

NAVOPT: Navigator Assisted Vehicular route OPTimizer

Wooseong Kim and Mario Gerla

Department of Computer Science, University of California, Los Angeles, USA

Email: {wooseong@cs.ucla.edu, gerla@cs.ucla.edu}

Abstract—The number of vehicles has been growing compared to limited road capacity, leading to severe traffic congestion during rush hour. Urban planners are looking for solutions to reduce traffic congestion. Realtime traffic information obtained from vehicles can help reduce congestion exploiting the wireless communications architecture recently established for safety and traffic control purposes. In this paper, we propose NAVOPT, a vehicular routing strategy assisted by the on board navigator as well as the navigation server. The on board navigator equipped with area map and GPS monitor, reports its own position to the server via wireless connection (WiFi or 3G); in turns, it acquires from the Server a minimum cost path (i.e. path with shortest travel time) under the current traffic conditions. Under NAVOPT, the Server uses a flow deviation (FD) algorithm to compute optimal vehicle routes by load balancing vehicle traffic over alternate routes. We establish an analytic model of NAVOPT; we also simulate it using a SUMO simulator to verify its accuracy. From the simulation results, the NAVOPT model shows robustness to delay and speed and efficacy in controlling the traffic as the number of vehicles increases.

Index Terms—Vehicular Networks, Flow deviation, Navigator

I. INTRODUCTION

Today, research in Intelligent Transport System (ITS) is becoming increasingly popular as inter-vehicular and vehicle to road side unit (RSU) communications are become available. In the past, ITS was limited to locally restricted operations because of the use of infrastructure sensors (e.g. induction loop, camera), which are available only in a few cities. A navigator equipped with wireless radios as well as GPS however can replace the infrastructure by providing vehicle density per road and vehicle speed. Vehicles exchange the traffic information among themselves or report it to a traffic management server to enable estimation of traveling times. Based on the traffic condition learned from the navigator reports, vehicles can be dispatched optimally over multiple routes.

Given a stationary vehicle traffic distribution, the optimal routing solution which minimizes the average aggregate trip delay can be found with conventional gradient methods, or steepest descent techniques. A good introductory paper is the flow deviation (FD) method by Gerla, Fratta and Kleinrock [1]. The FD scheme was developed for packet switched networks, but it can be applied to any multi-commodity flow network with a concave link cost (i.e. the cost grows more rapidly than the flow). This is true for the delay on a packet network link, and it is also feasible for the vehicle delay on a road

segment, which grows faster than the flow of cars and can become unbounded in a deadlock situation.

There is a difference however between the road segment and the Internet link. While on an Internet link the packet rate increases constantly until it reaches capacity (at which point the delay becomes infinity), on the road segment the flow rate initially increases and then peaks when the segment becomes congested. After that it decreases as a function of offered load all the way to zero. We observe however that there is another variable on the road segment that is monotonically related to delay, namely, traffic density (i.e. number of cars per square meter). In fact, the delay keeps increasing with density and becomes infinity when the density is equal to the "parking lot" capacity of the road.

In this paper, we propose NAVOPT, a new architecture for vehicular routing. In NAVOPT, vehicles cooperatively sense traffic parameters such as density and delay. These parameters become the inputs to the traffic optimization algorithm. The algorithm is inspired to the FD approach. It computes the marginal cost of each link as the derivative of delay versus vehicle rate on the road segment (aggregated over all lanes). When a vehicle requests a new route, the central Navigator Server, using the FD algorithm, finds a path (from current position to destination) that minimizes the sum of marginal delays on its links. This is the classic shortest path problem. The Server reroutes the vehicle on the new path if the marginal cost of the new path is less than the cost of the current path. Otherwise, the vehicle stays on its current route (sorry!). The intuitive interpretation of this approach is that it is a steepest descent approach that removing density from a congested path and it adds it to a clear path.

In the implementation, the navigator server keeps track of road delays (and densities) and computes weights (from vehicle speeds) for all road segments. We assume that for each road segment a delay has been computed ahead of time, which actually is available from popular traffic congestion papers. When a car requests a new route, the navigator server computes the best marginal path and dispatches it to the driver. If there is no improving path, the driver will be kept on the current path. In this paper, we establish a numerical model for the NAVOPT routing and flow algorithm and show a performance comparison between distributed traffics based on our algorithm and shortest path traffics by selfish manner. In addition, we simulated same scenarios using SUMO simulator to demonstrate feasibility of our approach.

The remainder of this work is structured as follows: Section II reviews related works. Section III describes our NAVOPT model and illustrates it on an example urban configuration. Numerical analysis and simulation results are shown in Section IV. The paper is concluded in section V.

II. RELATED WORK

In [2], a system framework using traffic and congestion driven vehicular navigation was introduced. Vehicles are used to estimate traffic conditions based on their position and speed. All measured data is reported to the traffic management center (TMC) via the cellular networks. This data is used to estimate traveling time on the main urban roads. After that, the estimated traveling time is disseminated back again to vehicles as a form of feedback through the cellular networks. Performance results from this approach shows the feasibility of probing traffic condition with only a small number of vehicles, around 4% , rather than with conventional fixed sensors like cameras or induction loops.

As probing nodes, vehicles sense and gather traffic condition and disseminate the information through the inter-vehicle communication (IVC) or vehicle-road side unit (RSU). A framework for vehicular sensing was introduced in [3] which optimized message dissemination in limited channel bandwidth using data aggregation and information aging. MobEye as a vehicular sensing platform harvests urban traffic and urban environment information and diffuses them opportunistically based on vehicle mobility [4]. Here high mobility of vehicles encourages data dissemination. It also proposes summary harvesting and summary diffusion to reduce duplicated data and save channel bandwidth.

TraNs [5] is an interactive vehicular simulator between communication and vehicular simulators (NS-2 [6] and SUMO [7]). Many studies about VANET have used a mobility model generated by traffic simulators in off-line state. TraNs however provides programmable interfaces between two simulators to exchange coordinate information of vehicles during simulation time. In network centric simulation, for example NS-2 can acquire new node coordinates which are updated by the SUMO simulator based on realistic vehicular mobility model. Also in the SUMO simulator, mobility parameters (e.g. speed, route and lane) can be changed by command from VANET applications like safety alarm in NS-2. Unfortunately, TraNs is not fully exploited in terms of effect of traffic information on vehicular routing. With another version integrating CARISMA as one of traffic simulators and NS-2, authors showed simulation results about vehicular routing in an accident scenario [8]. They stopped the vehicle intentionally before the accident emulation and the broadcast of warning messages through the network simulator. VANET instrumented with the safety or traffic control applications helped vehicles stuck in the place of an accident using detours that reduced congestion. In [9], authors did similar simulations in a simulator combining OMNet++ [10] and SUMO. They compared performance when using TCP and UDP as transport protocols delivering traffic messages in OMNet++. However,

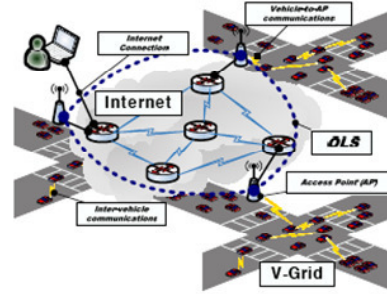


Fig. 1: Model of urban vehicular networks

vehicle traffic distribution for load balance in urban scenarios has never been demonstrated in network and traffic simulators.

III. NAVOPT: NAVIGATOR SERVER ASSISTED VEHICULAR ROUTING

A. NAVOPT architecture

In NAVERO, we assume vehicles are equipped with a smart navigation system which supports wireless communications and features geographical map information. The navigation system has geographical positioning system (GPS) and WiFi (i.e. 802.11a/b/g) or dedicated short-range communications (DSRC) (i.e. 802.11p and WAVE) radio interfaces [11]. DSRC is mainly aimed at the exchange of safety messages for inter-vehicle communications. Also, vehicles can share traffic information through IVC or VRC. Due to limited channels in DSRC, we can expect many vehicles to utilize WiFi channels since the spectrum is free for unlicensed users and the WiFi radio interface cost much less than the DSRC one. And there could be open WiFi access points that serve as RSUs to access the Internet.

Figure 1 illustrates NAVOPT architecture. The access points as RSUs constitute a wireless mesh network (WMN) and provide Internet access to moving vehicles as shown in the figure. The vehicles organize V-GRID, namely a mobile mesh network among the vehicles. In this NAVOPT architecture, the vehicles can communicate with each others as well as with servers located in the Internet through the RSUs, or; they communicate directly through the V-GRID using multi-hop communications like in VANET.

NAVOPT is implemented in centralized way; each geographical partition has a designated navigation server which provides optimal travel routes to drivers. Large-scale vehicular networks such as nation-wide networks can be segmented into many small-scale networks like city networks as a hierarchical system for scalability.

Vehicles gather traffic information such as vehicle density, speed, accident locations etc using various on board sensors such as camera and GPS. In a typical NAVOPT scenario, some vehicles may not report their traffic information due to the fact that they are selfish or do not have wireless connections. However, NAVOPT can improve vehicular routing even with partially observed traffic information and with control on

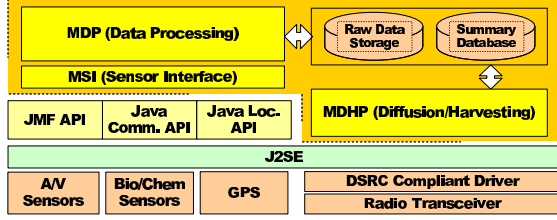


Fig. 2: Urban vehicular sensing platform

only a part of the vehicles, as we will explain in next subsection. Figure 2 shows a NAVOPT platform for urban vehicular sensing and communications [4]. It is composed of a communication module for sensed data distribution using DSRC or other radio transceivers, a sensing module, a processing module and a data base module for traffic information management. Vehicles report periodically their sensed information, eg position, average speed, travel direction etc, to a navigation Server. Based on such information, the navigation Server can estimate travel time as a function of average number of vehicles on the road and delay to travel the road segment. Actually, the delay can be derived from the density of the road based on experimentally validated numerical models. However, exceptional cases like accidents can alter the relationship between them. In this paper, we limit ourselves to traffic distributions that correspond to normal traffic conditions.

B. Multi-commodity flow problem in vehicular routing

In this section, we give a design overview of NAVOPT, a navigator assisted optimal vehicular routing. We can regard the stop-and-go traffic model at each intersection as a store-and-forward communication model. The intersection of roads is the same as a node (i.e. router) which decides a route and forwards packets in the communications network. In this context, we adopt a general queuing model of the communication network for our vehicular routing model.

Multi-commodity flow problem is finding best routes for multiple flows while minimizing total cost of routing (e.g. end to end delay) [1]. This vehicular routing optimization problem extended from store-and-forward design of packet networks can be formulated as follow:

$$\min \sum_{s,f} P(\Phi), \quad f \in F, s \in G$$

,where $P(\Phi)$ is a performance measure of multi-commodity flow (e.g. delay or cost) for a given flow configuration Φ . F and G are sets of flows and nodes in a given network topology. Requirement flow, r^f is an incoming flow into the given network partition from the outside world. The flow is constraint by the following network flow conservation rule and must be positive.

$$\sum_{j \in G(j \neq i)} x_{ji}^f + r_{(s(f) \text{ or } d(f)=i)}^f \leq \sum_{j \in G(j \neq i)} x_{ij}^f \quad \forall f, i, j \quad (1)$$

,where the r^f is more than zero if its source is the node i and less than zero if its destination is the node i .

$$x_{ij}^f > 0 \quad \forall f, i, j \quad (2)$$

When a multi-commodity global flow f satisfies the above constraints Eq.(1) and Eq.(2), the set of feasible flows, F spans a convex polyhedron. A local extreme point in F corresponds to a set on non-bifurcated (i.e. not split on multiple path) flows. [12].

From now on, we define the vehicular routing performance function, $P(\Phi)$ that can be expressed by delay cost, where the flow configuration, Φ is global flow assignment on each road. The average travel time, T as the end to end delay from a source to a destination is,

$$T = \sum_{e=1}^b \frac{\lambda_e}{\gamma} T_e \quad (3)$$

,where T is total average delay (hour/vehicle) of the global flows, b is number of road segments, λ is arrival rate of a flow at each road segment e and γ is requirement (total vehicle arrival rate to the system) from external networks. A requirement flow r^f from node i to j can be written by r_{ij} , where r_{ij} is a $n \times n$ matrix at the given network that consists of n nodes (i.e. intersections) and b roads (i.e. total road segments in network). Then, total incoming flows are,

$$\gamma[\text{vehicles/hour}] = \sum_f r^f = \sum_{i=1}^n \sum_{j=1}^n r_{ij} \quad (4)$$

T_e is average delay for road e which is composed of queuing (T_q) and propagation delay (T_p) as time to wait traffic signal at an intersection and time for moving respectively:

$$T_q = \frac{1}{C_e - \lambda_e} \quad (5)$$

,where λ_e is arrival rate (e.g. vehicles/hour) at each road segment and C_e is road capacity (e.g. vehicles/hour). As shown in 5, T_q as a main input parameter for vehicular routing algorithm is eventually determined by vehicle density of each road e .

$$T_p = \frac{\text{RoadLength}}{\text{MaxSpeed}} \quad (6)$$

With the average delay (T_e), the equation Eq.(3) is rephrased as follow:

$$T = \sum_{e=1}^b \frac{\lambda_e}{\gamma} T_e = \frac{1}{\gamma} \sum_{e=1}^b \left(\frac{\lambda_e}{C_e - \lambda_e} + \lambda_e \cdot T_p \right) \quad (7)$$

Figure 3 depicts an example of vehicle formation to help describe each parameter mentioned above. Road is segmented by 400m with a single lane for each direction. Maximum 60Km/h is allowed typically in urban roads according to speed policy. Thus, travel Delay (T_p) per each road segment for example can be $\frac{400m}{60km/h} = 0.0067[\text{hour}]$. Approximated maximum density in each lane of the road e is around $400m/4m$

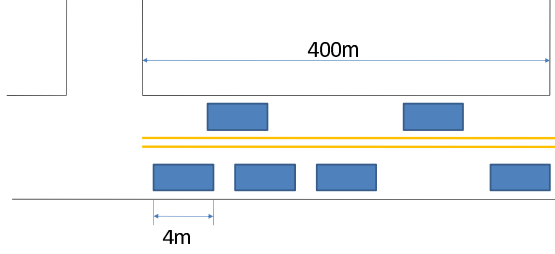


Fig. 3: Model of vehicle formation on a road

= 100 vehicles when we assume vehicle length is 4m. Thus vehicle density in the road is 250[vehicles/km]. Density can be varying with vehicle type (e.g. bus or trailer) and safety gap between vehicles. Based on this approximated density, road capacity C_e is derived as 100 vehicles/0.0067 hours = 14,925 [vehicles/hour] when we ignore micro-mobility model such as deceleration in intersection etc. Our model is useful to investigate the efficacy of the FD algorithm for vehicular routing. However, the actual traffic is quite different from the traffic estimated by this numerical approach because of the many extra factors that impact our analysis.

In this study, our objective is to find the optimal route assignment of the "multi-commodity" flows. For given topology and road capacity as described in previous paragraphs, the route optimization problem can be formulated as follow:

$$\min T(f) = \min \frac{1}{\gamma} \sum_{e=1}^b \left(\frac{f_e}{C_e - f_e} + f_e \cdot Tp \right), \quad f = \lambda \quad (8)$$

$$f_e \leq C_e \quad (9)$$

Here the primary problem is relaxed by replacing the capacity constraint with $C'_e = C_e - \varepsilon$ where the ε is acceptable positive tolerance. In NAVOPT, this unconstrained multi-commodity flow problem is solved by the FD method that finds the minimum cost solution using steepest descent in a differentiable problem. Flow deviation method is an iterative method. It evaluates at each iteration the direction of the steepest decent to find shortest path flow as the minimum cost solution as follows:

$$f^{n+1} = (1 - \alpha) \cdot f^n + \alpha \cdot v$$

, where α is step size ($0 \leq \alpha \leq 1$) and v is a minimum first derivative path flow. NAVOPT processes flow deviation repeatedly to find optimal traffic distribution which leads to stationary condition as $T(f^n) - T(f^{n+1}) < \varepsilon$.

IV. EVALUATION

We first have established a numerical model of NAVOPT using the flow deviation algorithm described in the previous section and have run simulations with the SUMO traffic simulator. Figure 4 shows a road topology to evaluate NAVOPT performance, namely a 10 x 10 Manhattan grid where each road segment is 400m and maximum allowed speed is 60 Km/h. Vehicle length is 4m. Acceleration and deceleration is

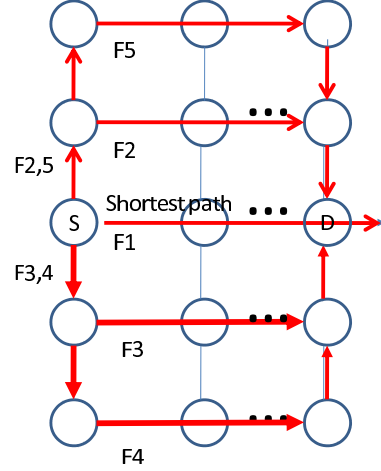


Fig. 4: Manhattan Grid topologies for evaluation

4 and 4.5 respectively. Driver imperfection (σ) is 0.5. Here we assume a flow from a source denoted by S to a destination, D as can be seen in the Figure 4. A set of routes (i.e. from F1 to F5) from the source to the destination consists of a shortest path and four alternate paths. Therefore, the routes have different path lengths. For example, the shortest path F1 has 9 road segments and F2 and F3 have 11 road segments.

A. Numerical analysis

Based on NAVOPT model, performance was evaluated in terms of average delay, average speed and average density under varying requirement of global flows. Figure 5 plots performance comparison between NAVOPT and shortest path routing algorithm that is common in conventional navigator-assisted driving.

As can be seen in Figure 5a, total average delay of each route in NAVOPT is almost constant as the incoming flow of the requirement increases while delay of the shortest path increases infinitely from around 14800 [vehicles/h]. In case of NAVOPT, the flow is deviated into different routes including the shortest path. In contrast, the shortest path algorithm suffers from increased number of vehicles along the path. In Figure 5c, average density has a similar curve with the average delay since the delay increases monotonically by number of incoming vehicles into the road each unit time.

According to fluid traffic model (FTM), the speed of vehicle can be derived by current vehicle density over the road [13].

$$v(t + \Delta t) = \max[v_{min}, v_{max}(1 - \frac{\rho(t)}{\rho_{max}})]$$

,where the v_{min} and v_{max} are allowed minimum and maximum speed, and $\rho(t)$ is current vehicle density and ρ_{max} is maximum vehicle density of the road.

Figure 5b depicts average speed of the road segment. Average speed in NAVOPT is close to maximum speed allowed in the road but speed in the shortest path algorithm decreases fast as the density increases. Once the density becomes saturated

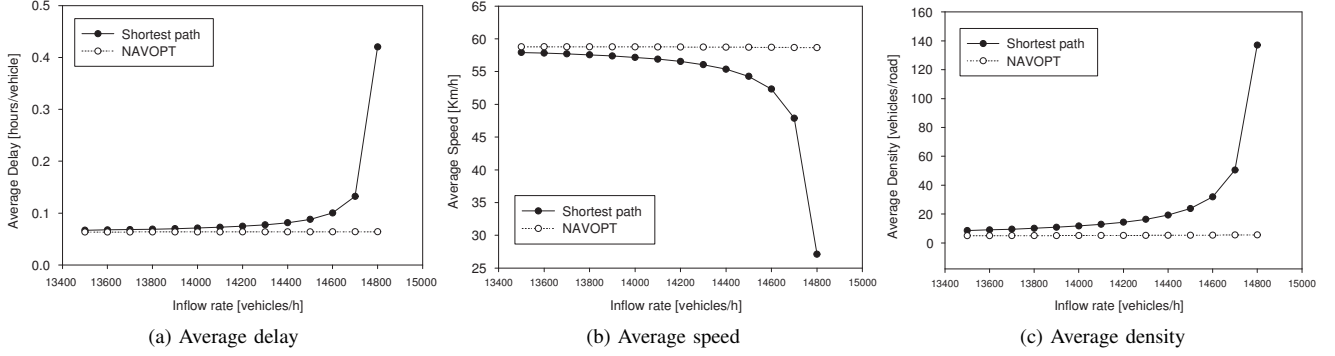


Fig. 5: Performance results from numerical model

in maximum density, 250 vehicles/Km like parking lot, speed of vehicles eventually becomes zero without moving.

B. NAVOPT with partial Navigator penetration

In a real situation, NAVOPT cannot control all vehicles on the road. The navigator that instructs a driver to move along a route prescribed by the NAVOPT server might not be available on some of the vehicles. It may also happen that the on board navigator does not have Internet connectivity or the driver does not follow NAVOPT instructions. Therefore the situation when only a fraction of the vehicles is NAVOPT capable must be anticipated. Based on these premises, the effect of "partial penetration" was investigated by varying the percentage of NAVOPT enabled vehicles (e.g. 10% controllable vehicles).

Figure 6a shows average delay for the Manhattan grid topology for incremental flow deviations. The step size was configured as 0.01 of the shortest path that is actually the minimum first derivative path. The average delay decreases as the number of steps increases under fixed incoming flow rate, 14800 [vehicles/h]. It is remarkable that at the first step, the delay drops from 0.4 to about 0.1. After that, the average delay still decreases but very slowly. This result implies that even only 1% deviation of the congested flow (or equivalently, 1

The average delay with varying control factors is shown in Figure 6b. Difference exists in the average delay according to ratio of controlled vehicles. Above the factor 0.4 (i.e. 40% of total vehicles), however the average delay does not change; the 3 curves are practically overlapped. According to the comparison of average delay as shown in Figure 6c, controlling only 10% of the vehicles achieves the same performance as controlling all the vehicles. From this result, we can conclude that NAVOPT is a very effective method to reduce traffic congestion even if the amount of controllable vehicles (i.e. navigator penetration) is rather small.

C. SUMO simulation results

We simulated the same scenario used for the numerical FD optimization using SUMO-0.12.0 simulator [7] which supports a micro mobility model for each vehicle as well as a flow model. We generated a Manhattan grid map and multiple flows

from same source and destination. First, all of the flows use the shortest path and they are optimally deviated into the multiple paths that were obtained from the NAVOPT numerical analysis. Performance is evaluated (in terms of average delay, speed and density) for increasing inflow rate values, both under shortest path and under multiple paths set by NAVOPT.

Average delay for each road segment is shown in Figure 7a. Average delay in a road increases drastically when incoming flow rate approaches 14800 [vehicles/hour] in case of the shortest path scenario. In contrast, NAVOPT shows almost constant delay even if the incoming flow rate increases. It is worth noting that the delay does not go to infinity since the simulator automatically decreases traffics at congested inlets. The same result can be seen in Figure 7c. The density of a road became saturated as we kept increasing the incoming flow rate. From this observation, another difference between numerical model and simulator is the maximum vehicle density on each road. Maximum 250 vehicle capacity per each road segment was speculated in our numerical model; the simulator peaks with only 40 vehicles. Also, vehicle density in NAVOPT increases very slowly as the incoming flow rate grows. As for average speed in each road, vehicles in NAVOPT maintain an average speed near 42 Km/h while average speed of vehicles on shortest path drops near 34 Km/h. Average speed also saturates since it is depends on vehicle density in the road.

V. CONCLUSION

In this paper, we proposed NAVOPT, a new vehicular routing architecture that exploits vehicular wireless communications and navigator services. NAVOPT utilizes traffic information (e.g. traffic flow, density, speed) gathered from vehicles to estimate traveling times and to find optimal routes. We first established a numerical model for vehicular delays in the urban grid. We the used the Flow Deviation routing algorithm in the NAVOPT server to compute optimal routes and to investigate key performance measures such as average delay, speed and density. Finally, we compared NAVOPT solutions to Dijkstra's shortest path solutions. To demonstrate feasibility of our approach, we also simulated the FD algorithm in a Manhattan

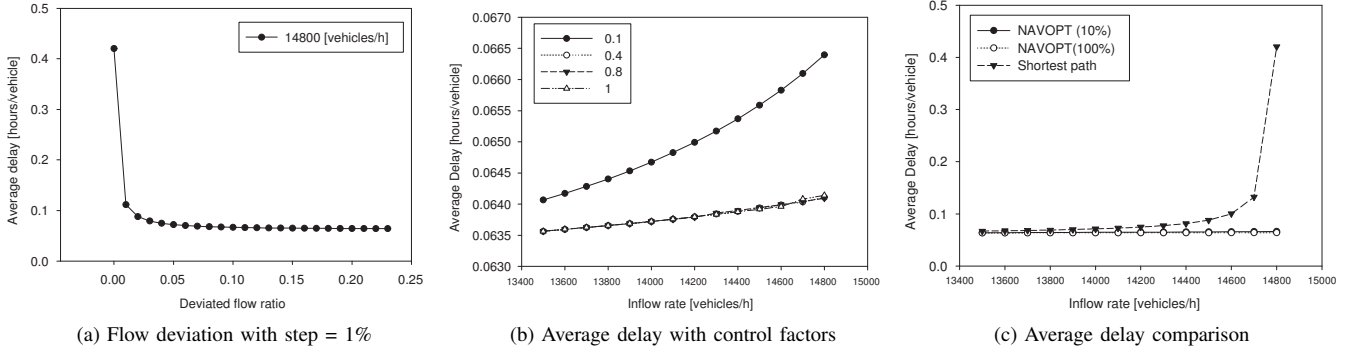


Fig. 6: Performance by partially controlled vehicles

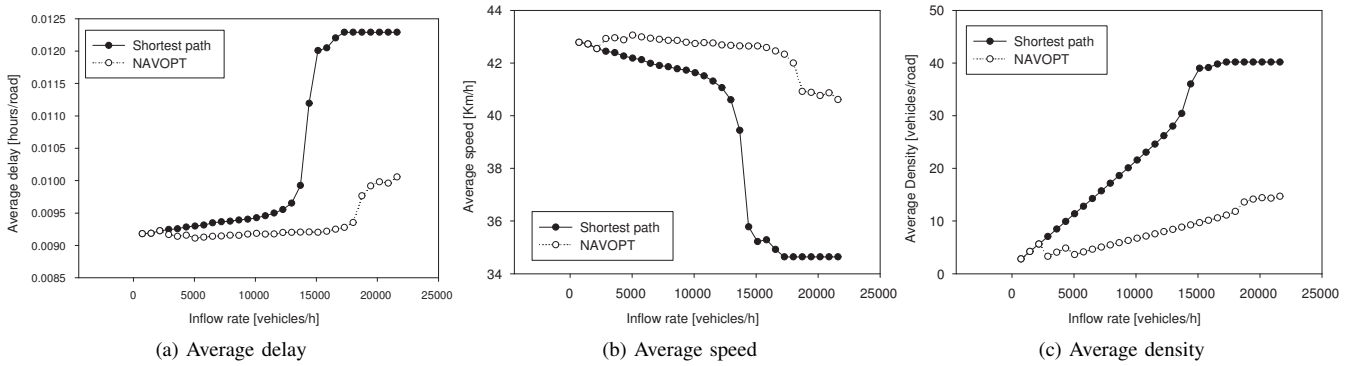


Fig. 7: Performance results from SUMO simulation

grid using the SUMO micro mobility simulator. According to simulation results, NAVOPT improved speed and throughput by about 25% compared to shortest path routing - namely the "selfish" routing implemented by individual drivers when they use alternative routes around the shortest path. Also, NAVOPT reduces total traveling time by 40%. NAVOPT showed it could improve throughput even with partial penetration based on numerical analysis. As for future work, we plan to implement our algorithm on top of a combined Qualnet and SUMO simulator.

REFERENCES

- [1] L. Fratta, M. Gerla, and L. Kleinrock, "The flow deviation method: An approach to store-and-forward communication network design," *Networks, an International journal*, vol. 3, no. 2, pp. 97–133, 1973.
- [2] K. Sanwal and J. Walrand, "Vehicles as probes," in *Technical Report, UCB-ITS-PWP-95-11*, 1995.
- [3] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode, "Trafficview: traffic data dissemination using car-to-car communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, no. 3, pp. 6–19, 2004.
- [4] U. Lee, E. Magistretti, M. Gerla, P. Bellavista, and A. Corradi, "Dissemination and harvesting of urban data using vehicular sensor platforms," *IEEE Transaction on Vehicular Technology*, vol. 58, no. 2, pp. 882–901, 2009.
- [5] M. Piorkowski, M. R. A., L. Lugo, P. Papadimitratos, M. Grossglauser, and J.-P. Hubaux, "Trans: Joint traffic and network simulator for vanets," in *Proc. of ACM MOBICom*, 2007.
- [6] "Ns-2 simulator," in <http://www.isi.edu/nsnam/ns/>.
- [7] "Sumo simulator," in <http://sourceforge.net/apps/mediawiki/sumo/>.
- [8] S. Eichler, B. Ostermaier, C. Schroth, and T. Kosch, "Simulation of car-to-car messaging: Analyzing the impact on road traffic," in *Proc. of IEEE/ACM MASCOTS*, 2005.
- [9] C. Sommer, Z. Yao, R. German, and F. Dressler, "Simulating the influence of ivc on road traffic using bidirectionally coupled simulators," in *Proc. of IEEE INFOCOM Workshop*, 2008.
- [10] "Omnet++ simulator," in <http://www.omnetpp.org/>.
- [11] "Ieee 802.11p," in <http://grouper.ieee.org/groups/802/11/Reports/>.
- [12] T. C. Hu, "Integer programming and network flows," in *Addison-Wesley*, 1969.
- [13] E. Kometani and T. Sasaki, "On the stability of traffic flow," in *Journal of Operations Research*, 1958.