Cars as a Main ICT Resource of Smart Cities

Florian Hagenauer[†], Falko Dressler[†], Onur Altintas[*] and Christoph Sommer[†]
[*]Toyota InfoTechnology Center, Tokyo, Japan
[†]Dept. of Computer Science, University of Paderborn, Germany
onur@jp.toyota-itc.com
{hagenauer,dressler,sommer}@ccs-labs.org

Abstract—**Inter-Vehicle Communication (IVC) is expected to play a central role in tomorrow's smart cities. Yet, even today, operators of infrastructure are struggling to keep up with the increasing data demand and infrastructure might not be available in more extreme conditions, such as natural or man-made disaster situations. We see a way out in the use of cars as a main Information and Communication Technology (ICT) resource of smart cities, in a technology agnostic architecture which we term Car4ICT. Cars can be envisioned to be available almost ubiquitously in future smart cities and are flexibly made use of to mediate between users offering and consuming services. We present a flexible and extensible scheme for service discovery, as well as how to set up and operate such an architecture. To illustrate the feasibility of the Car4ICT architecture, we conclude with an evaluation of the service discovery performance in our Car4ICT framework under progressively worsening conditions. This evaluation is based on a custom built Open Source simulation framework for evaluating such complex systems using multiple communication technologies in a heterogeneous vehicular network. We show that even in such conditions simple tweaks to protocol parameters can deliver good user experience in terms of low service discovery latencies.**

I. Introduction

Information and Communication Technology (ICT) is a foundation of today's and tomorrow's smart cities [1] and research activities span such diverse areas as smart buildings, smart urban planning, and smart healthcare. Figure 1 gives an overview of many of these activities, categorizing them roughly as pertaining to smart cities' *Infrastructure*, their *Administration*, and (more directly) to smart cities' *Inhabitants*.

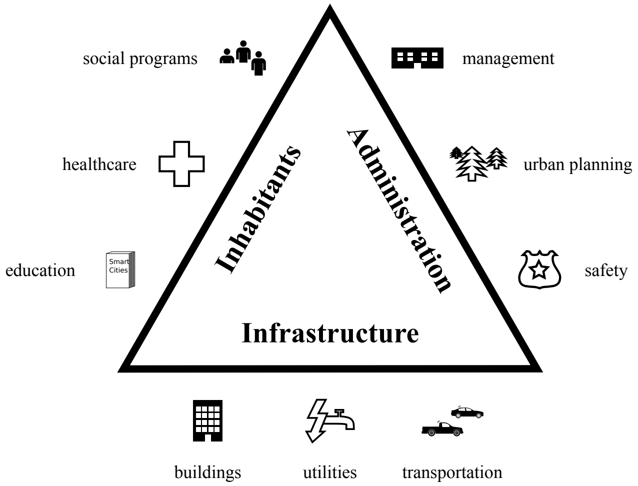


Figure 1. The different building blocks of smart cities.

Smart sensing systems and dynamic wireless network infrastructures make future smart cities possible – from enabling applications like environment monitoring, smart grid, and smart mobility. In addition they improve survivability in emergency situations. Technology-wise such applications demand robust and fault tolerant wireless communication and wide-spread sensing capabilities, that is, a tremendous amount of data. Yet, even today, operators of infrastructure are struggling to keep up with the increasing data demand, as evidenced by a massive push towards newer technologies and more bandwidth for moving data.

We see a way out in the use of cars as a main ICT resource of smart cities. We believe that, by acting in two capacities, cars will play a major role in smart cities of tomorrow [2]: First, while driving they can provide access to the services of a powerful vehicular network – in essence acting as mobile base stations. They can be accessed by people in their homes, waiting at the curb, by devices in smart buildings, or by sensors deployed throughout the smart city. Secondly, cars themselves form a network with incredible processing and storage capacities, as well as sensing capabilities beyond any possible roadside deployment. Thus, cars can not only offer access to services, but can offer services of their own. Further, by also including parked cars, both storage and processing capabilities can (to some degree) be made persistent in time and space.

Such a network formed by cars is different than typical Internet based solutions which are readily available only in a certain geographical context and during a certain time interval. On the contrary, cars are ubiquitous. This holds even in critical emergency situations such as after disasters like hurricanes or tsunamis [3]. Either way, these networked cars will play a big role in future cities by enabling a multitude of services in an architecture that we term *Car4ICT*.

Some first activities towards turning vehicles into a larger scale networked system are already taking place [4]. Such architectures, however, commonly require high market penetration rates of the system, which is a problem in early market introduction – and even more so after disasters. Further, current approaches often focus on only a single application. Our architecture, in contrast, is able to support a huge variety of new applications both independent of and in conjunction with infrastructure.

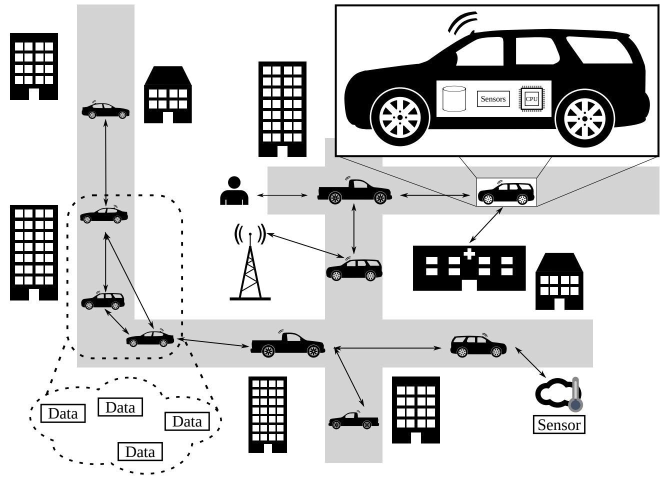


Figure 2. Concept of providing ICT resources in smart cities using mobile and parked vehicles.

Figure 2 outlines our concept. Cars are collecting data from smart sensors along the road or they connect to smart buildings. Cars also provide services like storage, processing power, or access to their sensors. They relay, distribute, and store data and connect people to this data or to the Internet. As a consequence of this design, our architecture is extensible and flexible. It can be used as a base for many applications such as distributed processing, monitoring, and sensing.

As Figure 3 outlines, the same system can just as well be envisioned to provide communication support in case of a disaster. Many large cities are built in areas where disasters (e.g., earthquakes, hurricanes) can cause widespread damage – also to infrastructure, particularly to installed communication infrastructure. If such an event occurs our architecture can help emergency services by providing communication. The figure shows an early concept how Car4ICT can support the rescue staff in case of a disaster. Inside the disaster area people can connect to cars via ad-hoc WiFi (or other communication means which need no infrastructure) and send messages to nearby cars. Any such car can then take messages to an evacuation site where other vehicles might be waiting. If no other cars are in range during the drive to the site, cars transport the messages in a store-carry-and-forward manner. If other cars are in range, messages can be sent to them via short-range communication (e.g., DSRC) in a multi-hop fashion. Finally, there might be a car which is close enough to a working cellular base station and sends the data to the Internet, delivering the messages to people outside the disaster zone. Such a scenario works with Car4ICT because the whole concept is envisioned to be able to work without any external infrastructure – although it might make use of infrastructure if and where available.

In the following, we describe the envisioned *Car4ICT* architecture and sample use cases. In particular, we motivate the use of cars as a main ICT resource in smart cities – in a network architecture going beyond current information-centric networking systems (based on [5]). We present a technique for the simulative performance evaluation of such a complex system (based on [6]) and conclude with a presentation of basic performance results that outline the capabilities of our architecture.

II. Related Work

The idea of using cars and their communication capabilities to form networks been investigated since more than a decade [4].

In much of this work the focus was on improving traffic safety, driving efficiency, and entertainment solutions based on data exchanged between cars. Early work on data exchange via ad hoc routing in vehicular networks demonstrated that routing works only in very local contexts, thus, clustering solutions have been investigated as an enabling technology [7]. Today, many network protocols and architectures are, in essence, broadcast protocols that have been augmented to be adaptive to network conditions – first and foremost to make them congestion aware. One example is Decentralized Congestion Control (DCC) [8]. The basis of these protocols are different communication technologies, of which a multitude has been investigated in the past years. Early solutions were based on the exclusive use of cellular networks such as UMTS or LTE; later research moved to using WiFi access points as Roadside Units (RSUs) (that is, hotspots along the streets). More recently, a dedicated short range radio solution based on WLAN was developed and standardized for vehicular networks as IEEE 802.11p. Standardization of Inter-Vehicle Communication (IVC) protocols is focusing on this in particular. In the U.S., one of the driving forces is the DOT, which plans rulemaking to make IEEE 802.11p based Dedicated Short-Range Communication (DSRC) systems mandatory for new cars [9]; consequently, a standard called IEEE WAVE is developed. In Europe, *ETSI ITS-G5* is planned to be rolled out as soon as cars are equipped with IEEE 802.11p radios. Japanese automakers have recently announced 760 MHz based vehicle-to-vehicle and vehicle-to-roadside devices as part of an optional package offered in some 2015 model cars in Japan. In parallel to IEEE 802.11p, many other communication technologies are researched which are viable candidates for IVC in future smart cities. This includes visible light communication [10], Bluetooth [11], and millimeter-wave communications superimposed on radar signals [12]. While, traditionally, all these technologies were seen as alternative ways of transporting data, most recently, we observe a trend towards heterogeneous vehicular networking, the use of multiple communication technologies in parallel – each bringing individual strengths and compensating others' weaknesses [13]. Along with many new challenges, this heterogeneous vehicular networking also offers great opportunities. It allows us to design applications and networks in a situation-aware and intelligent way.

Investigating a heterogeneous vehicular network, however, also means that appropriate simulation tool support needs to be guaranteed. Many established simulation tools for vehicular networking research fall short of providing the needed support for such a complex coupled system. First, vehicle mobility

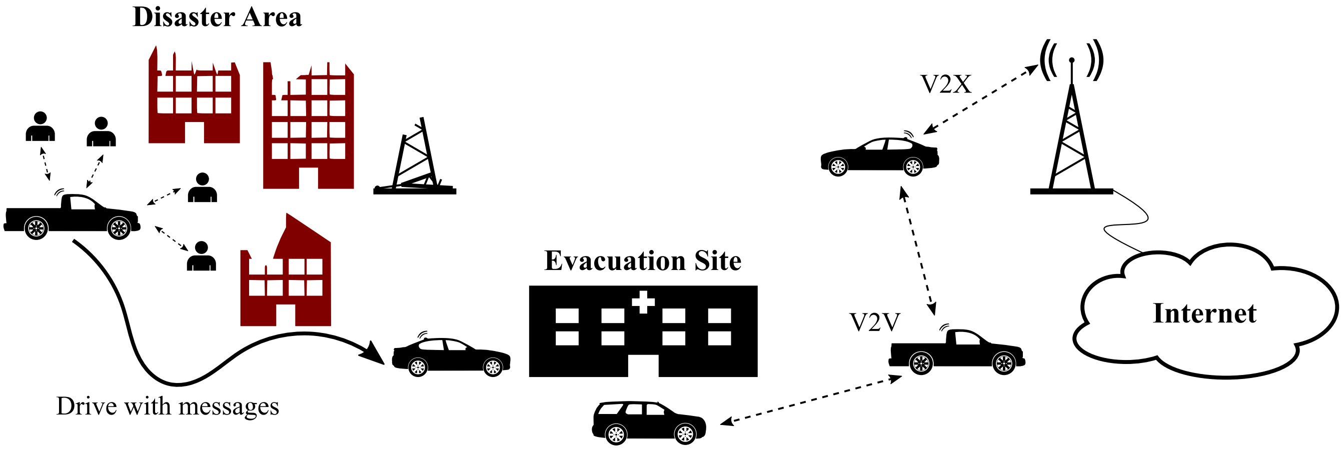


Figure 3. Concept how Car4ICT creates an infrastructure in case of a disaster and helps the emergency services.

has to be adequately modeled – particularly as far as the influence of additional information on the mobility is concerned, e.g., in terms of trips, routes, or driver behavior. Second, communication according to the different technologies has to be modeled – again, with a particular focus on the interrelation of packet transmissions using one technology on another. There exist well-established tools that can model network communication and its influence on vehicles' movement patterns: *Veins*, *iTETRIS* and *VSimRTI*. All of them have good support for mobility but are missing the tools to simulate heterogeneous networks, especially support for LTE networks is missing.

Similar to research on vehicular communication technologies, research on vehicular applications and architectures has been going on for many years now. In the most closely related proposals, cars are envisioned to become gateways for streaming multimedia content or simply providing Internet access to their occupants, or they become an enabler for emergency services. Gerla et al. propose the concept of a vehicular cloud [14], focusing on autonomous driving more than on services provisioning to users: The vehicular cloud is envisioned to provide sensor data and other information which is needed for autonomous vehicles. Other work investigated the establishing of clusters of parked vehicles to form an information hub [2]. Here, parked cars are organized using virtual coordinate based routing concepts. Users are then able to store data inside (and retrieve data from) this network. Interdomain routing is envisioned to provide network connectivity and data management for disconnected operation. Another, already existing system is proposed by Barros [15]. It uses mesh routing between and across cars to provide Internet access to users. A similar solution coordinates delay-tolerant transfer of bulk data, e.g., between data centers, via a vehicular network embedded in a centralized system [16]. In our Car4ICT architecture, we go one step further and enable a large-scale vehicular network to provide services to other cars, outside users, and even sensor systems in future smart cities.

Using Service Discovery Protocols (SDPs) in VANETs has been investigated in recent years, although with no consent on a good solution. An early proposal, designed for MANETs, is by Sailhan et al. [17]. By building a virtual overlay network they aim to reduce the load on the network. Based on network densities virtual Service Directories (SDs) are established or removed. Services are described using the Web Service Description Language (WSDL) and sent following four geographic trajectories. This allows to search for a service and find one by also search in the same four trajectories. As nodes in their evaluation randomly move with a speed of 1 m/s it is unclear how well this works for VANETs, as cars are usually moving much faster. A protocol specifically designed for VANETs is proposed by Abrougui et al. [18]. Their system relies on infrastructure in the form of RSUs and on clustering mechanisms to reduce network load. RSUs are able to move small distances and are clustered around a service. Three types of services are supported in their approach: moving services, fixed services, and migratory ones. The last kind are fixed services staying in a certain region and are passed to other vehicles in case the current vehicle leaves the area. Lakas et al. present a service discovery solution with four different actors [19]. Instead of the usual three actors, Service Provider, Service Consumer, and Service Directory, they also see cars as mobile Service Directories. Their requests are flooded via broadcast until a user providing the service is found or the lifetime of the message is over. Because they rely heavily on broadcast the scalability of the proposed SDP remains unknown.

For routing in vehicular networks, there exist a lot of suitable solutions for urban or rural environments. There are reactive and proactive approaches, as well as protocols focused on georouting. As evaluating and comparing them is out of the scope of this chapter we refer to [4], which presents the best known algorithms.

The Car4ICT system will need to be able to identify available services in a location-based approach. The task of content identification is most closely related to future Internet research. At its core, a basic solution is to associate content with unique names. Here, concepts such as Information-Centric Networking (ICN) [20] and Named Data Networking (NDN) [21] have been proposed. Applying the concepts of such ICN to vehicular networks has been proposed in [22], yet with a focus on a single-use, single-protocol network. By only focusing on IEEE 802.11p communication, the authors were able to move away from IP and towards a new custom protocol to enable efficient ICN networking. Other architectures, by Wang et al. [23] and Grassi et al. [24], apply the concept of NDN to the vehicular domain. Both approaches, however, are targeting only a limited set of applications, optimizing the system for a specific use case and a specific communication technology. More recently, *Internames* [25] was proposed, a system that is envisioned to work with all kinds of approaches including IP, cellular, and ad-hoc networks. At its core, Internames maps all names to locations as well as the needed protocol to reach them. Still, it focuses very much on one application domain. Our architecture tries to act as a base for a multitude of applications, building upon concepts described in these approaches such as the association of items with hash tags. Further, current ICN and NDN solutions commonly fall short when it comes to the specific requirements in the very dynamic and disruptive environment of vehicular networks, which we try to directly address in Car4ICT.

III. The Car4ICT Concept

Our concept makes no further assumption than that there are some network-enabled vehicles driving on the road. Such vehicles can be envisioned to be available almost ubiquitously in future smart cities. No pre-installed infrastructure is assumed to be available (or to offer free capacity), though if it is, the network is assumed to be able to exploit it for improved performance. As discussed, this network of vehicles is designed to complement the available WiFi access points, hotspots on the streets, and cellular networks based on UMTS (3G) or LTE (4G), where these are available.

Building on this assumption, we aim to make the additional network of connected vehicles available to users. One use case might be to exploit the movement of a car when there is no other connection available – thus having a car deliver data to another region simply by exploiting its movement, either to the intended destination of the data or to an area where connectivity to the destination is available.

This enables the car network to act as central components of future ICTs systems of smart cities. Our concept revolves around requesting *services* or offering them to users: either smart devices or, of course, people. Potential use cases are people offering storage space, unused CPU power, or devices offering access to smart sensors. Users are envisioned to request access to these services (e.g., for processing large amounts of data on the go). Service requests and offers are mediated by cars, which act as coordinator: storing service offers and matching them with service requests. Two possible methods are allowed in the architecture: reactive service fulfillment where neighbors are actively queried once a request is received and pro-active exchange of service tables. Once a request and one or more offers have been matched, the Car4ICT system compiles a list of offers and returns it to the requesting entity. People or devices can then elect to use a service, exchanging data with the entity offering the service – again mediated by the Car4ICT system, which employs the network of vehicles to relay the data between the user requesting and the one offering the service. Such data relaying can also be done in multiple possible ways, from routing to pure store-carry-forward.

A. Use Cases

As the architecture is designed to be used both by people and by machines, various different use cases arise for Car4ICT. Consider, for example, smart sensors of future smart cities detecting a natural disaster. Because data validation and post-processing is a resource intensive task and because infrastructure might already be failing, they might exploit the Car4ICT system to transport large amounts of raw data to a central authority – or even to offload the tasks of post-processing and evaluating the data to CPU resources offered by the network itself.

As another use case, consider a tourist who wants to quickly store more pictures than can be kept on her camera. Without having to rely on compatible infrastructure or on a data plan in a foreign country she can rely on any technology (WiFi or Bluetooth being the most likely ones) to connect to any Car4ICT-enabled car to access the system. The system will try to match the incoming request for secure replicated storage to an existing service. If one is found, the tourist can use the Car4ICT network to offload pictures. Back at the hotel, the tourist can query any passing Car4ICT-enabled car to download the pictures back to her device. Automated systems can also make use of the Car4ICT architecture, e.g. a weather forecast system. To acquire the necessary input data for a certain region the system can now exploit Car4ICT to search for temperature sensors in that area. The temperature data can be offered by sensors at fixed locations in that area or cars which just happen to drive through it. Choosing now from a vast range of possible sensors, the forecast system is able to get lots of data via Car4ICT and generate the temperature forecast. Note that searching for such sensors and finally acquiring the data are two steps and allows the forecast system to decide from which sensors to get the temperature. More and more cars in recent years come equipped with so called dashboard cameras which constantly record videos or take pictures. This provides another use case related to semi-live picture sharing. Cars can offer the dashboard camera data via Car4ICT and are therefore able to provide a semi-live view of certain areas. Depending on the storage and network capacities of a car, it can offer only recent pictures or also older ones to provide a view into the past. With cars offering this data it is now possible to search for a place and gather pictures from this place taken by dashboard cameras. This gives the possibility to see how the area looked recently or at some point in the past.

B. Basic Architecture

Two types of entities interact in the Car4ICT architecture. First, *members* are the nodes that are connected to form the Car4ICT network. Members are responsible for transporting all control and data messages through the network. Members will typically be cars driving in a smart city, but parked cars or temporary support can be envisioned to participate just as well. The second type of entities, *users*, rely on any nearby Car4ICT member to serve as their gateway to the Car4ICT network. They inject new service requests or offers into the network and use the network to consume services. A Car4ICT user might be a person (e.g., a tourist accessing a service via their smart phone's WiFi, Bluetooth, or a DSRC connection) or a machine (e.g., a sensor in a building connecting wirelessly or even a subsystem of the car itself, connected by wire). The choice of which communication technology to use for interacting with a member is left up to the user. Depending on metrics like delay, connectivity, and cost, the user might want to rely on provided ad hoc communication or on a more costly LTE data connection. While persons should be enabled to make conscious choices about the technologies they use, machines will need to be equipped with a way of automatically reasoning about the suitability of available technologies – thus enabling them to use the Car4ICT architecture in true Machine-to-Machine (M2M) fashion. To prevent abuse of the system every user must be prepared to undergo a verification step before being allowed to offer or request services via a member of the Car4ICT network. Using, e.g., a Public Key Infrastructure (PKI) any member is equipped to verify a user's credentials (presented wirelessly when connecting to the network) and to reply with an individual grant.

Figure 4 illustrates this architecture: Members advertise their availability for connections by means of *access broadcasts*. Upon receiving such an access broadcast by a member, a user may choose to establish a connection to the Car4ICT network via this member. The user sends security credentials to the member, which are checked for the user's eligibility to access the Car4ICT network. After confirming the eligibility, the member responds with an individual access grant. This access grant can be used to prove eligibility to access the network – not just at the member that generated the grant. How and where this grant is generated is left open at this time.

A user that is authorized for access to the network is free to send offers to any Car4ICT member, or to send requests for services that will be matched with existing offers before being returned to the user. It is left to any member to decide whether a valid grant is necessary or whether this step can be skipped (e.g., because the member is a well-known device in the same car connected via a trusted wired connection).

Figure 5 illustrates how the entities of a Car4ICT system interact to exchange control messages. Shown are potential users requesting access from a member, the access being granted, and finally the user accessing a service offered (in this case, directly on the member the user is connected to). Also illustrated is proactive exchange of service tables between members and (via an optional cellular network) with an (optional) central server.

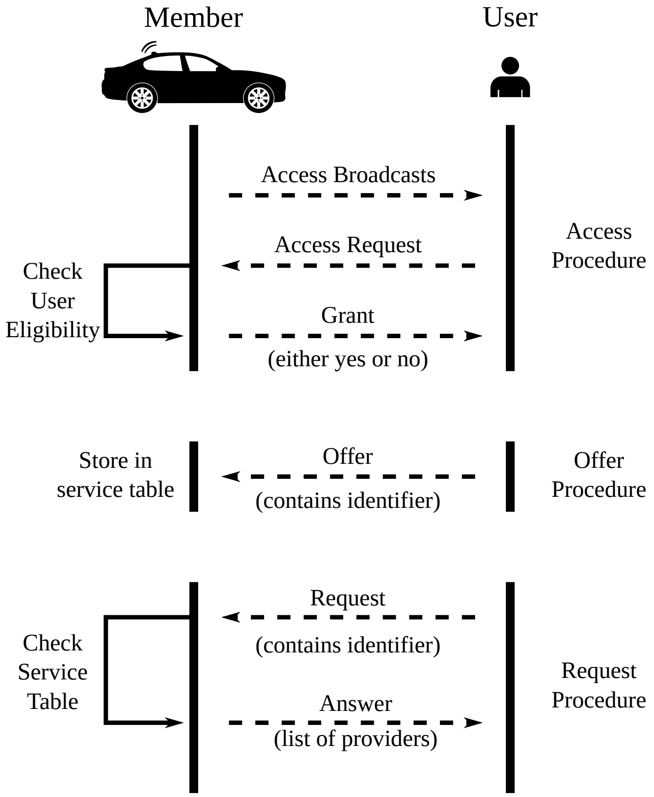


Figure 4. Entities of the Car4ICT architecture.

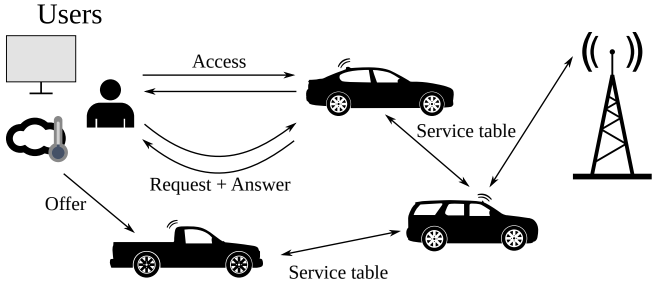


Figure 5. Illustration of control message exchange in the Car4ICT architecture.

Our architecture can serve as the base for providing a wide array of different services, e.g., data storage, distributed computing, sharing messages with a specific region, or retrieving sensor readings. Service offers can be stored on each member of the Car4ICT network by means of service tables, which can (but need not) be shared among neighbors. This enables each member to query its local table whenever a user requests a service. If the matching of requested service and known (or discovered) service offers was successful, an answer is sent back including the ID of the service. Naturally, not each service will be known everywhere, so there are multiple steps that can be taken to improve the chances of a query's success:

- A car could forward the request to neighboring cars which then in return could check their service tables for a potential match. This method probably reduces the overall network load but might increase the delay for a service discovery request.
- Another approach would be to share the service tables proactively. In such a case most likely the response time for requests decreases but the communication overhead would increase.
- As some of the cars also have a cellular link it could be helpful to store offers at some central server reachable via the Internet. If a car with such a connection receives a request which it cannot answer it can query this central service table and get an answer from there. In such a case the location information of the request will play an important role.
- Mixed approaches are also possible. A car may first execute an expanding ring search using only short-range radio communication. If this remains unsuccessful, and if sufficient incentive exists, a car might then use an LTE uplink to query more distant members.

When successful, this process yields a list of users offering the requested service, as well as metadata that indicates how these users might be reachable via the Car4ICT network. It is left up to the user to decide whether and which service(s) to use. For a person (e.g., using the system for replicated data storage), the incentives to use a given service might be different than those of a machine (e.g., for building a small scale weather, pollution, or allergen map). In any case, data transport to and from the service is taken care of by the Car4ICT network, that is, by its members.

C. Service and Neighbor Tables

As described, the operation of the Car4ICT architecture hinges on service tables, which are kept locally on Car4ICT members and which are continuously updated.

Such updates can occur by express messages, sent via connected users. Similarly, updates can be triggered by overhearing service announcements from other members. We believe that these service announcements can be sent in a bandwidth-conserving manner by exploiting other protocols. Current standardization by the IEEE and ETSI in the U.S. and Europe, to give two examples, assume the continuous exchange of Basic Safety Messages (BSMs) and Cooperative Awareness Messages (CAMs), respectively. They are employed to maintain neighbor information for road safety applications, but can just as well serve to support other applications, e.g., for maintaining neighbor tables. Similar to other applications, it might be feasible to piggyback (subsets of) service tables onto CAMs – if just to make vehicles aware that a certain service is still operating. If no updates for a known service are received for a certain duration (either via piggybacked information or in response to queries), it is the task of Car4ICT members to expire its corresponding entry in their service tables. Similarly, services that are only offered in (or are relevant to) a certain geographic region need to be purged upon leaving this area.

Services are identified, distinguished, and tracked using the following scheme: Each service is associated with a hash tag and meta data. The hash tag is a string of fixed length that uniquely identifies the type of service. This might be the hash of specific content being offered (e.g., a file), or it might be a well-known string referring to such resources as a certain type of sensor, CPU power, or replicated storage. In addition to the hash tag, a service is annotated with meta data pertaining to, e.g., geographic area or validity constraints. This scheme follows the concepts discussed in the ICN context [26] and has proven itself to be both flexible and extensible.

Table I shows an example service table, identifying multiple services by one hash tag each, along with meta data: Three different users (1, 3, and 7) are offering a certain video file, *file1* of a given length (2 GB), at different locations (*Paderborn* and *Tokyo*). Another file, an image named *file2*, is hosted by two users (1 and 12). User 7 is also offering both computation resources and storage, each with associated metadata describing the details of, e.g., the CPU and the storage offered.

Geographic position is considered to be a first-class metadata which has to be included in all messages. This can be exploited to use only services in a certain area or prevent service requests from leaving a certain area. For example, if a user wants to get a processing resource close to her location the request can be already stopped if it is outside of this area – therefore reducing the load on the network.

There are multiple ways to compare identifiers with each other. An identifier I_x is denoted by the tuple (h_x, M_x), where h_x is its hash and M_x a set of its metadata elements. We identified three ways of comparing two identifiers I_1 and I_2:

- **Hash only ($\overset{h}{=}$):** This comparison only uses the hashes, therefore $I_1 \overset{h}{=} I_2 \leftrightarrow h_1 = h_2$. It can be used for example if a user wants to search for storage but has no specific requirements.
- **Subset matching ($\subseteq$):** To match only a subset of an identifier with another the hashes have to be equal and all $m \in M_1$ have to be included in M_2, but not all $m \in M_2$ have to be in M_1. This can be used to search for processing power with some requirements like the type of CPU needed. Therefore $I_1 \subseteq I_2 \leftrightarrow (I_1 \overset{h}{=} I_2) \wedge (\forall m \in M_1 : m \in M_2)$.

Table I
A SERVICE TABLE EXAMPLE.

User	Hash tag	Meta data		
1	hash(file1)	location = Tokyo,	type = video,	size = 2GB
3	hash(file1)	location = Tokyo,	type = video,	size = 2GB
7	hash(file1)	location = Paderborn,	type = video,	size = 2GB
1	hash(file2)	location = Tokyo,	type = image	
12	hash(file2)		type = image,	size = 500MB
7	CPU	location = Paderborn,	type = ARM	
7	Storage	location = Paderborn,	type = hours,	size = 78GB

- **Equals ($=$):** This comparison compares both hashes to be the same and all $m \in M_i$ have to be in every other $M_j, j \neq i$. The comparison $=$ can be formulated as $I_1 = I_2 \leftrightarrow (I_1 \subseteq I_2) \wedge (I_2 \subseteq I_1)$.

As the geographical position is a first class metadata entry an Identifier I_x can be extended to a triple (h_x, pos_x, M_x). A user can now send a request with a certain position pos_u and a radius r. If the user requests now a so called *distance match* with the identifier I_u from the Car4ICT network a matching identifier I_x can be matched by the above comparisons. Only the first one has to be modified, as the others inherit the distance check from it: $I_u \overset{h}{=} I_x \leftrightarrow (h_u = h_x) \wedge (\text{dist}(pos_u, pos_x) <= r)$.

IV. Simulative Performance Evaluation

To validate and evaluate the proposed complex architecture a simulation framework is needed that can accurately model all of vehicle movement, user behavior, short range radio communication between users and members, as well as cellular communications where such infrastructure is available. We performed simulations using *Veins LTE* [6], a simulator developed with heterogeneous vehicular networking in mind. Such networks usually consist of cars communicating via multiple network technologies. For example, IEEE 802.11p based technologies (e.g., DSRC or ETSI G5) can be used to communicate with cars in close proximity, while a cellular network, e.g., LTE, is used to exchange data with cars farther away or to connect to the Internet. To be able to simulate such networks we developed *Veins LTE* and published it as Open Source software. Veins LTE, as the name implies, is based on *Veins*, which itself is based on the discrete network simulator OMNeT++ and connects to the mobility simulator *SUMO*. *Veins* has a so called feedback loop which allows the network simulator (OMNeT++) to react on events from the mobility simulator (SUMO) and vice versa. Such a feedback loop enables the simulation of scenarios which are dependent on information from the network and allows cars to change their route during the simulation (which would be not the case with trace-based simulations). While Veins already provides means to simulate IEEE 802.11p based networks, *Veins LTE* itself adds *SimuLTE* [27] to the mix – which provides the possibility to simulate the complete LTE stack. With this addition two network technologies (IEEE 802.11p and LTE) can be simulated in great detail by using a single simulator.

We conducted an initial study of using a heterogeneous vehicular network for cooperative intersection collision avoidance according to an algorithm proposed by Tung et al. [7] to ensure that the combined simulation framework was behaving as expected. As a simple indication, Figure 6 shows a sample result, the delay in the LTE uplink. As can be seen the delay increases with the frequency of sent messages. It also should be noted that the delay is shown independent of the transmission distance – this also adds longer delays for messages as more re-transmissions are needed.

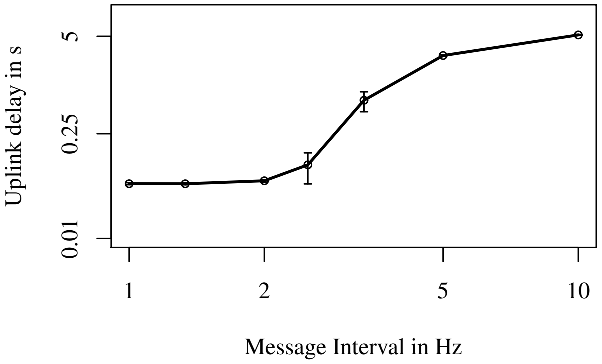


Figure 6. LTE message interval vs. LTE uplink delay (mean and 95 % confidence interval) for an allocation of 100 RB (20 MHz bandwidth). Note that both scales are logarithmic [6].

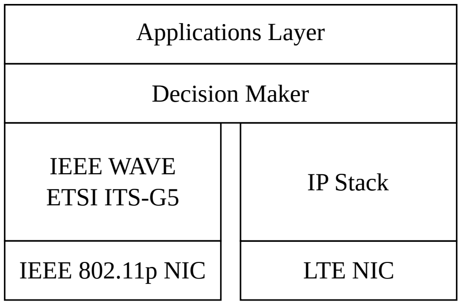


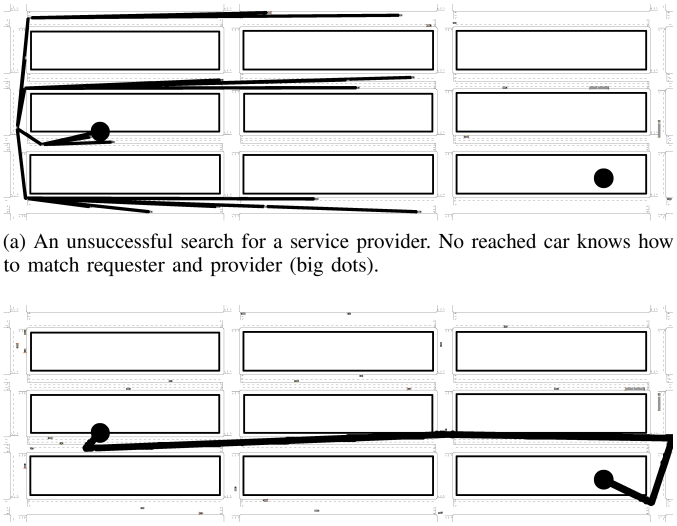
Figure 7. The heterogeneous stack of Veins LTE [6].

On top of *Veins LTE*, applications for such heterogeneous vehicular networks can be developed. Any application can decide to send the packet via a specific network stack or let a dedicated module – named *Decision Maker* – decide which network technology is better suited for the current transmission. The stack can be seen in Figure 7. As Car4ICT is meant to work in exactly such a heterogeneous vehicular network where different technologies can flexibly be used when and where available, this simulation environment is the perfect fit for the evaluation and the validation of Car4ICT.

We added a new *User* module to the simulator which is able to run multiple applications and has a *Connector* which is in charge of connecting the user (and its applications) to the Car4ICT network. Applications then send messages to the *Connector* which in turn forwards it, if a connection exists, to a car. On the car side we developed the *Car4ICT module*. This module has three main tasks:

1) If an offer an offer is received it is stored in the service table.
2) If a message arrives that is neither a request nor an offer, it should be, if possible, routed towards its destination.
3) As cars can also include applications, the *Car4ICT module* also offers an interface for such applications. Our simulation is built so applications can be easily used in cars and by users – only a single flag has to be set which defines the connection type (e.g., local or wireless).

Figures 8a and 8b show screenshots of a running Car4ICT simulation. Both times, a user requests a service provided by another user (big dots). In the first case, shown in Figure 8a,



(a) An unsuccessful search for a service provider. No reached car knows how to match requester and provider (big dots).

(b) A successful search for a service provider. Shown is an example path from provider to requester.

Figure 8. Screenshots of a running simulation.

service discovery remained unsuccessful. As the first car – the one the user connects to – does not have a corresponding entry in its service table, the request is sent further to other cars, but the request expires before reaching a valid destination. In the second case, shown in Figure 8b, a service is found and the user requesting it is informed. A thick line indicates the hops a data packet might take between the requesting user and the one offering the service. Note that there are two ways of answering to requests: *pro-actively* by sharing the service tables upfront and *reactively* by searching for a provider after a request can not be answered.

V. Simulation Study

To illustrate the feasibility of the Car4ICT architecture, we conclude with an evaluation of the performance of service discovery in our Car4ICT framework under progressively worse conditions.

We used the described simulation framework, implementing the outlined approach in detail. Each member (that is, each car) in the simulation periodically announces its presence to users nearby. We opted for implementing a *pro-active* Car4ICT service discovery variant, exchanging service tables in fixed time intervals. Users periodically request different services. As long as the request cannot be matched to an offer by the Car4ICT architecture the request is retried. When the request is successfully matched and a positive response is delivered to the user, we note the delay between first try and fulfillment as the *discovery latency*.

Our implementation supports all described ways of comparing service identifiers. The two most important ones for when a user is searching for a service are $\overset{h}{=}$ and $\subseteq$. The first one, $\overset{h}{=}$, can be used to get an overview of all offered services of a certain kind in an area. Secondly, $\subseteq$ helps a user to find a service with some prerequisites. The comparison $=$ is mostly used when comparing new offered services if they do not already exist in the service table.

Table II
Used simulation parameters.

Parameter	Value
simulated area	0.7 km^2
average number of equipped cars per km^2	35–415
total number of users	15
number of users requesting	5
number of users offering	5
IVC technology	IEEE 802.11p
IVC maximal transmit power	10 mW
simulation duration	80 s
service table broadcast interval	0.1–10 s
neighbor table entry lifetime	10 s
service table entry lifetime	10 s
user request interval	2 s
request timeout	30 s

As the simulated traffic scenario we chose a challenging Manhattan Grid topology, populated by as much as 415 and down to as little 35 equipped cars per square kilometer taking random trips. Both road and building dimensions correspond to downtown Manhattan. Figure 8 shows a small section of this scenario. Buildings next to each road shield radio transmissions across streets, thus cutting down further on the number of potential communication partners in low density scenarios. To make the scenario more challenging, we add exactly 5 users connecting to a close by vehicle and offering a service, 5 users requesting this service, and 5 more users offering unrelated services. This scenario closely mirrors the discussed use case of secure replicated storage, but with a tightly limited amount of available resources. All parameters are summarized in Table II.

We illustrate results of our discovery latency evaluation in the form of an eCDF as service requests can take anything from being instantaneously fulfilled to never being fulfilled (in increments of two seconds, the retry interval).

In Figure 9 we show results for different traffic densities (measured in vehicles per km^2), given a service table broadcast

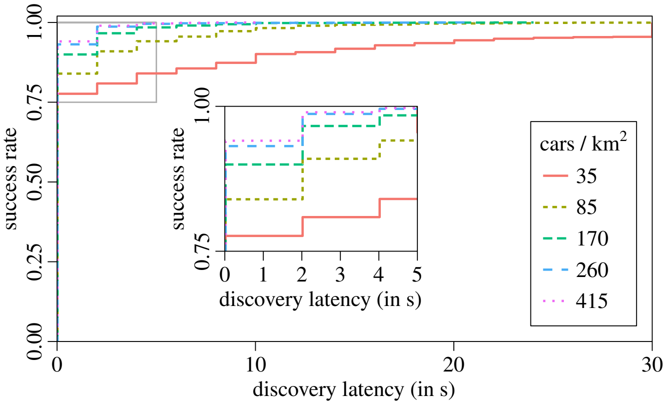


Figure 9. The discovery latency until a service is found for different traffic densities. The service table broadcast interval was set to 10 s. The inlay shows a zoom to the first 5 s.

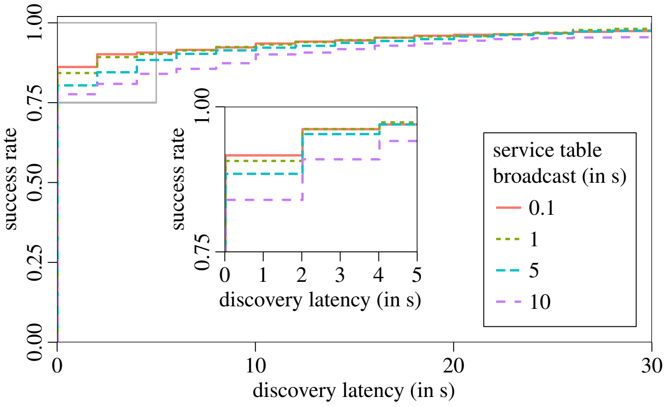


Figure 10. The discovery latency until a service is found for different service table broadcast intervals. The traffic density was 35 vehicles per km^2.

interval of 10 s. As could be expected, traffic density has an immediate impact on discovery latency. While, for high densities of 415 vehicles per km^2, it takes no more than 2 s to match 99 % of service requests to a corresponding offer, at slightly lower densities of 85 vehicles per km^2 this fraction already drops to as 90 %, as can be seen in the zoomed in part of the figure. At this density, fulfilling 99 % takes as much as 12 s. At lower densities, even 30 s are not enough to fulfill 99 % of requests. While this would be well in the acceptable range for M2M communication like in a distributed sensing use case, it would likely be above the tolerable threshold for a user facing service.

We thus look to smaller service table broadcast intervals as a potential solution at low traffic densities. Figure 10 illustrates our results, now using service table broadcast intervals as low as 0.1 s. We observed that adjusting the service table broadcast interval from 10 s to 1 s was enough to achieve a better success rate than with these lower rates. Even for medium to low discovery latency constraints, system performance can be seen to be in an acceptable range; not just for M2M services but also for user facing services like in the discussed secure replicated storage use case.

Decreasing the interval further yielded no substantial performance gain. Thus, while an adaptive service table broadcast interval would certainly help conserve channel capacity at high vehicle densities, configuring a moderate service table broadcast interval at any traffic density yields a system that delivers good performance at both high and low traffic densities.

VI. Conclusion

We outlined our idea for a novel architecture, which we term Car4ICT, that uses cars as a main ICT resource of smart cities. We showed how such an architecture can help cope with the forecasted massive data demand of smart cities – and how it can operate in disaster situations where no infrastructure might be available. Based on this architecture, a huge variety of new applications is made possible in a network architecture going beyond current information-centric networking systems. Our concept revolves around offering and requesting services, serving smart devices just as well as people. We highlighted use cases of this architecture like offering storage space, unused CPU power, or devices offering access to smart sensors. We presented a flexible and extensible scheme for identifying, distinguishing, and tracking services. The architecture is technology agnostic, using any of multiple available wireless technologies to connect cars to each other and with users. This necessitated a custom built simulation framework, which we published as Open Source. Based on this framework, we illustrated the feasibility of the proposed Car4ICT architecture, concluding with an evaluation of the performance of service discovery under progressively worse conditions. We were able to show that even in such conditions simple tweaks to protocol parameters can deliver good user experience in terms of low service discovery latencies.

References

[1] A. Caragliu, C. Del Bo, and P. Nijkamp, “Smart Cities in Europe,” *Journal of Urban Technology*, vol. 18, no. 2, pp. 65–82, 2011.

[2] F. Dressler, P. Handle, and C. Sommer, “Towards a Vehicular Cloud - Using Parked Vehicles as a Temporary Network and Storage Infrastructure,” in *15th ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc 2014): ACM International Workshop on Wireless and Mobile Technologies for Smart Cities (WiMobCity 2014)*. Philadelphia, PA: ACM, Aug. 2014, pp. 11–18.

[3] O. Altintas, K. Seki, H. Kremo, M. Matsumoto, R. Onishi, and H. Tanaka, “Vehicles as Information Hubs During Disasters: Glueing Wi-Fi to TV White Space to Cellular Networks,” *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 1, pp. 68–71, Jan. 2014.

[4] C. Sommer and F. Dressler, *Vehicular Networking*. Cambridge University Press, Nov. 2014.

[5] O. Altintas, F. Dressler, F. Hagenauer, M. Matsumoto, M. Sepulcre, and C. Sommer, “Making Cars a Main ICT Resource in Smart Cities,” in *34th IEEE Conference on Computer Communications (INFOCOM 2015), International Workshop on Smart Cities and Urban Informatics (SmartCities 2015)*. Hong Kong: IEEE, Apr. 2015, pp. 654–659.

[6] F. Hagenauer, F. Dressler, and C. Sommer, “A Simulator for Heterogeneous Vehicular Networks,” in *6th IEEE Vehicular Networking Conference (VNC 2014), Poster Session*. Paderborn, Germany: IEEE, Dec. 2014, pp. 185–186.

[7] L.-C. Tung, J. Mena, M. Gerla, and C. Sommer, “A Cluster Based Architecture for Intersection Collision Avoidance Using Heterogeneous Networks,” in *12th IFIP/IEEE Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2013)*. Ajaccio, Corsica, France: IEEE, Jun. 2013.

[8] A. Vesco, R. Scopigno, C. Casetti, and C.-F. Chiasserini, “Investigating the Effectiveness of Decentralized Congestion Control in Vehicular Networks,” in *IEEE Global Telecommunications Conference (GLOBECOM 2013)*. Atlanta, GA: IEEE, Dec. 2013.

[9] M. L. Wald, “U.S. Plans Car-to-Car Warning System,” *The New York Times*, p. B3, February 4 2014.

[10] W. Viriyasitavat, S.-H. Yu, and H.-M. Tsai, “Channel model for visible light communications using off-the-shelf scooter taillight,” in *Vehicular Networking Conference (VNC), 2013 IEEE*. Boston, MA: IEEE, Dec. 2013, pp. 170–173.

[11] W. Bronzi, R. Frank, G. Castignani, and T. Engel, “Bluetooth Low Energy for Inter-Vehicular Communications,” in *6th IEEE Vehicular Networking Conference (VNC 2014)*. Paderborn, Germany: IEEE, Dec. 2014.

[12] J. Hasch, E. Topak, R. Schnabel, T. Zwick, R. Weigel, and C. Waldschmidt, “Millimeter-Wave Technology for Automotive Radar Sensors in the 77 GHz Frequency Band,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 60, no. 3, pp. 845–860, Mar. 2012.

[13] F. Dressler, H. Hartenstein, O. Altintas, and O. K. Tonguz, “Inter-Vehicle Communication - Quo Vadis,” *IEEE Communications Magazine*, vol. 52, no. 6, pp. 170–177, Jun. 2014.

[14] E. Lee, E.-K. Lee, M. Gerla, and S. Oh, “Vehicular Cloud Networking: Architecture and Design Principles,” *IEEE Communications Magazine*, vol. 52, no. 2, pp. 148–155, Feb. 2014.

[15] J. Barros, “How to Build Vehicular Networks in the Real World,” in *15th ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc 2014)*. Philadelphia, PA: ACM, Aug. 2014, pp. 123–124.

[16] B. Baron, P. Spathis, H. Rivano, M. Dias De Amorim, Y. Viniotis, and J. Clarke, “Software-Defined Vehicular Backhaul,” in *IFIP Wireless Days Conference 2014*. Rio de Janeiro, Brazil: IEEE, Nov. 2014.

[17] F. Sailhan and V. Issarny, “Scalable Service Discovery for MANET,” in *Third International Conference on Pervasive Computing and Communications (PerCom 2005)*. Kauai, Hawaii: IEEE, March 2005, pp. 235–244.

[18] K. Abrougui, A. Boukerche, and R. Pazzi, “Design and Evaluation of Context-Aware and Location-Based Service Discovery Protocols for Vehicular Networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 717–735, September 2011.

[19] A. Lakas, M. A. Serhani, and M. Boulmalf, “A hybrid cooperative service discovery scheme for mobile services in VANET,” in *7th International Conference onWireless and Mobile Computing, Networking and Communications (WiMob)*. Shanghai, China: IEEE, October 2011.

[20] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, “A Survey of Information-Centric Networking,” *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, Jul. 2012.

[21] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, k. claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, “Named Data Networking,” *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 66–73, Jul. 2014.

[22] M. Amadeo, C. Campolo, and A. Molinaro, “CRoWN: Content-Centric Networking in Vehicular Ad Hoc Networks,” *IEEE Communications Letters*, vol. 16, no. 9, pp. 1380–1383, Sep. 2012.

[23] L. Wang, R. Wakikawa, R. Kuntz, R. Vuyyuru, and L. Zhang, “Data Naming in Vehicle-to-Vehicle Communications,” in *31st IEEE Conference on Computer Communications (INFOCOM 2012): Workshop on Emerging Design Choices in Name-Oriented Networking*. Orlando, FL, USA: IEEE, Mar. 2012, pp. 328–333.

[24] G. Grassi, D. Pesavento, L. Wang, G. Pau, R. Vuyyuru, R. Wakikawa, and L. Zhang, “ACM HotMobile 2013 Poster: Vehicular Inter-networking via Named Data,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 17, no. 3, pp. 23–24, Jul. 2013.

[25] N. B. Melazzi, A. Detti, M. Arumaithurai, and K. K. Ramakrishnan, “Internames: a name-to-name principle for the future Internet,” in *10th International Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine 2014)*. Island of Rhodes, Greece: IEEE, Aug. 2014, pp. 146–151.

[26] S. Farrell, D. Kutscher, C. Dannewitz, B. Ohlman, A. Keranen, and P. Hallam-Baker, “Naming Things with Hashes,” IETF, RFC 6920, Apr. 2013.

[27] A. Virdis, G. Stea, and G. Nardini, “SimuLTE - A Modular System-level Simulator for LTE/LTE-A Networks based on OMNeT++,” in *4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2014)*, Vienna, Aug. 2014.