# Exploring the Design and Implementation of Vehicular Networked Systems

Liviu Iftode
Department of Computer Science, Rutgers University

Cristian Borcea
Department of Computer Science, New Jersey Institute of Technology

Nishkam Ravi
Department of Computer Science, Rutgers University

Tamer Nadeem
Department of Computer Science, University of Maryland, College Park

## 1 Introduction

In an increasingly computerized and connected world, one of the most ubiquitous technological presences in our life, the car, has been "resisting" this invasion. At a time when gadgets exploiting computing and wireless networking bombard us from every direction, cars remain a conservative technological island with the inter-car communication mostly visual (break and reverse lights, turn signals, horn, etc), executed by the driver and invented almost a century ago [1]. When visibility is limited, such as on a foggy day, around the corner, or just further ahead on a congested road, there is no alternative source of information to assist the driver in his decisions. Moreover, beyond passive safety technologies such as seat belts and airbags, which have been substantially improved over the last decade, there is a tremendous opportunity to move forward to active safety using active accident avoidance technologies.

In spite of current advancements in vehicular technology, today, the tail of traffic safety remains the same. According to a recently released U.S. Traffic Safety Facts Report between 1973 and 2003 the number of reported traffic accidents has remained relatively constant. Furthermore, the number of reported fatalities has decreased by only 10 percent between 1988 and 2003 (42,130 vs. 38,252). Not surprisingly, this mirrors closely with similar statistics recently released by the EU. Beyond the cost in human lives, the financial impact of this is staggering. The annual costs associated with traffic accidents are nearly 1 Trillion US dollars. This is equivalent to almost 3 percent of the world's Gross Domestic Product (GDP). The picture is clear, as more vehicles are added to the world's roadways, the situation can only get worse unless the technology improves dramatically.

For this to happen, the adoption of wireless communication technology to support computer-based inter-car communication protocols is required. The vision is an intelligent transportation system composed of smart vehicles exchanging data. This exchange of data enables these vehicles to execute applications for safe driving, traffic condition monitoring, dynamic route planning, e-commerce, etc. The recently adopted Dedicated Short-Range Communication (DSRC) spectrum for vehicle-to-vehicle and vehicle-to-infrastructure multi-channel communication is an indication of the increasing interest and expectations from this emerging technology.

At first glance, the research community seems to have already accumulated the necessary results to solve this challenge. The Ad hoc networking research community finally appears to have a long awaited killer application domain, to which an impressive body of work can be directly applied. There are also significant results in other areas such as human-computer interaction, security, and computer vision that seem relevant

---

[1] Yellow brake lights were introduced in 1915; The reverse light was invented in 1921 and mandated only in the 1960s; Buick introduced the first electric turn signals in 1938; third brake light was invented in 1974 and mandated in 1986)

in this case. If indeed this is the case, what holds up rapid technological advances toward the desired new generation of vehicles? We believe that the answer is multi-dimensional: social, legal, psychological, but ultimately technical. There is still an immense engineering gap between the research studies and their implementation. Wireless communication is ubiquitous yet extremely unreliable, with serious bottlenecks, predisposed to congestion, causing non-negligible delays and connection failures. Node/car heterogeneity, short yet real time car-to-car interactions, trust and privacy are additional problems which seriously conflict with the strong guarantees expected by the manufacturers and the drivers; consequently, they make adoption of this technology more complicated. Vehicular computing and networking is an area that benefits from all of these pre-existent research fields, but it is much more than a simplistic integration of their results.

In this paper, we describe our vision of a inter-vehicular networking system focussed on car-to-car communication. We point out the key challenges and obstacles that stand in the way of vehicular computing becoming a reality. In particular, we present the design of a network protocol stack consisting of several new, vehicle-specific layers such as (i) Data Validation and Caching, (ii) Store and Forward Message Routing, (iii) In-Network Processing, and (iv) a Location-aware Publish-Subscribe framework, which uses GPRS/3G links to efficiently route high-priority messages between disconnected ad-hoc networks. The design decisions in this paper are based on our experience with real-life experiments over TrafficView [32, 13], which is a simple system for traffic data dissemination and visualization over wireless ad hoc networks.

Rest of the paper is organized as follows. We discuss related work in Section 2. In Section 3 we present the design of a hybrid vehicular network architecture. Section 4 describes a prototype implementation of the network stack that is currently underway. We draw preliminary conclusions in Section 5.

## 2 Related Work

Several major automobile manufactures have already begun to investigate inter-vehicular networks: GM research center in CMU [1], BMW Research Labs [11], and Ford Research Labs [30]. In Europe, a large consortium of car manufacturers and universities have received significant research funds to develop a platform for inter-vehicle and vehicle-to-roadside communication [6].

Dedicated Short-Range Communications (DSRC) is 75 MHz of spectrum at 5.9 GHz allocated by Federal Communications Commissions (FCC) to “increase traveler safety, reduce fuel consumption and pollution, and continue to advance the nation’s economy”. DSRC will support safety critical communications, such as collision warnings, as well as other valuable Intelligent Transportation System Applications such as Electronic Toll Collection (ETC).

One of the earliest studies on inter-vehicular computing was started by JSK (Association of Electronic Technology for Automobile Traffic and Driving) of Japan in the early 1980s. Later, well-known research results on platooning have been demonstrated by California PATH [17] and Chauffeur of EU [14]. The cooperative driving systems of Japan in the late 1990s and 2000 (e.g DEMO 2000 [39]) exhibit another set of important applications of IVC. The European Project CarTALK [37] tries to cover problems related to safe and comfortable driving based on IVC.

In recent years, traffic safety applications have been developed by several research groups. In [20], a wireless traffic-light system is presented, where information about current light status, location of intersection, and a reference point are broadcasted periodically. In [30], the authors present a collision warning system that exchanges beacon messages in a peer-to-peer fashion. In [40], the authors propose a method to avoid congestion in propagating warning messages.

The work presented in [12] describes an Inter-Vehicle Communication system (IVC) with Vehicle-Roadside Communication (VRC), where both moving vehicles and base stations can be peers in the system. Security mechanisms for inter-vehicle communication are presented in [5]. In [34], the authors examine resource discovery using an opportunistic approach in inter-vehicle ad hoc networks in an urban area, where moving vehicles communicate with each other via short-range wireless transmissions. The CarNet [31] project has focused on providing IP connectivity to radio nodes in vehicles with the help of a grid location service [29].

Traditionally, vehicular network research has assumed communication between vehicles and base stations that would relay the messages across the Internet. With the advances in mobile ad hoc networks research [24, 36, 41], the idea of vehicle-to-vehicle communication using short-range wireless network interfaces has become more attractive [23, 32, 35]. However, in addition to the similarities to mobile ad hoc networks such as short radio transmission range, low bandwidth, omni-directional broadcast (at most times), and low storage capacity, vehicular networks using short-range wireless communication have their unique characteristics and challenges. The network topology is continuously and rapidly changing. Because of the relative movement of the vehicles, the connectivity between vehicles is always changing. For example, if vehicles speed is around 60 mph (i.e., 25m/s) and the transmission range of wireless networks is 250m, the connectivity between two vehicles could last for at most 500/25 = 20 seconds.

Vehicular ad hoc networks will be frequently partitioned. In the case of low vehicle density, the gap between two vehicles might be several miles, far beyond the transmission range of wireless networks. In turn, the disconnection time could be minutes. Due to the fast movement of vehicles, and high dynamic traffic conditions, this situation is not uncommon.

Unlike general mobile ad hoc network where it is hard to predict the mobility of the nodes, vehicles normally run along roads which remain unchanged over years. Therefore, given the average speed, current position, and road trajectory of a specific vehicle, the future position of that vehicle can be predicted.

Energy is not a big issue in vehicular network as it is the case with sensor networks [25, 22, 18, 19]. In sensor networks, the nodes are battery-powered, and it is not easy to replace the battery after deployment. This limits the lifetime of a node in such a network. Therefore, a lot of effort has been made to conserve energy in sensor networks. On the other hand, in a vehicular network, the vehicle itself can be used as a source of electric power, and therefore, energy efficiency does not influence the design of network protocols.

Vehicular networks hold great promise for improving safety levels and avoiding traffic congestion. At the same time, they present great potential for abuse. Golle et al [15] classify adversarial attacks based on their nature, target, scope and impact. They propose a general, sensor-driven technique that allows nodes to detect sources of incorrect information, by validating information against a set of rules that are dictated by vehicular network models and physics. Their approach can thwart a limited number of attacks, primarily those that contradict with sensor information that the node directly obtains. Newsome et al [33] study the threat posed by Sybil attacks and propose identity based authentication, where identity could be the number plate of the driver or an electronic identity assigned by a trusted third party, or location itself. This raises serious privacy issues. Besides, as the number of participants in the network grows larger, the task of maintaining and revoking identities becomes unmanageable. Reputation or recommendation based security solutions do not scale up to highly mobile networks as the existence of groups is very short lived. A highly mobile node switches between different groups very frequently. Assigning and changing node identity and corresponding reputation information is practically impossible.

## 2.1 TrafficView

TrafficView [32, 13] is a scalable traffic monitoring system that works on top of vehicular ad hoc networks. In TrafficView, vehicles are equipped with short range wireless network interfaces (i.e., IEEE 802.11) and form ad hoc vehicle-to-vehicle networks to exchange traffic information. A device embedded in the vehicles receives this information and provides the drivers with a real-time view of the road traffic far beyond what they can physically see. Each TrafficView device installed in a vehicle is composed of three components: an embedded computer with display, a GPS receiver, and a short-range wireless network interface (IEEE 802.11b). The GPS receiver provides the location, speed, current time, and direction of the vehicle. Each vehicle equipped with TrafficView gathers and broadcasts information about other vehicles in a peer-to-peer fashion using the short-range wireless interface. The display shows a map of the road ahead of the vehicle, annotated with dynamic and real time information exchanged over the ad hoc wireless network.

Figure 1 presents the TrafficView software architecture. Each vehicle stores records about other vehicles in its local *validated* dataset. Local GPS readings are periodically generated and adjusted through the *navigation* module before storing them. The main job of the navigation module is to determine the road on which a car is moving. When a record is received through a broadcast message, it is stored in the *non-validated* dataset, since it might contain outdated or conflicting information. After these records are examined for validity, they are merged with the validated dataset. Periodically, the system displays data from the validated dataset.

In TrafficView, we have chosen to broadcast all data stored at a vehicle in a single packet. This simple data propagation model has three advantages: (1) it limits the bandwidth consumed by each vehicle, (2) it

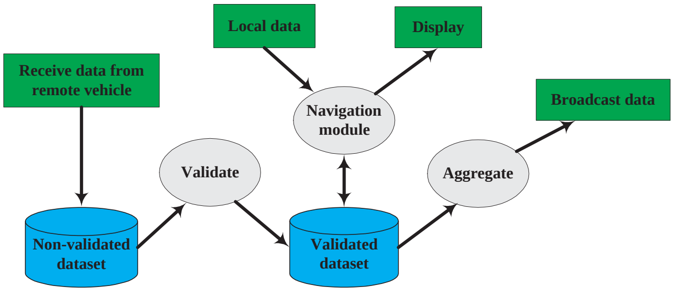

Figure 1: TrafficView Software Architecture

Figure 2: Video snapshot of TrafficView experiment with 3 cars on US Highway 1, in New Jersey. The left part shows the three cars on the road, while the right part shows the information displayed to the driver in the third car.

Figure 3: EZCab driver system consisting of an HP iPAQ equipped Orinoco 802.11 wireless card and Geko 201 GPS receiver

limits the number of re-transmissions due to collisions, and (3) it avoids dealing with flow control (which would be necessary if data would be split in multiple packets). Of course, the data stored at a vehicle is usually greater than the size of a packet. Therefore, data compression/aggregation techniques should be applied to the records exchanged. We have focused on semantic data aggregation. (which is implemented by the *aggregation* module). For example, the records about two vehicles can be replaced by a single record with little error, if the vehicles are very close to each other and move with relatively the same speed; in other their relative distance is always in a small range. The way data aggregation contributes to the TrafficView system is by delivering as many records as possible in one broadcast message. Thus, more new records can be delivered in a certain period of time, and the overall system performance is improved. Since the aggregation introduces certain errors, the challenge is to maintain good accuracy of the traffic data while increasing the "visibility" for the drivers.

We have implemented a TrafficView prototype on top of Linux. This prototype uses 802.11b PC cards for communication and disseminates data in the form of UDP packets. As testbed, we have used HP iPAQ PDAs. The prototype obtains the information about location, time, and speed from a GPS receiver attached to iPAQs through the serial interface. During our initial real-life traffic experiments on the roads, we have realized that having the iPAQs inside the cars decreases the communication range significantly. Therefore, we have attached omni-directional antennas which actually increase the communication range to up to 500 meters. On the other hand, we have observed that the speed does not influence the communication significantly.

The most difficult problem that we have been confronted with was how to accurately map the position of the cars to the roads. This is very important because we need to show only the traffic going on the same road with a given car. The issue was mainly generated by the inaccuracy of raw GPS data. Usually the accuracy that we got was between 5 and 10 meters. But sometimes, we have experienced double or triple values. Even with the usual accuracy, however, identifying the road close to intersections is not a trivial problem. A second problem involved the inaccuracy of the Tiger Line database [9] which we used as a digital road map of all the roads in the United States. Nevertheless, we have incorporated various smoothing techniques in the navigation module, and consequently, we have achieved 100% road identification when the roads have been divided in 5 meters long sub-segments [13].

We have run multiple experiments with up to five cars in real-life traffic conditions. Figure 2 presents a video snapshot from one of our experiments. Except for small intervals of time when some of the cars lost connectivity, TrafficView performed well. For instance, the drivers could see all the cars ahead of them on the display even though they where behind a curve on the road.

### 2.2 EZCab

EZCab [35] is a ubiquitous computing application built over vehicular ad hoc networks, which allows people to book nearby cabs in densely populated urban areas using their cell phones or PDAs equipped with short-range wireless network interfaces. Current cab booking systems rely on centralized schemes for cab dispatching such as making phone calls to a taxi company or sending short messages (SMSs) to a certain server over cellular links [2, 3]. Although under the traditional centralized solution cab dispatching is guaranteed, this solution is not scalable due to: 1) all requests have to go through one or multiple cab dispatchers, which introduces waiting time for the clients, especially during periods of peak cab requests, and 2) in order to dispatch the nearest cab to the client, all cabs in the city have to be monitored to find the closest one to the client's location.

EZCab consists of a mobile ad hoc network of computers embedded in taxis and client handheld devices, which communicate using short-range wireless network interfaces such as IEEE 802.11. Instead of booking cabs through a centralized dispatcher, EZCab clients book free nearby cabs by communicating directly with other EZCab nodes (i.e., taxis) over a mobile ad hoc network. This decentralized architecture provides a simple, cheap, and scalable solution to a real-life problem. However, EZCab presents new challenges which do not exist in traditional systems based on centralized dispatching centers. For instance, we need a distributed protocol to ensure that at most one cab arrives at the site of the client who requested the cab. Furthermore, we must ensure that any free cab accepts only one client request at any point in time. To provide an automatic booking mechanism, we also need accurate location information for both clients and cabs (e.g., the client's street address has to be present in the booking request). Finally, EZCab needs to provide a mechanism for the client and driver to authenticate each other when they meet.

We have implemented the EZCab prototype on top of the SM platform installed on HP iPAQs running Linux. The iPAQs use Orinoco's 802.11 cards for wireless communication, and each of them is connected to a Geko 201 GPS receiver. Figure 3 illustrates an EZCab driver system. The EZCab prototype has two types of graphical user interfaces, corresponding to client system and driver system, respectively. Each user interface communicates with the SM platform via special bi-directional I/O tags, called UI tags. These tags persist for the entire duration of the user interface process and behave similar to a producer-consumer circular buffer. We have used the Open Palmtop Integrated Environment (OPIE) to develop the user interfaces of EZCab. OPIE is an open source graphical user environment for PDA's and other devices running Linux.

## 3 A Practical Hybrid Vehicular Network Architecture

A common network architecture for inter-vehicular communication is necessary to realize an intelligent transportation system supporting safe driving, dynamic route scheduling, emergency message dissemination, traffic condition monitoring, etc. The envisioned applications for an intelligent transportation system will need three types of communication across vehicular networks:

- **Real-time dissemination of traffic information.** The network can disseminate traffic monitoring messages, road condition messages, or accident reports. For instance, a driver can receive traffic information from several miles ahead of her car on a highway. In a city, a driver can receive traffic

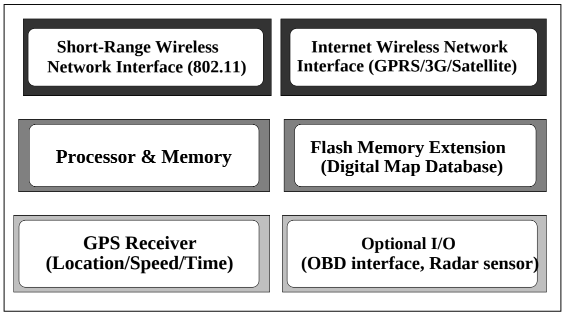

Figure 4: Vehicular networked embedded system

information from the current street as well as from several streets that intersect the current street. The network should avail this traffic information to every vehicle on the roads, with small delay and low bandwidth cost.

- **Traffic information queries.** Besides receiving the disseminated traffic information, a vehicle should be able to query information about specific targets which are not covered by the data already received. For example, a vehicle can query about the average vehicle speed in a region, road condition at a certain exit, or empty parking spots on a given street.
- **Reliable information exchange.** Although not the common case, reliable communication can be necessary in vehicular networks. For instance, the network must ensure that certain traffic warnings arrive to all the intended vehicles. Furthermore, reliability is necessary for certain queries or distributed applications (e.g., music sharing, back-seat network games).

To support these three types of communication, we propose a hybrid vehicular network architecture, where the communication between vehicles takes place over both ad hoc networks through short-range wireless network interfaces (IEEE 802.11) and the Internet (using cellular or satellite connections and intermediate "proxy" servers). In this network architecture, vehicles will form ad hoc networks using the short-range wireless interfaces and exchange ordinary traffic information over these networks. The cellular/satellite connectivity is used only for extraordinary situations where reliability is needed or network partitions preclude answering certain queries.

Each vehicle will be equipped with an embedded system as depicted in Figure 4. Besides the dual wireless connectivity, the system features a flash memory extension that will store a digital map of the roads. The digital map together with the location information provided by the GPS receiver will be used to position the vehicles on the roads. The GPS receiver also provides the speed of the vehicle and current time. Our preliminary work demonstrated that the accuracy of time synchronization using GPS data is suitable for the type of applications executing in vehicular networks. Optionally, the embedded system can incorporate more I/O devices such as radar sensors to detect obstacles around the car or the on-board diagnostics system (OBD) interface [7], present in all vehicles sold in US after 1996, which can be used to acquire mechanical and electrical data from sensors installed in vehicles.

## 3.1 Architectural Principles

This research tries to identify the main architectural principles that differentiate vehicular networks from other networks such as the Internet, sensor networks, and even general mobile ad hoc networks. Once identified, these principles can serve as guidelines for the design and implementation of a vehicular network architecture.

### 3.1.1 Scalability and Reliability through Hybrid Communication

A central theme of the vehicular network design is scalability. These networks cannot solely rely on Internet access (e.g., GPRS/3G, satellite, IEEE 802.20) because such a solution will not be scalable. First, the providers cannot offer enough bandwidth for all the vehicles trying to exchange continuously real-time information. Second, even if they can, the amount of network traffic generated by vehicles will put a huge burden on the Internet. Third, communication over the Internet will lead to a lost of neighborhood geographical information (i.e., vehicles care mostly about traffic in their proximity). Thus, back-end servers will be needed to manage vehicular data, and consequently, these servers will become a bottleneck.

To solve the scale issue, we propose that the vehicles will form mobile ad hoc networks and exchange traffic information directly through short-range wireless interfaces. The major problem with this approach is that network partitions can appear quite often. Therefore, we can use them for applications that require only best-effort semantics. An optimization for such applications is to avoid dropping the data when no connection is available. The data can be stored at intermediate nodes and forwarded as soon as the network partitions are re-connected.

The question that needs to be answered is what to do when applications need reliable delivery. We believe that only a few applications will need reliable communication in vehicular networks. Hence, broken communication paths for such applications will be rare events. In such cases, we can use the cellular/satellite interface to route data around areas of disconnectivity. A scalable protocol needs to be defined for the communication hand-off between ad hoc network and the cellular/satellite connection to the Internet. Furthermore, the network has to take care of the re-transmission of packets dropped due to congestion.

### 3.1.2 Location-based Communication

The characteristics of the vehicular applications render the traditional end-to-end communication paradigm unusable. The main reason is that in the common case we do not need data transfers between two nodes across multiple hops. In vehicular networks, the common case is the dissemination of traffic information between certain geographical regions. For instance, vehicles can disseminate traffic data to vehicles 20 miles behind on a highway. In a city, vehicles can disseminate traffic information not only on the street they run, but also on the streets that intersect the current street. For such applications, the individual identities of the vehicles do not matter. The important information is the location of vehicles, their speeds, and their relative positions with respect to each other.

Therefore, we can characterize the communication in vehicular networks either as region-based multicast (i.e., disseminate certain data to all the vehicles in a given region) or region-based anycast which is mostly used for queries (i.e., query any car from a given region). End-to-end communication can still exist, but this will be an uncommon case. This location-based communication can greatly benefit form the scalability and robustness against frequent topological changes of geographical routing algorithms [27, 28, 38, 16]. With these algorithms, there is no need to maintain routes from the sender to the destination because the forwarding decisions are based only on local knowledge.

### 3.1.3 Active Nodes

In the Internet and even in "traditional" mobile ad hoc networks, the communication takes place end-to-end, and the intermediate nodes just route packets. The routers are passive entities, in the sense that they do not look into the data portion of a packet. Vehicular networks, however, have a unique characteristic which leads to the idea of turning every node into an active node that processes the data before forwarding it. These networks continuously disseminate real-time traffic information. Therefore, we will face severe bandwidth limitations if nodes in the network do not perform any type of optimization. We believe that, similar to sensor networks, vehicular networks should perform in-network processing. In order to send as much information as possible within the available bandwidth, these networks can reduce the data either through compression or through aggregation. Furthermore, caching and suppression techniques can also be involved to reduce the traffic in the network.

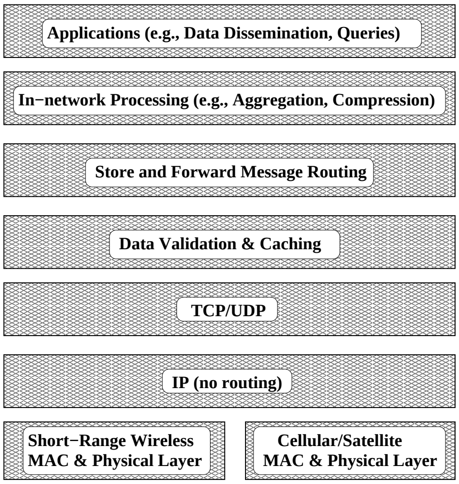

Figure 5: Vehicular Network Protocol Stack

### 3.1.4 Security and Privacy Protocols in the Protocol Stack

Unlike the Internet, where security was left to the application layer (with the consequences seen today), a network architecture for vehicular networks has the chance to incorporate security protocols in the network stack from the beginning. Security and trust are two crucial issues in such networks. A fraudulent vehicle could disseminate information about non-existent vehicles or broadcast bogus information about existing vehicles. It is necessry to investigate mechanisms that identify the fraudulent vehicles and prevent their data from being further propagated in the network.

Trusted computing is a proposal to make the commerce over the Internet less cumbersome by increasing the security of networked personal computers. At the heart of this approach is the idea of *attestation*. Attestation authenticates who built the platform hardware and what software was started at each layer of the software stack, from the firmware up to the virtual machine executing the application. Attestation requires building a certificate chain, from the tamper resistant hardware all the way to a virtual machine, to identify each component of the software stack. By ensuring that every software component running on the machine is trusted, the safety of the actions taken by that machine is ensured. This is a promising approach to build networks of trustworthy systems.

Privacy is another important issue in such networks. Different privacy levels should be available from which the drivers can select. One level of privacy could be to completely hide any information about the vehicle while it continues to participate in relaying other vehicles' information. Another level is to allow others to gain information about the vehicle without being able to identify it. For example, vehicles on the road may know about a group of vehicles, without being able to identify the exact location of any group member.

## 3.2 Protocol Stack

Based on the architectural principles discussed above, we have identified several services that have to be provided by a vehicular network architecture. These services can be either provided by application-specific implementations or by a common protocol stack. As demonstrated by the success of the Internet, a clearly defined protocol stack that incorporates the necessary network services can speed up significantly the mass deployment and acceptance of network technologies. However, defining the service provided at each network layer and the protocol interfaces between layers needs a thorough investigation. Our goal is to explore the design and implementation of a common protocol stack for vehicular networks. The starting point in achieving this goal is the layered architecture presented in Figure 5.

### 3.2.1 Inherited TCP/IP Support

By default the TCP/IP stack will be present in the vehicular systems because they connect not only to ad hoc networks, but also to the Internet when needed. Clearly, these systems do not need to run IP routing algorithms because they are just regular edge nodes in the Internet. At this time, we have decided to use the TCP/IP support for the ad hoc network communication as well. However, we will use TCP or UDP only for one-hop communication, while the routing will be performed at a higher layer. Although the ad hoc network performance can suffer a degradation compared to a potential implementation from scratch on top of the MAC layer, this solution has significant practical benefits. First, it is much easier to move forth and back between ad hoc and cellular/satellite communication if a common layer can take this decision. Second, implementing real-life prototypes will be simplified if we take advantage of the services already provided by TCP or UDP.

### 3.2.2 Data Validation and Caching

The first operation that needs to be performed once a message is received from the TCP/UDP layer (note that TCP/UDP are used only for one-hop communication) is data validation. No real-life vehicular networks will be deployed unless the data can be validated. Once data is validated, it can be cached locally for further uses.

Validation is necessary for two distinct reasons. One reason is security, more exactly preventing vehicles from sending bogus traffic information. For instance, a malicious driver can disseminate ”traffic jam” messages in order to keep the road empty for himself. Therefore, this layer should identify and discard such information. The other reason for validation refers to outdated or unnecessary data. For example, this layer can remove the data disseminated by vehicles located behind the current vehicle on a highway. Another example of a validity check is when there are multiple records containing information about the same vehicle. In this case, this layer decides to cache the most recent record and remove the older versions.

Both new and cached data must be examined to verify that they reflect the current state of the road and eliminated if they are outdated. For example, data about vehicles that have left the road have to be purged from the cache. Moreover, newly received messages might contain inaccurate information due to frequent changes in the speed of the corresponding vehicles and/or aggregation mechanisms applied to the data within relaying nodes. There are two main problems here: how should the value of the information in a message be assessed, and how can a balance between knowing inaccurate information about a vehicle, and having no knowledge about it, be achieved.

To solve this problem, we can exploit two aging mechanisms. The first mechanism associates a timer with each data item added in the cache. This timer is reset each time the data is updated. If the timer is expired, the data is dropped. The second mechanism deals with newly received data. Whenever a new message is received, the expected latency in receiving the message is calculated and compared to the actual latency. If the difference between these two is lower than a threshold, the data is stored; otherwise, it is considered outdated and ignored.

### 3.2.3 Store and Forward Message Routing

This layer performs routing in the vehicular network. Note that this is the layer that decides whether to send data over the ad hoc network or to send it over the cellular/satellite link. Section 3.2.6 describes our solution for using the Internet as a proxy for communication between vehicles when nodes cannot forward a message over ad hoc networks (i.e., network partitions can lead to such situations). Topically, dissemination messages are broadcast over the UDP in the ad hoc network to improve the performance. To deal with frequent disconnections encountered in ad hoc networks, this layer can help delay-tolerant applications by postponing the transmission of a message until connectivity is available.

Since virtually all vehicular information is location-dependent, geographical routing is used to deliver data over multiple hops. This routing is also a scalable solution for mobile ad hoc networks. Essentially, a message will contain the source location and the destination location (which can be expressed as any vehicle in a given geographical region). A result to a query can be sent to the expected geographical region of the source (which can be computed based on the original position, speed, and time). To deliver the message to

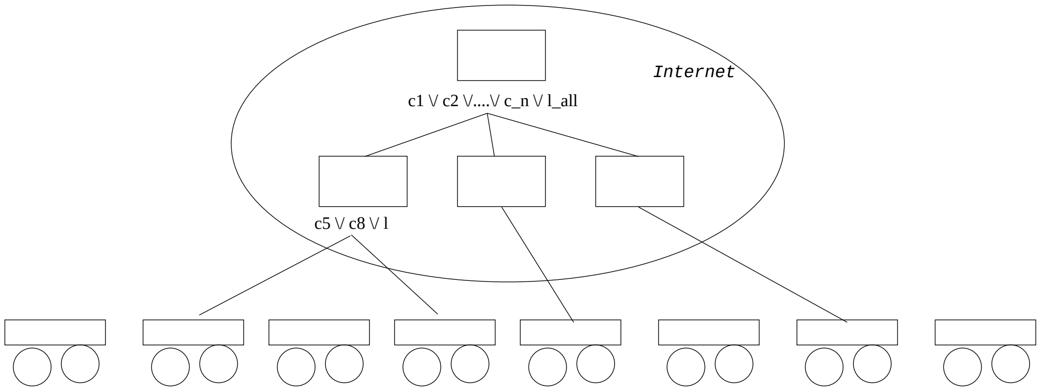

Figure 6: Hybrid Publish/Subscribe Framework for Vehicular Networks

a vehicle in a given region, the routing layer implements a region-bound flooding. This protocol can also be used to distribute a message to all the vehicles in that region (multicast). In this project, we plan to investigate if more complex routing algorithms that maintain routing tables at nodes can improve upon the performance provided by geographical routing and region-bound flooding. Additionally, we plan to design algorithms that take advantage of the predictive mobility patterns encountered in vehicular networks (i.e., bi-directional mobility on well defined paths). We also plan to study if this is the right layer to provide priority for critical messages such as accident warnings.

### 3.2.4 In-network Processing

To reduce the bandwidth consumption in vehicular networks, data compression and data aggregation can be used. Data compression is binary compression in the sense that it does not base its decisions on the semantics of the data. Moreover, data compression techniques require a lot of computation resources which might not be suitable for the systems embedded in the cars.

Data aggregation is based on the semantics of the data. For example, the records from two vehicles can be replaced by a single record with little error, if the vehicles are very close to each other and they are moving with relatively the same speed; in other words, the distance between them is always in a small range. The way data aggregation contributes to the efficient bandwidth consumption is by delivering as many records as possible in one broadcast message. Hence, more new records can be delivered in a certain period of time and the overall network performance is improved.

### 3.2.5 Application Layer

At this layer, we need to provide efficient data dissemination algorithms, a property-based naming that allows applications to refer to the desired locations and properties rather than individual vehicles, and support for distributed query processing in the network. Encryption can also be provided for applications that need to secure their data transfers.

At the application layer we implement a hybrid publish-subscribe framework to take care of disconnected operations. The publish-subscribe framework will serve to provide reliable information exchange for high priority information and traffic queries.

### 3.2.6 Publish-Subscribe Framework for Reliable Communication

Publish/Subscribe frameworks hold promise for mobile networks as pointed out in [21]. A typical publish/subscribe framework is composed of three kinds of entities: publishers, subscribers and brokers. Subscribers subsribe to content hosted by brokers, by specifying the topic or content of interest. The content is generated by publishers and uploaded on the brokers. Brokers provide storage and management of subscriptions and messages. The brokers are usually heirarchically organized in a tree-like structure with the subscribers as leaf nodes. The content flows from the root of the tree towards the leaves and gets filtered on the way by predicates that specify the subscription of the users. Typically, publish/subscribe frameworks are used for fixed networks or web services.

The publish/subscribe paradigm is different from the traditional point-to-point communication in a number of ways. In sublish/subscribe systems the communication between end points is anonymous, asynchronous and loosely coupled. In other words, publish/subscribe systems decouple publishers and subscribers in space, time and flow. This makes publish/subscribe frameworks highly scalable by removing all explicit dependencies between the interacting parties. It also helps the system to adapt quickly to the dynamic environment. Decoupling in space allows the subscriber to move from one location to another without informing the publisher while decoupling in time allows for disconnected operations of the subscriber.

We plan to extend publish/subscribe frameworks to vehicular networks which are characterized as dense, highly mobile and weakly connected networks. Assuming that GPRS/3G would serve to provide ubiquitous internet connectivity, a naive approach is to have all the brokers reside on the web and have all the vehicles subscribe with these web brokers. The sheer scale of vehicular networks would limit the scalability of such an approach. Another possible approach is to have all the brokers reside in the ad hoc network, where a broker would be the elected leader of a group of vehicles. Although scalable, this approach is limited by the mobility of vehicles. Frequent hand-overs would limit the efficiency and reliability of such an approach. With this in perspective, we propose a hybrid publish/subscribe framework with the following key properties: (1). Brokers exist in the ad hoc network as well as the internet (2). Vehicle subscriptions are grouped not only based on content but also location of vehicles. Taking a hybrid approach serves to minimize usage of limited GPRS/3G bandwidth providing reliability at the same time. Location based group subscriptions serve to provide a mechanism for handling dense networks.

For a hybrid publish/subscribe framework, the main issues to be dealt with are : broker election, broker discovery, subscribtion initiation, broker hand-off, broker inter-communication and location based subscription aggregation.

**Broker election:** In Figure 6, the brokers at the lowest level represent brokers in the ad hoc network. The square boxes represent brokers on the internet. Network topology on highways or cities can be characterized as a forest of weakly connected graphs (in a connected graph there is a path between every node), where nodes represent vehicles. Two nodes have an edge between them if and only if they are within transmission range of each other. The edges appear and disappear as nodes move. Each connected graph has a broker which is elected by the nodes of that graph. Election is initiated when a subscription is requested. Brokers are elected based on three main criteria : (1). reachability (2). GPRS/3G availability (3). proximity to the neighbouring connected graph.

**Broker discovery and subscription initiation:** When a node wishes to subscribe, it first tries to discover a broker in that region. If a broker is discovered the subscription is registered with that broker. If not, broker election is initiated. Broker discovery is initiated by broadcasting broker-discovery messages. If a broker exists, it replies to the broker-discovery message. Broker election is initiated by broadcasting election-initiation messages. Upon recieving the message, the nodes of the clique start a broker election protocol based on the aforementioned criteria.

**Broker hand-off and intercommunication:** Consider the following subscription request from car ABC 123: *Location of car BHV 57K* (where cars ABC and BHV are friends travelling together). Since ABC and BHV are travelling together, they would never get too distant from each other. Such a subscription request can be handled by brokers in the ad hoc network *primarily.* As the cars move, the subscription is handed off from one broker to the next. If the broker finds the subscriber unreachable at a certain point, it could either forward the subscription to a web broker or hand it off to another broker in the ad hoc network. It would then maintain a forward link to the new broker for the purpose of forwarding publisher's data or inform the publisher of the new broker. For such subscription requests, ad hoc network brokers are primary brokers while web brokers only serve to provide added reliability.

**Location based subscription aggregation :** Consider the following subscription request by car ABC 123 : *Traffic on 34th Street* (where car ABC is 10 miles away from 34th street). Such a subscription request is best handled by web brokers. The subscription is done directly with a web broker and the data is published directly to the web broker. For such subscription requests, ad hoc network brokers would serve as data distributers. Web brokers aggregate subscriptions based on location. Hence subscription requests from all

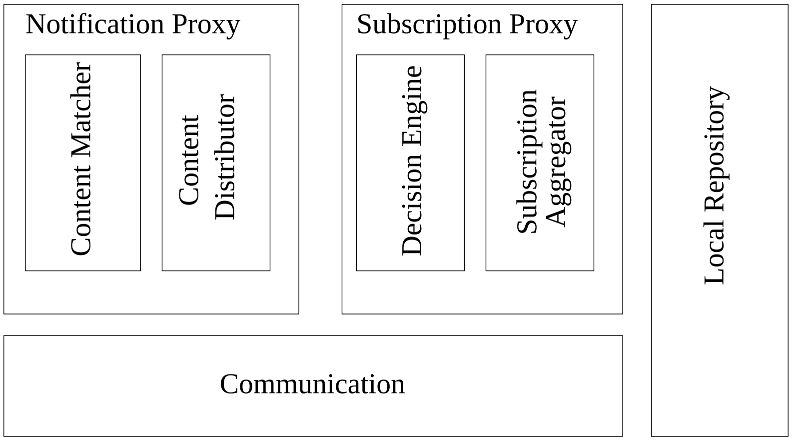

Figure 7: Publish/Subscribe Architecture

cars around milestone 23 on highway US-1 would be bundled together. The content corresponding to such subscriptions will not be pushed independently to every subscriber but to the broker/leader of the cars around that location. Hence, content is filtered down the tree not only based on subscription predicates but also location. The brokers would then broadcast the content to the nearby cars.

Let $c_i$ denote a subscription predicate (e.g *Traffic on 34th Street*). Let $c_{all} = c_1 \vee c_2 \vee c_3 \vee .... \vee c_n$. In a typical publish/subscribe framework, root of the forwarding tree filters content using $c_{all}$. An intermediate broker filters content using a subexpression of $c_{all}$, for example $c = c_5 \vee c_8$. In our publish/subscribe framework, every subscription predicate $c_i$ is appended with the subscribers location $l_i$. Hence $c_{all} = c_1 \vee c_2 \vee .... \vee c_n \vee l_{all}$ where $l_i \subset l_{all}$, and $c = c_5 \vee c_8 \vee l$ where $l_5 \subset l$, $l_8 \subset l$, $l \subset l_{all}$.

The decision of whether the subscription should be done with an ad hoc broker or with an internet broker is taken by the network stack on the user's car pc in a manner oblivious to the user. The decision is taken based on the space and time discontinuity between the subscriber and the publisher. If the publisher and subscriber are not far away from each other then ad hoc network brokers are preferred, otherwise web brokers are preferred. Similarly, if the subscription request is active for a short period of time (e.g *Location of car BHV 57K for the next 5 minutes*) then ad hoc network brokers are preferred otherwise web brokers are preferred. We would use ontologies (OIL + DAML) or XML for specifying subsciption predicates, in order to make them machine readable.

The network stack also takes the decision of when to switch between 802.11 and GPRS. This decision is based on the availability of neighbours around the vehicle. If the vehicle is not able to forward data because it is disconnected from the ad hoc network, then the data is transferred over GPRS.

Figure 7 gives a higher level view of the key components of our publish/subscribe architecture. The architecture sits on top of our network stack in the application layer on every vehicle, since every vehicle is a potential broker. The same architecture sits on the network stack of web brokers. *Local Repository* stores subscription predicates and content. *Content Matcher* and *Content Distributor* are part of the *Notification Proxy* which is responsible for the outflow of content. Content Matcher is responsible for matching content against subscription predicates and passing it onto Content Distributor which forwards the content to other nodes. *Decision Engine* and *Subscription Aggregator* are part of the *Subscription Proxy* which is responsible for handling incoming subscriptions. Decision Engine decides which broker to subscribe the request with and also interacts with the network stack for broker hand-offs. Subscription Aggregator aggregates subscriptions based on location.

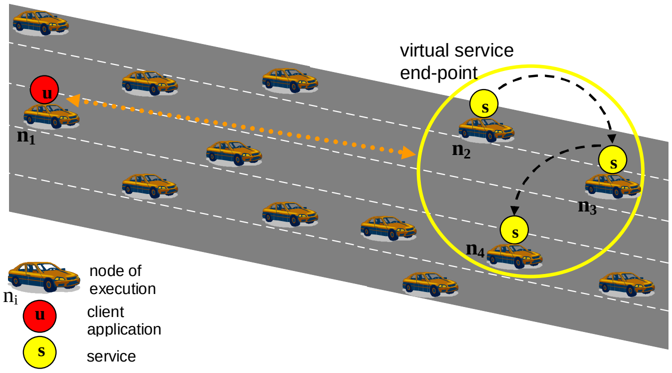

Figure 8: Example of Traffic Query as a Mobile Service

### 3.3 Mobile Traffic Queries

Executing queries in a vehicular network implies distributed query processing at multiple nodes. For instance, a vehicle may want to know, periodically, the average traffic speed 10 miles ahead of it on the highway. To answer such a query, we can contact periodically several vehicles located 10 miles ahead and compute the average speed at the local vehicle. This solution, however, is not efficient for two reasons: it generates too much traffic in the network, and the volatility of the network may lead to lost messages or multiple re-transmissions.

A more efficient solution would be to send the query as an autonomous task in the region of interest. This task can compute the average speed locally and send the answers back to the source. To achieve such a functionality, we propose mobile queries (similar to mobile agents), a novel model of query execution in ad hoc networks. A mobile query is an autonomous task, capable of migrating to different nodes in the network in order to effectively accomplish its function. This migration occurs transparently to the client and, except for a certain delay, no service interruption is perceived by the client. Although the destination end-point of a mobile query (i.e., the node where the query executes) is physically located on different nodes over time, the model presents a unique virtual end-point to the client.

Figure 8 illustrates an example of interaction between a client and a monitoring mobile query that periodically reports the average speed to the client. The query changes the node of execution by migrating from node *n2* to *n3*, and subsequently from *n3* to *n4*. During the entire period, the client application is not aware of the query migration on different nodes (i.e., it sees a virtual end-point) and the service provisioning is not affected by any interruption. Although not shown in the figure, we assume that any message between two nodes travels across multiple intermediate nodes. We plan to prototype such migratory traffic queries using Smart Messages [10, 26], a distributed computing platform based on execution migration that we developed for programming highly volatile ad hoc networks.

## 4 Architecture Validation

In order to validate the proposed protocol stack, we are building a prototype system that will be tested under real-life traffic conditions.

### 4.1 Prototyping

The goal is to design and implement a flexible and efficient vehicular computing platform, bottom-up starting from a car PC (that is being assembled in our lab), up to the application layer. This will help us understand better which are the necessary layers in the protocol stack and what type of services they need to implement.

The target embedded system prototype will have enough processing power and memory to support the required applications, advanced visualization capabilities (e.g., OpenGL), small form factor, and it should be easy to use (no keyboard, just touch screen and voice commands). It is important to be able to demonstrate a product to car manufacturers in order to convince them to install such embedded systems in the next generation of vehicles. Car manufacturers would be reluctant to put proprietary systems in production automobiles and hence is it important that off-the-shelf components be used as much as possible. This prototype will also allow us to collect real traffic traces, which can be used for larger scale emulations in the lab. The initial configuration that we plan to use contains: EPIA M10000 1GHz motherboard, Xenarc 700TSV 7" TFT LCD Display with VGA and Touchscreen, 256MB PC2100/DDR266 RAM, 512MB Flash Disk, IEEE 802.11 PC card, GPRS modem card, and GPS receiver. We expect that the final configuration will be different based upon the results of testing the initial prototype.

Dedicated Short-Range Communications (DSRC) is 75 MHz of spectrum at 5.9 GHz allocated by Federal Communications Commissions (FCC) for inter-vehicular networking, using a variant of 802.11a technology. The 2004 FCC ruling specifies that DSRC will have six service channels and one control channel. The control channel is to be regularly monitored by all vehicles. "Safety of life" messages have the highest priority whether originated by vehicles or road-side transmitters. In our prototype, IEEE 802.11 is just a temporary solution; we plan to use DSRC when it becomes available. We also plan to investigate the feasibility of sending safety messages from vehicle to vehicle in the DSRC control channel.

In developing this prototype, we will benefit from the experience gained with developing a prototype for TrafficView [32]. This prototype was developed over HP iPAQ PDAs running Linux and was tested over mobile ad hoc networks based on IEEE 802.11 communication. Additionally, our experience with developing Smart Messages [10, 26], a distributed computing platform for ad hoc networks of embedded systems, will help us with the prototype software. Furthermore, the Smart Messages platform can be used at the application layer to provide support for more complex applications over vehicular networks.

## 4.2 Modeling and Simulation

We plan to evaluate our architecture and the individual protocols through large scale simulations. The major problem with simulating vehicular traffic is that no realistic vehicle traffic generators exist (CORSIM [4] is a microscopic traffic simulator developed by The Federal Highway Administration, but it is not publicly available). It is clear that we need better mobility models than the traditional random way-point model typically used in mobile ad hoc network research. We have developed a scenario generator tool that we intend to make to public for the research community. The generator tool is used to generate traffic scenarios in different formats suitable for different simulators, for example, ns-2 simulator [8].

We are developing a version of the traffic generator to generate highway traffic scenarios (an initial version has been used in [32]). Such scenario generator accepts as parameters simulation time, length of the road, average speed of the nodes, number of lanes on the road, and the average gap length between vehicles. It uses a simplified traffic model as follows:

- *Entries and Exits:* The entries and exits are evenly distributed along the road. Vehicles may enter the road at each entry except the last one and leave at any consecutive exit. Vehicles enter the road at the front-end entry with a certain probability, and at side entries with another probability.
- *Speed Changes:* To model the changes to the speed of a node, the road between the entry point and exit point of a node is divided into regions of 50 meters, and a constant speed of $\text{max speed} \times (0.75 + \text{rand}(-2, 2) \times 0.125)$ is used for each region, where $\text{rand}(a, b)$ returns a uniformly distributed random integer between $a$ and $b$.
- *Changing Lanes:* Vehicles can change their lanes with no dependence on other vehicles. We can vary the probability of staying on the same lane and the probabilities of changing to the right or left lanes..
- *Vehicle Density:* The density of vehicles is an important factor because it determines the number of neighboring nodes in the transmission range of a vehicle, which has a great impact on the transmission delay and available bandwidth of the network. The scenario generator initially puts

$$\frac{\text{road-length} \times \text{number of lanes}}{\text{average gap}}$$

Figure 9: Sample histograms of average speed (left) and average number of lane changes per minute (right) in a scenario generated by the scenario generator tool

> *active* nodes, evenly distributed, on the road. Once a vehicle leaves the road at one of the exits, it is deactivated, and a new node is added (activated) to the road randomly. As soon as a node is deactivated, it will no longer affect our metric calculations introduced in the next section.

Figure 9 shows the histogram of the average speed and number of lane changes per minute for a scenario generate with average speed = 30m, and average gap = 100. The graphs show the percentage of vehicles that have that average speed and average number of lane changes per minute, respectively.

We are also extending the scenario generator to generate traffic scenarios for cities with grid roads such as Manhattan. In such generator, we use an even number of alternated single direction roads along each dimension of the grid. Vehicles can switch their roads at intersections. With equal probabilities, a vehicle chooses either to stay on the same road or change to the road it intersects with. When a vehicle reaches the last intersection on a road, it is forced to change to the road it intersects with. Initially, each vehicle moves toward a random destination along its road using its currently selected speed. Once a vehicle reaches its destination, it selects another random destination along its road as well as a new speed. If the selected destination is behind the next intersection on the road, the vehicle sets its destination to the intersection.

The traffic parameter values need more investigation to match the real-life traffic patterns. For example, the values used for the average gap length between vehicles must take into consideration highway vs. city traffics, rush-hour vs. regular traffics, etc. We plan to investigate such parameter values based on statistics from real traffic patterns.

## 5 Summary

We identified the different kinds of information that can be exchanged in these vehicular car-to-car networks, namely real-time, on-demand and high-priority. We identified location as a key enabler of information exchange. We presented the design of a network protocol stack specific to these networks. Some of the key functionalities that this network stack provides are: data aggregation, data validation, store and forward routing, and utilization of multiple network interfaces (e.g WLAN and GPRS/3G). We discussed the implementation of a prototype currently underway.

## References

[1] General Motors Collaborative Laboratory: http://sigmobile.org/mobicom/2004/demos.html.

[2] http://www.citycab.com.sg/services/nts/sms_booking.html.

[3] http://www.comfort-transportation.com.sg/booking_svcs.html.

[4] CORSIM User Manual, Ver. 1.01, The Federal Highway Administration, US Dept. of Transportation, 1996.

[5] Inter-Vehicle Communication Project. http://ivc.epfl.ch/.

[6] Network-on-Wheels Project. http://www.informatik.uni-mannheim.de/pi4/projects/NoW/publications.html.

[7] On-Board Diagnostic System. http://www.obdii.com/.

[8] The Network Simulator ns-2,
http://www.isi.edu/nsnam/ns/.

[9] US Census Bureau, Topologically Integrated Geographic Encoding and Referencing system (Tiger Line).
http://www.census.gov/geo/www/tiger.

[10] Borcea, C., Iyer, D., Kang, P., Saxena, A., and Iftode, L. Cooperative Computing for Distributed Embedded Systems. In *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS 2002)* (Vienna, Austria, July 2002), pp. 227–236.

[11] C. Schwingenschloegl and T. Kosch. Geocast Enhancements for AODV in Vehicular Networks. *ACM Mobile Computing and Communications Review 6*, 3 (July 2002), 96–97.

[12] Chisalita, I., and Shahmehri, N. A Peer-to-Peer Approach to Vehicular Communication for the Support of Traffic Safety Applications. In *Proceedings of the 5th IEEE International Conference on Intelligent Transportation Systems* (Singapore, Sept 2002).

[13] Dashtinezhad, S., Nadeem, T., Dorohonceanu, B., Borcea, C., Kang, P., and Iftode, L. TrafficView: A Driver Assistant Device for Traffic Monitoring based on Car-to-Car Communication. In *Proceedings of the 59th IEEE Semiannual Vehicular Technology Conference* (May 2004).

[14] Gehring, O., and Fritz, H. Practical results of a longitudinal control concept for truck platooning with vehicle to vehicle communication.

[15] Golle, P., Greene, D., and Staddon, J. Detecting and correcting malicious data in vanets. In *Proceedings of the 1st ACM Workshop on Vehicular Ad Hoc Networks* (2004).

[16] Hartenstein, H., Bochow, B., Ebner, A., Lott, M., Radimirsch, M., and Vollmer, D. Position-aware ad hoc wireless networks for inter-vehicle communications:the Fleetnet project. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing* (Long Beach, CA, 2001), pp. 259–262.

[17] Hedrick, J., Tomizuka, M., and Varaiya, P. Control issues in automated highway systems, 1994.

[18] Heinzelman, W. R., Kulik, J., and Balakrishnan, H. Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. In *Proceedings of the Fifth annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 1999)* (Seattle, WA, August 1999), ACM Press, New York, NY, pp. 174–185.

[19] Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., and Pister, K. System Architecture Directions for Networked Sensors. In *Proceedings of the Ninth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX)* (Cambridge, MA, November 2000), ACM Press, New York, NY, pp. 93–104.

[20] Huang, Q., and Miller, R. The design of reliable protocols for wireless traffic signal systems. Tech. rep., Department of Computer Science and Engineering, Washington University, 2003.

[21] Huang, Y., and Garcia-Molina, H. Publish/subscribe in a mobile enviroment. In *MobiDe '01: Proceedings of the 2nd ACM international workshop on Data engineering for wireless and mobile access* (2001), pp. 27–34.

[22] Intanagonwiwat, C., Govindan, R., and Estrin, D. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *Proceedings of the Sixth annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000)* (Boston, MA, August 2000), ACM Press, New York, NY, pp. 56–67.

[23] J. Jetcheva and Y. Hu and S. PalChaudhuri and A. K. Saha and D. B. Johnson. Design and Evaluation of a Metropolitan Area Multitier Wireless Ad Hoc Network Architecture. In *Proceedings of the 5th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2003)* (October 2003), pp. 32–43.

[24] Johnson, D., and Maltz, D. *Dynamic Source Routing in Ad Hoc Wireless Networks.* T. Imielinski and H. Korth, (Eds.). Kluwer Academic Publishers, 1996.

[25] Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L., and Rubenstein, D. Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet. In *Proceedings of the Tenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X)* (San Jose, CA, October 2002), ACM Press, New York, NY, pp. 96–107.

[26] Kang, P., Borcea, C., Xu, G., Saxena, A., Kremer, U., and Iftode, L. Smart Messages: A Distributed Computing Platform for Networks of Embedded Systems. *The Computer Journal, Special Focus-Mobile and Pervasive Computing 47* (2004), 475–494. The British Computer Society. Oxford University Press.

[27] Karp, B., and Kung, H. Greedy Perimeter Stateless Routing for Wireless Networks. In *Proceedings of the Sixth annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000)* (Boston, MA, August 2000), ACM Press, New York, NY, pp. 243–254.

[28] Ko, Y.-B., and Vaidya, N. H. Location-Aided Routing(LAR) in Mobile Ad Hoc Networks. In *Proceedings of the Fourth annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)* (October 1998), pp. 66–75.

[29] Li, J., Jannotti, J., Couto, D. S. J. D., Karger, D. R., and Morris, R. A Scalable Location Service for Geographic Ad Hoc Routing. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking* (Boston, MA, Aug 2000), pp. 120–130.

[30] Miller, R., and Huang, Q. An Adaptive Peer-to-Peer Collision Warning System. In *IEEE Vehicular Technology Conference (VTC)* (Birmingham, AL, May 2002).

[31] Morris, R., Jannotti, J., Kaashoek, F., Li, J., and Decouto, D. CarNet: A Scalable Ad Hoc Wireless Network System. In *Proceedings of the 9th ACM SIGOPS European Workshop* (Kolding, Denmark, September 2000), ACM Press, New York, NY, pp. 61–65.

[32] Nadeem, T., Dashtinezhad, S., Liao, C., and Iftode, L. TrafficView: A Scalable Traffic Monitoring System. In *Proceedings of the 2004 IEEE International Conference on Mobile Data Management (MDM 2004)* (Berkeley, GA, January 2004).

[33] Newsome, J., Shi, E., Song, D., and Perrig, A. The sybil attack in sensor networks: Analysis and defenses. In *Proceedings of the Third International Symposium on Information Processing in Sensor Networks* (2004).

[34] Ouksel, A., Wolfson, O., and Xu, B. Opportunistic Resource Exchange in Inter-vehicle Ad Hoc Networks. In *Proceedings of the 2004 IEEE International Conference on Mobile Data Management (MDM 2004)* (Berkeley, GA, January 2004).

[35] Peng Zhou and Tamer Nadeem and Porlin Kang and Cristian Borcea and Liviu Iftode. EZCab: A Cab Booking Application Using Short-Range. In *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom 2005). To Appear.* (March 2005).

[36] Perkins, C., and Royer, E. Ad Hoc On Demand Distance Vector Routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 1999)* (New Orleans, LA, February 1999), pp. 90–100.

[37] Reichardt, D., Miglietta, M., Moretti, L., Morsink, P., and Schulz, W. Cartalk 2000 - safe and comfortable driving based upon inter-vehicle communication, 2002.

[38] Tian, J., Han, L., Rothermel, K., and Cseh, C. Spatially Aware Packet Routing for Mobile Ad Hoc Inter-Vehicle Radio Networks. In *IEEE 6th International Conference on Intelligent Transportation Systems (ITSC)* (Shanghai, China, October 2003).

[39] Tsugawa, S., Tokuda, K., Kato, S., Matsui, T., and Fujii, H. An overview on demo 2000 cooperative driving. In *Proceedings of the IEEE Intelligent Vehicle Symposium* (2001).

[40] X. Yang and J. Liu and F. Zhao and N. Vaidya. A Vehicle-to-Vehicle Communication Protocol for Cooperative Collision Warning. In *Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous 2004)* (August 2004).

[41] Zong Da Chen and HT Kung and Dario Vlah. Ad hoc Relay Wireless Networks over Moving Vehicles on Highways, October 2001.