions [18, 78, 61, 17], this research concentrates on the communication among resource discovery components.

2.8.1 System components and functions

First of all, three system components and four main functions of resource discovery are defined, shown in Figure 2.6. This distinction is conceptual; each component may be installed on the same node.

System components and data elements

- **Resource Consumer (RC)** an entity trying to discover and consuming resources.
- **Resource Provider (RP)** an entity providing resources to RCs.
- **Resource Registry (RR)** an optional entity aggregating resource descriptions registered from RPs.

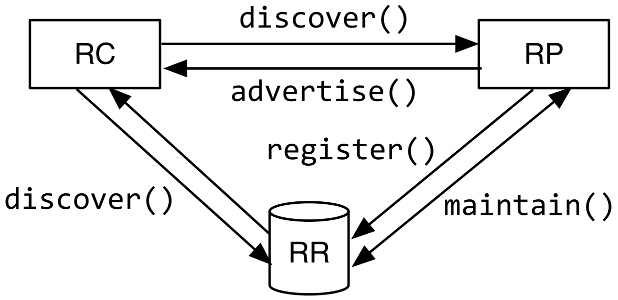


Figure 2.6: Resource discovery components and functions.

- **Resource description** a set of data elements representing a set of characteristics of a resource. Basically it contains a resource identifier, a list of communication endpoints of RP (e.g., IP address and port number), and optional attributes.
- **Discovery criteria** a set of data elements used to specify a requested resource, delivered from RCs to RPs and/or RRs. Basically it contains a resource identifier and optional attributes.

Functions

- **discover** a function issued by RCs with a discovery criteria to look up one or several resources in RPs and/or RRs. If RPs and/or RRs host appropriate resources, they return a list of resource descriptions to the RC.
- **register** a function issued by RPs to RRs with one or several resource descriptions.
- **advertise** a function issued by RPs with one or several resource descriptions. This function can be considered as a un-solicited resource registration from RPs to RCs and/or RRs.
- **maintenance** a function issued by RPs and RRs to manage registered resource descriptions to keep them be up-to-date.

2.8.2 Discovery scenarios

According to the applications' use cases, resource discovery scenarios are defined using *discovery mode* and *underlying network*. The discovery scenario shows which type of RC can try to discover resources in considered networks.

The *discovery mode* represents the type of RC and RP:

- **Mobile-to-Mobile discovery (MM):** a RC in a mobile node tries to discover resources hosted in other mobile nodes. Mostly the RC locates RPs according to geographical position. This mode can be used in any type of underlying networks.
- **Mobile-to-Static discovery (MS):** a RC in a mobile node tries to discover resources hosted in static nodes connected to a mobile network. The RC mostly locates RPs according to geographical position. The infrastructure-less mobile network does not support this mode.
- **Static-to-Mobile discovery (SM):** a RC in a static node connected to a mobile network tries to discover resources hosted in mobile nodes. Mostly the RC locates resource providers according to geographical position. The infrastructure-less mobile network does not support this mode.
- **Mobile-to-Internet discovery (MI):** a RC in a mobile node tries to discover resources hosted in the Internet hosts. As the resource resides in the Internet, the geographical scope is not supported as discovery criteria. The infrastructure-less mobile network does not support this mode.
- **Internet-to-Mobile discovery (IM):** a RC in the Internet host tries to discover resources hosted in mobile nodes. The RC may locate RPs according to geographical position. The infrastructure-less mobile network does not support this mode.

The *underlying network* shows the type of network where RCs and RPs are connected to. In other words, the source and destination networks where resource discovery procedures are taking place. In this research, the following three types of networks are defined: (i) infrastructure-less VANET, (ii) infrastructure-based VANET with Internet gateway, and (iii) infrastructure-based mobile network, such as a public access network or a cellular network. The following five pairs of underlying networks are therefore defined:

- **In-VANET:** a resource discovery in single VANET. Each RC tries to discover resources inside a VANET.
- **Inter-VANETs:** a resource discovery in interconnected VANETs. Each RC in each VANET tries to discover resources in other VANETs.
- **VANET from/to Internet:** a resource discovery between an infrastructure-based VANET and the Internet. Each RC in VANET/Internet tries to discover resources inside the Internet/other infrastructure-based VANETs.
- **In-public network:** a resource discovery in single public access network. Each RC tries to discover resources inside a public network. It can be performed via infrastructure nodes since some public access network does not support direct peer-to-peer communication.
- **Inter-public networks:** a resource discovery between public networks. Each RC in a public network tries to discover resources in other public networks.
- **Public from/to Internet:** a resource discovery between a public network and the Internet. Each RC in a Public network/Internet tries to discover resources in the Internet/other public networks.
- **VANET from/to public network:** a resource discovery between an infrastructure-based VANET and a public network. Each RC in a Public network/VANET tries to discover resources inside VANET/other public networks.

Consequently, 16 discovery scenarios are defined, as listed in Table 2.2. This table shows which types of *discovery modes* can be taking place in which types of *underlying networks*. For instance, the Mobile-to-Mobile discovery (shown as MM: a mobile RC discovers mobile RPs) cannot be taken place if RC/RPs are either under the *VANET from/to Internet* or *Public/Internet* situation. Note that the resource discovery among static nodes is out of focus of this research.

Table 2.2: Resource discovery scenarios. *Yes* means the resource discovery procedure can be taken place, whereas *No* means it is never invoked.

Discovery mode	Underlying network						
	In-VANET	Inter-VANETs	VANET/ Internet	In-Public	Inter-public	Public/ Internet	VANET/ Public
MM	Yes	Yes	No	Yes	Yes	No	Yes
MS	Yes	Yes	No	Yes	Yes	No	Yes
SM	Yes	Yes	No	Yes	Yes	No	Yes
IM	No	No	Yes	No	No	Yes	No
MI	No	No	Yes	No	No	Yes	No

2.8.3 Abstraction model

The abstraction model of resource discovery is defined according to*architecture*, *transmission mode*, *discovery scope*, and *discovery pattern*.

Architecture

- **Distributed** a RR-less architecture where RCs and RPs directly communicates with each other. RPs do not perform the *register* function, instead they may *advertise* resource descriptions.
- **Centralized** a RR-based architecture where each RP issues *register* available resources to a RR. Each RC issues *discover* to the RR instead of RPs.
- **Hybrid** a harmonization of the distributed architecture and centralized architecture. The structure of the architecture is the same as the centralized one, but RCs can communicate with either RPs or the RR depending on situations.
- **Clustered** a harmonization of the distributed architecture and centralized architecture. A number of nodes compose a distributed RR where each RR exchanges resource descriptions with other RRs as the distributed architecture. RPs and RCs communicate with one of nodes acting as the distributed RR.

Transmission mode

The transmission mode represents a set of messaging styles for the above-mentioned four functions: discovery messaging (DM), registration messaging (RM), advertisement messaging (AM), and maintenance messaging (MM).

- **Broadcast** all-node broadcast with which each component does not identify a specific node as a destination. The messages are propagated to every node inside a considered network.
- **Unicast** one-to-one communication among each component.
- **Multicast** selective one-to-multiple communication according to topological location, in which each component specifies a certain multicast group address.
- **Geocast** one-to-one or one-to-multiple communication according to geographical location. Each component delivers messages to one or multiple nodes inside a certain geographical area.

Scope

The scope represents the extent of the area in which resources are discovered.

- **Local** a small scope discovery including link local (single hop) scope and MANET/-VANET local (multi hop) performed inside a single network. The local scope resource discovery does not support to discover resources in other networks.
- **Global** a global scope discovery. Each component is able to discover resources in any type of network regardless of underlying network topology.

Discovery pattern

The discovery pattern shows how RCs obtain information of resources from RPs and/or RRs.

- **Pull** a request-reply style communication in which RCs deliver discovery messages to RPs and/or RRs as required. RPs and/or RRs reply to received messages only when they host a requested resource.

- **Push** in this pattern RPs and/or RRs advertise their resources to RCs without any request. RPs and/or RRs may advertise resources either periodically or triggered by a certain event.

2.8.4 Resource discovery frameworks

This clause defines possible resource discovery frameworks according to the above mentioned abstraction model, listed in Table 2.3 and 2.4. The following frameworks are mainly categorized by architecture, such as distributed, centralized, hybrid and clustered frameworks.

Distributed frameworks

A distributed framework is built on the distributed architecture. Depending on the supported scope, several sub-frameworks are defined. One of the benefits of this framework is fault tolerance because of its RR-less architecture, in addition, the RR-less architecture is suitable for mobile environments from the point of view of consistency of resource descriptions. However, this framework may suffer from scalability issues in large scale networks if it relies on one-to-multiple communication protocols. Furthermore some networks may not support this framework because of the one-to-multiple communication protocols. This category includes the following five types of sub-frameworks:

- **Topological broadcast framework**: RCs and RPs broadcast messages to all nodes in single hop distance without any routing mechanisms. The benefit of this framework is simplicity, however it is basically cost-intensive.
- **Topological flooding framework**: RCs and RPs propagate messages to a certain hop neighbors using some sort of flooding mechanisms in a MANET or VANET.
- **Topological multicast framework**: RCs and RPs send multicast messages to a subset of nodes in a considered network. Destinations are specified by multicast group addresses. Basically this framework is able to be applied to one administrative domain because of necessity of multicast routing protocol.

- **Geographical broadcast framework**: RCs and RPs deliver messages to all nodes inside a certain Geographical area using Geographical routing mechanisms. Destination RCs and RPs are specified according to their geographical location instead of topological one. This framework is suitable for location-based resource discovery in MANET and VANET in terms of bandwidth usage since unnecessary messages can be reduced compared to the topological broadcast.
- **Pure P2P framework**: this framework composes an application layer overlay network. RCs and RPs send messages using dedicated application layer routing protocols mostly relying on the UDP unicast. The benefit of this framework is that it is able to be applied to any type of networks/scopes regardless of underlying network topology. However, the application layer multi-hop routing may incur more bandwidth and latency compared to traditional network layer-only routing.

Centralized framework

A client-server style framework relying on one RR acting as a index server. The RR is used by RCs in single administrative domain or all RCs connected to the Internet. The communication endpoint of the RR is configured in RCs and RRs a priori, otherwise they may resolve it using an arbitrary mechanism, such as the DHCP option. The network architecture of this framework is relatively stable than other ones since the RR is hosted in a static node. One drawback in this framework is that the RR is single point of failure. Also the RR may be easily overloaded.

Hybrid framework

The hybrid framework relaxes the shortcomings in distributed and centralized frameworks by switching its behavior between distributed and centralized manner. Each RC and RP basically communicates with a RR as the centralized framework. When RCs need to discover a service within a subset of nodes in a considered network (or the RR is not reachable), RCs may directly communicate with RPs as the distributed frameworks. RCs and RPs must support an algorithm to switch these two frameworks.

As of now, most solutions of this framework are identified as a combination of the topological multicast and centralized framework.

Clustered framework

In the clustered framework, multiple nodes compose a group of distributed RRs. Each RR behaves as a proxy server for attached RCs and RPs, thus RCs and RPs only need to communicate with one of RRs to discover and register resources. Each RR compose a kind of overlay network, therefore messages received by one of RRs from attached RCs and RPs are propagated to the other RRs. The structure of the RR network may be flat (ring or meshed), tree, etc. This framework has more advantage than other ones in terms of scalability while it requires an efficient messaging mechanism to compose and maintain the RR network. This framework is identified as the following three sub-frameworks:

- **Mirrored cluster framework**: a framework relying on multiple RRs that share all information with each other. Resource descriptions sent from RPs are propagated all RRs, in other words, all the RRs share the same knowledge about available resources. Only RCs and RPs need to do is to attach one of RRs and communicate with it.
- **Hierarchical cluster framework**: a framework relying on multiple RRs composing a tree-like hierarchical network. Resource descriptions are distributed to RRs according to a certain criteria, such as organization, resource type, etc. Each RR manages one or multiple group of resource descriptions. Only RCs and RPs need to do is to attach one of RRs and communicate with it.
- **P2P cluster framework**: a variant of P2P frameworks relying on multiple RRs connected as an overlay network. RRs compose a virtual registry using a certain application layer overlay routing algorithm so that registered resources and discovery messages are delivered to a particular RR managing a corresponding information. Only RCs and RPs need to do is to attach one of RRs and communicate with it.

Table 2.3: List of frameworks (1)

		Topological broadcast	Topological multicast	Topological flooding	Geographical broadcast	Pure P2P
Architecture		Distributed	Distributed	Distributed	Distributed	Distributed
DM		Broadcast	Multicast	Flooding	Geo.Bcast	Unicast
RM		None	None	None	None	None
MM		None	None	None	None	Unicast
AM		Broadcast	Multicast	Flooding	Geo.Bcast	Unicast
Scope	Local	Y	Y	Y	Y	Y
	Global	N	N	N	Y	Y
Pattern	Active	Y	Y	Y	Y	Y
	Passive	Y	Y	Y	Y	Y
Additional function			Geo-routing	Ad-hoc routing	Multicast routing	Overlay routing

2.8.5 Taxonomy of resource discovery protocols

Table 2.5 and 2.6 list existing resource discovery solutions according to the above-mentioned discovery scenarios using the frameworks. They show the applicability of exiting solutions to the potential discovery scenarios in accordance with the generalized frameworks.

As shown in these Tables, basically, the cluster-based, hybrid, or P2P framework can be used to support all the discovery scenarios. Nonetheless, each potential solution has several issues. Next Section discusses the issues and solution.

2.9 Problem statement and potential strategy

Although existing resource discovery techniques support some of the discovery scenarios, to satisfy the requirements for wide-area geographical resource discovery mentioned in Chapter 1, several issues are raised. This section describes these issues with respect to each requirement.

Table 2.4: List of frameworks (2)

		Centralized	Hybrid	Mirrored cluster	Hierarchical cluster	P2P cluster
Architecture		Centralized	Hybrid	Cluster	Cluster	Cluster
DM		Unicast	Multicast, unicast	Unicast	Unicast	Unicast
RM		Unicast	Unicast	Unicast	Unicast	Unicast
MM		None	None	Unicast, multicast	Unicast	Unicast
AM		None	Multicast	Multicast	None	Unicast
Scope	Local	Y	Y	Y	Y	Y
	Global	Y	Y	Y	Y	Y
Pattern	Pull	Y	Y	Y	Y	Y
	Push	N	Y	Y	N	N
Additional function			Mode selection	Synchronization mechanism	Tree construction	Overlay routing

2.9.1 Discovery scope determination

As listed in Table 2.2, a number of discovery scenarios have been defined depending on the underlying network and discovery mode, however, applications cannot specify an appropriate scope of discovery for each scenario a priori; only they can do is to execute the *discovery* function by specifying the type and attributes of resources. Consequently RCs may send discovery messages to too large or too small number of nodes, including unreachable nodes (e.g., inside an infrastructure-less VANET, a global scoped communication does not make sense). Although wider scope solutions relying on the centralized or clustered frameworks are able to handle smaller scopes, it may takes larger bandwidth usage and latency. This issue is specifically significant in VCS, in which there is a number of VANETs. These VANETs may be partitioned according to region (e.g., city, country, continent), and density of vehicles. Therefore, even if the control overhead is negligible, the traditional broadcast or multicast-based solutions cannot handle VCS since point-to-multipoint communications designed for VANETs can basically not go out to the other types of networks.

One reason for this issue is that existing solutions are designed for a specific scope

Table 2.5: Taxonomy of resource discovery protocols (1)

Proposal	Framework	Supported discovery scenario										
		In-Mobile			Inter-Mobile			Mobile-Internet				
		MM	MS	SM	MM	MS	SM	MM	MS	SM	IM	MI
NDP [62]	Topological broadcast	Y	Y	Y	N	N	N	N	N	N	N	N
SLP [46, 44]	Topological multicast	Y	Y	Y	N	N	N	N	N	N	N	N
SLP+DA [46, 44]	Hybrid	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
UPnP [77]	Topological broadcast	Y	Y	Y	N	N	N	N	N	N	N	N
DNS	Hierarchical cluster	N	N	N	N	N	N	Y	Y	Y	Y	Y
mDNS [10, 4]	Topological multicast	Y	Y	Y	N	N	N	N	N	N	N	N
Wide-Area Bonjour	Hybrid + Hierarchical cluster	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
DHT	Pure P2P	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
LDAP [72]	Hierarchical cluster	N	N	N	N	N	N	Y	Y	Y	Y	Y
INS/Twine [19]	P2P cluster	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
GSD [25]	Topological flooding	Y	Y	Y	N	N	N	N	N	N	N	N
Konark [57]	Topological flooding	Y	Y	Y	N	N	N	N	N	N	N	N
mSLP [82]	Mirrored cluster	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
CAM [37]	Topological broadcast	Y	Y	Y	N	N	N	N	N	N	N	N

Table 2.6: Taxonomy of resource discovery protocols (2)

Proposal	Framework	Supported discovery scenario							
		Inter-public			Public/Internet		Public/Mobile		
		MM	MS	SM	MI	IM	MM	MS	SM
NDP [62]	Topological broadcast	N	N	N	N	N	N	N	N
SLP [46, 44]	Topological multicast	N	N	N	N	N	N	N	N
SLP+DA [46, 44]	Hybrid	N	N	N	N	N	N	N	N
UPnP [77]	Topological broadcast	N	N	N	N	N	N	N	N
DNS	Hierarchical cluster	Y	Y	Y	Y	Y	Y	Y	Y
mDNS [10, 4]	Topological multicast	N	N	N	N	N	N	N	N
Wide-Area Bonjour	Hybrid + Hierarchical cluster	N	N	N	N	N	N	N	N
DHT	Pure P2P	Y	Y	Y	Y	Y	Y	Y	Y
LDAP [72]	Hierarchical cluster	Y	Y	Y	Y	Y	Y	Y	Y
INS/Twine [19]	P2P cluster	Y	Y	Y	Y	Y	Y	Y	Y
GSD [25]	Topological flooding	N	N	N	N	N	N	N	N
Konark [57]	Topological flooding	N	N	N	N	N	N	N	N
mSLP [82]	Mirrored cluster	N	N	N	N	N	N	N	N
CAM [37]	Topological broadcast	N	N	N	N	N	N	N	N

of discovery. To realize a scalable and efficient resource discovery, resource discovery mechanisms must support multiple scopes of discovery and dynamically select an appropriate scope among them. With respect to this context, a possible solution is to rely on the hybrid framework. However, existing solutions built on the hybrid framework do not support the dynamic scope selection. In addition, this framework suffers from the issues of single point of failure due to the centralized RR. Consequently, a potential solution is to integrate a scope selection mechanism to one of the clustered frameworks.

2.9.2 Geographical location-based discovery

Even though *topological* location based discovery is supported in some of the existing solutions, *geographical* location based discovery is not yet natively supported. For mobile applications, geographical location is much more important than topological location since resource consumers try to discover resources according to their geographical position. One solution for the geographical location based discovery using existing mechanisms is the application-layer-only solution: RPs add their current geographical position as one of the attributes of resource description while RCs include the geographical position of necessary resource to discovery criteria. However, from the point of view of the network layer, this solution may propagate discovery packets to an unnecessarily large numbers of nodes because it relies only on the topological address-based routing. Multicast based solutions (e.g., SLP, mDNS) cannot solve this issue since they assign multicast groups acceding to the type of resources instead of geographical position.

Consequently, two approaches for this issue are identified: (i) organizing geographically separated distributed RRs in the hierarchical or P2P clustered frameworks, and (ii) exploiting a network-layer geographical routing mechanism.

2.9.3 Scalability

Existing solutions are not capable of supporting the above mentioned 1 billion of mobile nodes. By nature, the distributed frameworks are designed for small size of networks or single administrative domain, while the RR in the centralized framework cannot handle so many resources due to the single centralized RR. The clustered frameworks may support such number of nodes by installing a large number of distributed RRs, however they suffer from too many numbers of updates of resources from mobile RPs: as described in Section 2.9.2, the resource discovery system needs to support geographical location based discovery, in which RRs need to manage the geographical position of each RP. Consequently, RPs must send location update messages frequently to RRs to maintain consistently their geographical location.

2.9.4 Latency

C-ITS applications have tight latency requirements, described in Section 2.4, however, most resource discovery mechanisms do not satisfy these requirements. The main reason is that these mechanisms rely only on single or distributed RRs.

To meet the latency requirements, the resource discovery system must reduce unnecessary communications and localize its communication to exchange messages quickly. In addition, it is necessary to exploit the push style resource discovery in which RRs and RPs passively push resource descriptions to RCs. One of the benefits of the push style discovery is that it can efficiently notify RCs with the update of previously-discovered resources, such as, geographical position, availability, etc.

Chapter 3

Use case evaluation: integration of an in-vehicle driver support application into actual vehicles

This chapter explores a set of use cases of C-ITS applications to realize practical communication scenarios of C-ITS. In this chapter, some use cases are defined from the point of view of C-ITS as part of ITS, one of the most important and emerging system for recent years. Not only describing the practical use cases of C-ITS applications, this chapter also describes an evaluation of a pair of applications is implemented using the IPv6 GeoNetworking technology. Further, these applications are integrated into actual vehicles in the IMARA project team at INRIA. The applications are publicly evaluated in outdoor tests, which have been conducted as the final demonstration of the GeoNet project, an European international ITS project.

3.1 Background and motivation

To realize *what is a C-ITS application* in detail, it is necessary to study practical use cases of such applications in compliance with several ongoing efforts in addition to the basic use cases described in Chapter 2.3. This chapter therefore focuses on use cases of C-ITS applications as one of the most important distributed system in recent years. Specifically, road safety and traffic efficiency use cases are shown in this chapter, because these scenarios are primal usages of C-ITS applications. To discuss C-ITS applications in compliance with ongoing efforts, this chapter follows the series of C-ITS specifications published by international standard organizations, such as ISO [9] and ETSI [5].

Applications presented in this chapter are a pair of mobile applications aggregating and propagating messages from a node to a group of other nodes inside a particular geographical area. They are designed to evaluate some essential usages of C-ITS applications: providing road traffic information with vehicles for road safety and efficient drive in combination with in-vehicle Human Machine Interfaces (HMI). In addition, the applications show a solution of the harmonization of geographical addressing and routing mechanisms and traditional the IPv6 routing by means of *IPv6 GeoNetworking* initially developed in the GeoNet project, an European project started from 2008 and finished in 2010 [6], mentioned in Chapter 2.5.

3.2 In-vehicle driver support application using IPv6 GeoNetworking

As shown in Section 2.3, road safety and traffic efficiency are the main objectives of C-ITS applications. Use cases of these applications are basically considered in relatively small area, for instance, road safety applications are designed for single or a few hops distance in an intersection, a fleet of vehicles, etc [33]. Most of these use cases are able to be implemented using non-IP single hop broadcast or a few hops flooding. However, in practical scenarios, they may not always be achieved due to heterogeneous environments. In the case of propagating traffic hazard information, *receiver vehicles* may not be reachable from *sender vehicles* with the broadcasting or flooding depending on the capability of their access interface, distance between a sender and receivers. To handle such a situation, a practical use case exploiting Internet-based communication should be taken into account.

As practical usages of mobile applications, this section therefore presents two advanced use cases intended to support drivers using the Internet-based communication and IPv6 GeoNetworking, shown in Figure 3.1 and 3.2. The first one is *Road traffic event detection and dissemination*; when a vehicle detects a particular road traffic event (e.g., accident, approaching emergency vehicle), it reports the description of the event

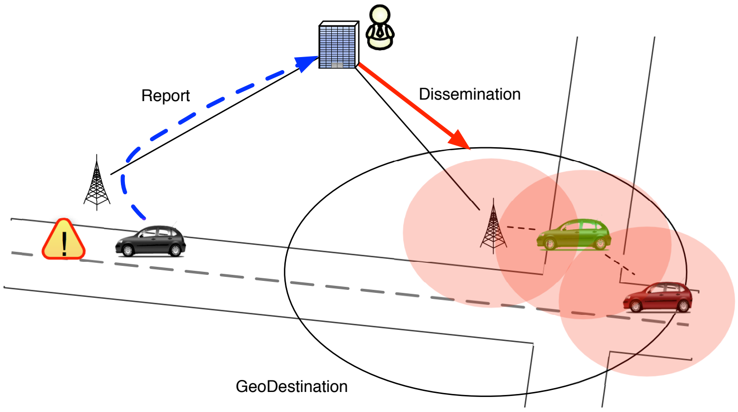


Figure 3.1: Road traffic event detection and dissemination

to an ITS center. A host in the ITS center shows the occurrence of the event to ITS operators and informs a set of vehicles inside a certain *relevant* area of the event. The second is *Road traffic information dissemination from ITS center*; an ITS operator in an ITS center informs about a particular road traffic event (e.g., road congestion, approaching ambulance) to relevant vehicles, which reside within a relevant area of the event.

3.2.1 Functional requirements

Dissemination messages must be delivered to vehicles within a relevant geographical area (namely *GeoDestination*), an appropriate range of roads through which receivers approach the point of event. The size of the area is configured depending on the type of event, driving speed, etc. For example, in urban area composed of narrow streets, it is assumed that the driving speed is relatively slow and it may regularly be congested. In such a situation GeoDestination can be small to reduce bandwidth usage, and vice versa.

Three types of functions are needed to realize these use cases: (i) detecting and reporting road traffic events (*Report*), (ii) aggregating and disseminating the road traffic event to relevant vehicles (*Disseminate*), and (iii) receiving messages from the Disseminator and notifying drivers of the event (*Receive and notify*). The *receive* and *report* functions must be integrated into vehicles, on the other hand, the *disseminate* function

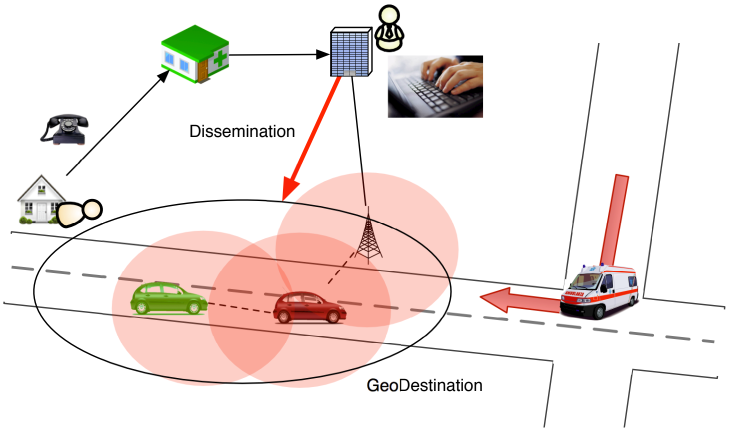


Figure 3.2: Road traffic information dissemination

must be in hosts in ITS center.

Roadside infrastructures must provide Internet access to vehicles so that applications in-vehicle and ITS centers communicate with each another. The *report* function needs to know a communication endpoint of host which executes the *disseminate* function. The *report* function is executed by in-vehicle hosts, a part of mobile nodes embedded in vehicles, this communication endpoint must be statically configured, otherwise, the *report* function must have some mechanism to dynamically locate it.

3.2.2 System architecture

To study how to implement C-ITS applications, the following sections actually design and implement a pair of applications supporting the previously described use cases. The application consists of two pair of software components: in-vehicle *clients* and a *server* in an ITS center. The client executes the *report* and *receive* functions, whereas the server performs the *disseminate* function. To deliver messages to a particular geographical area, applications exploit IPv6 GeoNetworking primally developed in the GeoNet project. Thanks to its geographical routing mechanism, messages are delivered to GeoDestination independently of routers' wireless communication range. The applications particularly exploit the GeoBroadcast mechanism that sends packets from a node to all nodes located inside a particular geographical area [76].

From the point of view of communication stack, in-vehicle mobile hosts only sup-

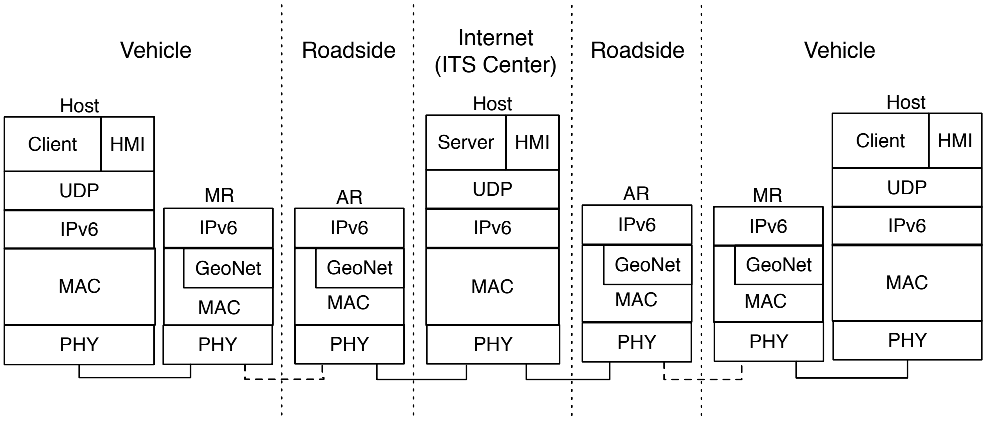


Figure 3.3: Protocol Architecture.

port IPv6, while in-vehicle mobile routers and roadside routers support IPv6 GeoNetworking so that combinational IPv6 packets constructed by applications are delivered to a certain relevant area through routers. In other words, any type of host supporting the combinational TCP/IP stack can be used to integrate the client application. Figure 3.3 shows the overall protocol architecture of the system. System configurations of each node are described as follows:

Vehicle: vehicles are equipped with a MR supporting IPv6 GeoNetworking and a host with the conventional IPv6 stack. MR and host are connected with Ethernet to perform in-vehicle communication. 802.11 WLAN is used for inter-vehicle communication between MRs and other routers. The client application consists of two main components and one support component: *reporter, receiver*, and *HMI*, respectively. The reporter and receiver execute the above mentioned functions; the reporter automatically detects road traffic events and unicasts them to the disseminator with UDP. The receiver receives messages from the disseminator. This dissemination message is a GeoBroadcast packet encapsulated from a IPv6 multicast packet. After receiving the dissemination packet, it notifies events with vehicle drivers though the GUI windows provided by the HMI.

Roadside: the roadside is equipped with ARs supporting both IPv6 GeoNetworking and the conventional IPv6 routing. ARs encapsulate regular IPv6 multicast packets delivered from the Internet and GeoBroadcasts to MRs through their WLAN interface. Even though an AR may not cover a necessary communication range to deliver a Geo-

Broadcast packet, the packet is forwarded by nearby vehicles thanks to the multi-hop geographical routing mechanism of the GeoNetworking protocol.

Center: ITS center contains a host connected to the Internet. The server application is integrated to this host which has one main component and one support component: *disseminator* and *HMI*, respectively. The disseminator performs the *disseminate* function; aggregating information of road traffic events reported by clients, and multicast them to a set of receiver clients inside a relevant area using IPv6 GeoNetworking. The disseminator also informs ITS operators of road traffic events though the GUI window of the HMI.

3.2.3 Communication sequence

As shown in Figure 3.4, the communication sequence of the applications is as follows:

1. ARs and the disseminator establish an IPv6 multicast tunnel. ARs are configured for forwarding IPv6 multicast from the Internet to VANET.

2. Reporter periodically obtains the vehicle's current geographical position from MR's GPS (Position_req, Position_ack) via TCP unicast.

3. Receiver joins a particular global scope IPv6 multicast group by issuing a MLDv2 Multicast Lister Report to the locally-connected MR.

4. In the case of *Road traffic event detection and dissemination*, when a reporter detects a road traffic event, it unicasts descriptions of the event to the disseminator using UDP (Report_event) via a reachable AR.

5. The disseminator receives the report and displays the position and descriptions of the event to ITS operators through HMI.

6. The disseminator periodically sends IPv6 multicast packets to ARs through the multicast tunnel (Propagate_Event) for a certain period of time.

7. ARs receive the IPv6 multicast packets and forwards them via GeoNetworking layer, which generates GeoBroadcast packets. The size of GeoDestination is determined from the IPv6 multicast group address.

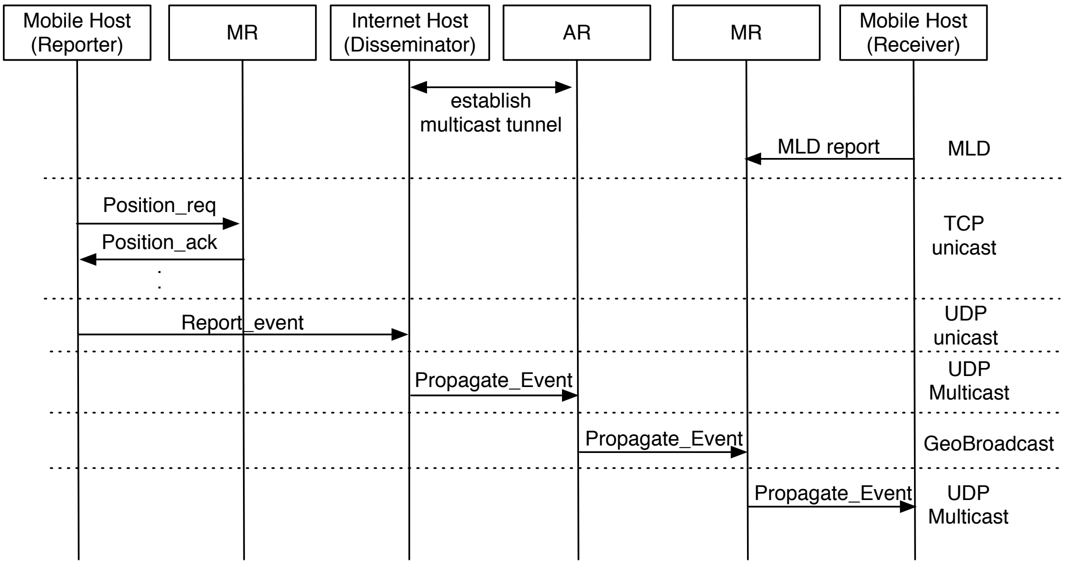


Figure 3.4: Messaging sequence. Report_event is used only in the Road traffic event detection and dissemination scenario.

8. MR's GeoNetworking layer receives GeoBroadcast packets and decapsulate them. Regular IPv6 multicast packets are stripped out of the GeoNetworking header. It passes the IPv6 packets to its IPv6 stack. The IPv6 stack finally sends the packets to receiver through regular Ethernet.

9. Receiver notifies its driver of the reception of messages.

3.3 Implementation

The proposed system was implemented and integrated into the actual vehicle testbed in INRIA rocquencourt, which includes three vehicles and roadside infrastructures. Each vehicle is equipped with a pair of MR and host, while roadside infrastructures are ARs. In addition to the vehicles and roadside infrastructures, a host in a building acting as an ITS center is used. The system is therefore capable of testing multi-hop communication in VANET. MRs and ARs are Alix3d3 embedded PCs equipped with an Ethernet and WLAN card (Atheros AR5414 802.11 a/b/g Rev 01). Ubuntu 9.0.4 (kernel 2.6.29.6) is installed to both of them. Each router is capable of obtaining its

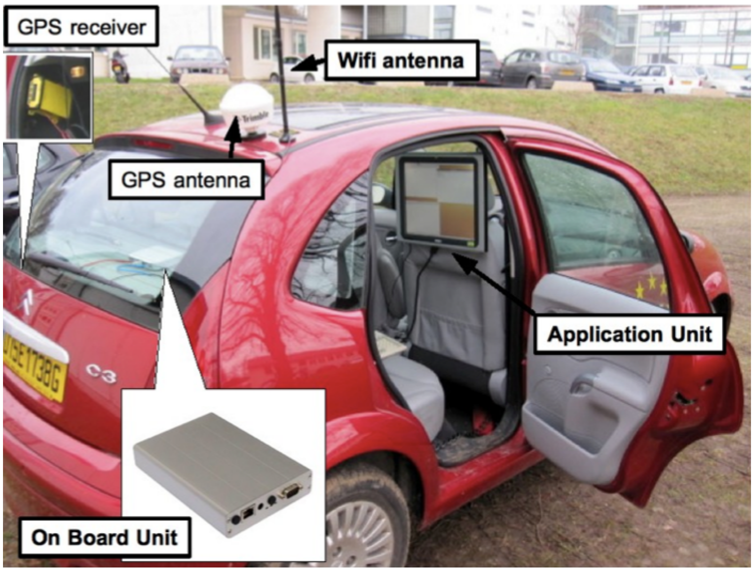


Figure 3.5: Equipments in real vehicle testbed.

current geographical position; a GPS receiver (Trimble AgGPS 323) and gpsd [7] are installed in each MR, while ARs are statically configured with their position. Hosts are conventional PCs. Figure 3.5 shows the equipment integrated into the vehicle.

MRs and ARs support the basic IPv6 routing, IPv6 multicast forwarding, and IPv6 GeoNetworking. At this moment the C2CNet layer developed by the GeoNet project was used as a IPv6 GeoNetworking implementation. To enable the IPv6 multicast forwarding, a simple multicast forwarding daemon was implemented. The IPv6 multicast tunnel between ARs and the ITS center is statically configured. Regarding hosts, although the applications can work on any operating systems as long as IPv6, MLDv2, and Java are supported, Ubuntu 9.0.4 was used for operational reasons. Each in-vehicle host is connected to its MR through Ethernet.

IPv6 GeoNetworking on MRs and ARs is implemented with the TUN virtual network device that communicates with IPv6 and C2CNet. Each router at first receives regular IPv6 packets on its ingress (egress) interface and passes the packets to C2CNet through the virtual device. Subsequent communication is then performed in C2CNet (the GeoNetworking layer). C2CNet determines a type of communication (i.e., GeoUnicast or GeoBroadcast) from the first 8 bits of the destination IPv6 address. Regarding the GeoBroadcast generated from IPv6 multicast, the destination area is mapped to a

Name
Vehicle1

Latitude
48.8377

Longitude
2.100294

Status
Accident detected!
Reporting to Center..

Message from Center

Figure 3.6: Client HMI (reporter).

particular IPv6 multicast address, such as *ff02::1* to a radius of 500m centered from sender. C2CNet in intermediate routers forward the packets. Destination routers receive the packets via their egress (ingress) interface and pass them to ingress (egress) interface through the virtual device as conventional IPv6 packets.

The client application is implemented in JavaSE 6 and JavaFX 1.2, as shown in Figure 3.8. The reporter and receiver components run on JavaSE whereas the HMI components are on JavaFX. The client HMI displays the vehicle's name, current geographical position, detected road traffic events, and disseminated messages, shown in Figure 3.6 and 3.7. The road traffic event detection mechanism is out of focus of this chapter, therefore the event detection is statically performed acceding to the configuration file by specifying the geographical position where the event is supposed to take place.

The server application is implemented in JavaSE 6, JavaFX 1.2 and Google Earth. The disseminator runs on JavaSE and the HMI works on JavaFX with Google Earth. The server HMI has a local http server overlaying reported road traffic information on Google Earth by generating KML sentences, shown in Figure 3.9.


Name Vehicle1

Latitude 48.8377

Longitude 2.099444

Status Fine

Message from Center

Accident

Figure 3.7: Client HMI (receiver).

3.4 Outdoor evaluation

A field demonstration of the proposed system was conducted under the scenarios described in Section 3.2. The main goal of this demonstration is to show effectiveness of integrating the C-ITS application on top of IPv6 GeoNetworking thanks to the following three features: GeoBroadcast generated from IPv6 multicast, Internet-to-Geographical area communication, and multi-hop geographical routing at the C2CNet layer. Following the scenario depicted in Figure 3.1, one of the three vehicles (the forwarder) acts as a stationary vehicle at the boundary of wireless communication range of AR so that it forwards GeoBroadcast packets to GeoDestination. Other two vehicles, which act as the reporter or receiver, go around a pre-specified route. When the reporter vehicle is close to the statically configured road hazard point, it reports the description of the event to the server running on a host in the ITS center via nearest AR. After receiving the report, the server in the center periodically sends the event notification messages (Propagate_event) with IPv6 multicast to a relevant AR. This relevant AR propagates the event to a GeoDestination with GeoBroadcast generated from the IPv6 multicast packets. The receiver vehicle receives the packets as long as it resides within the destination area. Regarding the other scenario, Road traffic information dis-

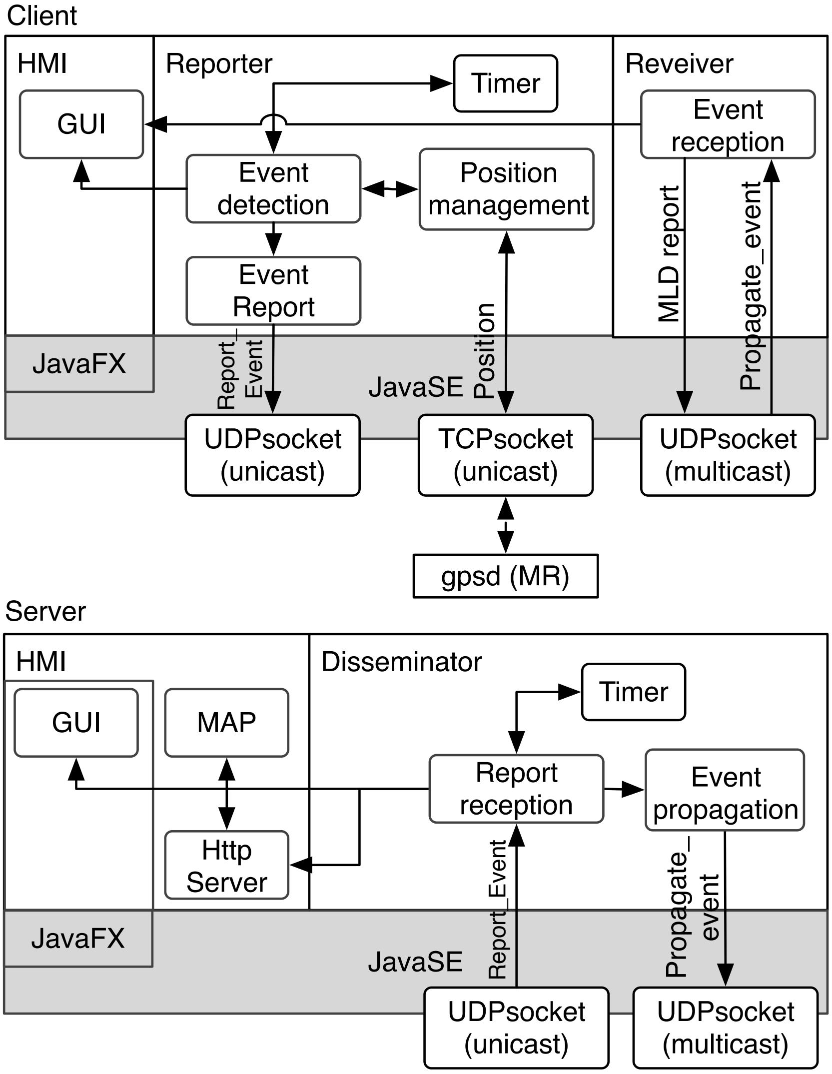


Figure 3.8: Structure of system components.

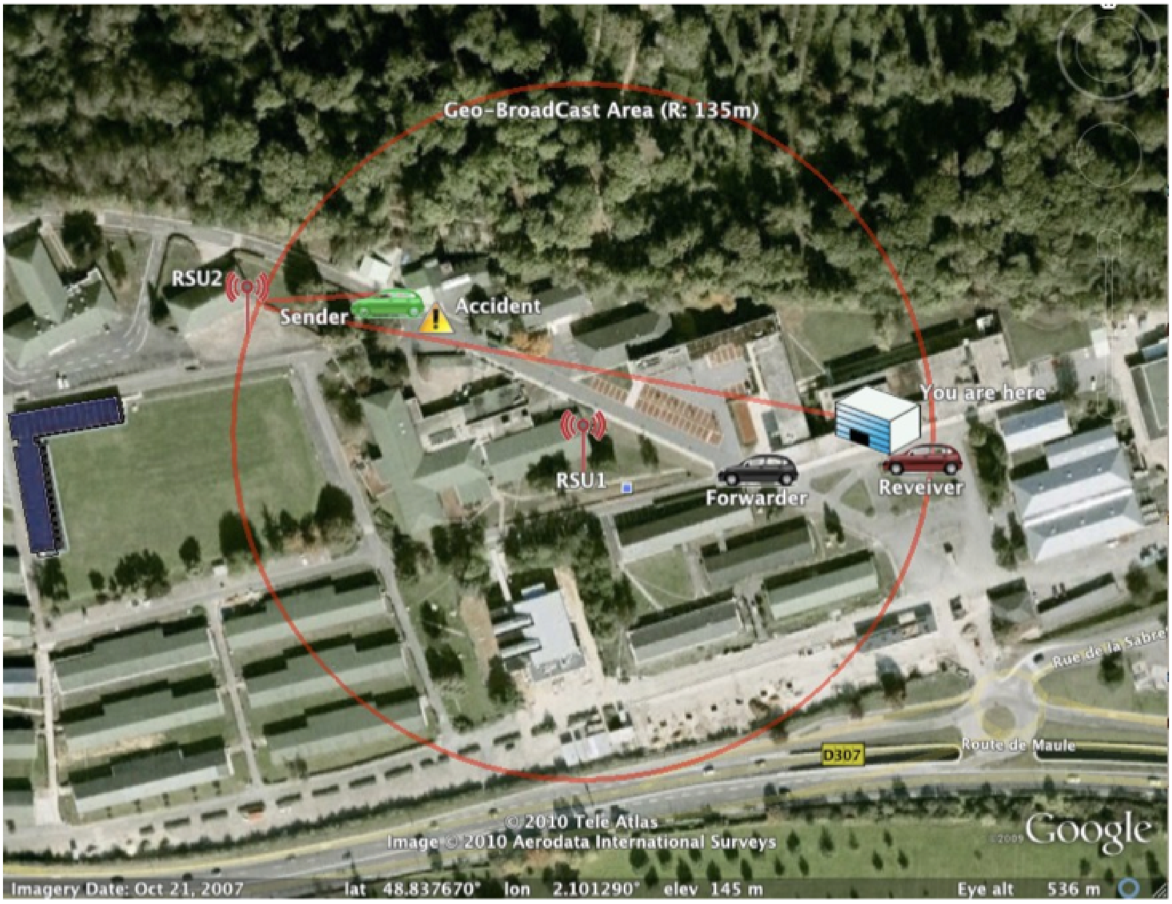


Figure 3.9: Server HMI in ITS center.

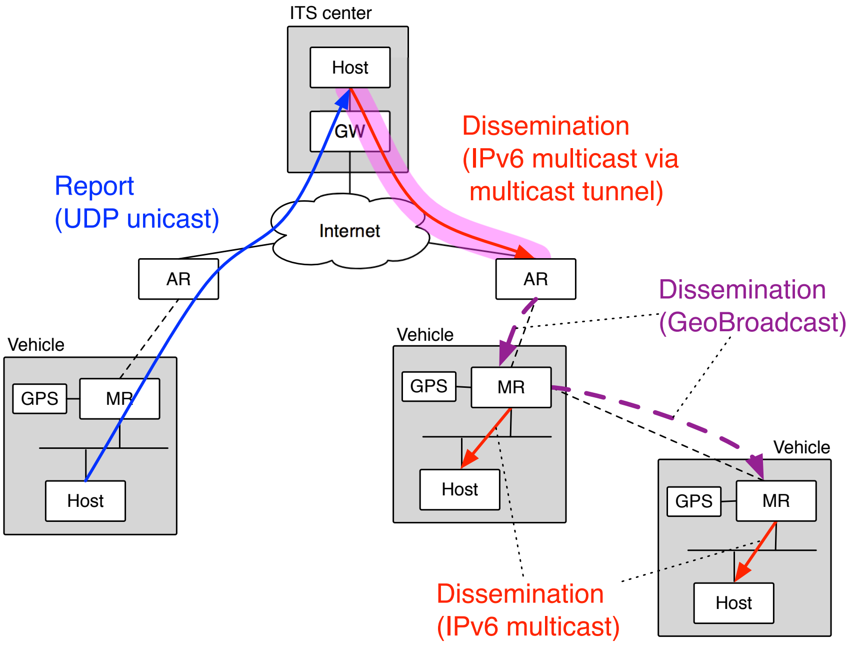


Figure 3.10: Network topology of outdoor evaluation.

semination, the server can also send the traffic information messages (Propagate_info) at any time without report from reporters. Figure 3.10 shows the network topology of the outdoor evaluation.

The system configuration is shown in Table 3.1. The frequency of event propagation, which represents the frequency of the transmission of event propagation messages (Propagate_event), is empirically determined through preliminary tests. It was configured to 10Hz to notify drivers of road traffic events as fast as possible via the client HMI. The GeoBroadcast range is configured to a radius of 150m of the location of AR. One of evaluations showed the average Round Trip Time (RTT) between AR and MR using IPv6 GeoNetworking was about 15ms, shown in Figure 3.11. It means, even though the GeoBroadcast packets are propagated through multi-hop connections, it is delivered within 7.5ms from AR to MR.

The proposed system has publicly been demonstrated at the final workshop of the GeoNet project at INRIA premises (January 2010, Rocquencourt, France) [43].

Table 3.1: System configuration

Entity	Parameter	Configuration
All	Multicast group address	FF0E::2
Host (Server)	Event propagation frequency	10Hz
AR, MR	GeoBroadcast radius	150m
Host (Client)	GPS position update interval	1sec
Host	HMI refresh interval	1sec
Vehicle	Driving speed	< 40km/h

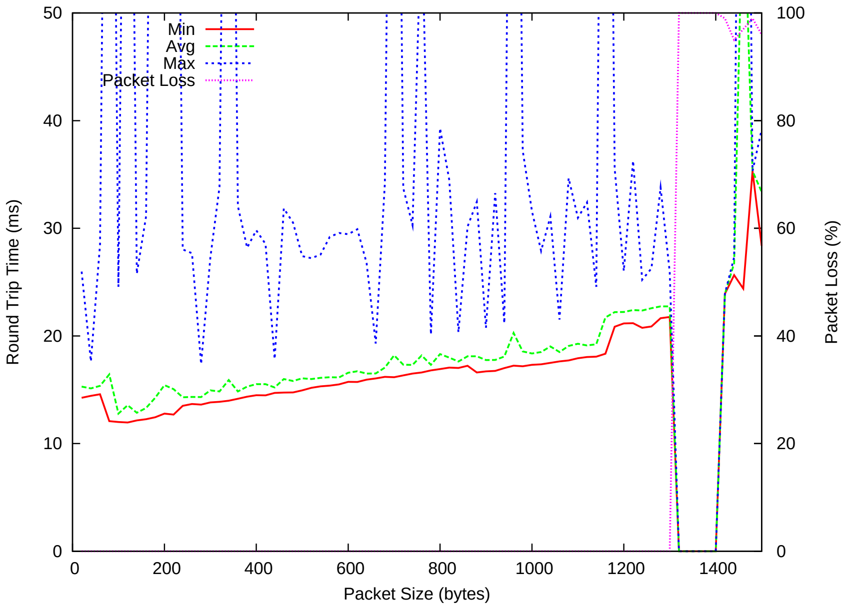


Figure 3.11: Latency between AR and MR (multi-hop).

3.5 Discussion

The implementation and evaluation of the proposed applications revealed that the following capabilities are essential for practical C-ITS applications using IPv6 GeoNetworking:

- **Resource discovery**: for large-scale VCS, messages sent from reporters must be delivered to an appropriate host (e.g., in ITS center) which provides a certain service to a certain geographical area. According to tight latency requirements for C-ITS applications (e.g., from 10ms to 1000ms for road safety [54]), locating such hosts must be done as quickly as possible. It might be configured statically, however the static configuration is inefficient in terms of flexibility and scalability. It is therefore necessary to integrate resource discovery technologies.
- **Topology independent Geocast**: ITS centers need to specify appropriate ARs to disseminate IPv6 multicast packets using GeoBroadcast. One potential solution is to send IPv6 multicast to all ARs, however, it consumes bandwidth in the case of large scale network. In the future it is necessary to investigate how to deliver Geocast packets through the Internet.

3.6 Summary

This chapter explored practical use cases of C-ITS applications and designed a pair of C-ITS applications relying on IPv6 GeoNetworking developed in the GeoNet project. It showed what are practical mobile applications and how to implement them from the context of C-ITS. Furthermore it evaluated the necessity and effectiveness of IPv6 GeoNetworking for C-ITS applications. The applications were implemented using Java as a server in an ITS center and clients in vehicles. They were integrated into the actual vehicles and tested in realistic scenarios as the final demonstration of the GeoNet project at INRIA.

Chapter 4

Geographical mobile resource discovery to support resource discovery locally

Chapter 3 showed the necessity of resource discovery for C-ITS applications and the effectiveness of IPv6 GeoNetworking. Because the goal of this research is to present a wide-area geographical resource discovery system for distributed mobile applications, this chapter first proposes a resource discovery mechanism for C-ITS applications that locates resources according to their geographical position in a mobile network such as a VANET. Specifically, this chapter focuses on a VANET as the most important types of mobile networks for emerging C-ITS. The proposed mechanism exploits an IPv6 multicast-based service discovery protocol on IPv6 GeoNetworking. Thanks to the GeoBroadcast mechanism, it efficiently propagates discovery messages to a subset of nodes inside a relevant geographical area. The proposed mechanism is implemented in ns-3, a network simulator, and is also developed as an actual system using CarGeo6, an open source implementation of IPv6 GeoNetworking, with openSLP. The simulation and field evaluation results shows that the system can discover resources rapidly without incurring bandwidth usage issues.

4.1 Background and motivation

As evaluated in Chapter 3, resource discovery is essential for C-ITS applications to orchestrate necessary resources mainly integrated into VCS. To achieve the goal of this research: presenting *a wide area geographical mobile resource discovery mechanism for C-ITS*, as a first step, it is necessary to design a geographical resource discovery method to support resource discovery locally in a VANET. In general, broadcasting (directly delivering data packets without any preliminary communication) is sufficient for latency demanding use cases within single-hop distances. However, if applications need to communicate with resources at a distance of several hops, broadcast-based solutions are obviously not applicable in terms of bandwidth usage. In the worst case, they cause applications to send data packets to all of the nodes inside the considered network. This issue is specifically significant for C-ITS applications because they are mainly operated in VANET. Although the basic strategy to reach the goal of this dissertation, as discussed in Chapter 2.9.2, is to exploit the *resource registry (RR)*, within a VANET, decentralized resource discovery should be taken into account because an infrastructure-less VANET may not be capable of a static centralized entity. Therefore, even in a VANET, it is sufficient that applications exclusively deliver data to necessary resources using a resource discovery mechanism.

A potential solution for geographical resource discovery in a VANET is to exploit IP multicast in cooperation with ad-hoc routing protocols so that it can efficiently react to changes of the network topology caused by the mobility of the nodes and also avoid broadcasting to all of the nodes in the network. IPv6 multicast-based solutions, specifically Service Location Protocol version 2 (SLPv2) [46, 45] and multicast DNS (mDNS) with DNS-based service discovery (DNS-SD) [10, 4], on traditional routing protocols may be a possible foundation for resource discovery in a mobile network such as a VANET. However, such solutions do not satisfy the main requirement, "geographical resource discovery." Yet, they are able to manage geographical location as one of the resource attributes. In this case, resource discovery messages must be delivered to all of the nodes joining a corresponding multicast group allocated to discover resources. Therefore, this solution is not applicable in terms of bandwidth consumption.

This chapter proposes a resource discovery mechanism that locates resources inside a particular geographical area in a single VANET. The main contributions of this chapter are as follows:

- Low latency, low-cost geographical resource discovery using a legacy IPv6 multicast-based service discovery protocol over IPv6 GeoNetworking
- Dynamic geographical destination management from the application layer
- Field evaluation of the proposed resource discovery mechanism using an actual implementation on Linux

The proposed mechanism is composed of the IPv6 multicast-based service discovery protocol in combination with geographical addressing and routing: SLPv2 with RFC3111 [46, 44] by IETF, and IPv6 GeoNetworking defined by the GeoNet project [6]. The SLP modifications for IPv6 specified in RFC 3111 enable SLP to use multiple IPv6 multicast groups, which allows one-multicast-address-for-one-service usage. IPv6 GeoNetworking makes it possible to deliver conventional IPv6 packets according to geographical location. Thus, upper layer entities can transparently use the geographical routing functionality as legacy IPv6.

4.2 Related work

Resource discovery mechanisms are identified as either *separate or cross-layer integration.* Separate integration is a combination of an application layer discovery protocol and a particular network layer routing protocol. On the other hand, cross-layer integration directly injects resource discovery functions into underlying routing protocols, e.g., [58]. Although the cross-layer solution efficiently discovers resources thanks to direct interaction with the resource discovery functions and underlying routing protocols, it loses the modularity of each protocol because they are tightly connected to a particular routing protocol. The separated solution is more feasible for heterogeneous networks, which may use many different communication technologies.

Regarding the selection of a resource discovery mechanism and underlying routing protocols, the combination of a UDP-based discovery protocol and link-local scope IP multicast is a traditional solution for small and static networks [77, 46, 10]. For mobile resource discovery within a MANET and VANET, the resource discovery mechanism needs to be integrated on ad-hoc routing protocols such as [69, 27, 41]. Because the main target of this research includes ad-hoc networks, this section investigates the harmonization of resource discovery protocols and ad-hoc routing protocols.

One well-known solution that relies on IP multicast is Service Location Protocol version 2 (SLPv2) specified by IETF [46, 45]. It uses three components: a User Agent (UA), Service Agent (SA), and optional Directory Agent (DA). In the context of this dissertation, these components correspond to the RC, RP, and RR defined in Chapter 2.8, respectively. A UA issues a unicast Service Request (*SrvRqst*), which contains the type and attributes of the requested resource, to SAs or DAs. If a service description managed by an SP and/or DA satisfies the request, the SP and/or DA returns a Service Reply (*SrvRply*), which contains a list of URL representations of all available resources in the considered network, to the UA. In addition to unicast, UAs can multicast *SrvRqst* to SAs if DAs are not available. If a DA is available, UAs must use unicast *SrvRqst* to the DA. SAs always unicast *SrvRply* to reply the UAs.

While the original SLPv2 used only one IPv4 multicast address as a communication channel for all SLP-enabled nodes, the modification specified in RFC3111 enabled SLPv2 to discover resources over IPv6 [44]. This modification allows the assignment of multiple IPv6 multicast addresses for each type of service (allocated address range is FF0x::1:1000/118 [8]). An SA joins one of these multicast groups corresponding to its available type of service. The multicast address is determined according to a hash algorithm, which generates a numerical value (0-1023: corresponding to the range of multicast addresses) from a string representation of the service type. The benefit of this modification is the ability to send *SrvRqst* to a specific subset of nodes that definitely manages a particular resource using the IPv6 multicast group assigned to each service. If there are a large number of SAs hosting several different resources in a network, this modification can significantly reduce bandwidth usage.

Multicast DNS (mDNS) with DNS-based service discovery (DNS-SD) is also a well-known resource discovery mechanism that uses IPv4 and IPv6 multicast [10, 4]. It locates resources using the regular DNS message on IP multicast by introducing a special DNS domain ".local" to look up resources in a local network. Resources are described in a list of DNS resource records using SRV, TXT, etc. In contrast to regular DNS, DNS servers (corresponding to RRs) are not needed because mDNS/DNS-SD nodes (corresponding to RPs) act as distributed DNS servers. Thus, DNS clients (corresponding to RCs) are able to directly communicate with RPs.

Unlike SLPv2 with the IPv6 modification, in mDNS/DNS-SD, both request and reply messages are delivered as IP multicast packets using only one IP multicast group (FF0x::FB [8]) for all communications. All of the clients inside a considered network are able to *listen* to the multicast reply for a particular discovery request. This mechanism enables to make a type of *cache* for discovering resources quickly.

Thanks to the link-local scope IP multicast, each protocol can locate resources in a static and/or local network. However, in the case of an ad-hoc multi-hop network (especially in VANET), they may consume significant bandwidth because they rely only on topological address-based routing. As pointed out in Chapter 2.9.2, they can manage the geographical position as one resource attribute in the application layer. The resource type-based IP multicast in SLPv2 does not solve this issue because the RCs in turn need to ask all of the RPs joining a multicast group in a considered network to discover a resource inside a specific geographical area.

Several routing protocols for MANET have been proposed and actually implemented, such as [69] and [27]. However, to achieve the geographical resource discovery in combination with multicast-based resource discovery protocols, GeoBroadcast of IPv6 GeoNetworking is the most appropriate mechanism to locate resources inside a certain geographical area, because it makes it possible to deliver a packet to all of the nodes inside a specific geographical destination (GeoDestination) using IPv6 multicast.

4.3 Assumptions

4.3.1 Network architecture

In addition to the basic assumptions shown in Chapter 2.5, this section describes the detailed assumptions for a VANET as a part of VCS. Each node installed in a vehicle or along the roadside complies with the *ITS station reference architecture* by ISO/ETSI [34, 50, 68]. An ITS station is assumed to be equipped with a router (MR or AR) and a host, which may be connected to the router. MRs and ARs are equipped with at least (i) one wireless egress interface to communicate with other routers and (ii) one wired/wireless ingress interface to connect to the station-internal network. IPv6 works as a mandatory network layer communication protocol and MRs/ARs provide network prefixes to their attached hosts. Each host therefore has a global IPv6 address configured from a network prefix assigned by its station-internal MR or AR. ARs provide Internet access to MRs. Although the ITS station reference architecture makes it possible to integrate the MRs/ARs with host functionality into an identical node, this chapter assumes that each node is separately installed as a router or host.

All of the MRs and ARs composing a VANET support IPv6 GeoNetworking. They communicate with each other as single-hop neighbors from the IPv6 point of view, because IPv6 GeoNetworking takes care of the multi-hop routing. Each MR/AR obtains its current geographical position through static configuration or GPS.

4.3.2 Discovery scenarios

This chapter discusses the in-mobile resource discovery defined in Chapter 2.8.2: Mobile-to-Mobile (MM) discovery, Mobile-to-Static (MS) discovery, and Static-to-Mobile (SM) discovery in a VANET.

In the MM discovery use case, an in-vehicle application communicates with resources in other vehicles. The location of resources is determined with the geographical position, type of resource, and resource specific attributes (e.g., type of vehicle, trajectory, and equipment). A typical discovery scenario is as follows: when a vehicle is going to merge into a lane, an application in the vehicle notifies relevant vehicles of its behavior. In this case, the application needs to discover *the presence services* in moving and/or parked vehicles inside a relevant area.

In the SM and MS discovery use cases, an in-vehicle or roadside application communicates with other roadside or in-vehicle hosts based on their geographical position, type of resource and resource specific attributes. A typical scenario is as follows: if a driver wants to park his/her vehicle near a destination (municipal office, supermarket, etc), an in-vehicle application tries to discover *parking management services* that manage public and/or pay parking lots near the destination. While selecting a parking lot, the application also locates *road congestion monitoring services* on the roadside along the path to the parking lot.

4.4 Geographical resource discovery on IPv6 GeoNetworking

This section proposes a geographical resource discovery mechanism for VANET. As mentioned above, the objective of this mechanism is to discover resources according to their geographical position, in addition to the resource type and its attributes. The main challenges of this mechanism are (i) *how to specify GeoDestination for each resource* and (ii) *how to reduce the bandwidth usage*. Regarding the former challenge, each RC should be able to separately specify an arbitrary GeoDestination. In addition, GeoDestinations should be dynamically configured by RCs. For the latter challenge, each component needs to avoid transmitting unnecessary messages to distant nodes outside of the requested geographical area.

Additionally, the solution should maintain the modularity of protocol layers so that each protocol is integrated separately for future improvement. The proposed mechanism is therefore composed of an application layer resource discovery protocol on IPv6 GeoNetworking. In contrast to existing solutions using application layer discovery protocols on traditional network layer routing protocols, in the proposed solution, IPv6 GeoNetworking supports geographical routing under IPv6 so that the proposed mechanism discovers resources according to geographic position transparently using geographic routing while avoiding propagating packets to the entire network. One of the challenges of harmonizing resource discovery protocols with IPv6 GeoNetworking is determining a method for the discovery protocol to assign application specific GeoDestinations to IPv6 GeoNetworking without directly merging resource discovery functions into the IPv6 GeoNetworking mechanisms. To tackle this issue, a cross-layer interface is introduced that enables the resource discovery protocol to configure GeoDestinations in IPv6 GeoNetworking.

The proposed mechanism uses SLPv2 as a foundation for the resource discovery protocol because of the compatibility of the IPv6 modification of SLPv2 with IPv6 GeoNetworking. This mechanism makes it possible to handle multiple IPv6 multicast groups, as mentioned in Chapter 4.2, which meets the above-mentioned design principles. To comply with the standardized specification, the following sections use the terms UA and SA from SLPv2 for RC and RP, respectively.

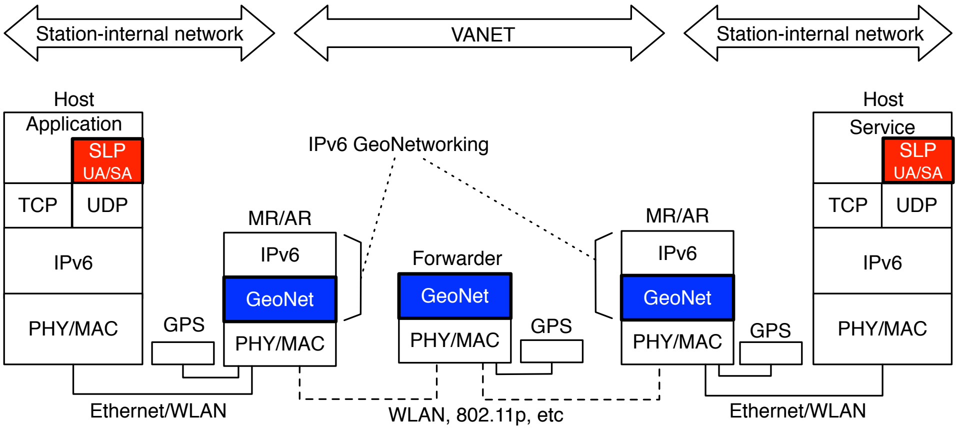


Figure 4.1: Communication stack.

4.4.1 SLP-based resource discovery over IPv6 GeoNetworking

SLPv2 and IPv6 GeoNetworking are integrated separately: SLPv2 components are integrated into attached hosts, while IPv6 GeoNetworking is integrated into MRs and ARs. The hosts can be conventional PCs as long as they support the regular TCP/IP stack, while routers do not necessarily support application layer entities, as shown in the proposed protocol stack in Figure 4.1. Applications and resource discovery components are integrated into the attached hosts.

The proposed mechanism uses SLPv2 components: UAs and SAs without DAs. When a resource is activated, it registers its type and attributes to an SA. SAs do not advertise and forward registered information to any other hosts but passively reply to discovery requests from UAs. As mentioned in Chapter 4.2, in SLPv2, SAs hosting a particular service join a corresponding IPv6 multicast group, which is determined from the service type using the SLPv2's hash function. Resource discovery in the proposed system is composed of the following three mechanisms:

- **IPv6 multicast *SrvRqst* over GeoBroadcast**: UAs try to discover resources using IPv6 multicast *SrvRqst* over GeoBroadcast. The multicast *SrvRqst* is delivered as a GeoBroadcast packet among MRs/ARs. Thus, it is received merely as a subset of nodes joining the corresponding IPv6 multicast group inside a particular geographical area [42].

  From the points of view of SLP components, *SrvRqst* is a legacy IPv6 multicast packet transmitted to a multicast group corresponding to the requested resource. However, as described in Chapter 4.2, the IPv6 GeoNetworking mechanism encapsulates the *SrvRqst* into a GeoBroadcast packet, which is disseminated to all of the nodes inside a specific GeoDestination. The MRs and ARs inside the GeoDestination decapsulate the packets to IPv6 multicast *SrvRqst* packets and deliver them to SAs. A GeoDestination is described using coordinates and the size of the requested area [42].

The decision to encapsulate or decapsulate packets from/to an IPv6 multicast and GeoBroadcast is made using a list of tuples stored in IPv6 GeoNetworking. A tuple is a pair of data elements mapping an 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 and 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 headers of received GeoBroadcast (i.e., multicast *SrvRqst*) packets. If the position is not available (e.g., it has expired), IPv6 GeoNetworking resolves the position using the *Location Service* protocol specified in [42]. The received GeoUnicast packets are decapsulated in an MR and AR to an IPv6 unicast *SrvRply* packet and delivered to the UA. The packet encapsulation mechanisms of *SrvRqst* and *SrvRply* are depicted in Figure 4.2.

- **GeoDestination management over TCP unicast**: the GeoNet specification describes one of the potential solutions for determining a GeoDestination by using the destination IPv6 address. For instance, regarding a GeoBroadcast, the IPv6 multicast address is statically mapped to a corresponding geographical area using a configuration file that assigns an IPv6 multicast address to a GeoDestination as a radius around the center of the area where the packet shall be propagated (i.e., FF0E::1 corresponds to a circle with a radius of 500 m centered on the sender). In this solution, users of IPv6 GeoNetworking need to specify all of the possible IPv6 multicast address and GeoDestination pairs beforehand.

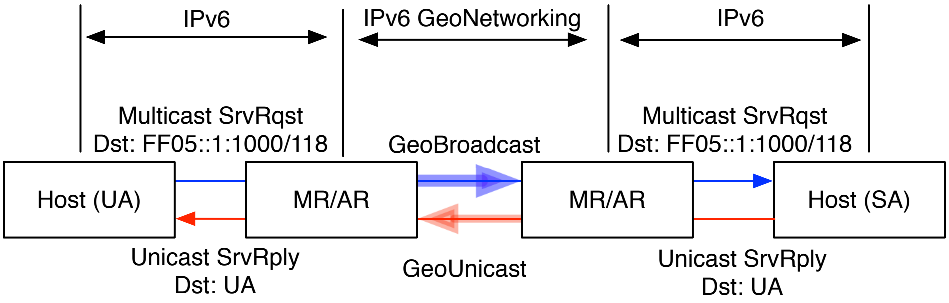


Figure 4.2: Encapsulation of SLP messages.

> However, a solution that relies on the static configuration cannot be applied to the proposed mechanism because the GeoDestination for each resource discovery trial cannot be configured statically. The necessary GeoDestination is different in each discovery scenario.
>
> To overcome this issue, this chapter introduces a mechanism into IPv6 GeoNetworking that allows external entities to dynamically configure the GeoDestination 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 also send a unicast packet that indicates the GeoDestination corresponding 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 for VANET.

Regarding the multi-hop IPv6 multicast routing from an attached host to the other hosts in *SrvRqst*, the MRs/ARs simply forward the multicast packets between their ingress and egress interfaces. They do not use any dedicated multicast routing mechanism because MRs/ARs can communicate with each other as one-hop IP neighbors in the considered VANET thanks to IPv6 GeoNetworking, as shown in Chapter 4.2. Therefore, the proposed mechanism does not require additional overhead to build and maintain the multicast routing topology. Note that the site-local scope IPv6 multicast (i.e. FF05::1:1000/118) was used instead of the link-local scope because UAs and SAs are out of the link-local scope; they are located in attached hosts behind different MRs/ARs.

4.4.2 Operation sequence

The proposed mechanism has three phases: the resource activation phase, resource discovery phase, and resource operation phase. The overall operations are described as follows:

Resource activation phase

1. When a resource is installed and activated, it registers its descriptions (e.g., resource type, attributes) to an SA running in the local host.
2. The SA joins an IPv6 multicast group determined from the resource type using the SLP's hash function. An MLD report is sent to the station-internal MR and AR.

Resource discovery phase

1. An application requests an UA in the local host to discover a resource. The application sends the description of the discovery request (i.e. the requested resource's characteristics and relevant geographical area) to the UA.
2. From the application's request, the UA calculates the corresponding IPv6 multicast address, as in the resource activation phase, and then sends a {*IPv6 multicast address, GeoDestination*} pair, to its station-internal MR/AR via TCP unicast (**GeoDestination management**).
3. IPv6 GeoNetworking in the MR/AR creates or updates the list of mapped GeoDestination entries with the received IPv6 multicast address and GeoDestination pair.
4. The UA then performs the **IPv6 multicast *SrvRqst* over GeoBroadcast**: it issues the multicast *SrvRqst* designated for 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 mapped entry created 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, GeoNetworking ID, and the requested GeoDestination. The GeoNetworking ID is obtained from the sender's IPv6 unicast address.

7. The MRs/ARs located inside the GeoDestination receive the GeoBroadcast packets on their egress interface. They check whether there are attached hosts belonging to the corresponding multicast group (i.e., an SA that operates the requested resource) on their ingress interface. If there are corresponding SAs, the MRs/ARs decapsulate the GeoBroadcast packets into the regular IPv6 multicast packets and send them to SAs via their ingress interface. At the same time, the MRs/ARs record the sender's geographical position and the GeoNetworking ID included in the header of the GeoBroadcast packets.

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

9. The MR/AR first resolves the IPv6 address of the destination MR/AR for the UA using legacy IP unicast routing. Then, the 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 them into IPv6 unicast packets. Finally, the UA receives the unicast *SrvRply*.

Resource operation phase Finally, the application and resource start to communicate with each other. Figure 4.3 shows the overall messaging sequence.

Suppose a set of all of the nodes in the considered network N, a group of all of the nodes that join the corresponding IPv6 multicast group G_{mc}, a group of all of the nodes inside the corresponding GeoDestination G_{gd}. Only a node $N_i \in N$ that satisfies

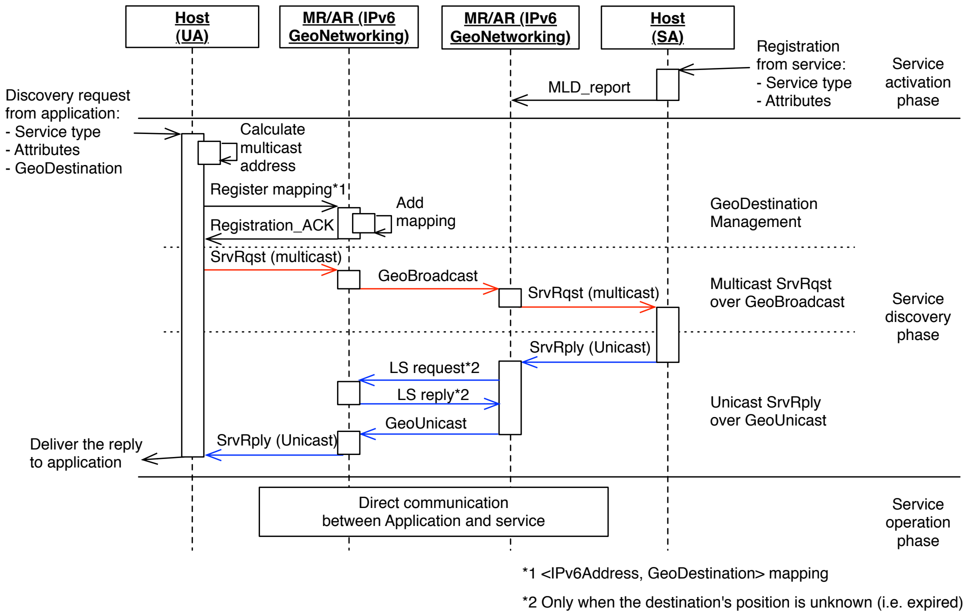


Figure 4.3: Messaging sequence.

$(N_i \in G_{mc}) \cap \left(N_i \in G_{gd}\right)$ can receive *SrvRqst* packets, shown in the selective propagation of multicast *SrvRqst* packets in Figure 4.4. Thanks to this mechanism with the SLP's per-service IPv6 multicast address assignment, the proposed mechanism can avoid unnecessarily propagating resource discovery messages to a large geographical area. In addition, the UAs and SAs do not necessarily take care of their current locations when discovering resources because the geographical position is managed by IPv6 GeoNetworking in the MR/ARs.

4.5 Experiments

To observe the cost and performance of the proposed mechanism, a prototype system is implemented. First, scalability evaluations are conducted using the network simulator ns-3 [11]; then, outdoor tests are also performed by integrating the system into actual hosts and routers.

The basic discovery scenario in each evaluation is as follows: a UA in a vehicle station (a router and host pair in a vehicle) periodically tries to discover resources within a VANET composed of several roadside stations, in which the attached hosts

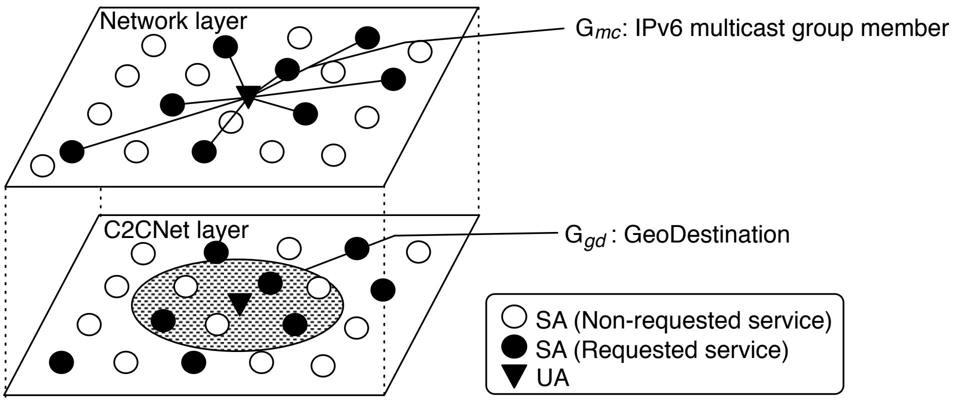


Figure 4.4: Propagation of multicast SrvRqst packets.

work as SAs with 100 resources. The UA sequentially issues *SrvRqsts* until it has transmitted 100 requests; it issues a *SrvRqst* 1 s after receiving the first *SrvRply* for the previously sent *SrvRqst*. In each request, the UA randomly determines a resource and GeoDestination. The vehicle moves inside a VANET. The MR in the vehicle station is always inside the radio communication range of at least one other station. Each router is equipped with a wireless egress interface with the standard IEEE 802.11b MAC layer. Its radio communication range is configured to 130m. IPv6 GeoNetworking is configured to the default setting specified in [42] (e.g., a beacon message is delivered every 500ms). SLP also uses the default configuration specified in [46, 44].

The performance metrics in each evaluation are as follows:

- Discovery success rate: the rate of successful replies to *SrvRqst*.
- End-to-end latency: the delay in the successful replies in the service discovery phases between the UA and SAs.
- Size of control messages transmitted per router: the number of bytes transmitted by the egress interface in each router. This metric represents the total size of the IPv6 GeoNetworking messages, including *Beacon, GeoUnicast, GeoBroadcast*, and *Location Service*. Note that the station-internal communication between the attached hosts and the routers are not taken into account because it is performed on Ethernet, which provides sufficient bandwidth and stability.

- Size 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*. This metric is used to show the ratio of the control messages transmitted to the number of *SrvRplys* delivered to the UA.

4.5.1 Simulation setup

The proposed system is integrated into ns-3 version 3-12.1. To conduct the simulation, one new application layer protocol and one new ns-3 model are implemented: SLP and IPv6 GeoNetworking. The SLP implementation works as a part of the ns-3 *application* model, while IPv6 GeoNetworking is an independent ns-3 model that works with the *Internet* model and *Netdevice* model. The SLP application only supports the limited functions required to evaluate the proposed system. Therefore, most of the features are not implemented, e.g., authentication, multiple scope handling, DA, etc. On the other hand, the IPv6 GeoNetworking model fully complies with [42], except for the geographical position management. Regarding geographical position, the ns-3 mobility model simulates the node's position. It is assumed that the position information is always accurate in the simulation. In addition, the UDP models are modified to support IPv6 because the model in ns-3.12-1 only supports IPv4.

The simulated VANET is composed of 100 stations in a 1000m × 1000m rectangular field. At first, the stations are located in a 2D-grid, in which the distance between adjacent nodes is 100 m. The vehicle moves according to a random waypoint mobility model using three velocity settings: low mobility (10 km/h), medium mobility (36 km/h), and high mobility (72 km/h). While the shape of GeoDestination is always circle, its center position and radius are randomly determined (the radius has a range of 75 - 150 m).

To compare the performance of the proposed system and the existing one, the same tests are performed without using the GeoDestination management mechanism. In these tests, the GeoDestination is fixed to cover all of the nodes in the VANET. This evaluation similarly corresponds to the solution in which the application layer manages the geographical position by relying on geographic-agnostic, all-node flooding.

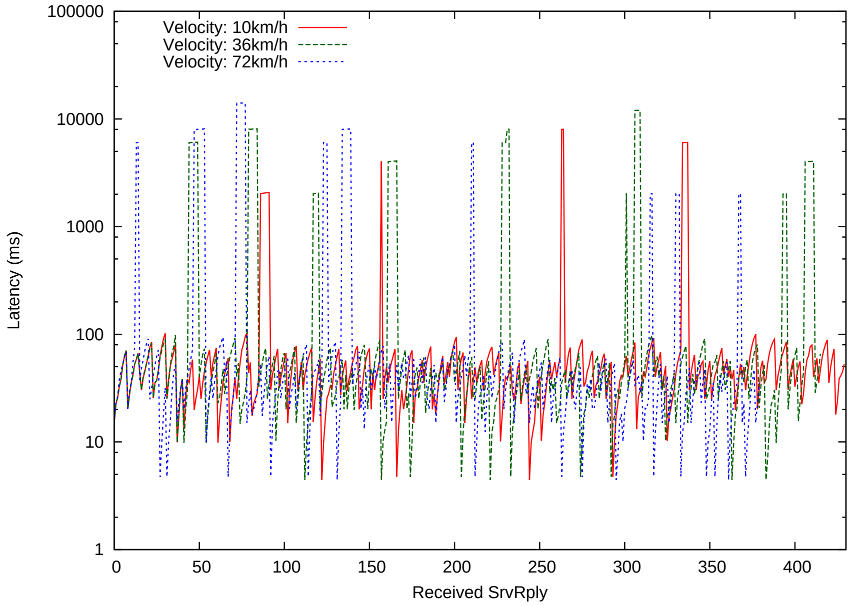


Figure 4.5: Distribution of latency in simulation with GeoDestination management.

4.5.2 Simulation results

Simulation results were obtained by averaging 10 different-seed runs for each setting. The discovery success rates were 100%, 96%, and 94% in the scenarios with velocities of 10 km/h, 36 km/h, and 72 km/h, respectively. The end-to-end latency were proportional to the velocities: 4/8022/175, 4/12034/592, 4/14068/678 ms (minimum/-maximum/average) for the different scenarios. Figure 4.5 shows the distribution of the end-to-end latency for each received *SrvRply*, and Figure 4.6 shows its CDF. The sizes of the control messages transmitted per router were 204, 174, and 167 bytes/s for the respective scenarios, while the sizes of the control messages transmitted per router per received *SrvRply* were 0.47, 0.41, and 0.43 bytes/s.

In the simulation without the GeoDestination management mechanism, the end-to-end latencies were 4/557/273, 4/593/286, and 4/10408/2807 ms (minimum/maximum/average) for the respective scenarios. Figure 4.7 shows the CDF of the end-to-end latency in each scenario. The sizes of the control messages transmitted per router were 1727, 1719, and 1729 bytes/s for the respective scenarios, while the sizes of the control messages transmitted per router per received *SrvRply* were 1.7, 2.1, and 1.2 bytes/s. Table 4.1 summarizes the simulation results.

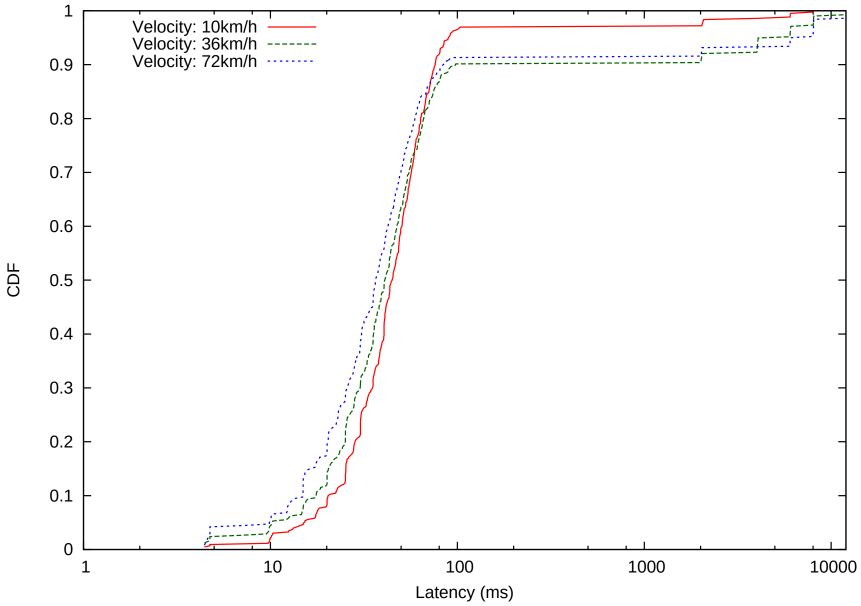


Figure 4.6: CDF of latency in simulation with GeoDestination management.

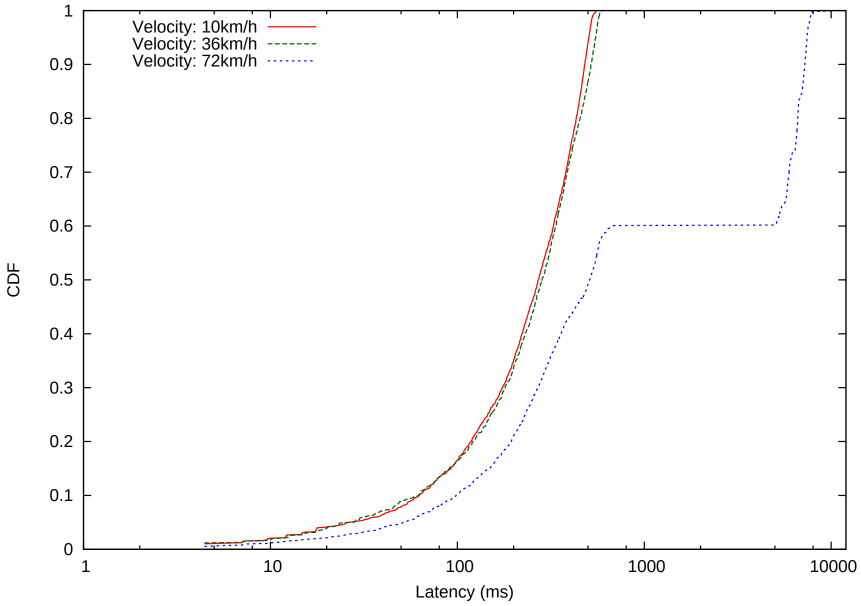


Figure 4.7: CDF of latency in simulation without GeoDestination management.

Table 4.1: Summary of simulation results.

Scenario		Success rate	Latency (ms)			Transmitted packet
GeoDestination management	UA's Velocity (km/h)	(%)	Min	Max	Average	(bytes/s/node)
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

4.5.3 Field evaluation setup

In addition to the simulation, the proposed system was actually implemented by extending *OpenSLP* 2.0 Beta 1 [12] and *CarGeo6* [2] on Linux. OpenSLP is an open-source implementation of SLP includes the modification for IPv6, and CarGeo6 is an open-source implementation of IPv6 GeoNetworking in compliance with the reference specification of the GeoNet project [41, 42]. To forward IPv6 multicast packets between the ingress and egress interfaces of each router, *si6mfd*, a simple IPv6 multicast forwarding daemon for Linux was also implemented.

The system was integrated into the field testbed at the NAIST campus in Japan, including three sets of routers with GPS and attached hosts. Each router was equipped with one Ethernet port as an ingress interface, and one wireless 802.11 b/g card on the 2.4 GHz frequency band as an egress interface. The data rate of the egress interface was configured to 6 Mbps. Ubuntu 10.10 (kernel 2.6.35.11) was used for all of the nodes. Each router was able to obtain its current geographical position through a GPS receiver via its USB-serial connection. In order to obtain coordinates, gpsd-2.96 [7] was installed as a local TCP server. While the routers executed CarGeo6 and the IPv6 multicast forwarding daemon to operate IPv6 GeoNetworking and IPv6 multicast forwarding, the attached hosts were conventional PCs supporting OpenSLP and the conventional IPv6 multicasting functions.

The following discovery scenario was used in this evaluation. As shown in Figure 4.8, three stations are located along the roadside. Each station has a router and an attached host: Station1 (MR, Host1), Station2 (AR1, Host2), and Station3 (AR2,

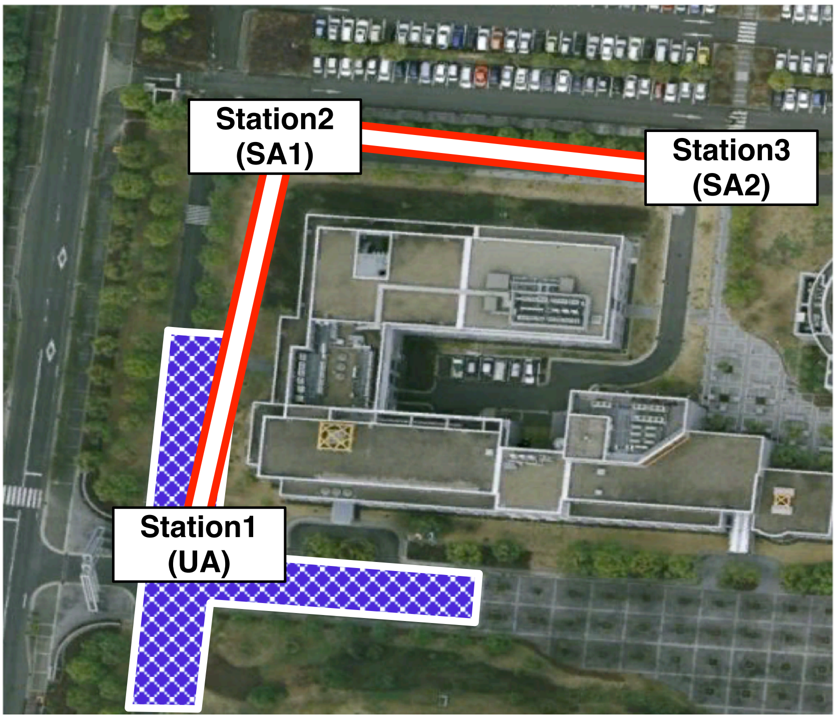


Figure 4.8: Network topology used in field evaluation.

Host3). Station1 acts as a vehicle station. Thus, it moves around the other two stations. A UA operates on Host1 and periodically attempts to discover resources using a randomly selected GeoBroadcast radius. In contrast, the other stations are stationary. Host2 and Host3 operate SAs (SA1 on Host2, SA2 on Host3). Station1 moves inside the direct communication range of Station2, while it does not enter Station3's direct communication range. This means the UA in Station1 can communicate with SA1 via a single-hop, whereas SA2 requires multiple-hops.

In the field evaluation, the center position of GeoDestination for GeoBroadcast is fixed to the position of the MR, because the CarGeo6 implementation at the moment only supports *Vehicle-Centred GeoBroadcast*. Therefore, the UA only specifies a randomly selected radius for GeoDestination (the range of this radius is: 50 - 200m). The system configuration is shown in Table 4.2.

Table 4.2: System settings for 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(Multiple hops) (m)	120 - 200

Table 4.3: End-to-end latency and discovery success rate of field evaluations per hop.

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

4.5.4 Field evaluation results

The discovery success rate was 86%. In the single-hop case, it was 96%, while it was 77% in the multi-hop case, and the end-to-end latency was 3.5/170/28.2 ms (minimum/maximum/average). Figure 4.9 shows the CDF of the end-to-end latency of the received *SrvRply*. The field evaluation also evaluated the end-to-end latency per hop: the single-hop latency between the UA and SA1 was 3.5/23.3/7.8 ms, whereas the two-hop latency between the UA and SA2 was 27.9/170/48.6 ms, as shown in Table 4.3.

The size of the control messages transmitted per router was 243 bytes/s. The MR in Station1, in which Host1 operates the UA, transmitted 300 byte/s, while that in Station2 was 223 bytes/s, and Station3 was 204 bytes/s. Figure 4.10 shows the proportions 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 service: 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.

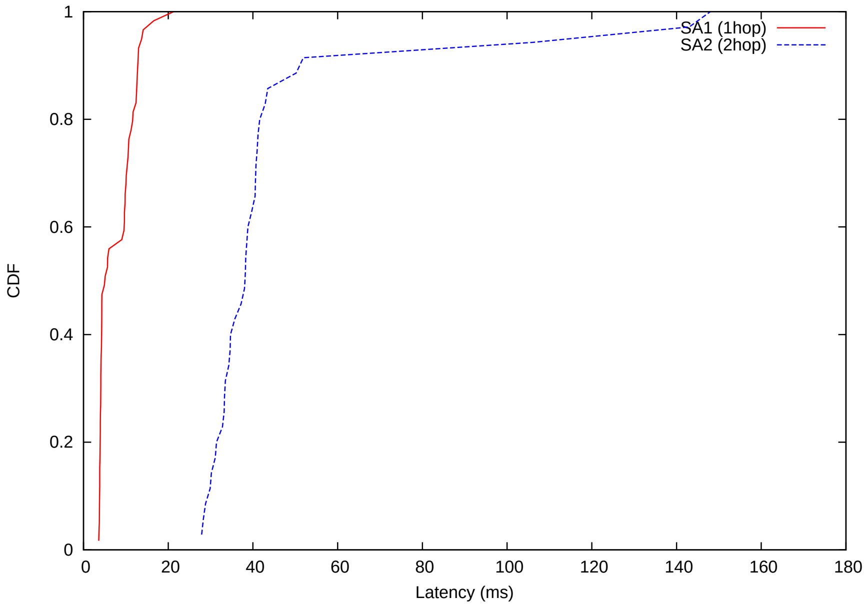


Figure 4.9: CDF of latency in field evaluation.

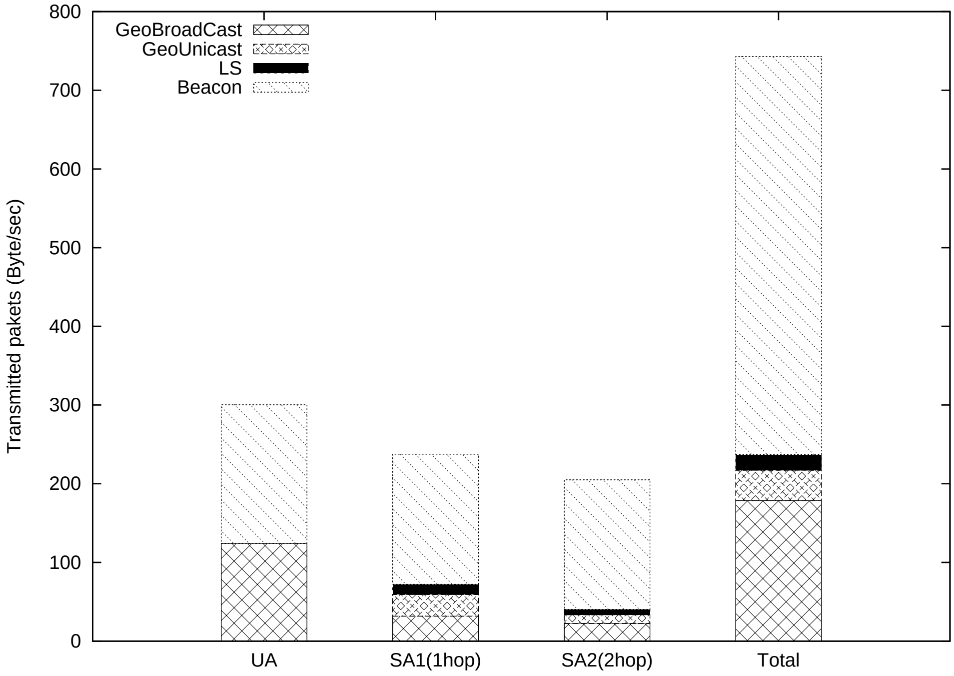


Figure 4.10: Message overhead.

4.5.5 Analysis

According to Figure 4.5, the proposed system stably worked in every evaluation. This result shows that the trend for 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 Figure 4.6 and Figure 4.9 show that 90% of the successful *SrvRqsts* were completed within 100 ms in any mobility setting. Even in the actual environment, the UA mostly discovered services within 20 ms via a single-hop, and within 60 ms via multiple-hops, as plotted in Figure 4.9. The high latency that appeared in the simulations (i.e., 2000 ms or more) was caused by the SLP's retransmission algorithm, which exponentially increased the wait interval for each discovery trial from 2 to 15 s [46]. Such retransmissions may have occurred when discovering services in distant GeoDestinations that required multiple hops. Consequently, the evaluation results show that the proposed mechanism mostly discovers resources within the required acceptable latencies specified in ITS related researches and standards (mainly from 100 to 500ms) [68, 23].

On the other hand, in the application layer solution with all node flooding, more than 80% of the discovery attempts needed more than 100 ms to discover services, as shown in Figure 4.7. 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 the discovery attempts 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. This is considered to have been caused by the Location Service mechanism of CarGeo6, which needs additional communications between routers. The Location Service mechanism was used to determine the geographical position of the UA to send back a *SrvRply*. Therefore, there the following two possibilities exist: (i) the AR2 in Station3 could not successfully record the UA's position from the received *SrvRqst* or (ii) the recorded position had expired because of the UA's mobility.

The overhead for the control messages in each successful discovery in the simulation was about 0.4 bytes/s for each mobility scenario. This result shows that the proposed system requires consistent costs to deliver a *SrvRply* to a UA, even in the high-mobility scenario. It efficiently reduces the overhead of the control messages compared to the all node flooding-based solutions, as depicted in Table 4.1, thanks to the GeoBroadcast management mechanism, which makes it possible to deliver *SrvRqst* to only a limited number of nodes inside the dynamically assigned GeoDestinations. The field evaluation results also show that the distant nodes (nodes in Station3 located at a two-hop distance from Station1) successfully avoided processing unnecessary packets sent to a GeoDestination that did not cover the position of the nodes.

As shown in Figure 4.10, a large portion of the overhead was caused by Beacon messages, which were periodically sent to single-hop neighbors at 500 ms intervals by each node regardless of the proposed resource discovery mechanism. The overhead caused by the resource discovery (except Beacons) in the actual environment was 178 bytes/s in each node and 124/31/22 bytes/s in MR/AR1/AR2 (connected to UA/SA1/SA2, respectively). According to a preliminary test evaluating the available throughput in the field testbed, the maximum bandwidth in the outdoor test setting was about 5 Mbps (625 Kbytes/s). This result shows that the bandwidth usage of the proposed system is fairly small.

4.6 Summary

This chapter presented a geographical resource discovery mechanism to support resource discovery locally. The proposed mechanism is a harmonization of SLPv2 and IPv6 GeoNetworking developed in the GeoNet project. Furthermore, this research integrated a cross-layer GeoDestination management mechanism that enables IPv6 GeoNetworking users (i.e., applications) to dynamically configure GeoDestination. Using the ns-3 network simulator, evaluation results showed that the IPv6 multicast-based resource discovery using GeoBroadcast with the GeoDestination management rapidly discovers resources without unnecessarily propagating discovery packets to the entire network.

In addition to the simulations, the proposed mechanism was actually implemented in Linux using the OpenSLP and CarGeo6 implementations. Its evaluation was performed in a field testbed at the NAIST campus in Japan. The evaluation results showed that the overhead of the proposed system is fairly low. As a next step, it is necessary to extend the proposed mechanism as a part of the wide area geographical resource discovery system.

Chapter 5

Wide-area hierarchical publish/subscribe resource discovery using IPv6 GeoNetworking

Previous chapters have discussed the importance, issues, requirements, and potential solutions of resource discovery for C-ITS applications. Chapter 4 presented a geographical resource discovery mechanism to support resource discovery locally in a VANET. This chapter finally tackles the key challenge of this dissertation: *wide-area geographical resource discovery* that is applicable to not only a VANET but also the entire Internet. This chapter presents a solution for wide area geographical mobile resource discovery for C-ITS applications mainly integrated into VCS composed of numerous mobile networks such as VANETs connected to the Internet. The proposed system relies on a hierarchical publish/subscribe architecture and geographic routing so that resource consumers can locate resources according to geographical coordinates without the scalability issue. The proposed system also includes a location management mechanism for mobile resources, which enables to reduce the periodic geographical location updates. Numerical analysis and simulation results show that the system can locate mobile resources without overloading in both VANETs and the Internet.

5.1 Background and motivation

While Chapter 4 presented a geographical resource discovery mechanism for a VANET by harmonizing an IPv6 multicast-based discovery protocol and IPv6 GeoNetworking, a significant issue remained for wide area geographical mobile resource discovery: the diversity of the underlying network topology and protocol. As mentioned in Chapter 2.8.2, in a heterogeneous environment, applications and requested resources may communicate with each other via several different types of networks. For instance, they can be connected (i) within a single wireless mobile network using a multi-hop ad-hoc routing protocol, (ii) via a WLAN access point using conventional IP routing protocols, (iii) with single-hop broadcasting, or (iv) via several different networks through the Internet. Even if there are numerous geographic routing protocols, wide-area inter-domain Geocasting, which only relies on a specific routing protocol, is not a feasible solution.

This chapter proposes a wide-area mobile resource discovery system that supports not only a VANET but also global scope discovery according to geographical position. The system is built on a hierarchically-distributed publish/subscribe architecture, which enables scalable resource discovery in combination with the GeoNetworking protocol. The loose coupling nature of the publish/subscribe scheme [40], which enables applications to communicate with potential resource providers asynchronously and anonymously, is suited to mobile resource discovery. To handle a large number of mobile resources, the proposed system relies on the distributed publish/subscribe architecture, where numerous RRs comprise a distributed registry in core (the Internet) and VANETs. The intra-domain geographical resource discovery is supported by IPv6 GeoNetworking, as described in Chapter 4, while the inter-domain scenario is taken cared for the core RRs .

This chapter focuses on two main issues in the distributed publish/subscribe mobile resource discovery. The first is a scalability issue caused by the frequent location updates by resource providers. In the case of geographical mobile resource discovery, resource providers need to publish available resources with their current geographical positions. Because this information must be frequently updated, depending on the publishers' mobility, in order to maintain the consistency of the registered information, the huge number of location updates needed for publishers may cause a scalability issue. The second issue is how to select an appropriate discovery scope: the smallest scope is a single hop inside a VANET, whereas a larger scope may be from a VANET to another VANET via several networks through the Internet. The appropriate scope is not known a priori, because applications issue discovery queries that only contain requested resource type, geographical position, and additional attributes, which do not help to determine the topological scope. Although a larger scope is certainly preferable to discover resources, large-scope discovery may cause a scalability issue. On the other hand, a small scope is desirable to discover resources rapidly, but it may fail to discover potential publishers.

The main contributions of this chapter are therefore as follows:

- **A geographically-distributed publish/subscribe system**: to reduce control messages, dedicated servers comprise a group of publish/subscribe RRs on top of a geographically-distributed overlay network (namely, *core RRs*), while mobile nodes supporting GeoNetworking act as *mobile RRs* and connect to one of the core RRs. These two types of RRs localize communications between publishers and subscribers according to their geographical and topological locations by utilizing the boundary information of the GeoNetworking-enabled network. This mechanism contributes to the discovery of resources without incurring bandwidth usage.

- **Adaptive scope selection according to topological and geographical location**: to reduce unnecessary queries from subscribers to RRs, mobile RRs adaptively select an appropriate discovery scope according to the geographical destination of the subscription, while taking into consideration the boundary of the geographical routing-enabled network.

- **Mobility aware, event driven location update**: to reduce the number of location updates from publishers, a location update is issued only when a publisher changes its mobility pattern, i.e., speed and/or heading. This mechanism reduces bandwidth usage that results from a large number of periodic location updates.

- **Integration of the standardized IPv6 GeoNetworking**: the proposed system exploits the standardized IPv6 GeoNetworking mechanism [38, 35], an adaptation of IPv6 to the geographical addressing and routing protocol designed for emerging VCS. This chapter evaluates the effectiveness of this IPv6 GeoNetworking mechanism.

5.2 Related work

5.2.1 Standardized GeoNetworking protocol

After the conclusion of the GeoNet project, the European Telecommunications Standards Institute (ETSI) specified a GeoNetworking protocol and a mechanism to adapt it to IPv6 (GeoNetworking to IPv6 Adaptation Sub-Layer, namely *GN6ASL*) [39], which enables IPv6 operation over the network-layer GeoNetworking protocol [38]. The GeoNetworking protocol is a full set of non-IP geographical messaging technologies, including a geographical addressing scheme, geographical routing algorithms, and geographical location resolution mechanism (namely Location Service). The GeoNetworking protocol is located between the network layer and the link layer, and GN6ASL provides interfaces to IPv6 so that upper layer entities can transparently use GeoNetworking through conventional IPv6. Several types of geographical routing have been specified, i.e., GeoUnicast, GeoBroadcast, and GeoAnycast. In particular, GeoBroadcast delivers packets to all of the nodes inside a certain geographical area, described as a circle, rectangle, or ellipse.

5.2.2 Publish/subscribe systems

Publish/subscribe is a mature paradigm; there have been a wide variety of publish/-subscribe systems such as [24, 75]. A mobility support mechanism for a distributed publish/subscribe system was proposed in [22]. It can handle mobile clients by means of service proxies, which act as interfaces for clients to the publish/subscribe system. In this mechanism, subscriptions and publications are managed by proxies on behalf of clients. Thanks to the buffering and synchronization of subscriptions/publications, clients are able to move from one access point of the system to others, which minimizes the inconsistency of information.

5.3 Geographically distributed mobile publish/subscribe resource discovery

5.3.1 Assumption and system model

It is assumed that a set of networks is composed of two types of mobile nodes: one that supports IPv6 with mobility support protocols and IPv6 GeoNetworking, while the other only supports IPv6 with mobility support protocols. A subset of the first type of nodes comprises a wireless ad-hoc network with GeoNetworking (namely GeoNetworking domain) when they are within one or a certain number of hops. On the other hand, the latter type is not capable of ad-hoc multi-hop communication according to geographical position. Because each node supports IPv6 mobility support protocols, at least one permanent IPv6 address is assigned to each node regardless of the point of attachment and underlying communication stack. These mobile nodes may be temporally disconnected from the Internet. In this case, mobile nodes communicate with each other using temporary IPv6 addresses (e.g., assigned by the access router of a foreign network). Each mobile node can obtain its geographical position from a GPS device, static configuration file, etc. It is assumed that the obtained geographical position is accurate.

The proposed system consists of four components: *publisher*, *subscriber*, *core RR* and *mobile RR*, as depicted in Figure 5.1. Publishers act as resource providers, while applications are subscribers. Core RRs are connected to each other, while mobile RRs are attached to one of them. Publishers and subscribers only communicate with topologically neighboring core or mobile RRs; for instance, an in-vehicle publisher/subscriber is attached to a mobile RR installed in an in-vehicle mobile node. Regardless of the type of RR, publishers and subscribers just send publication (information about available resources) and subscription (resource discovery request) information to the connected RR. Note that the distinction between the proposed system components is conceptual; some components may be installed in an identical node (e.g., a smartphone may be a publisher, subscriber, and mobile RR at the same time).

To manage a large number of publications and subscriptions, the core RRs act as a distributed database on top of an overlay network, where each node is assigned an identifier in an overlay that is generated using, for example, Z-order curve [56]. Thanks

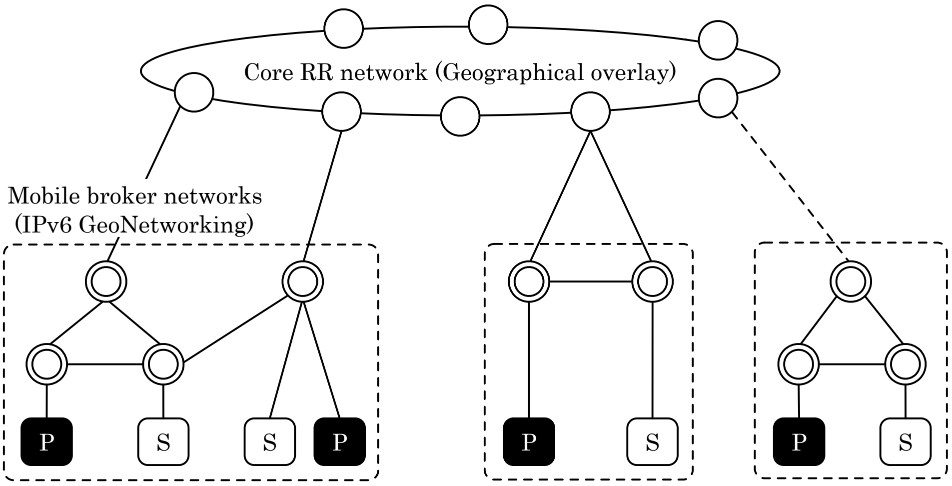


Figure 5.1: System model. S and P represent subscriber and publisher, respectively.

to the space-filling curve, which maps a multi-dimensional space (in this paper, the 2D surface of the earth) to one dimension while maintaining its locality, each core RR manages publications according to geographical area using geographical coordinates. In other words, publications in a neighboring area are likely to be stored on the same core RR.

The main requirements for the overlay network as a foundation of the core RR network are therefore as follows: *geographic search*, *range search*, and *scalable search*. As previously described, a geographic search is a key requirement for the proposed system. A range search is needed to search stored resources by *area* instead of *point*; the range-based lookup capability is necessary rather than the exact-matching-based lookup, for better performance. Regarding a scalable search, it is necessary to finish a lookup with a reasonable cost, even when the number of nodes is large. For instance, a broadcast-based approach, sequential lookup for all nodes, is not applicable.

While the proposed mechanism itself is independent of the overlay networking mechanism as long as it supports the above requirements, it is assumed that [59] is the foundation of the core RR network, which is a structured overlay network using the Z-order curve-based IDs, which can search for a stored value by $O(logN)$, according to the geographical position, and supports a range search.

In the proposed system, resource discovery is performed using three main functions: publish, subscribe, and notify, as in existing publish/subscribe systems, e.g., [24]. Resource providers publish descriptions of resources to one of the RRs, while applications subscribe to a certain resource from the RRs. Brokers match the publications and subscriptions, and then notify subscribers if publications satisfy the subscriptions.

5.3.2 Data model

A resource is described using a *Resource Description*, a set of data elements containing the geographical position, socket address, resource identifier, and resource-specific attributes (e.g., organization, type of vehicle, priority, etc.). Subscribers try to discover resources using the geographical area and at least one of these elements.

To reduce the bandwidth usage, the proposed mechanism separates the resource descriptions in two types of data sets: *static profiles* and *dynamic profiles*. A static profile is delivered only once from a mobile to a core RR, while a dynamic profile is generated to update the static profile. The static profile is composed of tuples of the form (HP, SA, rd_1, ..., rd_i), where HP points to the *Home Position*, the primary geographical position of the resource provider: the position when a resource provider is activated. SA, *Socket Address*, shows the communication endpoint of the resource provider. rd_i consists of the ith *Resource Description*. A Resource Description is represented as a tuple (RI, A, at_1, ..., at_j), where RI is the *Resource Identifier*, a unique identifier for the resource; and A is the *Availability*, a binary integer that represents whether the resource is active or not. at_i is the ith *Attribute*.

The dynamic profile basically represents the mobility of the resource providers. To describe the current status of mobile nodes while protecting their location privacy, the mobility of resource providers is described as a sequence of coordinate differences compared to HP. The dynamic profile is composed of tuples of the form (DP, rdm_1, ..., rdm_k), where DP is the *Difference of Position* that describes the difference in the coordinates compared to the last update, and rdm_i is the ith *Resource Description Modifier*, which comprises the corresponding resource identifier and optional updates. It enable to modify the availability and attributes of the corresponding resource registered with the static profile.

While a publication message is composed of a static profile or a dynamic profile, a subscription message is composed of tuples of the form (SA, TA, rd) where SA is the socket address of a resource consumer. TA is a target area that contains the requested resources. On the other hand, a notification message is composed of a list of data elements (SA, rd) which contains the socket address of a discovered resource provider.

5.3.3 Geographical locality-aware discovery scope selection

The key issue in wide-area resource discovery is the selection of the discovery scope. The topological scope of the subscription should be sufficiently small, while still covering the geographical destination. However, it is difficult to specify an appropriate topological scope corresponding to the requested geographical destination. Although GeoNetworking enables such routing, it is basically only applicable to a single GeoNetworking-enabled network.

The proposed system thus exploits the *Geographical Virtual Link (GVL) area* concept of GN6ASL. A *GVL* is a logical link covering multiple physical links, separated by geographical boundaries [39]. GVL is used by creating associations with a corresponding geographical area (*GVL area*). Because each GVL is associated with a certain geographical area, the Geocast packets delivered to a certain GVL are propagated only within its associated geographical area. This feature can be used by mobile RRs to determine whether the destination area is reachable using the geographical routing. In other words, the nodes in a place covered by a GVL area can be reachable using the geographical routing through its associated GVL.

By exploiting the above feature of GVL, the proposed discovery scope selection is performed as follows. When a mobile RR receives a subscription message containing a description of a requested resource and a destination area, the mobile RR checks whether it has a network interface supporting GeoNetworking. If there is a GeoNetworking-enabled interface, the mobile RR calculates the *intersection area* of the destination area for the subscription and the GVL area of the network interface. If the size of the intersection area is larger than a certain threshold, the mobile RR directly delivers the subscription as a GeoBroadcast packet to the mobile RRs inside the destination area.

On the other hand, if the size of the intersection area is smaller than the threshold, or the mobile RR does not have a GeoNetworking-enabled interface, the subscription is delivered as a unicast packet to a core RR instead. For instance, in Figure 5.2, subscriber S knows the size of geographical area GVL_i. Because area gd_i, an intersection

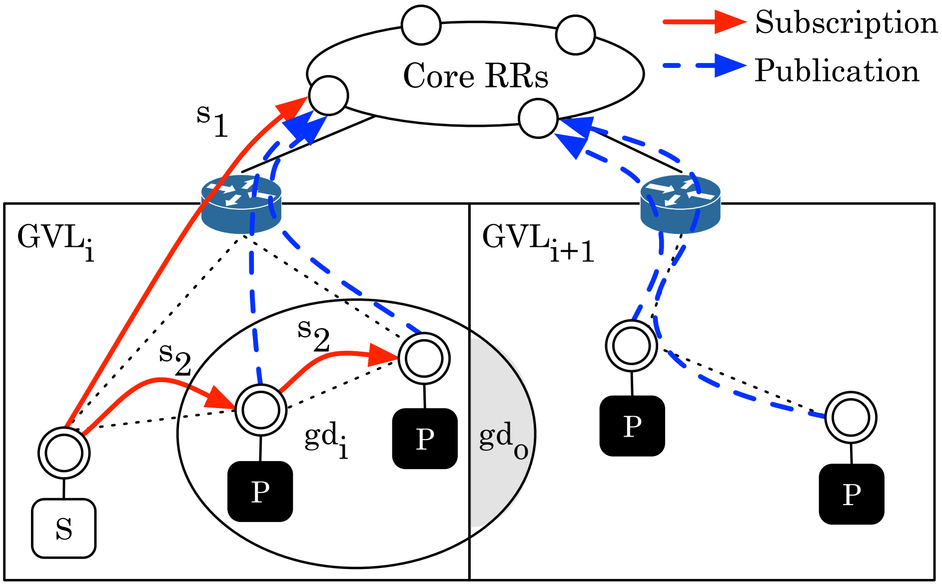


Figure 5.2: Discovery scope selection. Geocast-based local area discovery (path s_2) is performed only when gd_i is larger than a certain threshold.

area of $gd = gd_i + gd_o$ and GVL_i is large enough, the mobile RR connected to S delivers a subscription message to gd_i (path s_2) instead of the core RRs (path S_1).

The algorithm for the scope selection invoked in mobile RRs is therefore described as follows:

```
$s \leftarrow$ received SUBSCRIPTION$\{rd, gd\}$
**for all** interface $i_k \in I$ **do**
  **if** $i_k$ supports GeoNetworking **then**
    $g \leftarrow$ GVL area of $i_k$
    **if** $s.gd \cup g \neq \phi \wedge \frac{|s.gd \cup g|}{|s.gd|} \geq r_{gd}$ **then**
      **geocast** $s$ to the area $s.gd$
    **end if**
  **end if**
**end for**
**if** $s$ is not delivered yet **then**
  **unicast** $s$ to a core RR
**end if**
```

where *rd* is the resource description, *gd* is the geographical destination of the corresponding subscription, I is a list of active network interfaces, and $r_{gd}(0 < r_{gd} \leq 1)$ is the minimum ratio of the size of the intersection area for *gd*, which is used as the threshold.

Some of the existing solutions also support multiple discovery scopes, such as [46, 44]. However, they need a static configuration. The proposed mechanism, on the other hand, can dynamically select the scope according to descriptions of subscriptions and underlying network topologies.

5.3.4 Mobility aware event-driven location update

In the proposed system, the geographical locations of publishers are stored in core RRs in a distributed manner. Mobile RRs update their locations by sending their current locations to one of the core RRs. Then, the core RR propagates these locations to the other core RRs. These RRs manage locations using two mechanisms: *Home position registration*, and *Event driven location update*, which are performed as a part of the publication. The description of each mechanism is described as follows:

Home position registration is performed only on the boot up of mobile RRs: when a mobile RR is connected to one of the core RRs, it registers to the core RR he home position P_{home}, which consists of its current latitude and longitude (obtained e.g., by in-vehicle GPS) as one of the data elements in the *static profile* described in Chapter 5.3.2. Note that it is assumed that each publisher registers its resources to a mobile broker a priori.

Event driven location update is performed as a part of the publication when mobile RRs change speed, heading, and/or position over certain thresholds. Subsequent to *Home position registration*, each mobile RR sends the difference in position Δp between P_{home}, and its current position. In the next update, it also sends Δp which contains the difference in position from the last update. Using P_{home} and the Δp sequence, core RRs can calculate the geographical position of a publisher at a certain time t. The geographical position at a certain time P_t is calculated as $P_{t+1} = P_t + \sum_{i=1}^{t} \Delta p_i$, where Δp_i is the Δp at the ith location update. P_0 corresponds to P_{home}. Figure 5.3 and 5.4 show the pseudo-code for the location update mechanisms invoked in mobile and core RRs, respectively.

In addition to these location update mechanisms, to recover from location update failures (for example, from packet loss), a set of error correction mechanism is performed in the core and mobile RRs: at a certain interval, both RRs exchange the position difference between P_{home} and the current up-to-date position. If this difference contains a certain amount of error, each mobile RR sends error correction information to a core RR: how much the difference needs to be adjusted. These three mechanisms are sequentially invoked so that the above-mentioned two location updates and the error correction do not conflict with each other.

The main advantage of the proposed mechanism is that, compared to the periodic update, it can reduce the number of updates. Furthermore, it can avoid propagating the complete geographical position; the geographical coordinates are delivered to core RRs only once. Malicious nodes cannot obtain the complete geographical position unless they obtain P_{home} and the entire Δp sequence. It thus contributes to location privacy protection.

5.4 Performance analysis

This section evaluates the location update accuracy and the control overhead of the proposed system via a numerical analysis and simulations. This section presents the following two performance studies:

- **Accuracy evaluation using the ns-3 simulator:** first, the accuracy of the location update mechanism is studied using the ns-3 network simulator. This evaluation focuses on the impact of the location update frequency and a failure in the accuracy of the updated location on core RRs.
- **Cost analysis using numerical model:** next, the control overhead of the proposed system among RRs is studied by describing a set of simple analytical models. This evaluation focuses on the impact of the size of the network (e.g., the number of mobile nodes, size of the GVL area, etc.) on the data size of the control messages generated by the core and mobile RRs.

5.4.1 Accuracy of location update

The accuracy of the location update from the mobile to core RRs depends on the mobility of the mobile nodes, frequency of the location update, and location update failure

Table 5.1: Mobility settings for accuracy evaluation.

Parameters	Medium mobility	High mobility
Maximum speed	60 km/h	80 km/h
Turning speed	10 km/h	20 km/h
Acceleration	1.5 m/s^2	2.5 m/s^2
Deceleration	1.5 m/s^2	2.5 m/s^2

Table 5.2: System settings for accuracy evaluation.

Parameters	Value
Simulation duration	1 h
Speed threshold (ΔS)	5 km/h
Heading threshold (ΔH)	5°
Distance threshold (ΔD)	0.5 km
Error correction threshold (th_{error})	25 m
Error check interval	60 s
Error check time after turning	5 s

ratio (for example, from packet loss). This research thus evaluates the location update accuracy of a pair of mobile and core RRs under different mobility patterns, as well as the reliability ratio of successful location update r_p, using the ns-3 network simulator.

Simulation settings

In this simulation, two mobility patterns are used: medium and high mobility in terms of the node velocity. Under each mobility pattern, the mobile RR moves in a randomly selected direction. Its behavior is based on the random walk model, but it constantly accelerates/decelerates after/before corners. This model is implemented by extending the random walk model in ns-3. Table 5.1 shows the settings for each mobility pattern. Three settings for $r_p = \{0.1, 0.3, 0.5\}$ were used for each mobility pattern. This means that a location update from the mobile to core RRs did not succeed with probabilities of 10%, 30%, and 50%. The other settings are shown in Table 5.2.

Evaluation results

In the medium mobility scenario, the average error sizes are 9, 15, and 24 m for r_p values of 0.1, 0.3, and 0.5, respectively. In these cases, the numbers of successful location updates are 846, 662, and 464, while the numbers of error corrections are 51, 57, and 76, respectively. For the high mobility setting, the average error sizes are 10, 15, and 23 m for the r_p values, respectively. The numbers of successful location updates are 974, 762, and 541, while the numbers of error corrections are 82, 78, and 84, respectively. The distribution of the location update error for each mobility is shown in Figure 5.5 and 5.6. These results show that the size of the location update error is 25 m in the worst case, which corresponds to (i) the size of an edge of a small intersection (double lane with sidewalk on each side), (ii) around the length of 5 passenger vehicles without the inter-vehicular distance, or (iii) around a typical inter-vehicle distance in a suburban area. Consequently, the proposed system is sufficiently capable of locating resources inside an intersection, and within an inter-vehicle distance.

Compared to the periodic update mechanism that relies on GPS, in which a location update is invoked every time nodes obtain a new position from GPS (typically, every second), the proposed mechanism can reduce the number of packets. For instance, in the medium mobility scenario, it reduced the number of update by about 80%: 662 + 57 updates for the proposed mechanism (r_p = 0.3) compared to 3600 updates for the periodic update mechanism.

5.4.2 Control cost analysis

System parameters

The control overhead of the proposed system is evaluated by presenting a simple analytical model. This evaluation focuses on the impact of the network size (e.g., the number of mobile nodes, size of the GVL area, etc.) on the data size of the control messages generated by core and mobile RRs. The analytical model is designed to study the total size of the control messages generated by the proposed system, including publication, subscription, and notification messages for each discovery scope. It models the data size of the control messages for (i) one VANET, and (ii) the core RR network. All of the notations and settings used in the analysis are found in Table 5.3 and 5.4.

Table 5.3: Notation for model variables.

Symbol	Definition
N_{cb}	Number of core RRs in the system
N_p	Number of publishers in the system
N_s	Number of subscribers in the system
$N_p^{(m)}$	Number of publishers in one VANET
$N_s^{(m)}$	Number of subscribers in one VANET
$N_p^{(gd)}$	Number of mobile RRs inside geographical destination
p_{gd}	Probability that the ratio of the intersection area of destination area and the local network is larger than r_{gd}
r_{local}	Local area discovery invocation ratio
C_{mobile}	Total control overhead for one VANET
C_{core}	Total control overhead for all of the core RR networks

The system is composed of N_{cb} core RRs, N_p publishers, N_s subscribers, and an arbitrary number of mobile RRs. The number of mobile RRs is not specified in this analysis because the control overhead depends only on the number of publishers and subscribers. Mobile RRs are connected or directly installed to mobile routers, which support both ad-hoc multi hop routing by GeoNetworking and conventional IP routing protocols (possibly using multiple access interfaces for each routing, e.g., WLAN and cellular). Publishers and subscribers are connected to one of the mobile RRs. Each publisher has at least one resource. Mobile RRs compose GeoNetworking-enabled VANETs bounded to a GVL area. It is assumed that the entire geographical area in this evaluation is separated into 2D-grid shaped GVL areas, and each area is assigned to one of the VANETs. The size of each GVL area is assumed to be the same. The mobile RRs are uniformly distributed to the networks.

The scope for each discovery, (i) in-network geocast-based discovery or (ii) unicast-based discovery to core RRs, depends on the location and size of the destination area for the discovery, the number of mobile networks, and the size of each VANET (GVL area). For simplicity, it is assumed that the destination area has a circular shape with a fixed radius l_r, and the location of the destination area is randomly selected from all of the VANETs. r_{gd} is fixed at 0.5. In this scenario, the probability that a subscriber specifies a destination area inside its own VANET (local network) is p_l. Inside the

Table 5.4: Notation for model parameters and settings.

Symbol	Definition	Value
p_l	Probability that a subscriber specifies a destination inside its own VANET	0.5
λ_s	Ratio of subscriptions per unit time	1
λ_n	Ratio of notifications per unit time	1
λ_p	Ratio of publications (location update)	0.26
λ_{pe}	Ratio of location update error correction	0.01
h_{mm}	Hop count between mobile nodes	5
h_{mg}	Hop count (mobile node/access router)	1
l_e	Length of edges of a VANET	500 m
l_r	Radius of the destination area	50 m
l_s	Packet length of subscription message	256
l_n	Packet length of notification message	512
l_{sp}	Packet length of publication (static profile)	640
l_{dp}	Packet length of publication (dynamic profile)	128
l_{lu}	Packet length of overlay lookup message	64

local network, the subscriber uniformly specifies a certain destination area. The probability that the ratio of the intersection area of the destination area and the local network is larger than 0.5 r_{gd} (in other words, the probability that a subscription message is delivered as geocast packets inside a VANET p_{gd}) is thus given by

$$p_{gd} = \frac{{l_e}^2 - 2{l_r}^2}{l_e^2},$$

where l_e is the length of the edge of a VANET. Consequently, the probability that a subscriber performs a local area discovery r_{local} is $r_{local} = p_l p_{gd}$. In other words, $(1 - r_{local})$ indicates the probability of wide area discovery.

It is assumed that the average hop counts among the mobile RRs is 5 hops, according to the default hop limit of 5 specified in [38]. Note that the settings for the length of each packet include the underlying layers' protocol headers, such as LLC, MAC, GeoNetworking (GeoBroadcast), IPv6, and UDP. The publication and *location update* *error correction* ratios are derived from the above-mentioned accuracy evaluation, because it shows the amount of location update (publication) and error collection.

Cost for mobile network

The control overhead for a VANET is described as the sum of the subscription and notification messages between mobile RRs in the local and wide area discovery, and publication from mobile to core RRs. The number of mobile RRs that reply to the subscriber is derived according to the density of the mobile RRs and the size of the geographical destination for subscription. The control overhead for each VANET per unit time C_{mobile} is expressed by

$$\begin{aligned} C_{mobile} &= N_s^{(m)}\{r_{local}h_{mm}(\lambda_s l_s + N_p^{(gd)}\lambda_n l_n) \\ &\quad +(1 - r_{local})h_{mg}(\lambda_s l_s + N_p^{(gd)}\lambda_n l_n)\} \\ &\quad +N_p^{(m)}\{h_{mg}(l_{sp} + \lambda_p l_{dp} + \lambda_{pe} l_{dp})\}, \end{aligned}$$

where $N_s^{(m)}$ and $N_p^{(m)}$ are the numbers of subscribers and publishers in a single VANET. λ_s and λ_n are the subscription and notification ratios per unit time, while λ_p and λ_{pe} are the publication and location update error correction ratios per unit time, respectively. h_{mm} is the average hop count between mobile nodes, and h_{mg} is the hop count between mobile nodes and the gateway of a VANET (e.g., access router). l_s and l_n are the subscription and notification packet lengths, whereas l_{sp} and l_{dp} are the lengths of the packets containing the static and dynamic profiles, respectively. $N_p^{(gd)}$ is the number of mobile RRs inside the geographical destination. Thus, $N_p^{(gd)} = \pi {l_r}^2 \cdot N_p^{(m)}/{l_e}^2)$.

Cost for core RRs

The control overhead among the core RRs is the sum of the subscription and notification messages from/to mobile RRs in the wide area discovery, publication messages from mobile RRs, and lookup messages within the core RR network. The cost for the core RR network per unit time C_{core} yields

$$\begin{aligned} C_{core} &= N_s\{\lambda_s(1 - r_{local)} \cdot (l_s + l_{lu}(\log N_{cb} + 1)) + N_p^{(gd)}\lambda_n l_n\} \\ &\quad + N_p\{l_{sp} + \lambda_p(l_{dp} + l_{lu}(\log N_{cb} + 1)) \\ &\quad + \lambda_{pe}(l_{dp} + l_{lu}(\log N_{cb} + 1))\}, \end{aligned}$$

where l_{lu} is the length of the packet used to look up the core RR managing the requested geographical area using the subscription message. The control overhead for each core RR $C_{core}^{(i)}$ is therefore obtained as $C_{core}^{(i)} = C_{core}/N_{cb}$.

Numerical results

Cost for mobile network: Figure 5.7 shows the impact of the number of publishers in a VANET on the control overhead (C_{mobile}) under different numbers of subscribers. The total data size for the messages in one VANET is about 35 Mbytes/s at a maximum number of publishers and subscribers. Accordingly, as the number of subscribers increases, the data size for messages rapidly increases because the number of publishers in a particular area represents the density of the nodes replying to subscription messages.

In this analysis, it is assumed that the maximum number of nodes (i.e., publishers and subscribers) in one VANET is 1000, which corresponds to a typical ITS scenario in a congested urban area with 100% penetration (all vehicles support the proposed system). Suppose that there are 1000 vehicles in one VANET consisting of a 500 m^2 rectangular area, in which there are 20 grid-shaped, double-lane streets at 50 m intervals. It is assumed that the length of a vehicle is 5 m the inter-vehicle distance is 15 m. Therefore, there are 25 vehicles in a lane.

This result shows, for ITS, the impact of the penetration of the proposed system on vehicles (how many vehicles support the proposed system). It also shows the control overhead in a single VANET with penetration ratios of 10%, 25%, 50%, and 100% in a congested urban area: even if all the vehicles support the system, the data size of the control messages for each node is 35 Kbytes/s. The proposed system therefore needs a fairly low data size for the control messages.

Cost for core RRs: Figure 5.8 shows the impact of the number of core RRs on the control overhead in the entire core RR network under several different numbers of publishers and subscribers.

In the following analysis, two different sets of settings are used, which represent the *small* and *large* scale scenarios. In the small scale scenario, the proposed system is assumed to be deployed in a single country, in which the maximum number of nodes (i.e., publishers and subscribers) in the system is 75 million. This setting corresponds to the number of vehicles in use in Japan as of 2012 [53]. On the other hand, in the *large* scale scenario, the maximum number of nodes is 1 billion, which corresponds to the number of vehicles in the world, or the number of smartphones in use in the world as of 2012. These results therefore show the control overhead in the entire core RR network with penetration ratios of 10%, 25%, 50%, and 100%. They show that the total control overhead for the entire core RR network does not depend so much on the number of core RRs, but rather on the number of publishers and subscribers. This result implies that system administrators can add an arbitrary number of core RRs to reduce the load of each broker according to the population of users.

Figure 5.9 shows the impact of the number of core RRs on the control overhead for each core RR. In the small scale scenario shown in Figure 5.9(a), by installing 500 core RRs, even in with the maximum (75 million) number of modes, the overhead for each core RR can be reduced to less than 500 Mbytes/s. On the other hand, in the large scale scenario shown in Figure 5.9(b), using 10,000 core RRs, the control overhead for each core RR can be limited to under 200 Mbyte/s in the case of the maximum (1 billion) number of nodes.

Figure 5.10 and 5.11 show the impact of the number of publishers and subscribers on the control overhead for each core RR. According to Figure 5.10(a) and 5.11(a), in the small scale scenario, if the overhead for each core RR must be reduced to 100 Mbytes/s, it is necessary to install 100 core RRs in the 100% penetration scenario. In the case of the maximum number of nodes (1 billion) for the large scale scenario, the cost of each core RR is excessively high if there are only 1000 core RRs, while when using 10,000 core RRs, the cost can be less than 300 Mbytes/s.

5.5 Summary

This chapter presented a wide-area geographical publish/subscribe resource discovery system focused primarily on vehicular communication systems. The proposed system combines a geographically-distributed overlay network and IPv6 GeoNetworking. It relies on a hierarchical publish/subscribe architecture based not only on conventional routing technologies but also on the geographic routing so that users can rapidly locate resources according to geographical coordinates. Evaluation results showed that the proposed system reduced the control messages for the publication and subscription processes without losing the accuracy of publishers' locations. It was observed that the control overheads for both VANETs and the core RR network were fairly small.

```
Location update (invoked on boot):
  initialize $s_{prev}$, $h_{prev}$, $p_{prev}$
  $rds \leftarrow$ descriptions of available resource
  $p_{home} \leftarrow$ current geographical position
  unicast $p_{home}$, $rds$, and timestamp to a core RR
  loop
    $s_{curr} \leftarrow$ current speed
    $h_{curr} \leftarrow$ current heading
    $p_{curr} \leftarrow$ current geographical position
    $\Delta s \leftarrow s_{curr} - s_{prev}$
    $\Delta h \leftarrow h_{curr} - h_{prev}$
    $\Delta p \leftarrow$ difference of coordinates between $p_{curr}$ and $p_{prev}$
    if $|\Delta s| > th_s \vee |\Delta h| > th_h \vee |\Delta p| > th_p$ then
      unicast $lu$ ($\Delta p$ and $rd$) to a core RR
    end if
    $s_{prev} \leftarrow s_{curr}$
    $h_{prev} \leftarrow h_{curr}$
    $p_{prev} \leftarrow p_{curr}$
  end loop

Error correction (invoked after changing mobility or after a certain period of time):
  $p_{curr} \leftarrow$ current geographical position
  request and receive error correction message $ec_{rep}$ from core RR
  $\delta p \leftarrow$ difference of coordinates between $p_{curr}$ and $p_{home}$
  if $|\delta p - ec_{rep}.\delta p| > th_e$ then
    unicast error correction message $ec_{corr}$ with $\delta p$
  end if
```

Figure 5.3: Pseudo-code of location update in mobile RRs.

```
Location update:
  initialize resource description list $L_{rd}$
  loop
    receive location update $lu$ with resource description $rd_m$ from mobile RR
    if $lu$ contains $\Delta p$ then
      $rm_c \leftarrow$ get an entry from $L_{rd}$ (key = resourceID in $rd_m$)
      $rm_c.pd_{curr} \leftarrow m_c.pd_{curr} + lu.\Delta p$
    else
      add an entry to $L_{rd}$ as $(rd, lu.p_{home})$
    end if
  end loop

Error correction:
  receive error correction request $ec_{req}$ from mobile RR
  $\delta p \leftarrow$ difference of coordinates between $p_{curr}$ and $p_{home}$ of corresponding resource
  unicast $deltap$ as $ec_{rep}$ to mobile RR
  receive error correction message $ec_{corr}$ from mobile RR
  correct corresponding position of resources using $ec_{corr}.\delta p$
```

Figure 5.4: Pseudo-code of location update in core RRs.

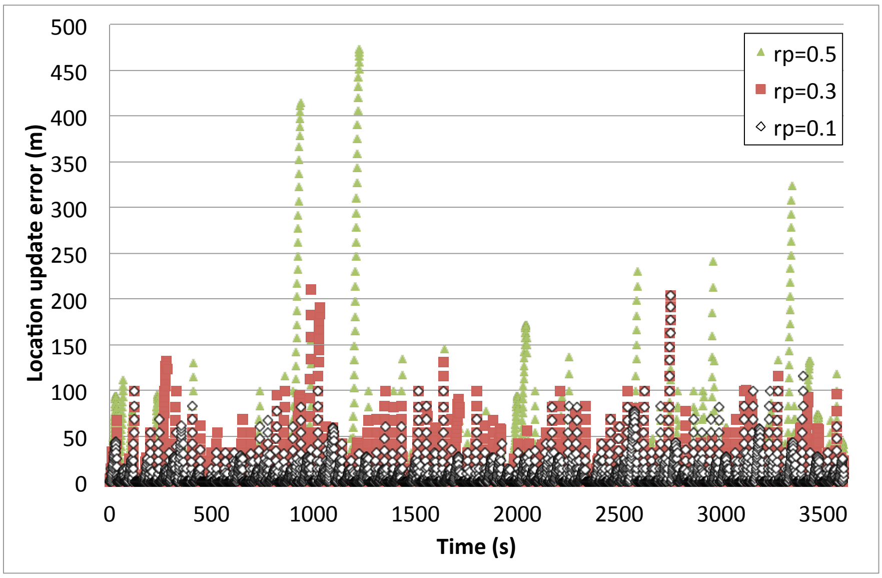


Figure 5.5: Accuracy of location update (Medium mobility).

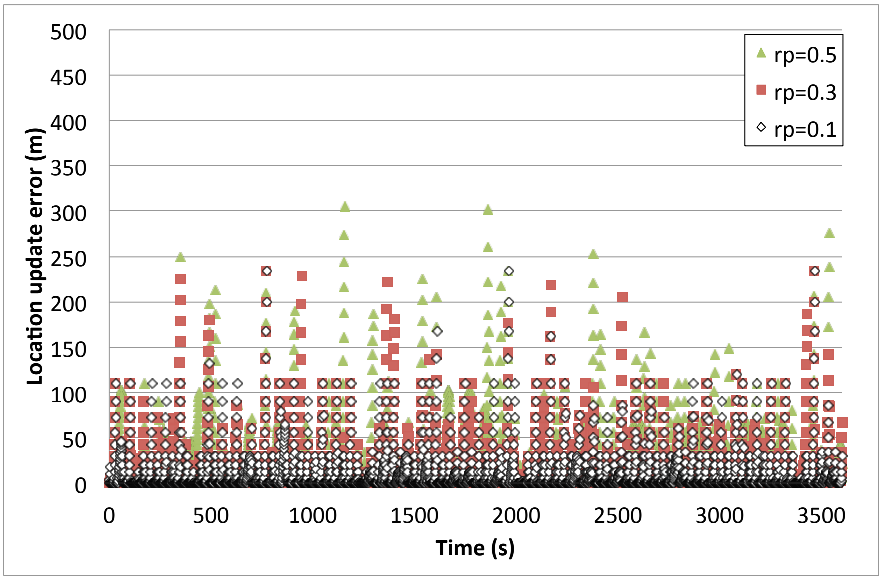


Figure 5.6: Accuracy of location update (High mobility).

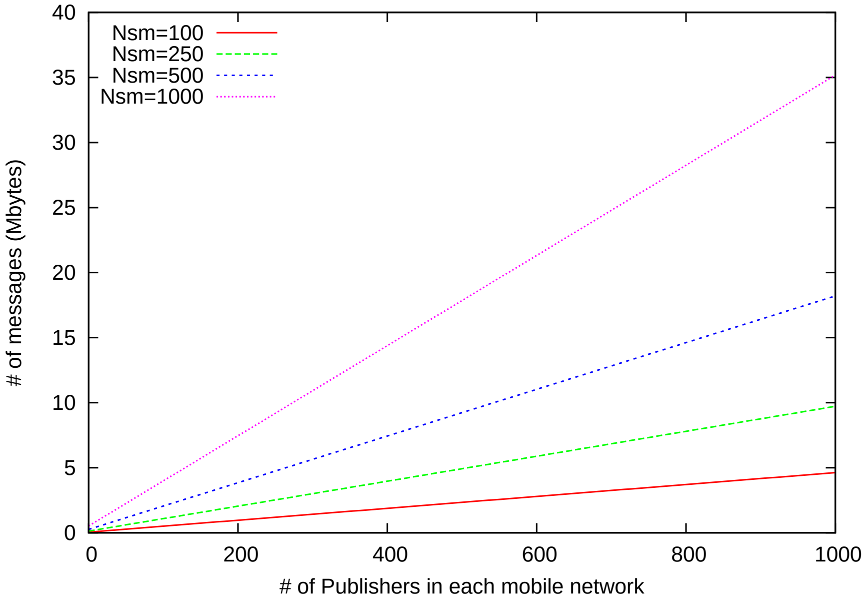
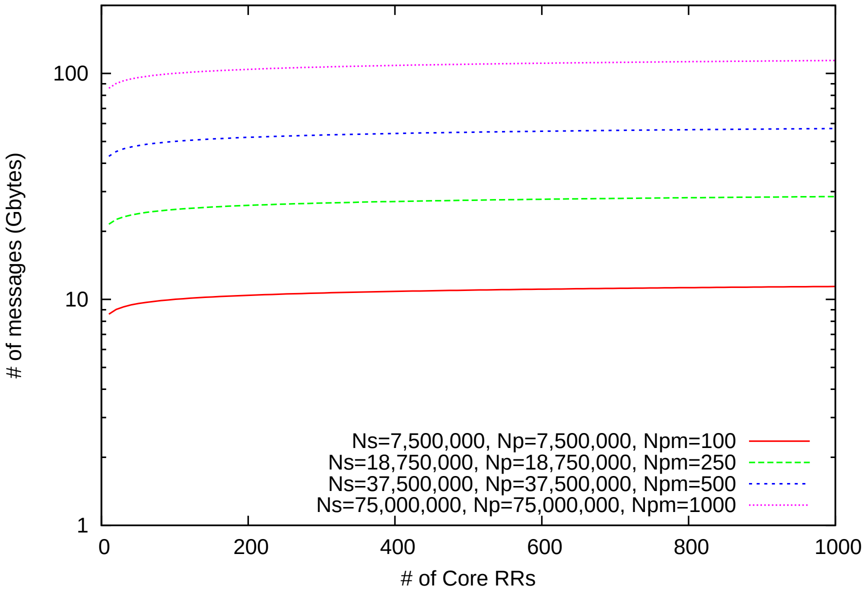
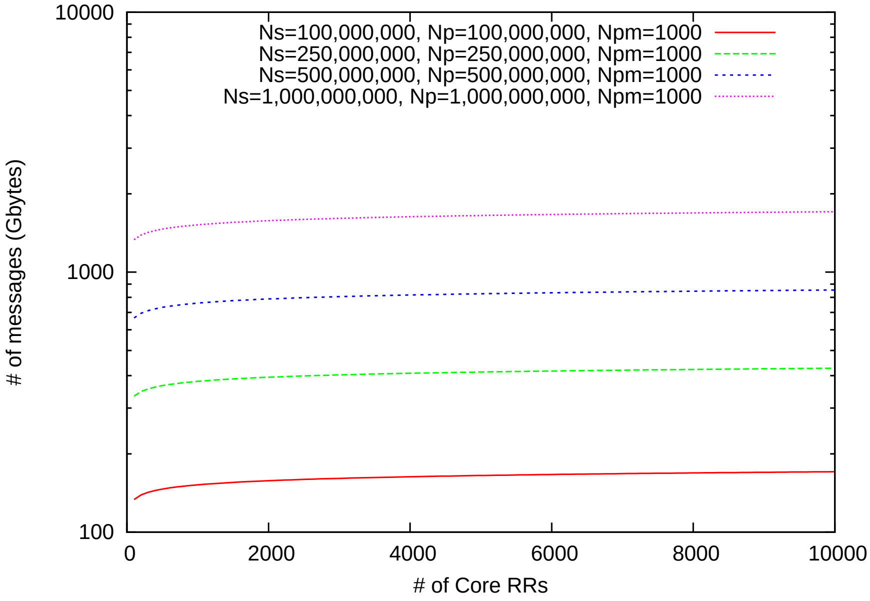


Figure 5.7: Impact of number of publishers on control overhead in one VANET.

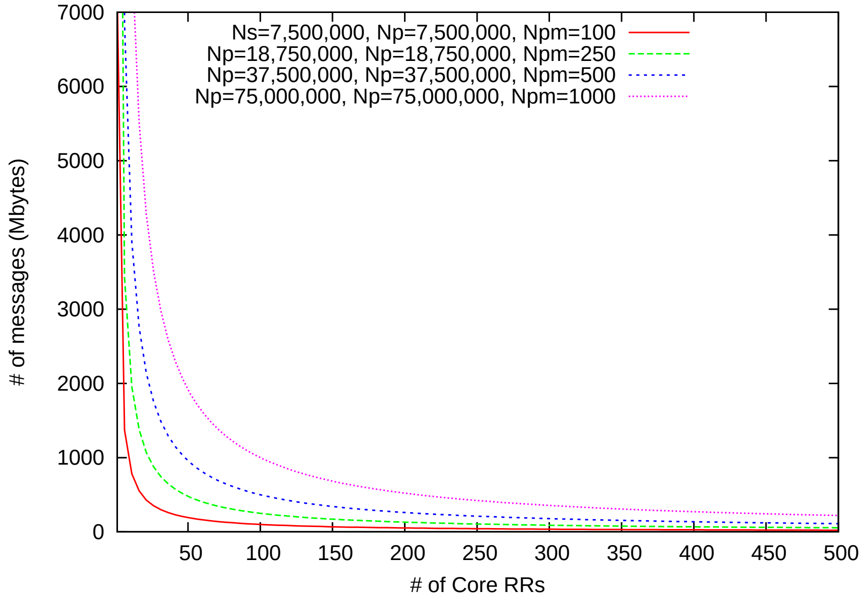


(a) Small scale scenario

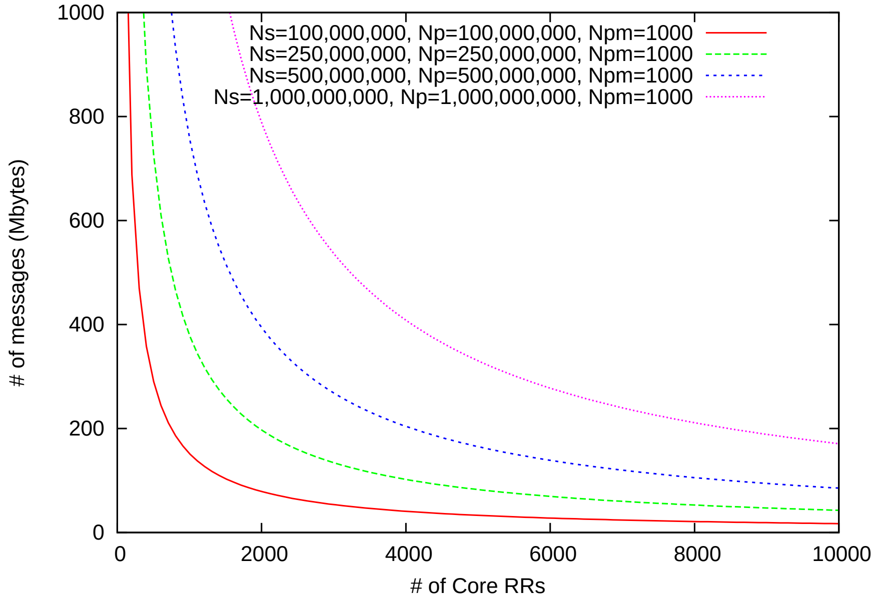


(b) Large scale scenario

Figure 5.8: Impact of number of core brokers on control overhead to entire core RR network.

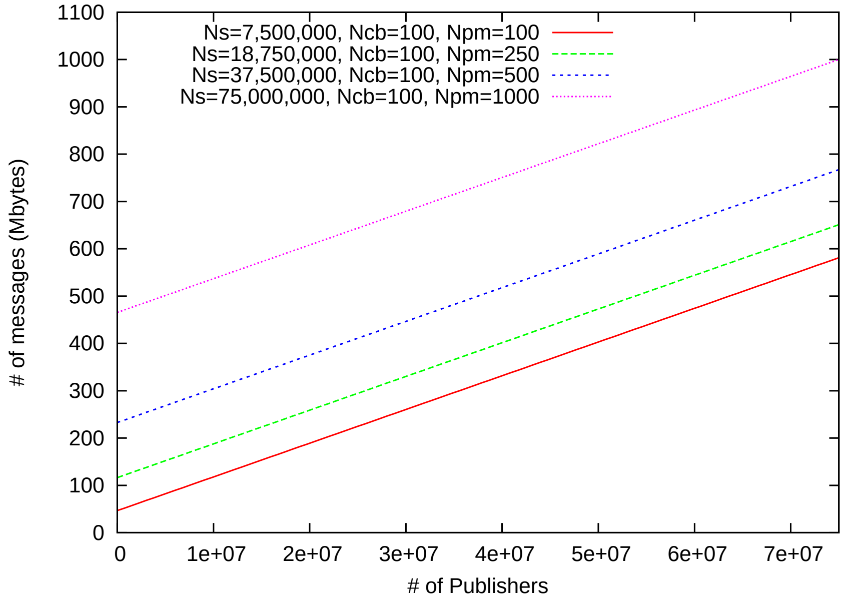


(a) Small scale scenario

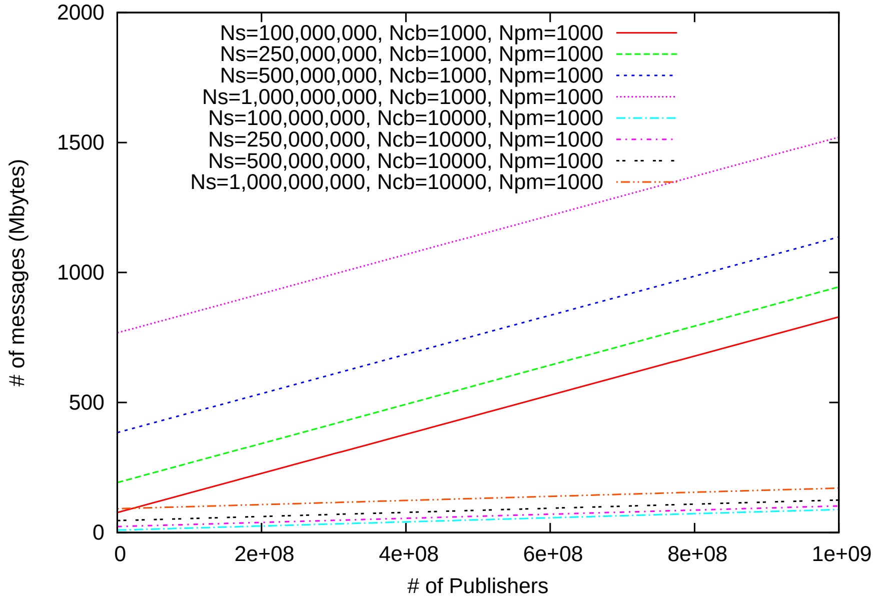


(b) Large scale scenario

Figure 5.9: Impact of number of core brokers on control overhead to each core RR.

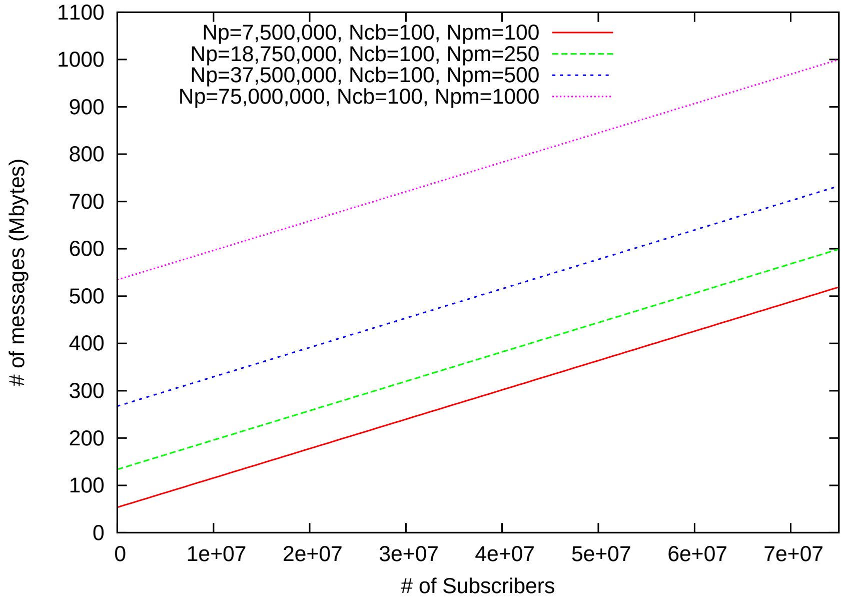


(a) Small scale scenario

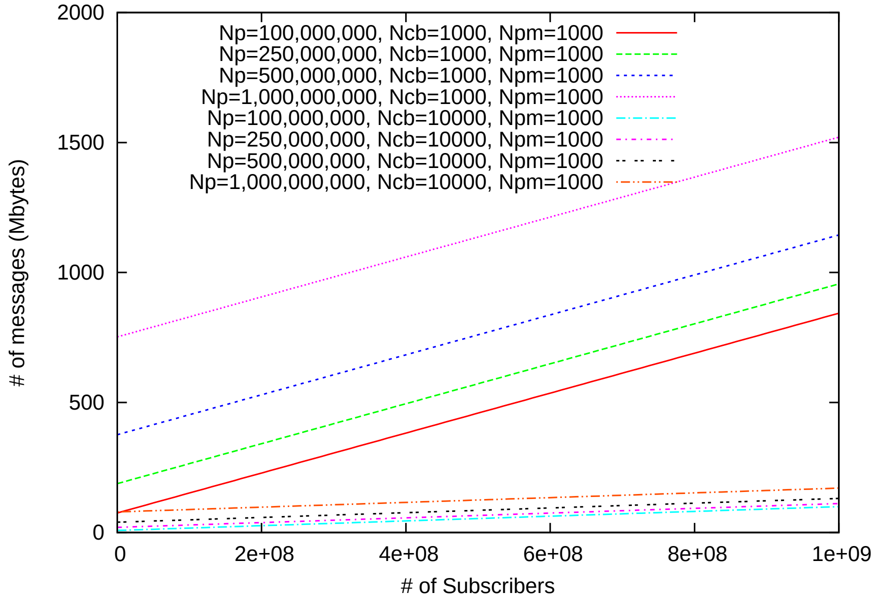


(b) Large scale scenario

Figure 5.10: Impact of number of publishers on control overhead to each core RR.



(a) Small scale scenario



(b) Large scale scenario

Figure 5.11: Impact of number of subscribers on control overhead to each core RR.

Chapter 6

Discussion

This dissertation has proposed a geographical resource discovery system for C-ITS applications mainly integrated into VCS, which exploits the publish/subscribe scheme and IPv6 GeoNetworking. Before concluding the dissertation, this Chapter evaluates the proposed system from the point of view of the requirements described in Chapter 2 to show whether it can be applied to the considered scenarios. This Chapter also discusses the possibility to improve the system towards a common infrastructure of resource discovery in the future.

6.1 On functional requirements

The proposed system has been designed at first to support resource discovery locally, e.g., in mobile networks such as VANETs, then extended to support wide-area resource discovery. Its basic design criteria have been defined in Chapter 2.2: *universal visibility, geographical location based discovery, incremental and extensible deployment*.

In terms of the *universal visibility*, the proposed system is able to discover any resource as long as its resource description is registered to one of the resource registries thanks to IPv6 and hierarchically clustered resource registries. Since the system is built on the hierarchically-clustered architecture where core and mobile resource registries compose a number of groups, and each mobile resource registry is attached to one of the core resource registries, each resource registry works as a part of a large-scale distributed database. By exploiting the structured overlay networking technologies, registered information in a registry is propagated to the core registries, therefore any registry can access any information.

Although mobile resource registries are designed to use IPv6 GeoNetworking to connect to other mobile resource registries, other networking protocols can be used as long as they support IPv6. This capability allows using multiple access technologies, for instance, mobile nodes hosting resource registry are able to use both WLAN with GeoNetworking and cellular with the traditional IP routing. Such a usage can also helps the incremental deployment, because not all networks may be able to support IPv6 GeoNetworking at the early stage of deployment.

Regarding the *geographical location based discovery*, the proposed system allows resource consumers to discover resources according to geographical position mainly using IPv6 GeoNetworking. Since resource providers, resource registries or mobile resource registries append their geographical coordinates when a resource description is registered, all resource description contain current position of the resource provider or closest resource registry. In the resource discovery phase, the geographical coordinates are used as a key field to lookup resource registries. While existing resource discovery solutions can take care of geographical position as an optional attribute, the proposed system uses it as a mandatory data element. In addition to using geographical position as a search key, by using IPv6 GeoNetworking, the proposed system also contributes to rapid and scalable discovery; geographical coordinates of resource consumers and providers are also used for the geographical routing in IPv6 GeoNetworking.

For the *incremental and extensible deployment*, as discussed above, the proposed system employs the hierarchical crusted architecture in which each resource registry can dynamically be added and deleted. The core resource registry network is composed as a structured overlay network where each peer (resource registry) can join and leave the network at any time; structured overlay routing algorithms enable peers to manage registered information in a distributed manner, therefore even if a peer leaves the network, registered data on this peer are moved to other peers staying in the network. These characteristics are well suited for the extensible deployment; the core resource registries can be allocated in accordance with the population of the resource consumers and providers. Further, the proposed system does not require any change of the existing Internet architecture: it merely adds mobile and core resource registries, keeping traditional IP routing as is. Therefore existing core routers do not have to be re-configured. In other words, intermediate routers do not have to support IPv6

GeoNetworking and the proposed mechanism.

6.2 On performance requirements

The performance requirements are also defined in Chapter 2.2 in terms of scalability and latency. Regarding the scalability in a mobile network, even if there are 1,000 resource consumers and providers in one mobile network, the proposed system only generates about 35 Mbytes/s, as shown in Chapter 5. In the case of the control overhead on the core resource registry network, although the total overhead is linearly increased according to the number of nodes, the overhead on each core resource registry is distributed by adding core resource registries because of the characteristics of the geographical overlay network: registered information is stored to geographically-neighboring resource registries, and resource consumers in general try to discover resources in their neighboring area (or along their driving path). It can be observed from these characteristics that the traffic load to core resource registries is not directly related to the total number of resource consumers and providers. Even in the case of a maximum number of nodes (1 billion nodes in the system), the numerical results in Chapter 5 showed the cost on each core resource registry can be less that 200 Mbytes/s when there are 10,000 core resource registries. In the single-country case (7.4 million nodes in the system), it can be 500 Mbytes/s by installing 20 core resource registries.

In terms of the latency requirement, in the simulation conducted in Chapter 4, more than 90 % of the resource discovery requests completed within 100 ms even in the high mobility scenarios. In the outdoor evaluation using the actual implementation on Linux PC, all the discovery requests have been terminated within 200 ms and 90% of them have been finished within 100 ms. These results show that the proposed system meets the latency requirement for each class of application, such as less than 100 ms for road safety, 200 ms to 500 ms for traffic efficiency and less than 500 ms for others.

6.3 Applicability

6.3.1 To ongoing ITS standardization efforts

The proposed system can be applied to the ongoing ITS standardization efforts, i.e., the ITS station reference architecture mentioned in Chapter 2.5. In this architecture, resource discovery mechanisms of the proposed system are able to be integrated into the new *facilities* layer, which corresponds to the TCP/IP application layer, the OSI application layer, presentation layer and the session layer. The facilities layer has been designed to provide sharable and common functions to C-ITS applications, such as common database of neighboring area, communication support, etc. Since the resource discovery is a common function for all applications, the proposed system meets the objective of the facilities layer. While the facilities layer only considers a local-area resource discovery by using broadcast-based protocols such as [37] by now, it can be extended to support the wide-area resource discovery.

As the ITS station reference architecture has been designed to support main components of the proposed mechanism, integration of the proposed mechanism to this architecture can be smoothly achieved: IPv6 and GeoNetworking is supported in the networking and transport layer [51, 52], while the database for registered resources is in the facilities layer (called *Local Dynamic Map*) [36].

The design of a possible integration of the proposed system to the ITS station reference architecture is outlined as follows. Mobile resource registries are integrated into *ITS-S routers* in vehicles and personal ITS stations because the ITS-S router is assumed to support IPv6 GeoNetworking, while core resource registries are in ITS-S hosts in central ITS stations. Resource consumers and providers are implemented as applications, therefore they can be integrated into both in *ITS-S host* or in *ITS-S router* in any type of ITS station, including vehicle, roadside, ITS center and personal device. Figure 6.1 and 6.2 show two potential integration models of the proposed mechanisms to the ITS station reference architecture: the one is a single host integration in which all functions are integrated into one ITS-S router, whereas the latter is a separated integration, in which resource consumer/providers (i.e., applications) are integrated into ITS-S hosts and they use the resource registry in ITS-S routers via station-internal or external network.

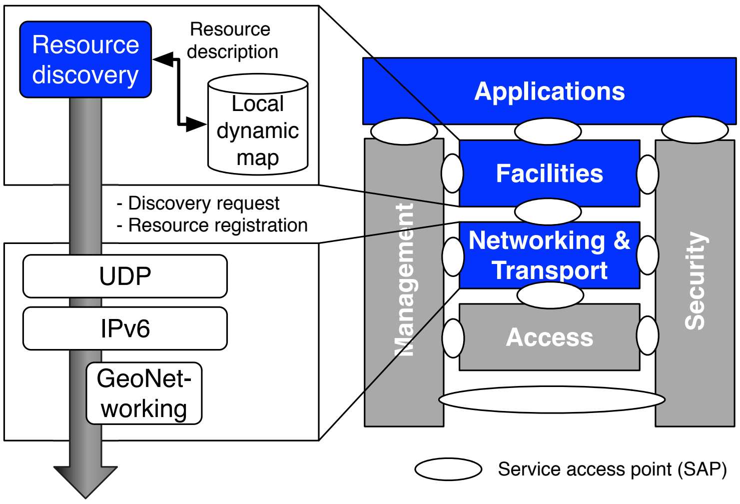


Figure 6.1: Design of a possible integration of the proposed system to the ITS station reference architecture (single node integration).

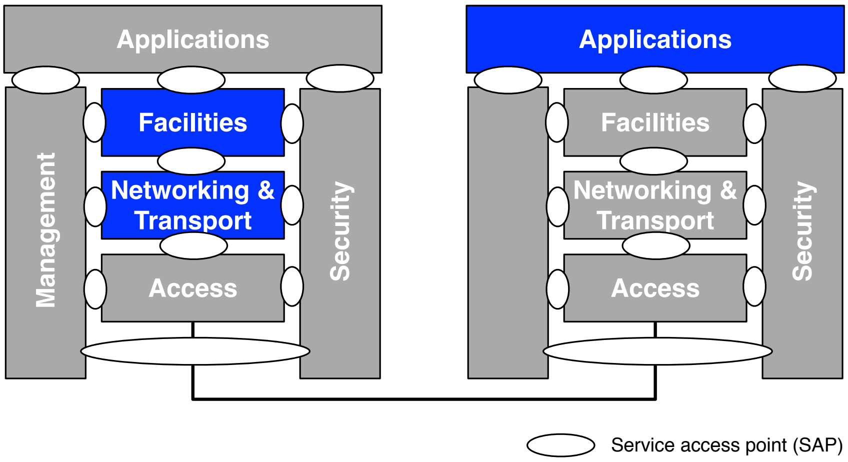


Figure 6.2: Design of a possible integration of the proposed system to the ITS station reference architecture (separated integration).

6.3.2 For other types of applications beyond C-ITS

While the proposed system is explicitly designed for C-ITS applications, applying the system to other types of scenarios is worth considering. A common infrastructure to discover mobile resources is obviously beneficial for any mobile applications. However, its applicability is carefully analyzed in terms of functionality and performance. As the conclusion of this Chapter, this Section discusses what kind of scenarios can be supported by the proposed system.

As described in the above Section, in summary, the proposed system is capable of mobile resource discovery with universal visibility. Users can dynamically register and discover mobile resources according to their geographical location. In addition, the system can be deployed incrementally. Further, it supports dynamic configuration of resources. Regarding the cost and performance, the proposed system can discover resources within 100ms at quickest, and supports 1 billion users at maximum. By installing a number of core resource registries, the cost on each registries is reasonably distributed. Under medium (36km/h) and high (72km/h) mobility with random walk model, the accuracy of location using the core resource registry is 23m and 25m, respectively.

According to these characteristics, the proposed system may also be used for other location based services for e.g., urban railway services which has similar mobility patterns to vehicles: scheduled trajectory and limited speed. In the future, it is necessary to analyze further usage to extend the proposed system as a truly-universal resource discovery infrastructure.

As described in the above Section, from the point of view of deployment, even without the ITS station reference architecture, the proposed system can be seamlessly integrated to the existing Internet, because it does neither require any re-configuration nor re-construction of the existing Internet architecture. By adding mobile resource registries and core resource registries to IPv6-enabled networks, resource consumers and providers can make use of basic functions of the proposed system.

Therefore, the proposed system can be deployed to the actual environments even if there are not enough access routers on the roadside by adjusting the size of the mobile network. The system is designed on an assumption that each mobile network is connected to the Internet via access router on the roadside, however, institutional and financial constraints may prevent this assumption coming true. Even in this case, the system can be deployed by configuring the *hop limit* parameter of GeoNetworking: a mobile network is composed using the GeoNetworking protocol which supports multi hop ad-hoc routing, therefore the size of the mobile network can be configured with the hop limit parameter. In the case where where there are not enough access routers to cover the entire zone, the connectivity can still be provided by adaptively reconfiguring the hop limit parameter. The means to perform this adaptive configuration is not described in this dissertation, but it can be implemented using a digital map which contains the location of infrastructure.

6.4 Open issues

The proposed scheme enables C-ITS applications to rapidly discover mobile resources in a large number of vehicles, specifically, this dissertation has focused on the basic architecture and communication mechanisms as a foundation of the future resource discovery system for C-ITS. This section therefore addresses open issues beyond the basic architecture and communication mechanisms to deploy and operate the resource discovery system in actual environments.

Location privacy: in the proposed scheme, resource providers need to publish their geographical position with descriptions of available resources. The geographical location of a resource represents the location of a node hosting the resource, therefore it must be protected by malicious users. In the proposed scheme, the full description of the geographical coordinates is delivered only once to a core resource registry. In other words, the geographical coordinates cannot be spoofed over the air thanks to the location update mechanism described in Chapter 5. However, the location information can be stolen from core resource registries because they aggregate the registered information from resource providers. For better location privacy protection, it is necessary to protect both (i) published information from resource registries over the air, and (ii) registered and stored information in resource registries from resource providers.

Data integrity: must also be considered since malicious users may not only steal published data but also modify them or add erroneous data. If the geographical position of a resource is skewed, several critical incidents may happen; it is easy to cause traffic accident, road congestion, wrong navigation etc. Potential solutions are authentication of subscribers, encryption of published packet and stored data in resource registries, however further solutions might be needed due to the latency and scalability requirements described in Chapter 2.

Resource selection: although this dissertation has addressed *how to discover resources* in the considered network, at the same time it is necessary to consider *how to select a resource in a number of discovered candidates*. Resource consumers discover resources according to a set of data elements including geographical position and resource specific attributes, in other words, the proposed discovery mechanism is a type of multi-key search; all resources matching the requested keys are noticed to resource consumers. In the case of applications using point-to-point or point-to-multipoint (except broadcast) messaging, it is therefore necessary to select an (a set of) appropriate resource(s) among the discovered candidates.

Obviously, *how to determine the most appropriate resource* is a task of applications instead of the discovery system, nonetheless, the system should provide some *metrics* to evaluate the applicability of resources to the usage of applications. For instance, even if several resources match a specific set of search keys, the suitability of these resources to an application can be different: a resource may be about to going out the requested geographical area, turing off the engine, or under unstable connectivity, etc.

Resource representation: there can be a large number of types of resource consumers and providers which share the proposed resource discovery scheme. Accordingly, a well-organized representation of resource should be specified so that different resource consumers and providers efficiently discover and register resources in the same manner.

The rule of *how to describe the resource* is realized as a *resource description language*, which specifies the format of the set of data elements for presenting a resource. The main requirements for the description language is the expressiveness, extensibility and size. The size is also important since it directly affects the bandwidth usage cause by publication and subscription. A number of potential solutions have been specified, such as DNS resource record [4], dedicated URL scheme [45], and XML [47]. Although these solutions are well organized to describe a resource, there is a missing capability: representation of relationship among other resources. It is necessary to investigate how to describe the relationship of resources by introducing some technology to represent semantics such as OWL [80].

Chapter 7

Conclusion

Recent development of mobile networking technologies has enabled a wide variety of mobile devices to communicate with each other. By connecting a huge number of mobile devices to the Internet, C-ITS applications, a new mobile location-based services, have now emerged. The capability of discovering software and hardware resources has became essential for C-ITS applications designed for these devices and services.

This dissertation studies how to discover resources based on their geographical position in VCS and designs a wide area geographical mobile resource discovery scheme used by C-ITS applications by taking into account the generic issues in the mobile resource discovery such as (i) heterogeneous connectivity, (ii) complex user demands, and (iii) a huge number of nodes. To conclude the research, this chapter summarizes the contributions presented in this dissertation and finally discusses the future perspective of resource discovery.

7.1 Contributions

First, Chapter 2 presented the context of this research by defining *what is resource discovery* and by describing VCS as part of C-ITS, which is the target of this research. Then, the existing resource discovery schemes were explored from the point of views of both small and large scale network, including LAN, MANET, VANET, and the Internet. These solutions were categorized using an abstraction model of resource discovery. This abstraction model was defined to present a taxonomy of the existing resource discovery schemes, which enables to investigate the solution for the wide area geographical resource discovery. Consequently, this chapter described a possible solution, which exploits the clustered framework in combination with the distributed framework.

Chapter 3 presented some of the practical use cases of C-ITS applications to show and investigate what is a C-ITS application and how to deploy it. Through the investigation, this chapter clarified the necessity of resource discovery and its underlying issues. The two use cases were presented, the first one was *Road traffic event detection and dissemination*, and the second was *Road traffic information dissemination from ITS center*. To evaluate these use cases, an application was actually implemented to Linux. This application was an in-vehicle driver support application using IPv6 GeoNetworking, a pair of GUI applications integrated into vehicles and an Internet host (acting as an ITS road traffic center). It relied on IPv6 GeoNetworking, a set of mechanism primally specified by the GeoNet project. Thanks to IPv6 GeoNetworking, which enables to deliver conventional IPv6 packets to a certain geographical area, the application realized the above-mentioned use cases. The outdoor evaluation was conducted as the final demonstration of the GeoNet project at INRIA, which showed the effectiveness of the application. Through the evaluation, this chapter showed the necessity of resource discovery: the static configuration of destination hosts was not applicable in the face of a huge number of mobile nodes.

Chapter 4 proposed the small area geographical mobile resource discovery mechanism for a single VANET. As described in Chapter 2, the basic strategy to the goal of this dissertation is to exploit the clustered framework. Therefore, Chapter 4 presented how to discover resources in a edge of the entire networks based on their geographical position. The presented solution is to harmonize the IPv6 multicast-based resource discovery protocol and IPv6 GeoNetworking. The IPv6 multicast-based resource discovery protocol, i.e., SLPv2, is an appropriate protocol to cooperate with IPv6 GeoNetworking, because both protocols efficiently use IPv6 multicast: SLPv2 delivers a discovery message as an IPv6 multicast packet, while IPv6 GeoNetworking encapsulates the IPv6 multicast packet as a GeoBroadcast packet. In addition to the combination of these protocols, this chapter presented the cross-layer GeoDestination management mechanism, by which SLPv2 can configure the size of GeoDestination stored in the GeoNetworking protocol. The proposed system was evaluated using simulation and actual measurements at NAIST. Evaluation results showed the proposed system could locate mobile resources rapidly without overloading the network.

Chapter 5 presented the wide area geographical mobile resource discovery scheme. The proposed mechanism is built on a publish/subscribe architecture in combination with the previously proposed mechanism in Chapter 4. Specifically, this chapter tackled one of the main research questions, *how can resources be discovered in a wide variety of heterogeneous networks?*: although Chapter 4 presented the geographical mobile resource discovery mechanism for a single VANET, the remaining issue is that a specific routing mechanism (e.g., GeoBroadcast) may not be allowed in some administrative domains. In addition, managing a huge number of resource descriptions generated from mobile nodes is a critical issue in terms of scalability and also consistency. Therefore, the proposed system exploited a hierarchical publish-subscribe architecture and a geographical routing so that resource consumers can locate resources according to their geographical position without scalability issue. Even if the geographical routing is not supported in a network, resource consumers can use traditional IP unicast instead of geocast. In addition, the proposed system includes a location management mechanism for mobile resources, which reduces periodic location updates from mobile publishers. Evaluation results showed that the proposed system discovered mobile resources without overloading both mobile and core networks.

In summary, this dissertation started from the investigation of the State of the Art of resource discovery techniques and their issues, potential solutions, and detailed evaluation of discovery scenarios and practical and actual implantation/field tests. Then the proposed system has been designed, as a first step, the resource discovery mechanism for a single VANET by harmonizing an IPv6 multicast-based discovery protocol and IPv6 GeoNetworking. Through the simulations and outdoor evaluations using actual implementations. The wide area geographical mobile resource discovery system was proposed, which relies on the publish-subscribe architecture and on the structured overlay routing mechanism. The performance evaluations showed the scalability and effectiveness of the proposed geographical resource discovery system.

7.2 Future perspective

This dissertation has focused on a wide-area geographical mobile resource discovery for C-ITS. The proposed mechanisms were at first designed to support resource discovery locally (e.g., in mobile networks such as VANETs), then extended as a system supporting the wide-area resource discovery in VCS. These mechanisms were evaluated in a number of communication scenarios of C-ITS applications for road safety and traffic efficiency. The proposed system can be deployed as a part of the emerging ITS station reference architecture. As of now each service provider deploys its own resource discovery mechanism, however, in the future, it must be an open and common infrastructure spanning the entire Internet so that every entity connected to the Internet can be rapidly discovered in the emerging era of all-mobile devices. Likewise DNS, such an open and universal infrastructure is essential for the robust and sustainable information system.

To become a common infrastructure for mobile devices, the author believes that the proposed resource discovery system needs to be extended to support *real-time* and *passive* discovery. Because mobile devices tend to be equipped with a wide variety of sensors such as video camera, microphone, accelerometer, laser sensor, weather sensor, and in-vehicle sensors, it can be much easy to obtain real-time environmental information from nearby areas. By aggregating and analyzing these real-time data, the resource discovery system can be a context aware service; the system detects a specific *situation*, then, offers a set of resources to relevant nodes prior to their discovery requests. This real-time and push-based discovery is considered to enable a much low-cost and rapid discovery, in which each node does not have to send discovery request but merely shares real-time sensing data among surrounding nodes. The system detects significant events and delivers necessary information of resources only to a set of relevant nodes. Therefore, it can reduce redundant requests and the length of time to detect resources. In many cases, it is not clear when a resource is actually necessary because the necessity of resources highly depends on contexts. The real-time passive resource discovery system can *offer* a potential resource, which is actually needed in a corresponding situation.

A possible solution of the real-time and passive discovery is to exploit the complex event processing (CEP) and context awareness technologies in combination with a cross layer harmonization: CEP is a type of event processing method, which aims at analyzing data streams to detect a certain event. It handles multiple data sources so that it can detect patterns of events in complicated situations. Conceptually, by applying CEP to the resource registries in the proposed scheme, it can support the real-time and passive resource discovery. While further improvement is also needed to handle the large amount of publications (e.g., real-time sensing data) from resource providers, the author believes this solution is a first step to the next-generation resource discovery system.

References

[1] CAR 2 CAR Communication Consortium. `http://www.car-to-car.org/`.

[2] CarGeo6. `http://www.cargeo6.org/`.

[3] CVIS project. `http://www.cvisproject.org/`.

[4] DNS-based Service Discovery. `http://www.dns-sd.org/`.

[5] ETSI TC ITS. `http://www.etsi.org/website/technologies/intelligenttransportsystems.aspx`.

[6] GeoNet project. `http://www.geonet-project.eu/`.

[7] gpsd-a GPS service daemon. `http://www.catb.org/gpsd/`.

[8] IPv6 Multicast Address Space Registry. `http://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xml`.

[9] ISO TC 204 Intelligent transport systems. `http://www.iso.org/iso/iso_technical_committee?commid=54706`.

[10] Multicast DNS. `http://www.multicastdns.org/`.

[11] ns-3. `http://www.nsnam.org/`.

[12] openSLP Service Location Protocol. `http://www.openslp.org/`.

[13] UDDI Version 3.0.2. `http://uddi.org/pubs/uddi_v3.htm`.

[14] UPnP Forum. `http://www.upnp.org/`.

[15] Zero Configuration Networking (Zeroconf). `http://www.zeroconf.org/`.

[16] William Adjie-Winoto, Elliot Schwartz, Hari Balakrishnan, and Jeremy Lilley. The design and implementation of an intentional naming system. *ACM SIGOPS Operating Systems Review*, 34(2):1–22, April 2000.

[17] Reaz Ahmed and Raouf Boutaba. A Survey of Distributed Search Techniques in Large Scale Distributed Systems. *IEEE Communications Surveys and Tutorials*, 13(2):150–167, 2011.

[18] Reaz Ahmed, Noura Limam, Jin Xiao, Youssef Iraqi, and Raouf Boutaba. Resource and service discovery in large-scale multi-domain networks. *IEEE Communications Surveys and Tutorials*, 9(4):2–30, 2007.

[19] Magdalena Balazinska, Hari Balakrishnan, and David Karger. INS/Twine: A Scalable Peer-to-Peer Architecture for Intentional Resource Discovery. In *Pervasive '02: Proceedings of the First International Conference on Pervasive Computing*, pages 195–210, London, UK, 2002. Springer-Verlag.

[20] T.D. Camacho, M. Foth, and A. Rakotonirainy. Pervasive technology and public transport: Opportunities beyond telematics. *IEEE Pervasive Computing*, 12(1):18 –25, jan.-mar. 2013.

[21] Celeste Campo and Carlos Garcia-Rubio. DNS-Based Service Discovery in Ad Hoc Networks: Evaluation and Improvements. *Personal Wireless Communications*, 4217:111–122, 2006.

[22] Mauro Caporuscio, Antonio Carzaniga, and Alexander L. Wolf. Design and evaluation of a support service for mobile, wireless publish/subscribe applications. *IEEE Transactions on Software Engineering*, 29(12):1059–1071, 2003.

[23] CAR 2 CAR Communication Consortium. CAR 2 CAR Communication Consortium Manifesto, Overview of the C2C-CC System, Version 1.1, August 2007.

[24] Antonio Carzaniga, David S. Rosenblum, and Alexander L. Wolf. Design and evaluation of a wide-area event notification service. *ACM Transactions on Computer Systems*, 19(3):332–383, August 2001.

[25] Dipanjan Chakraborty, Anupam Joshi, Yelena Yesha, and Tim Finin. GSD: a novel group-based service discovery protocol for MANETS. In *4th International Workshop on Mobile and Wireless Communications Network (MWCN 2002)*, pages 140–144, 2002.

[26] Dipanjan Chakraborty, Anupam Joshi, Yelena Yesha, and Tim Finin. Toward Distributed service discovery in pervasive computing environments. *IEEE Transactions on Mobile Computing*, 5(2):97–112, February 2006.

[27] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). RFC 3626 (Experimental), October 2003.

[28] COMeSafety project. European ITS Communication Architecture,Overall Framework, Proof of Concept Implementation, Version 3.0, March 2010.

[29] CVIS Project. D.CVIS.3.2 High Level Architecture, March 2007.

[30] CVIS Project. D.CVIS.3.4 Final Architecture and System Specifications, June 2010.

[31] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert. Network Mobility (NEMO) Basic Support Protocol. RFC 3963 (Proposed Standard), January 2005.

[32] Thierry Ernst. The information technology era of the vehicular industry. *SIGCOMM Computer Communication Review*, 36(2):49–52, 2006.

[33] ETSI. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definitions, ETSI TR 102 638 V1.1.1, June 2009.

[34] ETSI. Intelligent Transport Systems (ITS), Communications Architecture, ETSI EN 302 665 V1.1.1, September 2010.

[35] ETSI. Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 2: Scenarios, ETSI TS 102 636-2 V1.1.1, March 2010.

[36] ETSI. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM); Rationale for and guidance on standardization ETSI TR 102 863 V1.1.1, June 2011.

[37] ETSI. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service, ETSI TS 102 637-2 V1.2.1, March 2011.

[38] ETSI. Intelligent Transport Systems (ITS); Vehicular communications; GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 1: Media-Independent Functionality, ETSI TS102 636-4-1 V1.1.1, June 2011.

[39] ETSI. Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 6: Internet Integration; Sub-part 1: Transmission of IPv6 Packets over GeoNetworking Protocols, ETSI TS 102 636 6-1 V1.1.1, March 2011.

[40] Patrick Th Eugster, Pascal A Felber, Rachid Guerraoui, and Anne-Marie Kermarrec. The many faces of publish/subscribe. *ACM Computing Surveys*, 35(2):114–131, June 2003.

[41] GeoNet Project. D1.2 Final GeoNet Architecture Design, June 2010.

[42] GeoNet Project. D2.2 Final GeoNet Specification, January 2010.

[43] GeoNet Project. D7.1 GeoNet Experimentation Results, June 2010.

[44] E. Guttman. Service Location Protocol Modifications for IPv6. RFC 3111 (Proposed Standard), May 2001.

[45] E. Guttman, C. Perkins, and J. Kempf. Service Templates and Service: Schemes. RFC 2609 (Proposed Standard), June 1999.

[46] E. Guttman, C. Perkins, J. Veizades, and M. Day. Service Location Protocol, Version 2. RFC 2608 (Proposed Standard), June 1999. Updated by RFC 3224.

[47] T. Hardie, A. Newton, H. Schulzrinne, and H. Tschofenig. LoST: A Location-to-Service Translation Protocol. RFC 5222 (Proposed Standard), August 2008.

[48] International Telecommunication Union,Telecommunication Standardization Sector. The Directory – Overview of concepts, models and services", X.500 (also ISO/IEC 9594-1:1994), 1993.

[49] ISO. ISO/IEC 7498-1:1994 Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model, Also published as ITU-T Recommendation X.200, November 1994. 1, 11, 189. International Organization for Standardization, 1994.

[50] ISO TC 204. Intelligent Transport Systems - Communications access for land mobiles (CALM) - Architecture, ISO21217:2010. International Organization for Standardization, April 2010.

[51] ITSSv6 Project. D2.1 Preliminary System Recommendations, May 2012.

[52] ITSSv6 Project. D2.2: Preliminary System Specification, May 2012.

[53] Japan Automobile Manufacturers Association INC. The motor industry in Japan. `http://www.jama-english.jp/publications/MIJ2012.pdf`, 2012.

[54] G Karagiannis, O Altintas, E Ekici, G Heijenk, B Jarupan, K Lin, and T Weil. Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions. *Communications Surveys & Tutorials, IEEE*, 13(4):584–616, 2011.

[55] J. Kempf and J. Goldschmidt. Notification and Subscription for SLP. RFC 3082 (Experimental), March 2001.

[56] Jonathan K Lawder and Peter J H King. Using Space-Filling Curves for Multidimensional Indexing. In *in Proceedings of the 17th British National Conferenc on Databases: Advances in Databases*, pages 20–35, London, UK, UK, 2000. Springer-Verlag.

[57] Choonhwa Lee, Abdelsalam Helal, Nitin Desai, Varun Verma, and Bekir Arslan. Konark: A system and protocols for device independent, peer-to-peer discovery and delivery of mobile services. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 33(6):682–696, November 2003.

[58] Li Li and Louise Lamont. A lightweight service discovery mechanism for mobile ad hoc pervasive environment using cross-layer design. In *in Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW '05)*, pages 55–59, March 2005.

[59] Satoshi Matsuura, Kazutoshi Fujikawa, and Hideki Sunahara. Mill: A geographical location oriented overlay network managing data of ubiquitous sensors. *IEICE Transactions on Communications*, E90-B(10):2720–2728, October 2007.

[60] Petar Maymounkov and David Mazières. Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. In *In Proceedings of IPTPS2002*, pages 53–65, 2002.

[61] Adnan Noor Mian, Roberto Baldoni, and Roberto Beraldi. A survey of service discovery protocols in multihop mobile ad hoc networks. *IEEE Pervasive Computing*, 8(1):66 –74, Jan 2009.

[62] T. Narten, E. Nordmark, W. Simpson, and H. Soliman. Neighbor Discovery for IP version 6 (IPv6). RFC 4861 (Draft Standard), September 2007.

[63] Satoru Noguchi, Satoshi Matsuura, and Kazutoshi Fujikawa. Performance Analysis of Mobile Publish-Subscribe Service Discovery on IPv6 over GeoNetworking. In *Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS 2012)*, pages 535–540, July 2012.

[64] Satoru Noguchi, Manabu Tsukada, Ines Ben Jemaa, and Thierry Ernst. Real-Vehicle Integration of Driver Support Application with IPv6 GeoNetworking. In *IEEE 73rd Vehicular Technology Conference (VTC 2011 Spring)*, pages 1–5, May 2011.

[65] Satoru Noguchi, Manabu Tsukada, Thierry Ernst, Astuo Inomata, and Kazutoshi Fujikawa. Design and field evaluation of geographical location-aware service discovery on IPv6 GeoNetworking for VANET. *EURASIP Journal on Wireless Communications and Networking*, 2012(1):29, February 2012.

[66] Satoru Noguchi, Manabu Tsukada, Thierry Ernst, Atsuo Inomata, and Kazutoshi Fujikawa. Location-aware service discovery on IPv6 GeoNetworking for VANET. In *11th International Conference on ITS Telecommunications (ITST 2011)*, pages 224–229, August 2012.

[67] OASIS. OASIS reference model for service oriented architecture 1.0, October 2006.

[68] Panos Papadimitratos, Arnaud de La Fortelle, Knut Evenssen, Roberto Brignolo, and Stefano Cosenza. Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation. *Communications Magazine, IEEE*, 47(11):84–95, November 2009.

[69] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental), July 2003.

[70] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. A scalable content-addressable network. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM Request Permissions, August 2001.

[71] Antony Rowstron and Peter Druschel. Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In *Middleware '01: Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, pages 329–350. Springer-Verlag, 2001.

[72] J. Sermersheim. Lightweight Directory Access Protocol (LDAP): The Protocol. RFC 4511 (Proposed Standard), June 2006.

[73] M. Smith and T. Howes. Lightweight Directory Access Protocol (LDAP): Uniform Resource Locator. RFC 4516 (Proposed Standard), June 2006.

[74] Ion Stoica, Robert Morris, David Karger, M Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 149–160, New York, NY, USA, 2001. ACM.

[75] David Tam, Reza Azimi, and Hans-Arno Jacobsen. Building Content-Based Publish/Subscribe Systems with Distributed Hash Tables. In *in International Workshop On Databases, Information Systems and Peer-to-Peer Computing*, pages 138–152, September 2003.

[76] Manabu Tsukada, Ines Ben Jemaa, Hamid Menouar, Wenhui Zhang, Maria Goleva, and Thierry Ernst. Experimental evaluation for IPv6 over VANET geographic routing. In *in Proceedings of the 6th International Wireless Communications and Mobile Computing Conference (IWCMC'10)*, pages 736–741, 2010.

[77] UPnP Forum. UPnP device architecture 1.1, October 2008.

[78] Christopher N Ververidis and George Polyzos. Service discovery for mobile Ad Hoc networks: a survey of issues and techniques. *IEEE Communications Surverys and Tutorials*, 10(3):30–45, 2008.

[79] P. Vixie, S. Thomson, Y. Rekhter, and J. Bound. Dynamic Updates in the Domain Name System (DNS UPDATE). RFC 2136 (Proposed Standard), April 1997. Updated by RFCs 3007, 4035, 4033, 4034.

[80] W3C. OWL Web Ontology Language Overview, W3C Recommendation. `http://www.w3.org/TR/owl-features/`.

[81] K. Zeilenga. Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map. RFC 4510 (Proposed Standard), June 2006.

[82] W. Zhao, H. Schulzrinne, and E. Guttman. Mesh-enhanced Service Location Protocol (mSLP). RFC 3528 (Experimental), April 2003.

[83] Feng Zhu, Matt Mutka, and Lionel Ni. Splendor: A Secure, Private, and Location-Aware Service Discovery Protocol Supporting Mobile Services. In *PERCOM '03: Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, pages 235–242. IEEE Computer Society, March 2003.

Appendix

Abbreviations

AR Access router

C-ITS Cooperative intelligent transport system

DA Directory agent in SLP

GeoDestination a set of data elements that describes a particular geographical area

ITS Intelligent transport system

ITS Station A set of nodes which includes ITS related entities

LAN Local area network

MANET Mobile ad-hoc network

MR Mobile router

RC Resource consumer

RP Resource provider

RR Resource registry

SA Service agent in SLP

SLP Service location protocol

SrvRqst Service request message in SLP

SrvRply Service reply message in SLP

UA user agent in SLP

VANET Vehicular ad-hoc network

VCS Vehicular communication systems

WLAN Wireless local area network

Achievements

Journal

- Satoru Noguchi, Manabu Tsukada, Thierry Ernst, Atsuo Inomata, and Kazutoshi Fujikawa, ”Design and field evaluation of geographical location-aware service discovery on IPv6 GeoNetworking for VANET”, EURASIP Journal on Wireless Communications and Networking, vol. 2012, no. 1, pp.1–29, February 2012.
- Satoru Noguchi, Satoshi Matsuura, Atsuo Inomata, Kazutoshi Fujikawa and Hideki Sunahara, ”Wide-area publish/subscribe mobile resource discovery based on IPv6 GeoNetworking”, IEICE Transactions on Communications, 2013 (**to be published**).

International conference

- Satoru Noguchi, Satoshi Matsuura, and Kazutoshi Fujikawa, ”Performance Analysis of Mobile Publish-Subscribe Service Discovery on IPv6 over GeoNetworking”, in Proceedings of the Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS 2012) workshop (MoMoPE2012), pp. 535—540, July 2012
- Satoru Noguchi, Manabu Tsukada, Thierry Ernst, Atsuo Inomata, and Kazutoshi Fujikawa, ”Location-aware service discovery on IPv6 GeoNetworking for VANET”, in Proceedings of the 11th International Conference on ITS Telecommunications (ITST 2011), pp. 224—229, August 2011
- Satoru Noguchi, Manabu Tsukada, Ines Ben Jemaa, and Thierry Ernst, ”Real-

Vehicle Integration of Driver Support Application with IPv6 GeoNetworking", in Proceedings of the IEEE 73rd Vehicular Technology Conference (VTC 2011 Spring), pp. 1—5, May 2011

- Satoru Noguchi, Atsuo Inomata, Kazutoshi Fujikawa, and Hideki Sunahara, "Efficient data management using the session log in DHT and its evaluation", in Proceedings of the 6th IEEE Conference on Consumer Communications and Networking Conference (CCNC09), in CD-ROM, January 2009.

National conference

- 野口 悟, 猪俣 敦夫, 藤川 和利, 砂原 秀樹, "DHT におけるノードの動作履歴を用いたデータ管理手法の提案と実装", インターネットカンファレンス (IC 2008), Oct, 2008

Joint paper: international conference

- Yoshihiro Okamoto, Satoru Noguchi, Satoshi Matsuura, Atsuo Inomata, Kazutoshi Fujikawa: "Koshien-Cloud: Operations of Distributed Cloud as A Large Scale Web Contents Distribution Platform", in Proceedings of The 12th IEEE/IPSJ International Symposium on Applications and the Internet (SAINT 2012) Workshop (C3NET 2012), pp.363–368, July 2012

- Ryota Miyagi, Satoshi Matsuura, Satoru Noguchi, Atsuo Inomata and Kazutoshi Fujikawa, "A divide and merge method for sensor data processing on large-scale publish/subscribe systems", in Proceedings of the 3rd Workshop on Enablers for Ubiquitous Computing and Smart Services (EUCASS 2012), pp.424–429, July 2012

- Satoshi Matsuura, Ryota Miyagi, Satoru Noguchi, Hideya Ochiai, Kazutoshi Fujikawa, Hideki Sunahara, "A method of sensor data processing on large scale publish/subscribe systems", in Proceedings of the 5th International Conference on Engineering and Technology (ICET-2011), May 2011

- Atsuo Inomata, Taiki Morikawa, Minoru Ikebe, Yoshihiro Okamoto, Satoru Noguchi, Sk. Md. Mizanur Rahman, Kazutoshi Fujikawa, Hideki Sunahara, ”Proposal and Evaluation of a Dynamic Resource Allocation Method based on the Load of VMs on IaaS”, in Proceedings of 4th IFIP International Conference on New Technologies Mobility and Security (NTMS2011), IEEE IFIP, Vol.Next generation systems & Service-oriented techniques - I (In CDROM), February 2011