VGrid: Vehicular AdHoc Networking and Computing Grid for Intelligent Traffic Control

Joey Anda[1], Jason LeBrun[2], Dipak Ghosal[1], Chen-Nee Chuah[2], Michael Zhang[3]
University of California at Davis
Davis, California USA
[1]{anda,ghosal}@cs.ucdavis.edu, [2]{jblebrun,chuah}@ece.ucdavis.edu, [3]hmzhang@ucdavis.edu

Abstract— **We propose VGrid: an ad hoc networking and computing grid formed by leveraging inter-vehicle and vehicle-to-roadside wireless communications. In addition to exchanging data between vehicles, VGrid actively uses pertinent data to perform computations for solving traffic-related problems. The goal is to evolve intelligent transportation engineering from a centralized to a distributed approach, in which vehicle equipped with wireless networking and computers can cooperate and solve vehicular traffic-flow control problems autonomously. We present an example application: the merging of two lanes into one. We explore various algorithms to compute the optimal schedule for vehicle arrivals at the merge point using velocity/position information exchanged between vehicles. Our simulation results, using realistic vehicle mobility patterns, show that the proposed VGrid framework and algorithms can increase the system throughput and decrease the latency through the merge point.**

Keywords-Vehicular traffic flow; grid computing; ad-hoc networking.

I. INTRODUCTION

The FCC has recently allocated the 5.85-5.925GHz portion of the spectrum for inter-vehicle communications (IVC) and vehicle-to-roadside communications (VRC), known as Dedicated Short Range Communications (DSRC). This has fueled significant interest in designing new applications, including driver-vehicle safety applications, infotainment, and mobile internet services for passengers [1, 3]. However, there is a huge untapped opportunity to leverage vehicular ad hoc networks (VANET) to revolutionize the intelligent transportation systems (ITS). Unlike other mobile ad hoc networks (MANETs) that consist of power- and computing limited nodes, such as wireless sensor motes or hand-held devices, VANET has notably different design characteristics. The vehicles have ample power/energy and can be equipped with computing resources (e.g., processor and storage space). On the other hand, their high mobility results in very dynamic channel conditions.

In this work, we propose a new paradigm called VGrid (Vehicular-based Networking and Computing grid), where we leverage DSRC-enabled vehicles to perform *data sensing, relaying, and computing* to support *distributed monitoring and control of vehicular traffic flow*. Many kinds of road-side infrastructures (e.g., fixed sensors) for monitoring highway conditions have been in place. They are used to support obstacle detection/avoidance, speed monitoring, or meter-light control. For example, the existing ITS collect traffic statistics from roadside sensors and send it back to a central location, where computers and/or humans analyze the data to decide the optimal traffic-light schedules or to plan construction and detour routes. A distributed traffic management architecture can drastically reduce this "feedback loop"-from a time period on the order of weeks or months, to something on the order of seconds or minutes.

In VGrid, vehicles play the role of both mobile sensors (collecting data) and mobile routers (relaying data), and are linked together to form a global grid computer. A high density of cars results in a higher density of potential nodes that can be temporarily organized on the fly to perform a distributed computation to solve a single problem. This networking/computing capability can enable vehicle-driver safety applications. For example, to ease the merging of traffic from two lanes to one, the affected vehicles can exchange velocity/position information and schedule the best arrival time at complex highway interchanges. The mobile computer grid can also monitor and control ramp metering. Other applications may include: 1) Analyzing traffic congestion on the fly; 2) Computing optimal detour routes for vehicles, based on their destinations; 3) Collaborative tracking of vehicles; 4) Providing a virtual front view under poor weather conditions; and 5) Locating accidents and alleviating congestion through traffic metering or early warning messages.

II. THE VGRID FRAMEWORK

We envision an architecture in which the results of computations of various nodes are shared with neighbors, thereby influencing those neighbors' computations. Applications that utilize VGrid can operate in a number of different scopes: ranging from a single car to a platoon of cars that form a peer space, as discussed in [2]. The *platoon scope* is important since many potential VGrid applications will require knowledge of data contained in a local area of interest. However, for some applications, we may need a larger scope that combines a collection of peer spaces to achieve wider-scale results. Finally, we may think in terms of wide-area networking, in cases where data is sent back to a central server, or Internet services are utilized.

First, we need to consider, at the high level, the types of applications that might be implemented in this system. They fall into the following categories: 1) Local grid-style computations (platoon scope); 2) Large-scale distributed problems (inter-platoon scope); 3) Grid-style computations on behalf of another organization (SETI@Home, on the road) (inter-platoon scope); 4) Inter-personal communications (platoon scope); and 5) Providing Internet access to vehicles (wide-area scope). In this paper, we focus our discussion on the first two classes of applications, which will most likely evolve into interesting and completely-distributed autonomous control systems.

III. VGRID ARCHITECTURE

The proposed architecture is a hybrid architecture that consists of the following functional elements

- **Fixed roadside sensors:** These are sensors that are deployed along the highways including loops, lasers, and video cameras. These sensors can used to obtain aggregate information such as flow and density as well as (using more advanced sensors) the velocity and type of the vehicles passing through. These sensors can communicate with a central coordination center and also among themselves and with the in-vehicle sensors.
- **In-vehicle sensors:** These are sensors that are in the vehicles. These sensors can provide information on the instantaneous speed and position of the vehicle. The position information is obtained using in-vehicle GPS capability. The sensors can also provide information on the type of vehicle and the instantaneous front and back headway. The in-vehicle sensors can communicate with other in-vehicle and roadside sensors that are within the transmission range. Finally, each in-vehicle sensor is also equipped with processing and storage capability.
- **Central Coordination Center:** The central coordination center can gather data from roadside and in-vehicle sensors and provide advisory speeds on the changeable message signs (CMS). It also contains historical databases of past traffic flow attributes.
- **Changeable Message Signs:** These are sparsely located roadside signs that display changeable messages.

Figure 1 shows all the components of the proposed functional architecture. The fixed infrastructure components (fixed sensors, central coordination center, and CMS) already exist. From an architectural perspective, our focus is on the in-vehicle sensors and their integration with the fixed infrastructure. The in-vehicle sensors can communicate with other in-vehicle sensors and form a vehicular ad hoc network. While previous research on VANETs focused on their communication aspects, in this study we extend the notion of the VANET to a vehicular ad hoc grid computer *referred to as VGrid: vehicular-based grid computing* since the participating vehicles in the traffic stream are equipped with local processor and memory. VGrid can be used to perform some computations on demand and disseminate the results to the participating vehicles and/or fixed sensors along the road.

Compared with fixed sensor network or computing grids, VGrid offers several advantages. Because of its mobility, VGrid can be deployed wherever and whenever it is needed in a road network. Because of its ad hoc nature, information can permeate along many directions. Because of its finite information propagation speed, information is disseminated in a staggered fashion, avoiding over-flooding or the so-called herding effect in broadcast based information systems. VGrid can analyze problems on the fly using its cmputing resources. Moreover, VGrid can join a fixed sensor network and/or a global grid easily.

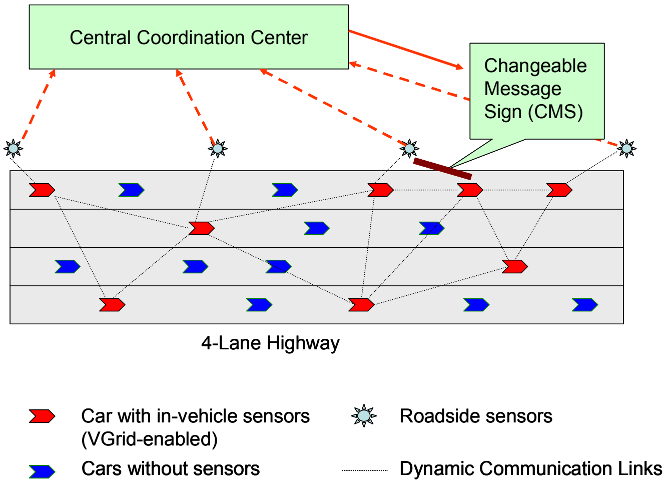

Figure 1 A framework architecture for VGrid illustrating the various components.

Grid computing involves the coordination of resources and problem solving in dynamic environments. Resources include computational power, data storage, and communication bandwidth. While grid computing itself is an established field, deploying it over the highly dynamic vehicular environment to solve transportation applications pose interesting new challenges. The VGrid protocol architecture will use the standard TCP/IP protocol suite consisting of the application layer, the transport layer, the network layer, and the physical layer. The application layer mainly concerns with how tasks are accomplished, given a well defined transparent interface to the grid-computing layer. The core of VGrid is the grid computing layer. It serves as a bridge between the applications, the network, and the available computing resources.

IV. APPLICATION : MERGING OF TWO LANES

We consider an application scenario where two lanes merge into a single lane as shown in Figure. Given certain traffic intensities in each lane, the goal is to determine the speed of each car such that average latency is minimized and the average throughput is maximized.

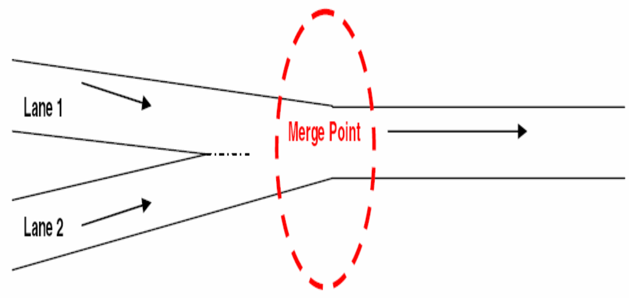

Figure 2 Two lane merging scenario.

Throughput is defined as the number of cars to successfully complete the merge in a given amount of time. Latency is defined as the time it takes for a vehicle to get past the merge point once it has entered the system. Unlike traditional scenarios, each vehicle knows the location and the velocity of all the other vehicles using VGrid. We have considered the following lane merging algorithms and compared them using our simulation tool.

1. *Weighted Fair Queuing (WFQ):* This algorithm is based on the WFQ algorithms proposed for queue management in IP routers [5, 6]. In this algorithm, each vehicle uses the information received from the surrounding vehicles to determine whether their current pace will result in a collision. If it is determined that a collision will occur, a dynamic weight is computed for each lane dependent upon their current congestion levels and priority is granted to the vehicle in the more congested lane.
2. *Platoon Scheduling (PS):* In this algorithm, the VGrid platform is used to coordinate the platooning of vehicles in one lane (referred to as the platoon lane) and the scheduling of vehicles in the other lane to fit between the groups at the merge point. When a vehicle enters the system in the platoon lane, it attempts to find vehicles in proximity to which it can form a platoon. Vehicles in the other lane proactively adjust their speed in order to fit between platoons and successfully merge.
3. *Take Turns:* This is a simple form of meter light scheduling. The vehicles merging are scheduled by following a strict alternation between the lanes. We consider this is a base case for comparison.
4. *No Restriction without Collisions (NRI):* The final algorithm is an ideal form of a no-restrictions algorithm (NRI), where vehicles are allowed to continue at their current rate into the merge point, with speed modifications occurring only when necessary to avoid collisions. The speed modifications are made such that the vehicle must only decelerate as much as is necessary so a collision is prevented.

V. SIMULATION TOOL

We have developed a java-based simulation tool to study the concept of VGrid, where the grid nodes are automobiles that can communicate through the wireless medium to share information and solve problems. The traffic part of the simulation tool is based on the Cellular Automaton Traffic Simulators applet developed by Kai Bolay [4]. The vehicular flow models developed in the original simulator are used to create a realistic merging environment where vehicles on two separate highways merge into one [14], as well as a free-flow environment of five lanes that allows lane changing. Because the original implementation simply implemented a vehicle as an integer to represent its speed, we needed to heavily modify the simulator to correspond to message-passing, position-aware, computationally intelligent nodes. We implemented a message passing architecture as a simplified approach where all vehicles within range are allowed to send and receive messages to each other without any interference/collisions from surrounding vehicles. Using the information received, vehicles can use the corresponding VGrid computational processing power and software to calculate their next moves.

We assume that the vehicles are equipped with a global positioning system (GPS), and are able to acquire and transmit their location information to surrounding vehicles. VGrid utilizes this location information, along with the current speed, lane number, and acceleration characteristics of the vehicles, to compute a schedule for merging vehicles that maximizes fairness and throughput, while minimizing latency (delay) and collisions.

VI. RESULTS AND DISCUSSIONS

We have simulated the four lane-merging algorithms previously discussed in Section 4 (NRI, TakeTurns, WFQ, Platoon). In the simulations, we examined a normal freeway driving scenario (i.e., probability slowdown of 0.5), as well as an Automatic Cruise Control (ACC) scenario (i.e., probability slowdown of 0, where vehicles always attempt to stay as close to the vehicle ahead of it as possible, without exceeding the maximum speed), since ACC can easily be achieved with the VGrid architecture.

The comparison of the four algorithms with respect to the latency and the throughput for a normal freeway driving scenario (i.e., probability slowdown = 0.5) are shown in Figure 3 and Figure 5, respectively, while the graphs for the ACC scenario (i.e., probability slowdown = 0) are shown in Figure 4 and Figure 6, respectively. Note that the throughput is calculated as the number of vehicles in 50,000 time steps that are able to enter and exit the system. We have chosen not to include colliding vehicles in computing the total throughout, with the collisions also having no adverse effects of the subsequent vehicles in the simulation. The latency is calculated as the average number of time steps from which a vehicle enters the system to which it passes the merge point.

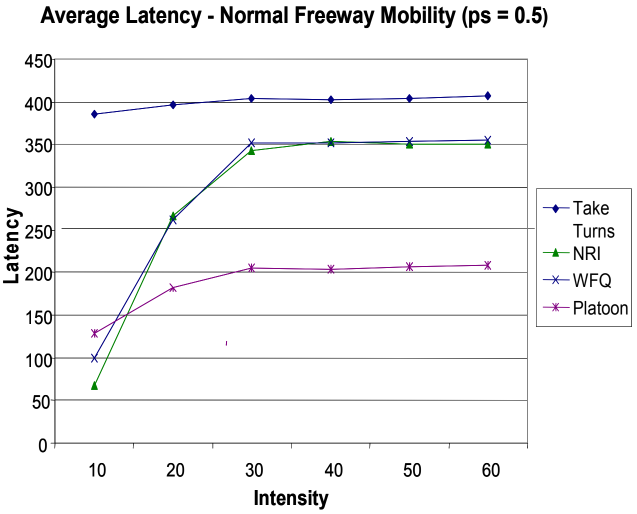

Figure 3 A comparison of the latency averages for the 5 different scheduling algorithms under normal freeway driving conditions (i.e., probability slowdown = 0.5) when simulated for 50,000 time steps. The intensity levels of Lane 2 vary, with Lane 1 maintaining an incoming intensity level of 25%. The curves for NRI and WFQ are identical.

Since the TakeTurns algorithm is dependent on a traffic light to enforce strict lane alternation, it performs significantly worse than other algorithms in terms of latency (until the traffic congestion levels becomes too great to make a difference). The latency performance of the WFQ and NRI algorithms is very similar under both driving conditions. Their performance is ideal until the congestion levels become too great to matter, at which point their performance is comparable to TakeTurns. The Platoon Scheduling performance stays rather consistent (and near ideal) through all of the varying intensities under both driving conditions. This is due to the fact that the congestion of the platoon lane forms at the start of the lane and not at the merge point. When the congestion increases, the vehicles schedule themselves to form strict platoon sizes, while maintaining a specified inter-platoon spacing for vehicles in the other lane to merge into.

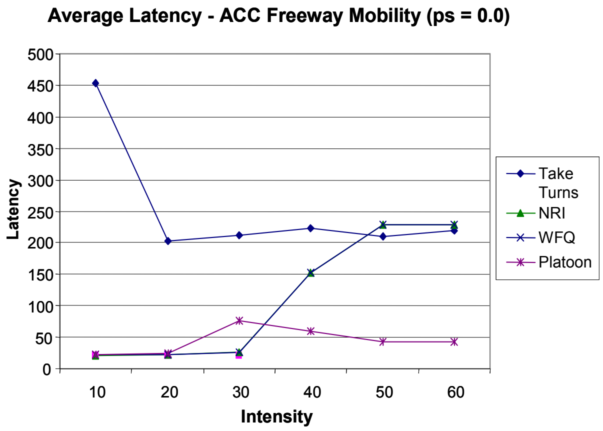

Figure 4 A comparison of the latency averages for the 5 different scheduling algorithms using ACC driving conditions (i.e., probability slowdown = 0.0) when simulated for 50,000 time steps. The intensity levels of Lane 2 vary, with Lane 1 maintaining an incoming intensity level of 25%. The curves for NRI and WFQ are identical.

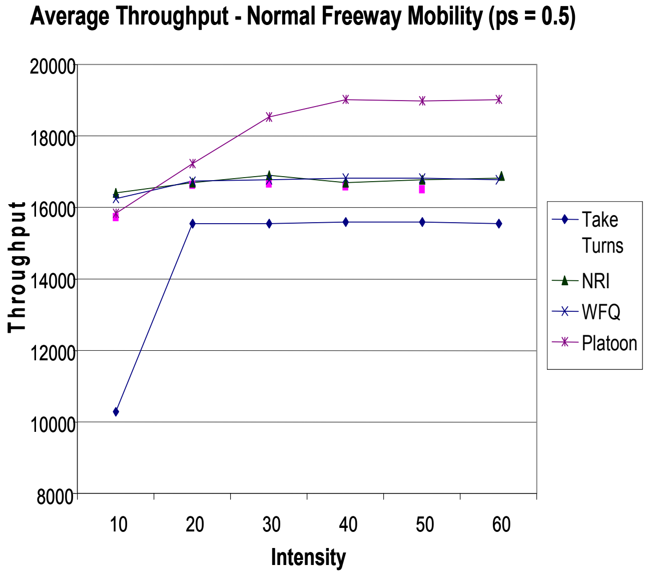

Figure 5 A comparison of the throughput averages for the 5 different scheduling algorithms under normal freeway driving conditions (i.e., probability slowdown = 0.5) when simulated for 50,000 time steps. The intensity levels of Lane 2 vary, with Lane 1 maintaining an incoming intensity level of 25%. The curves for NRI and WFQ are nearly identical.

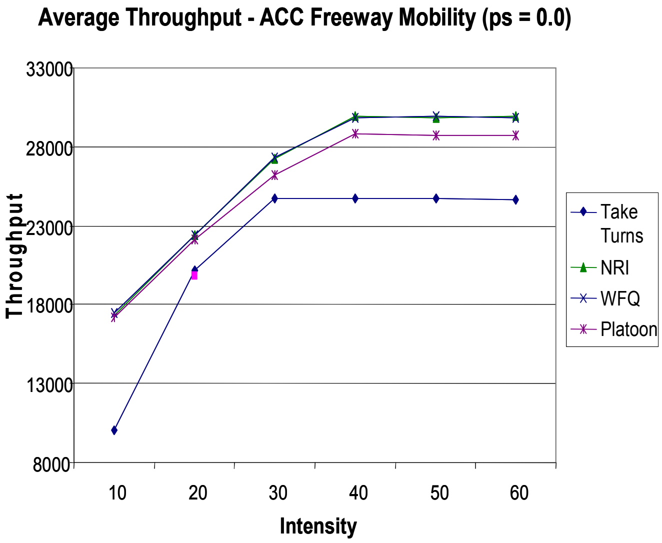

Figure 6 A comparison of the throughput averages for the 5 different scheduling algorithms under ACC driving conditions (i.e., probability slowdown = 0.0) when simulated for 50,000 time steps. The intensity levels of Lane 2 vary, with Lane 1 maintaining an incoming intensity level of 25%. The curves for NRI and WFQ are nearly identical.

As the congestion becomes significant, platoons are formed at the start of the platoon lane. Once a vehicle enters the platoon lane it will experience only slight delay before it can head towards the merge point (it must wait for the vehicles ahead of it to form platoons and allow for the specified inter-platoon spacing). Once the platoons are formed and the spacing is adequate, the vehicles can continue at their constant speed through the merge point as the other lane fits into the gaps.

It is interesting to note that the ACC driving conditions result in a significant decrease in terms of delay for all merging models, decreasing the latency by almost 50% for all cases. Therefore, ACC along with platoon scheduling appears to be an ideal combination for latency performance.

Figures 5 and 6 show the throughput results. Since the TakeTurns algorithm follows the strict alternation, it results in definite congestion at the merge point, corresponding to a decrease in throughput. It is interesting, however, to note that both NRI and WFQ schemes have a rapid transition region from which latency and throughout increases. This region corresponds to the transition from the free-flow branch to the congested flow branch on the fundamental traffic flow diagram. It is precisely in this region where traffic control (scheduling) can make a big difference.

The NRI and WFQ algorithms perform similarly, and have the best throughput performance of all the algorithms examined under ACC conditions, with the Platoon Scheduling performing the best under normal freeway conditions. In terms of overall performance, PS performs the best; it has achieves both high throughput and low latency. In the next section, we will examine a simplified form of PS to investigate its performance different parameter settings.

VII. RELATED WORK

Many kinds of road-side infrastructures (e.g., fixed sensors) for monitoring highway conditions and road-to-vehicle communications have been in place. They are used to support obstacle detection/avoidance and speed control. Adding wireless networking capability to vehicles further expands the telematic services in the automobile industry.

There are several projects that focus on developing {\em intelligent vehicles} based on DSRC, including Electronic TollCollection service (ETC), Advanced Cruise-Assist Highway System (AHS), and Vehicle Information and Communication System (VICS) [7], FleetNet [8], AutoNet [9], and Path [10]. AHS aims at reducing traffic accidents, enhancing safety, improving transportation efficiency, and reducing the operational involvement of drivers. On the other hand, VICS allows drivers to obtain road and traffic information in real time. FleetNet [8] is a research project which involves the cooperations from both industry (DaimlerChrysler, NEC, Bosch, siemens, etc) and academic units (University of Mannhein, University of Hannover, etc). FleetNet attempts to develop a wireless multi-hop ad hoc network for inter-vehicle communication to improve the driver's and passenger's safety and comfort. Location awareness and position data play a crucial role for FleetNet applications and the communication protocols deployed. NEC Network Labs Europe and University of Mannheim design and evaluate position-based routing and forwarding strategies [11, 12] for vehicular ad hoc networks within the framework of FleetNet.

The AutoNet project focuses on developing a distributed and self-organizing transportation management and control system. The autonomous information network is composed of vehicles containing GPS-equipped computers that communicate with each other information about traffic conditions. The California PATH project examines how vehicle automation technology can be used to relieve traffic congestion. The premise is that vehicles operating in tight coordination (platooning) in an automated highway system can provide a significant increase in highway throughput (vehicles per lane per hour moving along the highway). Also, aerodynamic drag is significantly reduced at close spacing, which can lead to major reductions in fuel consumption and exhaust emissions. The high-performance vehicle control system also increases the safety of highway travel, reduces driving stress and tedium, and provides a very smooth ride.

Another related project, funded by NSF, entitled Zero-Infrastructure [13], introduces and studies a fully decentralized traffic information system that is based only on data exchanged by equipped vehicles and does not require any infrastructure support. Vehicles exchange traffic information as they move through the network, which allows drivers to adjust their routes and avoid or be prepared for congestion, incidents or other hazards. While the concept is outside of the current policy philosophy of government agencies, the investigated system is quite simple: participating vehicles equipped with on-board computing and communication devices will be able to determine the vehicle's current location and past spatio-temporal trajectory as they traverse the network and exchange traffic measurements with other equipped vehicles.

VIII. CONCLUSION

While there has been a lot of research into ways to allow computational devices on vehicles to communicate with each other, there is often a necessary interaction with some sort of infrastructure in order to successfully execute traffic safety applications. In our work, we outline a framework for networking and computing grids to allow fully distributed traffic control via vehicular ad-hoc networks. We also outline a case study based on the problem of scheduling vehicles that are trying to merge from two lanes into one. Simulating various methods for scheduling these cars gives preliminary indications that using scheduling algorithms rather than driver insight can improve the throughput and latency of traffic merging scenarios.

REFERENCES

[1] Marc Bechler, Walter J. Franz, and L. Wolf. Mobile Internet Access in FleetNet. http://www.fleetnet.de/.

[2] Ioan Chisalita and Nahid Shahmehri. A peer-to-peer approach to vehicular communication for the support of traffic safety applications. In *ITSC*. IEEE, 2002.

[3] Drive-thru Internet: IEEE 802.11b for Automobile Users. http://www.drive-thru-internet.org/index.html.

[4] Kai Bolay. Cellular Automaton Traffic Simulators based on the work of Nagel-Schreckenberg (1992), Takayasu (1993), Helbing and Schreckenberg(1999). http://rcswww.urz.tu-dresden.de/~helbing/RoadApplet/.

[5] Alan Demers, Srinivasan Shenker, and Scott Keshav. Analysis and simulation of a fair queueing algorithm. In *Internetworking: Research and Experience*, pages 3–26, September 1990.

[6] Hui Zhang and Jon C.R. Bennett. Wf2q: Worst-case fair weighted fair queuing. In *INFOCOM*, pages 120–128. IEEE, 1996.

[7] Japanese Ministry of Land, Infrastructure and Transport, Road Bureau ITS. http://www.its.go.jp/ITS/index.html.

[8] FleetNet http://www.et2.tu-harburg.de/fleetnet/index.html.

[9] Autonet: Ad hoc Peer to Peer Information Technology for Traffic Networks, http://www.its.uci.edu/~mcnally/mgm-autonet.html.

[10] Partners for Advanced Transit and Highways (PATH), http://www-path.eecs.berkeley.edu/.

[11] Holger Fusler, Jorg Widmer, Michale Kasemann, Martin Mauve, and Hannes Hartenstein, Contention-Based Forwarding for Mobile Ad-Hoc Networks, Elsevier's Ad Hoc Networks, Nov. 2003, pp. 351-369.

[12] Christian Lochert, Hannes Hartenstein, Jing Tian, Holger Fusler, Dagmar Herman, and Martin Mauve, A Routing Strategy for Vehicular Ad-Hoc Networks inCity Environments, 58th IEEE Semiannual Vehicular Technology Conference VTC 2003-Fall, October 2003, pp. 156-161.

[13] A Zero Infrastructure Traffic Information System, http://trans.civil.northwestern.edu/~thanasis/research/projects/zi.html

[14] M.J. Cassidy and R.L. Bertini. Possible explanations of phase transitions in highway traffic. *Transportation Research*, 1999.