NEXT GENERATION VEHICLE NETWORK (NGVN): INTERNET ACCESS UTILIZING DYNAMIC DISCOVERY PROTOCOLS

Rami Baroody, Asif Rashid, Nizar Al-Holou, Salim Hariri
rbaroody@ecs.syr.edu, email@asifrashid.com, alholoun@udmercy.edu, hariri@ece.arizona.edu
University of Detroit Mercy

ABSTRACT

In the near future, vehicles are expected to become a part of the Internet, either as a terminal in a mobile network, as a network node, or as a moving sensor (providing environmental information, cars status, streaming video, etc.) or a combination of the three. This is partly due to the steadily growing interest of vehicles' passengers in location-based information. Drivers and passengers that would want to receive information about traffic jams or accidents in their vicinity will likely be interested in accessing Internet services from within the vehicular network. Access can be gained by using roadside installed Internet Gateways (IGs), which are able to communicate with the vehicles. However, several difficulties must be addressed in such a scenario. Examples are the communication efficiency, mobility support, the discovery of Internet Gateways, and the handover of connections from one gateway to the next. In this paper, we are focusing on the aspect of accessing the Internet from within the vehicle network using a service discovery protocol. We therefore developed a Dynamic Discovery Service (DDS) protocol to discover Internet Gateways which is suitable for the characteristics of future vehicular ad hoc networks. Besides the benefit of efficient service discovery, our protocol is able to choose the most suitable Internet Gateway among others.

Keywords: NGVN, DDS, JINI, SLP

I. INTRODUCTION

Electronics are used in automobiles to improve vehicle performance, serviceability, pollution control, product differentiation, reliability, safety, and convenience. As customers continue to demand additional functions and features in their automobiles, (e.g., navigation systems, intelligent highway system supports, automated collision avoidance systems, Internet access, remote diagnostics) there arises a need for a large number of devices. Such devices include smart sensors, mobile phones, hands-free devices and various computers to be deployed in future cars. These devices need to be interconnected on appropriate network speed with appropriate protocols that will allow automobiles to be able to perform these functions at a reasonable cost and with high reliability and availability. Furthermore, in automobiles, many subsystems need extensive interactions for information exchange. For example, an engine management system, transmission controller, and collision avoidance may work closely together for proper gear change. These kinds of interactions are becoming so numerous that the traditional view of vehicle electronics, as independent sub-systems may not be adequate [1-3].

The concept of local area network (LAN) has been introduced in automobiles to improve communications and interactions among different electronics systems. In applying a LAN to automotive electronics systems, an optimal protocol has been adopted for each system which in turn may run different protocols that satisfy the data rate for their applications.

In future automobiles, many subsystems will need extensive interactions for information exchange. For example, an engine management system and transmission controller may work closely together for proper gear change. This is enabled via multiple sub-networks (e.g., CAN, MOST, IP) [4-6].

Future cars will support computing facilities and will be expected to communicate with other cars and with the fixed network. Furthermore, future developments in the automobile domain will also include the utilization of new communication technologies. The major goals are to provide increasing automotive safety, to achieve smooth traffic flow on the roads, and to improve the convenience of vehicles' passengers by providing them with information and entertainment. In order to avoid communication costs and to be able to guarantee the low delays required for the exchange of safety related data between cars, Next Generation Vehicle Network (NGVN) systems based on ad hoc networks are a promising solution for future road communication scenarios.

NGVN systems will comprise roadside installed gateways to the Internet. The Internet Gateways (IGs) are integrated into the NGVN system, as well as connected to the Internet. The IGs provide a cheap, but timely restricted access to the Internet for passing vehicles. In order for vehicles in the ad hoc network to use the IGs, they must discover them first. In contrast to conventional ad hoc networks formed by most other mobile devices such as PDAs or laptops, vehicle ad hoc networks are highly mobile and dynamic, (i.e., the network topology changes frequently). As a result, the availability of IGs changes frequently as well, and several gateways may be available simultaneously.

In this paper, we propose an efficient and scalable solution for the discovery of Internet gateways in such a highly mobile vehicular environment. Our approach deploys an intelligent DDS for deciding which gateway currently fits best according to the requirements of the vehicle. In the next section, we will introduce the basic communication schemes and their Internet-working in future vehicular network. We will also discuss possible solutions to identify the IGs. Section III will then introduce our approach in detail. Section IV evaluates SLP and the NGVN DDS. Finally, section V concludes this paper.

II. COMMUNICATION IN NGVN

The key design requirements for Next Generation Vehicle Network are the capability to distribute locally relevant data and the satisfaction of the needs of drivers and passengers of vehicles for location-dependent information and services. Vehicles are able to communicate either directly or using intermediate vehicles as relaying nodes (Figure 1). Hence, vehicles are able to communicate with distant Internet Gateways using multi-hop communication. In order to discover and access Internet resources, we need an adequate DDS and communication architecture for Internet integration, as described in the following section.

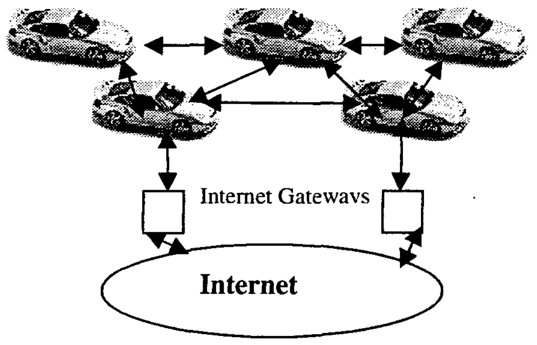


Figure 1: Future Vehicular Communication Scenario

II.1 Related work

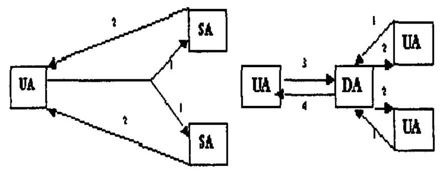
The concept in discovering IGs in automotive communication systems of the future is similar to that of finding services in ad hoc networks. This identification could occur at the network layer or the application layer.

1) Network Layer Solutions: Two methods are possible on the network layer. The first approach is to use multicast for the identification of IGs, which in turn forms a pre-defined multicast group. As a result, a vehicle sends its Internet traffic to this group. This group may simply consist of all IGs within its virtual communication range. For *n* IGs in the range, the data will be transmitted to the Internet *n* times. One optimization technique is to use anycast, which supports the addressing the nearest or most appropriate member of a group. Although multicast and anycast seem straightforward in theory, they are not simple to implement. Both need specific support for the routing protocol deployed in the ad hoc network. Additionally, with such a dynamic and mobile environment, the group maintenance requires a great deal of overhead. The second approach is the modification of router discovery process. This approach is the one used to implement the MIPMANET [7]. The premise is that routers advertise themselves within the local network and thus, each mobile node is able to configure its default router accordingly. With this approach, the router advertisements must be flooded in the network because each vehicle itself acts as a router. As a result, the overall ad hoc network will be flooded periodically with router advertisements. The main disadvantage of this method is that it is not very scalable. For example, if the number of participating vehicles doubles, the system complexity is quadrupled. Unless the number of participating vehicles is very small, there would be an enormous amount of traffic transmitted through the Internet and thus, this approach is also impractical for future automotive communication scenarios.

2) Service Discovery Protocols: Configuring and using mobile devices could be very inconvenient in large-scale networks. For example, a new user would have to configure his mobile device manually to print a document on the nearest printer. This problem is resolved by service discovery protocols. These protocols enable users, mobile devices, and applications to discover services in a network with minimal prior knowledge. Several service discovery protocols are available currently, including the Service Location Protocol (SLP), Universal Plug and Play (UPnP), Jini, and Salutation [8, 9]. SLP and Jini are described below in order to illustrate the basic mechanisms of service discovery protocols. SLP was standardized in RFC 2608 [10] and origins from the Internet community. SLP consists of three types of agents that appear on behalf of service providers, users, and service directories, respectively:

- *Service Agent (SA):* Service Agents act on behalf of service providers and advertise the addresses and characteristics of the provided services.
- *User Agent (UA):* Applications use User Agents to find specific services within a network. User Agents communicate with both, Service Agents and Directory Agents.

- *Directory Agent (DA):* Directory Agents implement service directories to manage the available services in a network.



1. Service Request (SrvRqst, Multicast) 1. Service Register (SrvReg)
2. Service Reply (SrvRply) 2. Acknowledgement (SrvAck)
3. Service Request
4. Service Reply (SrvRply)

(a) Service Discovery without DAs (b) Service Discovery Using DAs

Figure 2: Protocol Interaction in SLP

The two interaction modes SLP uses for discovering services are illustrated in Figure 2. The first discovery process (Figure 2(a)) shows the interactions that occur among the elements in a network without DAs. Each SA has to join a predefined multicast group and a UA queries available SAs by sending a service request (SrvRqst) to the multicast address. Next, each SA replies to the query in the second step with a service reply (SrvRply). Note that SLP uses Service URLs (Uniform Resource Locators [10]) to specify the location of a service (e.g., an IP address). The second interaction mode uses at least one DA (Figure 2(b)). In this mode, the first step consists of each SA within the network registering its services with the DA using a service registration message (SrvReg). In the second step, the DA acknowledges the registration with a SrvAck message. In the third step, a UA immediately requests a service from a DA (SrvRqst). The DA then replies with its registered services (SrvRply) for the fourth step. In this mode, the DAs need to be discovered only once as illustrated in the first mode. The classical service discovery model is reactive, and thus requires mobile devices to start the service discovery. Service requests are transmitted either immediately to a directory service or are transmitted via multicast to all participants in the network. This technique works well in most local area networks, because the number, type, and configuration of services are generally static. The mobility of devices is quite low in corporate networks, as well. As a result, service requests will likely occur when a new device joins the network or before a service is used for the first time. In large-scale networks, the use of service directories further improves the scalability of service discovery process by requiring multicast only for the discovery of the service directories. However, vehicles usually travel at high speeds and thereby have a continually changing service landscape. This feature has significant implications when considering that IGs are stationary. Due to this, vehicles must continually discover gateways to achieve a consistent view on the available IGs. In scenarios including numerous vehicles, the discovery heavily burdens the vehicular ad hoc network. With an inefficient routing algorithm, the multi-hop feature could amplify the complexity.

II.2 JINI

Jini technology has been developed by Sun Microsystems as an extension of the Java programming language. It addresses the issue of how devices connect with each other in order to form a simple ad hoc network (a Jini "\community"), and how these devices provide services to other devices in this network. Jini consists of architecture and a programming model. Each Jini device is assumed to have a Java Virtual Machine (JVM) running on it. The Jini architecture principle [11] is similar to that of SLP. Devices and applications register with a Jini network using two processes called *Discovery and Join.* To join a Jini network, a device or application places itself into the lookup table on a lookup server, which is a database for all services on the network (similar to the directory agent in SLP). Besides pointers to services, the lookup table in Jini can also store Java-based program code for these services. This means that services may upload device drivers, interfaces, and other programs that help the user access the service. When a client wishes to utilize the service, the object code is downloaded from the lookup table to the JVM of the client. Whereas a service request in SLP returns a service URL, the Jini object code offers direct access to the service using an interface known to the client. This code mobility replaces the necessity of pre-installing drivers on the client. The Jini specifications are open source and may be used freely. However, Sun charges a licensing fee for commercial use. A reference implementation may be downloaded at *http://www.sun.com/jini.* The Jini code can be implemented in 46K of Java binaries. Due to its portability and flexibility, we developed a DDS based on Jini technology, a new protocol to discover IGs in future vehicular networks. The following section describes it in more detail.

III. NGVN – BASED SERVICES

The service discovery of Internet Gateways must be highly scalable because vehicle densities can become very high in large cities at peak times. In order to discover IGs efficiently, we developed and implemented a DDS based on Jini as previously stated. Besides scalability and efficiency, another important objective is to determine the most suitable IG among the available IGs in communication range. Hence, the service discovery process of DDS comprises two functional tasks: the discovery of available IGs and the selection of the most suitable IG.

III.1 NGVN Dynamic Discovery Service architecture

At the Computer Networking Laboratory, a proof-of-concept prototype of the Next Generation Vehicle Network has been implemented using Java and Jini Lookup Services [12-13] as well as DDS technology.

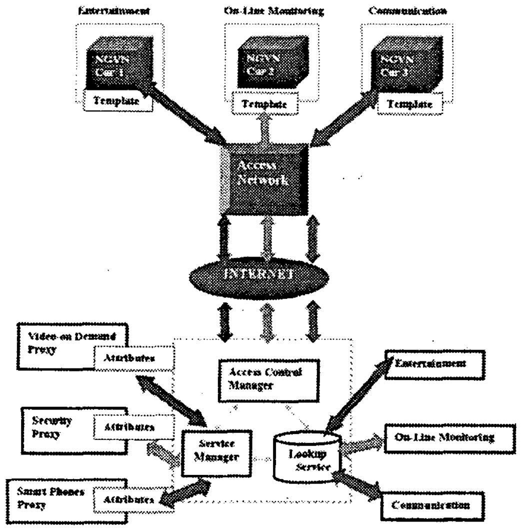


Figure 3: DDS Architecture

This project adopts the current prototype to implement the required automotive NGVN Internet services. Figure 3 illustrates the architecture of a NGVN DDS, which consists of six components: Network Service Proxy, Network Service Attributes, Service Manager, Lookup Service, Service Templates and Access Control Manager. These components, their roles in the system, and their interaction to provide DDS system functionality are described in the following sections.

<u>Network Service Proxy</u>

NGVN information services access automotive information and entertainment services through objects called *proxies*, which provide all the code needed to connect to a particular network service. Proxies are analogous to device drivers as they allow an application program to interact with a service while encapsulating it from the implementation details of that service. However, unlike device drivers, which are typically installed by a system administrator before the device can be used, proxies are carried by the services themselves, and are dynamically downloaded by the clients when they invoke a particular network service. Client applications do not have to know the implementation details of these proxies and they do not have to be "compiled in" when the applications are written. These proxy objects typically communicate over the network with the backend portion of the service using whatever protocols supported by the backend system.

<u>Network Service Attributes</u>

Attributes are Java objects that are attached to service proxies. Services attach these attributes when they publish their proxies, and NGVN services or clients can search for proxies by looking for certain attribute patterns. Attributes are used to associate descriptive information and state with a service. Attributes are important to distinguish between similar services (i.e., services that implement the same interface). Name, address, and location are some of the standard attributes that are associated with services. But sometimes we need to create our own attributes to reveal service-specific information. There are some attributes which are static, like name, that do not change often with time, but there are some attributes, like state, that are dynamic and can be updated by the service itself. These changes have to be reflected in all the lookup services that the proxy object is registered with, to ensure that clients have access to the most updated services. Figure 4 details the DDS architecture.

<u>Service Manager</u>

The service manager maintains updated information about the available NGVN services. The service manager detects the changes in the service attributes, compares it against threshold values. If the threshold is exceeded, it updates this information in all the lookup services where the proxy object had previously registered. The services can also have properties, which are not service attributes. These properties can be dynamic and come into effect during the service lookup process. The service manager tracks these dynamic properties and helps the client by providing the most updated values of these dynamic properties.

<u>Lookup Service</u>

The lookup service is a special service that keeps track of all other NGVN services in the community. When a provider of a NGVN service wishes to make itself available, it "publishes" its proxy by storing it in a lookup service. Clients (NGVN users) can then connect to the lookup service and ask what services are available. The lookup service can also inform interested parties when new services appear or when services leave the community through remote events [20-22]. Lookup services announce their existence using the *multicast announcement protocol.* In multicast announcement, interested parties listen on well-known multicast address for announcements about the existence of lookup services. Periodically all lookup services send a multicast message to this address. Once an interested party has received an announcement of the existence of one, it can ask it for its service proxy. It contacts the lookup service using the direct unicast connection, and the lookup service replies with the proxy object that the client can use to communicate with it. The lookup service is the main module in the DDS.

Service Templates

Service templates are Java objects that NGVN clients use to construct their search criteria. The service template matching system is extremely flexible and services can be found based on the set of attributes associated with them using type-based and content-based attribute matching rules. NGVN users/clients are the consumers of services. These clients discover the lookup service using the *multicast request protocol.* Once clients find the lookup service, they search it for services that implement particular services with certain attributes, using templates to construct their search criteria. The client downloads the service proxy and then uses it for communication with the service.

Access Control Manager (ACM)

Every lookup service has an ACM that maintains and updates the list of NGVNs (cars that have subscribed to certain NGVN services) that can access the services. When the clients try to download the proxy object from the lookup service, the ACM checks the list to see if the client is authorized to use the service. If the client is authorized, the ACM allows the proxy to be downloaded; otherwise an "access denied" message is sent back to the NGVN client. The client can request the lookup service, which in turn requests the service for access. If the service accepts the request it updates the access control list through the ACM. Thus, the ACM provides security in the DDS architecture.

Dynamic Lookup

In traditional architectures, services are registered with the lookup service once during a session and continue to use the resources till they are freed explicitly. However, these architectures do not reflect the dynamic characteristics in the system, which is very important in the NGVN mobile environment. Two methods have been developed to achieve dynamic lookup: Dynamic Update and Dynamic Filtering. In the dynamic update scheme, the service manager detects the changes in the service attributes and compares them with certain threshold values. These threshold values determine that the Jini Lookup Service does not support any inequality matching (like greater than, less than or logical operators) for the attribute values. This is a very rigid constraint in our application. In the dynamic filtering scheme, properties of services that change frequently with time are maintained separately and not defined as service attributes. The service manager has functionalities to retrieve the current values of the dynamic properties when requested by the client.

III.2 Discovery of Available Internet Gateways

The key concept of NGVN dynamic discovery service is the commutation of the roles of service user and service provider. This way, IGs do not wait for requests from vehicles requiring Internet access. Instead, they themselves advertise their service proactively. Vehicles looking for a service assume a passive role, and thus do not discover the gateways actively. Hence, we call this process "passive discovery" further on. In order to enable passive discovery in NGVN, we divided the functionality of a service provider into two distributed units. The first unit is situated on the IGs and announces its service periodically. The service advertisements are transmitted through multicast to the group of nodes in the ad hoc network. If a vehicle moves into the (virtual) transmission range of an IG, the second functional unit within the vehicle will receive the gateway's service advertisements. It then extracts the service information and stores it in a local database. Due to the periodicity of the service advertisements, a vehicle is able to infer whether an IG is available or not. From a vehicle's point of view, the discovery of IGs is reduced to a search in the vehicle's local database. If the search is successful, the in-vehicle functional unit will respond with the respective IG. Otherwise, the user must assume that a gateway is currently not available.

III.3 Selection of Suitable Internet Gateways

As a result of the multi-hop nature of the vehicle network, several IGs might be within a vehicle's communication range simultaneously. The NGVN Dynamic Discovery Service must decide the most suitable IG to use for connecting to the Internet. This decision will highly influence the quality of service experienced while utilizing the selected gateway for Internet access. For example, choosing an IG that is within direct communication range of a vehicle's wireless network hardware will enable the communication delay between itself and the IG to be very low. Often times, a new IG has to be discovered after the vehicle is out of direct coverage area of the previous IG. This risks leaving the vehicle with a period of disconnection during the handoff process. For an optimal selection, the selection process should be based on the requesting application's expectations regarding the quality of service. A simple classification scheme allows NGVN to estimate these expectations. Each application can specify its requirements using a combination of the following properties: *Interactivity*, *Streaming* and *Real-Time*. Along with these properties, NGVN's selection methodology also takes into consideration information about the state of the vehicle network and the available IGs. The vehicle network would provide data about the current traffic density in the vicinity of a vehicle. However, the greatest chunk of state information is within the IG's periodical service advertisements, including information about the following:

- The number of clients using the current IG.
- The utilization of the IG's available bandwidth.
- The IG's location for the estimation of the connection duration and the communication delay.

III.3 In-Vehicle NGVN Services

The NGVN clients/users will have access to a wide range of automotive-related information and entertainment services that can be added and removed dynamically. The various suppliers of NGVN services advertise their products through an interface that defines the type and features of their services. There are one or more lookup services running in the network waiting for the NGVN service providers to register with them. A provider locates the lookup service using a combination of multicast announcement and unicast response protocols. This process is known as *discovery*. Then the supplier sends the service proxy object to the lookup service to register itself. This is the *join* process. For a car user, there will be a number of NGVN services and every supplier registers its proxy object with the lookup service it discovers. There can be a number of attributes associated with each service. Some of the important attributes are location, service ID, subscription fee/price, and period. There is a HTTP server running, so that the proxy object class can be downloaded on demand when the NGVN client/user makes a request to access one of the provided services. The NGVN service provider that satisfies the request is discovered, and then the lookup service sends the matching provider's proxy to the car NGVN SOAP server. The NGVN server establishes a connection with the service provider and the transaction is started.

IV. EVALUATION

The performance evaluation is an essential step in the system design process in order to ensure the success of the system. Any system that is being designed, whether it is an industrial plant or the Next Generation Vehicle Network, must satisfy certain constraints. Performance metrics such as reliability, availability, bandwidth usage, and throughput are critical to take into account. System evaluation can be done using three different methods: measurement, simulation, or using analytical models. Mathematical analysis was used to evaluate performance. In this paper, we evaluated SLP and our DDS protocol primarily by calculating the bandwidth usage. We will investigate the worst-case scenario, assuming the flooding routing algorithm without duplicate detection and the following parameters: number of vehicles, number of Internet Gateways, frequency of request for Internet Gateways, frequency of Internet Gateway advertisements, duration time, request message size, reply message size, advertisement message size, number of hops, and the number of links.

Within a five hundred meter radius and an average of five meters per vehicle per lane, there would be one hundred vehicles per lane. With a three-lane highway in each direction, there would be six hundred vehicles. Generally, one Internet Gateway would be sufficient for most scenarios. However, in some high-demanding situations, more Internet Gateways would be required. Three IGs should be sufficient even under the worst-case scenario.

Number of Vehicles:	nCars	= 600
Number of Internet Gateways:	nIG	= 3

In a reactive system, vehicles send a request for an IG periodically. In contrast, in a proactive system, IGs send advertisements periodically. Since properly allocating resources is more critical to an IG because of the many clients it services, it should send advertisements less frequently than a vehicle sends requests.

Frequency of IG Request:	fIGReq	= 2/s
Frequency of IG Advert.:	fIGAdv	= 1/s

With a duration time of one second, bandwidth can be viewed in terms of bits (or megabits) per second.

Duration Time:	T	= 1s

Three different types of packets can be sent out. The first is a vehicle request for an IG. The second is the reply from the IG to the vehicle and the third is the message the IG advertises. Each packet is 125 bytes or 1000 bits long.

Request Message Size:	bReqSz	= 1000 bits
Reply Message Size:	bRepSz	= 1000 bits
Advert. Message Size:	bAdvSz	= 1000 bits

The number of hops in a given system increases its complexity exponentially, depending on the maximum number of links a node can have. Thus, using too many hops would quickly overburden the network. On the other hand, not using hops at all would decrease the coverage area substantially. Using a small number of hops such as two or three is a compromise. The same argument applies for the number of links. An inadequate number would not only reduce the coverage area, but would also defeat the purpose of multi-hopping.

Number of Hops:	nHops	= 2
Number of Links:	nLinks	= 50

IV.1 Bandwidth Usage in SLP

In the simplest case, bandwidth is the mere product of the number of messages sent and the size of the message. For SLP, each vehicle would send a request for an IG with a given packet size at a certain frequency times the duration time (which is one second). In the "worst-case" scenario, each IG would receive the message and consequently send a reply. The following equation summarizes this, but does not take multi-hopping into account:

BW SLP (w/o multi-hop) = nCars * fIGReq * T * nIG * (bReqSz + bRepSz)

The number of hops exponentially increases the complexity of the system. The previous equation was modified appropriately to include multi-hop:

BW SLP = (nCars * fIGReq * T * nIG * (bReqSz + bRepSz)) * (nLinks ^ nHops)

Using the given values, the equation reduces to:

BW SLP = (600 * 2/s * 1s * 3 * (1000 bits + 1000 bits)) * (50 ^ 2)

BW SLP = 1.8 * 10^10 bits or 18,000 megabits

This means, that on average, the bandwidth usage of each car is approximately BW SLP / nCars = 18,000 / 600 = 30 megabits/second. This is a high bandwidth usage for mere Internet connection.

IV.2 Bandwidth Usage in NGVN DDS

The bandwidth calculation for the NGVN DDS is quite a bit different from the SLP one. With this protocol, each IG sends an advertisement of a given packet size periodically (fIGAdv * T). The highest bandwidth usage would occur if each vehicle received the message. Unlike in SLP, no replies are necessary for this stage.

BW DDS = (nIG * fIGAdv * T * nCars * (bAdvSz)) * (nLinks ^ nHops)

With the given parameters:

BW DDS = (3 * 1/s * 1s * 600 * (1000 bits)) * (50 ^ 2)

BW DDS = 4.5 * 10^9 bits or 4500 megabits

This translates approximately to an average bandwidth usage of 7.5 megabits/second per car, which is four times less than that of SLP and is also more practical for use in vehicular ad hoc networks. Since this is the worst-case scenario, the expected bandwidth usage would be much lower.

V. CONCLUSION

Access to Internet resources in future road communication scenarios will be a necessary element to provide the safety and convenience for passengers. This access will be possible using stationary Internet Gateways at the roadside, which open up the Internet for vehicular networks. The passing vehicles must discover these gateways in order to use them. In this paper, we described possible solutions for the realization of such a discovery process. We showed that neither network layer approaches nor classical service discovery protocols are able to fulfill the requirements of the targeted scenario. Hence, we proposed the NGVN DDS protocol for discovering Internet Gateways in future road communication scenarios. One key concept of the DDS is the deployment of service announcements, which advertise the presence of the Internet Gateways. Due to the proactive nature, the DDS scales well compared to classical service discovery protocols such as SLP. The bandwidth usage caused by the NGVN DDS basically depends on the number of Internet Gateways and the number of vehicles. Another significant feature of the DDS is the automatic selection of the most suitable Internet Gateway, which in turn could be accessed either directly or through multi-hopping, via intermediate vehicles. The use of multi-hopping increases the coverage area of communication but increases the bandwidth exponentially. Fortunately, the bandwidth usage of the NGVN DDS with a small number of hops is practical for vehicular ad hoc networks, even in the worst-case scenario.

VI. REFERENCES

[1] Syed Musbahuddin, Syed Mahmud, and Nizar Al-Holou " Development and Performance Analysis of a Data Reduction Algorithm for Automotive Multiplexing," *IEEE Transaction on Vehicular Technology, January 2001, Vol. 50, Number 1, pp. 162-169.*

[2] Nizar Al-Holou and Syed Musbahuddin "Availability Modeling of Hierarchical Distributed Processing Systems," International Journal of Modeling and Simulation, International Association of Science and Technology for Development (IASTED), Vol. 19, Number 2, August 1999, pp. 137-143.

[3] Syed Musbahuddin, Nizar Al-Holou, and Syed Mahmud "Data Reduction Algorithm for Automotive Applications," *SAE 1998 Transactions, Section 6 - Vol. 107, pp. 1667-1670.*

[4] Friedrick H. Phail and David Arnet, "In-Vehicle Networking-Serial Communication requirements and directions, SAE paper 8603901998.

[5] Bosch, "CAN Specification", ver2.0, Robert Bosch GmbH, Stuttgart 1991.

[6] R. Tappe, C. Thiel, R. Konig, "MOST Media Oriented Systems Transport". Elektronik. July 2000.

[7] U. J¨ornsson et. al, *MIPMANET – Mobile IP for Mobile Ad Hoc Networks*, Proc. ACM International Symposium on Mobile Ad Hoc Networking and Computing , Boston, USA, Aug. 2000.

[8] C. Lee and S. Helal, *Protocols for Service Discovery in Dynamic Mobile Networks*, Int. Journal of Computer Research, Special Issue on Wireless Systems and Mobile Computing, 2002.

[9] G. G. Richard III, *Service Advertisement and Discovery: Enabling Universal Device Cooperation*, IEEE Internet Computing, Sep./Oct. 2000.

[10] E. Guttman et al., *Service Location Protocol, Version 2*, Request for Comment 2608, Internet Engineering Task Force, Jun. 1999.

[11] Sun. Technical White Paper, "Jini Architectural Overview", http://www.sun.com/jini/, 1999.

[12] Erik Guttman, James Kempf, "Automatic Discovery of Thin Servers: SLP, Jini and the SLP-Jini Bridge", *IEEE*, Los Alamitos, CA, pp. 722-727.

[13] W. Keith Edwards, Core Jini, Second Edition, Prentice Hall, 2001, ISBN 0-13-089408-7.