V2V Communication for Analysis of Lane Level Dynamics for better EV Traversal

Akash Agarwal and Praveen Paruchuri[1]

Abstract— **Slow moving traffic in heavily populated cities, can many times result in loss of lives due to emergency vehicles not being able to reach their destination hospitals on time. Recent advances in the field of Intelligent Transportation Systems (ITS) makes it increasingly likely that vehicles in the near future will be equipped with advanced systems that allow inter vehicular communication. In this paper, we assume the usage of such a system to optimize the lane level dynamics for an emergency vehicle (EV), traversing a multi lane stretch of road under a variety of traffic settings. In particular, we present the Fixed Lane Strategy (FLS) and the Best Lane Strategy (BLS) for EV traversal and perform an extensive agent based analysis to study their strengths and weaknesses. Through a series of experiments performed using the well-known traffic simulator SUMO, we could show that: (a) BLS performs better than SUMO strategy on all traffic settings we tested. (b) BLS performs better than FLS in settings that capture real-world traffic conditions involving congestion and uncertainties while FLS performs better in well-behaved conditions and (c) BLS was found to be the best strategy for the setting calibrated using real world data (obtained from NYCDOT).**

I. Introduction

Recent advances in the field of **Intelligent Transportation Systems (ITS)** make it increasingly likely that vehicles in the near future will be equipped with advanced systems that allow inter vehicular communication [1]. This is being made possible using VANETS (vehicular ad hoc networks) [2], a key component of ITS. Vehicle to vehicle (V2V) communication will allow for several innovative methods of traffic management. In this paper, we will study one application of this technology that can improve the traversal time of **emergency vehicles (EVs)** [3]. Note that we only assume usage of inter-vehicular communication system (V2V) but not the usage of any other road side infrastructure [4].

Improving the travel time of EVs can potentially save many lives [5], [6]. For example, [5] presents statistics related to thousands of people being affected by EV delays in UK along with detailed location wise statistics. Medical guidance, according to this article, says that immediately life-threatening (Category A) calls should receive a response within 8 minutes for atleast three quarters of the cases. This target was set because the chance of surviving a heart attack reduces by 10% with every minute that passes. [6] presents the following EV response times breakdown for Wales (part of UK): 42.6% of Category A calls received an emergency response within 8 minutes, 47.4% within 9 minutes, 52.0% within 10 minutes, 68.9% within 15 minutes, 79.1% within 20 minutes and 89.8% within 30 minutes. The key takeaway point from all this information is that, we are looking for savings in EV traversal time of the order of few minutes or even seconds, that can result in a life saving difference.

Simulators have long been useful aids where the understanding of phenomena, that can be simulated, are quite difficult. Traffic is one such domain where simulations have been found to be very helpful and have a rich history [7], [8]. Traffic simulations facilitate the evaluation of infrastructure and policy changes before implementing them on roads. For example, the effectiveness of dynamic traffic management systems [9] or autonomous intersection and traffic light control mechanisms [10], [11], can be tested and optimized in a simulation before being deployed in the real world.

At a broader level, traffic simulators can be classified into macro (involves modeling the general aspects of system like the average speed of vehicles on the road, vehicle density) and micro (involves modeling each vehicle at an individual level) level simulators [12]. In this paper, we perform our analysis using a free and open microscopic traffic simulation suite named **Simulation of Urban MObility (SUMO)** [13], to identify the best possible traversal strategy for an EV in a variety of settings. SUMO allows modeling of traffic systems and has a wealth of supporting tools which can handle tasks such as route finding, visualization and network import. We will use its rich feature set to simulate a variety of realistic traffic scenarios, then introduce the EV and allow its behavior to play out in the simulation.

Our assumption regarding availability of V2V communication infrastructure and its interaction with SUMO ensures that the EV can interact with vehicles upto a particular communication distance that allows it to sense other vehicles, find about their speeds and other details, send lane change requests and other such interactions. This interaction process is enabled by running a communication simulation which runs in parallel to SUMO and interacts with SUMO using its TraCI interface [14]. For example, upon receiving a lane change request, the vehicle requested only initiates the lane change action while the dynamics of lane change are handled by SUMO. By design, our algorithms are abstracted from all the low level operations like vehicle acceleration and deceleration, to focus specifically on high level planning part, i.e., strategies to intelligently pick lanes for the EV.

Roads in heavily populated cities like Hong Kong, Manila, Mumbai, Dhaka and Seoul suffer from extremely slow and dense traffic. Furthermore, in cities with relatively faster traffic like New York, London and Singapore, there is a wide

[1]Akash Agarwal and Praveen Paruchuri are with Center of Data Engineering, IIIT Hyderabad, Gachibowli, Hyderabad 500032, India akash.agarwal@research.iiit.ac.in, praveen.p@iiit.ac.in

variation in traffic speeds and many times the traffic can be quite slow. Such slow moving traffic can result in loss of lives due to EVs not being able to reach their destination hospitals on time. In this paper, we specifically focus on scenarios where the roads are crowded with slow moving traffic but our strategies are general enough to be applied to any type of road settings including free-ways and highways.

The rest of the paper is organized as follows: Section II presents in detail the related work. Section III presents the description of the SUMO Strategy, the **Fixed Lane Strategy (FLS)** and the **Best Lane Strategy (BLS)** algorithms. It is followed by Section IV, which explains the setup of the traffic simulator. We then illustrate the results of the experiments in Section V. Finally, Section VI presents the conclusion and some insights on future work.

II. Related work

We identified the following different threads of work related to improving the efficiency of EV traversal:

(a) Optimal routing problem: Finding the appropriate route from base station to the location of the EV, and to other destinations (e.g., hospitals) thereafter. ([15], [16]) identify the best path to be taken by the EV but do not describe the actual traversal details (like selection/changing of lanes).

(b) Determining lane-level dynamics: Determining the actual traversal details once an EV enters a stretch of road on its route to destination, including computing the best lane to travel on and the consideration of lane changes.

(c) Control of traffic lights: During emergencies, traffic lights may be controlled to reduce wait times at intersections for a faster journey. [17] suggests methods for preempting traffic lights using vehicle to infrastructure communication and [18] presents a warning system using radio communication to warn other vehicles and notify traffic lights.

We focus specifically on point (b) in this paper i.e., computing the lane level dynamics for an EV. We envision that the low level traversal strategies demonstrated in this paper will work in conjunction with both (a) high level planning of routes and (c) generation of a green wave for the EV by preempting traffic lights.

There are several lane changing models that are used by vehicles in microscopic traffic simulation suites like SUMO [19]. These models perform very well for vehicles in general, but for an EV we are able to obtain significant traversal time improvements by using BLS in comparison to previous models. This is because EVs can use specialized routing strategies to take advantage of the fact that other vehicles make way for EVs and also to account for the accurate information available about other vehicles such as speed, vehicle position etc., obtainable due to ITS assumptions.

Most lane changing models incorporate the following types of lane changes: (a) Lane changes due to route requirements of the vehicle under consideration [19], [20] (b) Lane changes to obtain faster speeds. This is mainly based on speed of the leader(s) (the vehicle(s) immediately in front of the vehicle under consideration). For example, models like [21], [22], are rule/discrete-choice models which consider lanes with faster leader(s) as better. However, they do not consider the overall advantage of being on a lane where the leader is not the fastest but maybe a better lane on average. [23], [24] introduce incentive/utility based models for lane-changing behavior which considers the incentive/desire to change lanes and the gaps/risks involved. But, the incentive/utility itself is based mainly on attributes like velocity of the leader(s), gap between the vehicle under consideration and each of the leader(s) and the possible acceleration/deceleration if lane change happens. [25] describes a lane changing model which uses information from surrounding vehicles and infrastructure to anticipate beforehand, the possibility of a slow leader. But like in other models, it has no provisions to anticipate better lanes based on vehicles clearing out due to EV.

In this paper, we introduce the BLS algorithm which instead of deciding on lanes based on a faster leader(s) uses a sophisticated strategy that can adapt to varying traffic patterns. This is achieved by using a utility function that includes average speeds, slowest speeds and normalized free space, and considers both the possibility of an immediate faster lane and the clearing time of other vehicles. We also introduce the FLS algorithm which acts as a good baseline. Our paper includes extensive experimentation on a number of settings developed to model real world traffic environments including one setting calibrated using real-world data obtained from NYCDOT, which hasn't been focused in prior works. We compare our strategies with the SUMO strategy, which includes a state of the art Tactical lane-changing model (lane-change maneuvers where a vehicle attempts to avoid following a slow leader) [20] and show that our strategies perform much better as compared to the SUMO model.

III. The Strategies

We present here two strategies for the EV to handle lane level dynamics while traversing on a multi-lane stretch of road namely **FLS (Fixed Lane Strategy)** and **BLS (Best Lane Strategy)**. Our strategies were developed, specifically for the EV, to take advantage of the higher priority an EV has and the accurate information available about other vehicles due to the presence of inter-vehicular communication. Hence, these strategies only specify the lane that the EV should travel in while the low-level dynamics of traversal namely speed, acceleration/deceleration and the dynamics involved in changing the lane for the EV are handled by SUMO. Another task these strategies perform is to identify the appropriate vehicles to send lane change requests.

In a similar vein, the traffic behavior for all the other vehicles is entirely generated by SUMO depending on the initial parameters which are picked using five different settings described in Section IV-C (one of these is calibrated on data from NYCDOT). Once the initial parameters are picked, we do not make any modifications to the general traffic behavior except for the two specific interventions presented in Section IV-F to incorporate lane change requests and lane modeling related parameters which SUMO does not provide otherwise. The key idea here is to let traffic patterns evolve using the underlying models SUMO has with minimal intervention and focus on lane picking strategies. This is also the same reason for why we let SUMO control the low level dynamics of EV. We then study the strengths and weaknesses of the lane picking behavior generated by these two strategies under a variety of experimental settings and benchmark it against SUMO strategy with suitable modifications described below.

A. The SUMO strategy

SUMO provides an “emergency” vehicle class which combined with the default simulator strategy to control the dynamics of vehicular traversal allows the modeling of EV traversal behavior. In real-world, EVs have sirens and lights to indicate other vehicles to make way. To model this into SUMO, we add communication on top of the SUMO default strategy. This will allow EV to send lane change requests to vehicles on its current lane up to a distance c_d in front of it. We assume that all lane change requests sent will reach the intended vehicles and vehicles will try to clear immediately upon receiving them (We also analyze situations when such a clearance is not possible). Similar communication facility is assumed for FLS and BLS strategies (not to model the effect of sirens/lights but) due to availability of V2V communication. Availability of communication in SUMO provides an advantage to the EV since in real traffic scenarios sirens may not always be heard or drivers may not always know the lane to clear until the EV is in sight. Hence, the SUMO strategy we use for bench-marking is expected to be favorable to the EV in terms of travel time.

B. The Fixed Lane Strategy (FLS)

The idea behind FLS is that the EV should identify the lane, that is fastest on an average, based on prior information and pick that as the fixed lane for its entire journey. This would on an average ensure the minimum possible traversal time unless exceptional situations arise. Assuming a right handed driving system, in most cases leftmost lane is the fastest, as faster vehicles tend to move on the left lanes while the slower ones move on the right lanes (the vice-verse typically holds true for left handed driving systems). When using FLS the EV therefore moves to the leftmost lane from its current position and then tries to clear out the vehicles from that lane. As described earlier, we assume V2V communication is feasible in the ITS framework. The EV can therefore clear a lane, by sending lane change requests to other vehicles traveling on its lane within its communication distance, c_d. It sends a lane change request after every fixed re-computation interval, t. We also analyze situations when clearance of other vehicles is not feasible.

C. The Best Lane Strategy (BLS)

The idea behind BLS is that the EV should identify the best lane at the current time and move to that lane from its current position. In order to do this the EV must calculate the utilities of the current lane and the other lanes using the utility computation function described below and then take a decision to switch if it is beneficial to do so. Computing this utility function would need the EV to make use of V2V communication and obtain the needed information. Similar to FLS the EV tries to clear out vehicles from the lane it currently is in (and like for FLS we analyze when such clearance is not feasible). Traffic being a dynamic domain once the EV moves onto the best lane, the traffic patterns on different lanes may change. BLS should therefore be able to switch to better lanes.

1) BLS Utility Computation: A key step in BLS is to compute the utility u_l, of a lane l. We envision u_l to be a function of the following factors: (a) Normalized speed of the slowest vehicle (since traffic on a lane eventually moves at speed of the slowest vehicle) (b) Normalized average speed (many times a vehicle (or vehicles) might be temporarily slow since it is just about to change a lane or near an intersection i.e., give some weight to average rather than decide entirely on temporary phenomenon) and (c) Normalized free space (since not all vehicles may be able to switch lanes immediately after a clear lane message is received). Here, normalized lowest speed is calculated as (speed/maximum possible speed) and is denoted by $\frac{a}{m}$. Normalized average speed, $\frac{b}{m}$, is calculated as (average speed/maximum possible speed). Here, maximum possible speed, is the speed limit of the road (different lanes can have additional speed restrictions). Normalized free space is an approximation computed as $\frac{n-c}{n}$, where c is the number of vehicles present on the lane l upto distance c_d, and n is the maximum number of vehicles that can be on the lane upto c_d. (n-c) therefore represents the number of vehicles that can be added in the free space available upto c_d. To compute n we assume an average length for vehicles (l_v) which makes the computation an approximation. Combining the terms:

$$u_l = w_a * \frac{a}{m} + w_b * \frac{b}{m} + w_c * \frac{n-c}{n} \quad (1)$$

where w_a, w_b, w_c are the weights of each of the terms. At the beginning of the simulation an EV starts on the lane with maximum value of u_l. Utilities are recomputed every t seconds and lane changes happen when the utility of the best lane u_b, exceeds the utility of the current lane u_c, by at least δ to compensate for lane switching overheads:

$$u_b - u_c > \delta \text{ (} Condition\ for\ lane\ change \text{)}$$

Summarizing the notation used so far:

- a - Lowest speed of a vehicle on lane l within the communication distance, c_d
- b - Average speed of vehicles on lane l (within communication distance c_d)
- c - Number of vehicles on lane l (within distance c_d)
- w_a, w_b, w_c are the weight-age of the factors a,b and c respectively where $0 <= w_a, w_b, w_c <= 1$ and $w_a + w_b + w_c = 1$
- m - Maximum speed limit of the road (Different lanes may have different speed limits)
- n - Maximum number of vehicles possible within distance c_d, $n = \frac{c_d}{m_v + l_v}$, where m_v is minimum gap between vehicles and l_v is length of a vehicle

- If there are no vehicles on the lane up to c_d, i.e., $c = 0$, then, $a = b = s_l$ (the speed limit of the lane).
- u_b, u_c - Utility of moving on the best lane and the current lane respectively
- δ - Minimum utility difference for lane switch to happen

Key steps in BLS : For brevity purposes we avoid presenting the entire algorithm but mention the key steps.
(a) Initially the EV computes utility of all the lanes using equation 1 and assigns current lane to the one with highest utility. Using appropriate values for w_a, w_b and w_c (Section IV-D), we weigh the different factors suitably to identify the lane with the highest utility (and hence the fastest traffic).
(b) Every t steps, the EV re-computes the utilities. If the difference between the maximum utility, u_b and the current lane utility, u_c exceeds δ, the EV changes its lane to the one with utility u_b. When the utility u_b is just slightly better, it may not actually be beneficial to change lanes. A value of $\delta = 0$, means that a lane change will take place whenever there is a lane with better utility than the current lane. However, changing a lane has some cost in terms of changing the traffic pattern that was used to compute the utilities of lanes e.g., deceleration of EV/other vehicles when making a lane change and other disturbances to traffic. This cost incurred due to the changed traffic pattern need not always be worth the advantage gained from increased utility due to being on a faster lane, hence we use an overhead cost,δ.
(c) Additionally, if lane clearing is allowed, it sends lane change requests (every t seconds) to vehicles on the current lane, that are within its communication distance, c_d.

The best case scenario for the EV is to be in a lane that is empty till distance c_d, i.e., $c = 0$. In this case, a and b both equal s_l. If $s_l = m$, $\frac{a}{m} = \frac{b}{m} = \frac{n-c}{n} = 1$, hence, $u_l = w_a + w_b + w_c = 1$, the maximum utility possible for a lane. If $s_l \neq m$, i.e., when different lanes have different speed limits, then $\frac{a}{m} = \frac{b}{m} = \frac{s_l}{m}$. Thus, different lanes will have different utilities depending on the values of s_l.

While FLS may appear to be a strict subset of BLS it should be noted that, BLS uses only locally available information from V2V communications to compute the best lanes while FLS uses historical information to pick the fastest lane for the entire journey. Also, in some of the environmental settings described later, the simulation setup results in much higher average speeds in the leftmost lane, giving FLS a significant advantage by design over BLS.

IV. Experimental Setup

We present here our experimental set-up for testing the EV strategies in a variety of environmental settings and parameters using the SUMO simulator. Our simulations are run on an environment that is common on many roads of heavily populated cities like Hong Kong, Manila, Mumbai, Dhaka and Seoul. Road segments in such cities have multiple lanes that are quite crowded and are in general 1 km to 10 km in length, at a stretch, before hitting an intersection. Thus, unless stated otherwise, for all the experiments shown in this paper, we use a 2 km one way stretch of road with 4 lanes (varying the number of lanes between 2-6 did not cause significant difference in the results). We tested with several traffic patterns on the road and our results are reproducible in all reasonably long road segments ($>$ 500 meters).

Modeling Communication : We simulate a V2V single hop communication model for our experiments. In particular, the EV can get the position and speed of a vehicle in any lane up to a fixed distance, c_d via V2V communication. It can also send lane change requests upto c_d. We model communication delay i.e., the time taken by the EV's request to reach a vehicle, using the variable c_j for the j^{th} request. We fix c_j's to 1 second for all the requests unless stated otherwise (as most realistic delays are less than 1 second).

Vehicular parameters : For each vehicle i, we have the parameters v_i, a_i and d_i - The maximum speed, acceleration and deceleration of the vehicle respectively in addition to other parameters we introduce later.

A. Modeling Human behavior

To make strategies work with autonomous vehicles and human drivers, we model the following parameters in SUMO.

1) v_{i_p}: The preferred maximum speed of drivers, for each vehicle i, is set as $v_{i\_p} <= v_i$ (A driver can't go faster than the vehicle maximum speed, hence the upper bound). For an autonomous vehicle we set $v_{i\_p} = v_i$.

2) s_{dev}: Speed deviation (speedDev in SUMO) models the deviation of vehicles from lane speed limits. The maximum allowed speed of vehicles, v_{i_mas} is generated using a normal distribution with a mean of s_l (the speed limit of the lane) and a standard deviation of $s_l * s_{dev}$ with $0 <= s_{dev} <= 1$. Typically, $s_{dev} = 0$ for autonomous vehicles i.e., they follow the lane speed limits while > 0 for humans (or a mix). Humans are also restricted by their own preferred maximum speed, v_{i_p}. Thus, the maximum speed of i^{th} vehicle is the minimum of v_{i_mas} and v_{i_p}.

3) σ: SUMO parameter to model driver imperfection. Vehicles do not always accelerate or decelerate at their maximum possible values depending on the driver at wheel. Using the car following models in SUMO, setting $\sigma > 0$ results in acceleration to be less than a_i. Precisely, the model sets acceleration at each time step for every vehicle, as $a_i - (\sigma * a_i * rand())$, where $rand()$ generates a random value between 0 and 1. For an autonomous vehicle σ is set to 0 which means the vehicle can accelerate at the maximum possible value a_i. Similar structure exists for deceleration.

Our experiments also use a parameter, re-computation time, t, which is the time interval after which lane change requests can be sent again. In BLS, t also represents the time interval after which the utility of each lane, u_l, is recomputed for taking a lane change decision. All the parameters and their default values are summarized in Table I, which we arrived at via domain knowledge and experimentation.

B. Simulation traffic distribution

New vehicles get introduced in our simulation any time before 250 seconds (i.e., between 1 and 250 seconds) and the start times for these new vehicles are generated randomly using a uniform distribution. The distribution is such that

Parameter	Description	Defaults
v_i	Vehicle maximum speed	120 km/h
v_{i_p}	Preferred maximum speed	variable
a_i	Vehicle acceleration	0.8 m/s^2
d_i	Vehicle deceleration	4.5 m/s^2
c_j	Request delay	1 second
s_{dev}	Speed deviation	$0, 0.2$
σ	Driver imperfection	$0, 0.5$
w_a	Lowest speed weight-age	0.4
w_b	Average speed weight-age	0.4
w_c	Free space weight-age	0.2
δ	Min utility difference between lanes	0.10
t	Recomputation Interval	10 seconds
c_d	Communication Distance	100 meters

TABLE I

EXPERIMENTAL PARAMETERS

a new vehicle enters the simulation every second with a 60% probability. This corresponds to 60 vehicles entering the simulation, every hundred seconds, on average. The EV enters the simulation at 200 seconds. The simulation run ends when all the vehicles that had entered during the 250 second period reach the end of the road segment.

C. Environmental Settings

For experimentation purposes, we modeled the following different environmental settings.

1) Setting 1: In this setting, we try to model an idealistic traffic pattern by setting all the vehicles to have the same preferred maximum speed, v_{i_p} of 90 km/h. The speed limit of all lanes is 60 km/h. The speed deviation factor, s_{dev}, is set to 0 and the driver imperfection factor σ is also set to 0. Hence, we expect a behavior where vehicles enter the road and travel at the lane speed limits throughout the journey. This is a possible scenario, when all the vehicles on the road are autonomous vehicles of the same kind, traveling without any obstructions or road damages.

2) Setting 2: In this setting, we try to add realism by: (a) Typically all the vehicles on a dense road do not travel at same speed. Hence, using a uniform distribution, we generate v_{i_p} of vehicles to vary from 20 to 60 km/h. The lane speed limits are 60 km/h here. (b) While speed deviation, s_{dev} is set to 0 which is an idealistic behavior we do model driver imperfections by setting σ to 0.5. To sum up, we expect this to be more realistic than Setting 1, due to the distribution over maximum preferred speeds and σ parameter.

3) Setting 3: We try to make this setting closest to most roads in densely populated cities, by adding the following behaviors: (a) As in Setting 2, v_{i_p} of vehicles varies from 20 to 60 km/h. (b) We model deviation from lane speed limits by setting s_{dev} to 0.2. (c) Driver imperfections are modeled by setting σ to 0.5. (d) To model congestion, we put limits on traffic speeds in different lanes by using a Gaussian distribution for lane speed limits with mean = 60 km/h and standard deviation = 30 km/h. We expect this to result in traffic speeds that vary between very low sometimes (as happens during congestion) to slow moving traffic range most times. (e) Modeling different speed limits for different lanes as in (d), also introduces significant planning uncertainties since vehicles would not know beforehand the traffic patterns that may arise. As described earlier, the maximum speed of a vehicle is derived as $min(v_{i\_mas}, v_{i\_p})$ where v_{i_mas} is generated using lane speed from distribution in (d) and s_{dev}.

4) Setting 4: In this setting, we model the traffic patterns corresponding to cities with relatively faster moving traffic like New York City. We test here on a wider range of lane speeds than setting 3, modeled using actual traffic speeds of a few roads. In particular, we use the data available from the City of New York Department of Transportation(NYCDOT) [26] to perform the modeling. More details in Section IV-E.

5) Setting 5: In this setting, we do not allow communication of lane change requests. Other than this all other parameters are same as Setting 2. This setting represents the current situation where typically the system does not have the infrastructure to facilitate V2V communication for sending lane change requests. Even if communication is allowed, this represents the situation for (lower priority) vehicles such as cars, buses etc., not having the permission to clear a road.

D. Deciding on weights in BLS

As described in Section III-C, weights in BLS are very important to determine the utility of different lanes. We therefore performed experiments using several values of w_a, w_b and w_c which combined with domain knowledge led to default weights of 0.4, 0.4 and 0.2, as they resulted in best traversal time for EV in most environmental settings. Although these weights are suitable for most scenarios, a different set of weights maybe used by the EV in some radically different road settings (e.g., w_c could be set to 0 for a road which is always very sparse, causing the effect of free space to be insignificant). We verified the robustness of solution for the weights we picked, by perturbing the weights by (-5%,+5%) from defaults. The maximum difference in EV traversal times, on average, was 1.64 seconds or 0.704%.

E. Mapping NYCDOT data to Setting 4

The NYCDOT data feed that we use to model Setting 4, contains real-time traffic information from sensor feeds, mostly from major arterials and highways of New York City (NYC). This data feed is updated every minute for each road over a total of 153 roads. We developed our model using the following: (a) We collected data for about a week (9670 minutes) from the real-time data feed (b) For any road we simulate, we use a segment of 2 kms, irrespective of the actual length of the road segment. (c) For (a few) roads with varying number of lanes, we use the maximum number of lanes for the entire length we simulate. (d) We included 151 roads having $>= 2$ lanes each (2 single lane roads were excluded). (e) The road speed data we have is converted into individual lane speed using the procedure described below that would capture the salient features of NYC traffic.

Procedure to calibrate Setting 4 for NYC road speeds:
Modeling groups: We first classify roads into groups based on number of lanes in the road. Hence all roads with 2 lanes are classified into one group g_2, roads with 3 lanes into group g_3 and so on till g_6 (maximum number of lanes for any road in the data). We therefore obtain 5 groups in total $g_2 - g_6$.

Modeling buckets: Within each group g_j, the average road speeds are classified into buckets using intervals of 5 m/s (18 km/h) each. Therefore we have buckets b_{1_j} for 0-5 m/s, b_{2_j} for 5-10 m/s till b_{10_j} for 45-50 m/s. The average speed for each bucket, $bavs_{i\_j}$, is taken as the mean of the bucket. For example, for the bucket 0-5 m/s, the average bucket speed, $bavs_{1\_j}$ is considered 2.5 m/s.

Obtaining weights for buckets: We then obtain the weights for each bucket in the following fashion: For each road, we have 9670 data points corresponding to average speed of the road at that minute. Each of these 9670 points are then classified into buckets depending on the speed the data point represents. The weight of a bucket is incremented by 1 for each data point that falls under this bucket. For example, if the average speed of a road (with j lanes) is 0-5 m/s for 500 minutes (out of the 9670 minutes) then we increment the weight w_{1_j} by 500. This procedure is repeated for all the roads to obtain the total weight for each bucket.

The SUMO simulation: Each b_{i_j} has an average bucket speed $bavs_{i\_j}$ and a weight w_{i_j}. $bavs_{i\_j}$ is taken as the average road speed in simulation. To simulate lanes, $bavs_{i\_j}$ is converted into lane speeds: Each lane is set a maximum speed, picked using a uniform distribution between the average road speed $\pm$ 40%. Hence, the mean of maximum speed across lanes is the average road speed on expectation. We then let the EV traverse on a 2 km stretch of road in the SUMO simulation and calculate it's run time. As mentioned in Section V, each setting is run for 100 times, hence the same lane will have different maximum speeds across the 100 runs and we obtain 100 different EV run times.

Summary of SUMO parameters used: (a) The number of lanes in this setting ranges from 2 to 6. (b) The maximum preferred speed of vehicles, v_{i_p}, varies from 60 to 120 km/h. (c) As in Setting 3, we model deviation of drivers from lane speed limits by setting s_{dev} to 0.2 and model driver imperfections by setting σ to 0.5. (d) All the other parameters were set to default values mentioned in Table I.

EV run time per bucket: We average the 100 different EV run times obtained to get the EV run time EVr_{i_j} for a bucket i and group j. It represents the time an EV would take (on average) if the number of lanes is j and the road speed corresponds to $bavs_{i\_j}$.

Computing mean run time: For each group j, we compute gr_j = EVr_{1_j} * w_{1_j} + EVr_{2_j} * w_{2_j} + $\ldots$ + EVr_{10_j} * w_{10_j}. We repeat this procedure to obtain $gr_2, gr_3, \ldots, gr_6$. Here $(gr_j)/(w_{1\_j} + \ldots + w_{10\_j})$ represents the average time an EV needs to travel a 2 km road with j lanes. The mean run time is obtained as $(gr_2 + \ldots + gr_6)/((w_{1\_2} + \ldots + w_{10\_2}) + (w_{1\_3} + \ldots + w_{10\_3}) + \ldots + (w_{1\_6}+\ldots+w_{10\_6}))$. This mean run time represents the average time an EV needs to cover a 2 km stretch of road with speeds corresponding to NYC roads and is used as run time for different strategies and experimental parameters for Setting 4.

F. Other Details

As described earlier, apart from the EV lane picking strategy we let SUMO control most of the traffic behavior (to generate realistic traffic patterns), except for the following changes/interventions: (a) In Settings 1,2,3 and 4 our strategies involve changing the lanes of some of the vehicles to implement lane change requests. (b) In Setting 3 and 4 we (also) modify lane speed limits to implement the varying speed limits for different lanes. Modification of behavior of vehicles and lane speed limits is done using the TraCI interface of SUMO [14]. Apart from these specific interventions, SUMO handles all the behaviors and we do not interfere in any way. Another detail related to experiments is that, we compare the FLS and BLS strategies against two baselines: One is the **"SUMO Strategy"** described earlier. The second is an **"Empty road baseline" (ERB)**, which is the time taken by the EV when there are 0 vehicles on the road apart from the EV for the entire simulation period. This acts as a lower bound for the EV traversal time.

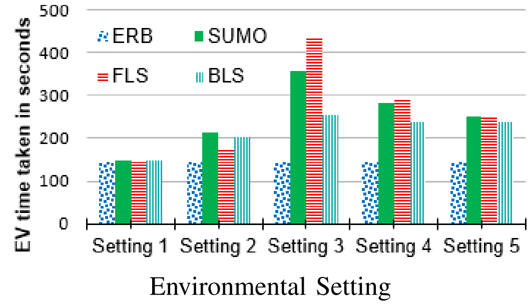

Fig. 1. Comparison of strategies

V. Experimental Results

All the results of our experiments presented in this section are an average over 100 runs. Each run has vehicles entering the simulation of a 2 km stretch of road with different starting times, generated using the description in Section IV-B.

A. Comparison between strategies

The first experiment studies how the four different strategies ERB, SUMO, FLS and BLS perform on each of the five settings described earlier. Results of this experiment are summarized in Figure 1. The figure shows the five settings on x-axis, and for each setting the four different strategies shown as bars. The y-axis represents the time taken in seconds by the EV to travel from start to destination using each of the strategies. Our results show that in Setting 1, all the strategies need similar time (as expected) as the empty road baseline (ERB) since vehicles are able to move at the lane speed limits. In Setting 2, FLS performs better than BLS on an average (12.39% faster EV travel time). Both of them perform significantly better than the SUMO Strategy (BLS is 6.18% faster while FLS is 17.81% faster). FLS is able to perform better here as the simulation setup leads to leftmost lane (chosen by FLS) being the fastest. The average speed of vehicles in the rightmost lane was 22.68 km/h while those in the leftmost lane was 29.88 km/h (31.7% more).

In Settings 3 and 5, BLS performs better than FLS (with 41.37% and 5.17% improvement) and the SUMO Strategy (with 29.44% and 6.13% improvement). Setting 3 allows deviation from lane speeds limits and has congestion, thus

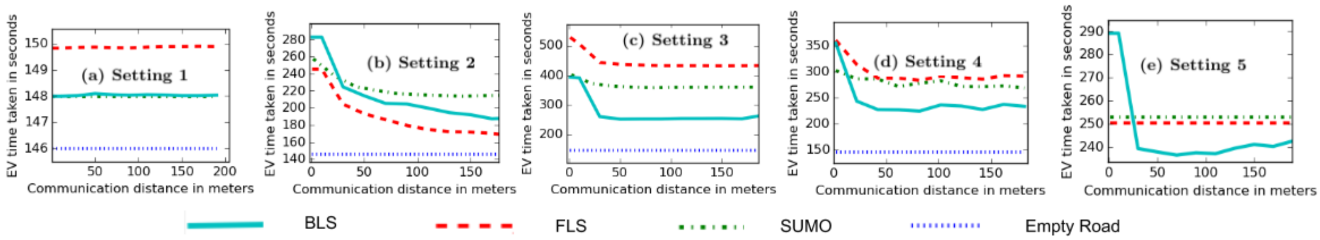

Fig. 2. Varying the communication distance, c_d

adding a lot of dynamicity to the traffic patterns. BLS handles it better as it uses the information obtained from V2V communication to perform a better optimization of its lane selection given the current traffic situation. On the other hand, FLS due to its static nature, is not able to handle this setting well and has a lower performance than even the SUMO strategy (SUMO strategy is 16.90% faster here).

In Setting 5 we do not allow clearance of vehicles. Here, BLS performs slightly better than FLS as it is able to find gaps on the road. In particular, BLS performs significantly better than FLS on less dense roads due to its ability to pick up the gaps (gaps have more utility) and about the same in empty roads as picking of gaps is not really needed. In this setting, both FLS and BLS perform a little better than the SUMO strategy (1.01% improvement and 6.13% improvement). The travel time trends for Setting 4 are similar to Setting 3. BLS performs significantly better than FLS and the SUMO strategy (with 18.90% and 16.42% improvement). FLS has a lower performance here too (SUMO strategy is 2.96% faster). Hence, we see that BLS performs better on the setting modeled using the New York city traffic data.

B. *Varying the communication distance, c_d*

Figure 2 plots the effect of changing the communication distance, c_d, in different environmental settings. Each sub figure of Figure 2, represents a setting and shows the different values picked for the parameter c_d on the x-axis (in meters) and the time taken by the EV on the y-axis (in seconds). As described earlier, c_d represents the distance upto which the EV can send lane change requests to vehicles in front of it (if allowed to). In addition, c_d also represents the distance in BLS upto which requests for information about other vehicles are sent to compute the utilities of lanes.

For Setting 1 in Figure 2-(a), there is no benefit from additional communication as vehicles are able to travel at the lane speed limits (and cannot go any faster). For settings 2,3 and 4, FLS, BLS and the SUMO strategies gain advantage from an increase in c_d till a certain threshold value, with maximum benefit gained by BLS and least by SUMO. We believe this happens at a lower threshold for Setting 3 because: There is lesser relevant information to be gained by communicating with farther vehicles, since due to congestion, there are more vehicles on the road within relatively short distances. BLS gains maximum benefit with increase in c_d since apart from sending lane change requests to farther vehicles, the EV also uses the information about speed and position of other vehicles, to compute better lanes.

Setting 4 has similar travel time trends as Setting 3 except that the run time difference between the FLS and SUMO strategies is less and becomes close to zero for higher c_d due to lesser congestion in Setting 4 resulting in more free space for lane clearance (in FLS). In Setting 5, lane clearing is not allowed. Since FLS and SUMO strategies use communication only for lane clearance, c_d has no effect here. BLS uses communication for utility computation too. Hence, for small values of c_d (less than 30 meters), the EV performance suffers as there is information about too few vehicles to analyze the lane dynamics. However, $c_d > 100$ meters does not lead to much better utility computation since vehicles further away may not add much information.

C. *Other experiments*

For completeness of analysis we performed several other experiments which we summarize here for brevity purposes.

1) Effect of varying the re-computation interval t: Parameter t specifies the time interval after which lane change requests to other vehicles are sent again (if clearance is allowed) for FLS and BLS. For BLS, it also specifies the time interval after which the utilities are recomputed to decide if the EV should change its lane. We varied t from 0-80 seconds and found that, it is beneficial to increase t till around 10-15 seconds. A value of $t < 10$ seconds causes frequent utility computations and lane changes, resulting in high traversal time overheads (along with frequent clearing of vehicles and its associated communication overheads) while for $t >$ 10-15 seconds, increasing t results in decreasing performance as the dynamicity of the traffic situation may go unaccounted for too long. Thus, $t = 10$ seconds is set as the default.

2) Effect of vehicles not following lane change requests: In Settings 1,2,3 and 4 vehicles always follow lane change requests while in Setting 5 they never follow. We study the intermediate case where vehicles may or may not accept lane change requests. The probability of vehicles following lane change requests p_{lcr}, was varied from 0 (never follow) to 1 (always follow). Within a run, a particular vehicle will always either follow or reject the lane change request (i.e., a probability of 0.3 denotes that 30% of vehicles picked randomly will always follow lanes change requests while 70% will always ignore them). We found that the EV run times decrease with increase in p_{lcr}, with BLS significantly outperforming FLS and SUMO. The only time BLS gets outperformed is by FLS for Setting 2 when $p_{lcr} >= .8$ due to leftmost lane being fastest which shows the ability of BLS to quickly adjust when vehicles do not follow requests.

3) Effect of Communication Delay, c_j: Default value of c_j is 1 second. However, for this experiment we generated c_j's using a uniform distribution ranging from 0-6 seconds in case 1 and 0-60 seconds for case 2 and tested FLS and BLS for Setting 3. To summarize, BLS is affected lot more in both cases via large delays since the information that it uses for computation of best lanes may get outdated due to continuous change in traffic patterns.

4) Effect of EV on other vehicles: Summarizing the result for Setting 2 (similar trends in Settings 3 and 4): In BLS, on an average, 56.64 vehicles were affected out of 153.3 vehicles in the simulation causing an average delay of 1.36 seconds per vehicle (0.50%). In FLS, 41.88 vehicles are affected with 1.00 second (0.37%) of average delay per vehicle. Overall, the total delay across all the vehicles in a run, caused by BLS on an average, is 77.03 seconds while FLS causes 41.88 seconds of delay. Thus, lane level optimizations do not seem to cause significant disruption to other vehicles in both FLS or BLS.

VI. Conclusions and future work

In this paper, we introduced the FLS and BLS strategies to improve the travel time of EVs by modeling lane level dynamics. In our experiments (performed using the well-known traffic simulator SUMO), we simulated a variety of traffic patterns and human behaviors by varying a wide range of parameter settings. We could show that: (a) BLS performs better than SUMO strategy on all the traffic settings we tested. (b) BLS performs better than FLS in circumstances involving congestion and uncertainties while FLS performs better in well-behaved traffic conditions and (c) BLS is the best strategy for the setting calibrated using real world data obtained from NYCDOT.

We believe this work can be expanded in a variety of ways in future: (a) The EV can communicate only to a single hop distance currently. We expect addition of multi-hop communication to provide further significant run-time improvements. (b) We presented here an experiment which analyzes how the different strategies perform when drivers do not follow lane change requests. However, the behavior is not intrinsic to the driver but the drivers have been randomly picked to reject requests. We therefore plan to perform a more realistic modeling of human behaviors (e.g., attribute intrinsic behaviors to drivers). (c) Include the effect of intelligent lane picking on traversal through intersections too (currently limited to straight sections of roads).

References

[1] P. Papadimitratos, A. La Fortelle, K. Evenssen, R. Brignolo, and S. Cosenza, "Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation," *Communications Magazine, IEEE*, vol. 47, no. 11, pp. 84–95, 2009.

[2] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (vanets): status, results, and challenges," *Telecommunication Systems*, vol. 50, no. 4, pp. 217–241, 2012.

[3] F. J. Martinez, C.-K. Toh, J.-C. Cano, C. T. Calafate, and P. Manzoni, "Emergency services in future intelligent transportation systems based on vehicular communication networks," *Intelligent Transportation Systems Magazine, IEEE*, vol. 2, no. 2, pp. 6–20, 2010.

[4] M. L. Sichitiu and M. Kihl, "Inter-vehicle communication systems: a survey," *Communications Surveys & Tutorials, IEEE*, vol. 10, no. 2, pp. 88–105, 2008.

[5] T. Telegraph, "Figures expose ambulance delays in life and death calls," http://www.telegraph.co.uk/news/health/news/10962298/Figures-expose-ambulance-delays-in-life-and-death-calls.html, 2015.

[6] T. B. B. Corporation, "Ambulance response times in wales are 'worst on record'," http://www.bbc.com/news/uk-wales-31008978, 2015.

[7] N. Cetin, A. Burri, and K. Nagel, "A large-scale agent-based traffic microsimulation based on queue model," in *IN PROCEEDINGS OF SWISS TRANSPORT RESEARCH CONFERENCE (STRC), MONTE VERITA, CH*. Citeseer, 2003.

[8] M. Balmer, N. Cetin, K. Nagel, and B. Raney, "Towards truly agent-based traffic and mobility simulations," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*. IEEE Computer Society, 2004, pp. 60–67.

[9] Q. Yang, H. Koutsopoulos, and M. Ben-Akiva, "Simulation laboratory for evaluating dynamic traffic management systems," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1710, pp. 122–130, 2000.

[10] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *Journal of artificial intelligence research*, pp. 591–656, 2008.

[11] A. L. Bazzan, "A distributed approach for coordination of traffic signal agents," *Autonomous Agents and Multi-Agent Systems*, vol. 10, no. 1, pp. 131–164, 2005.

[12] P. Paruchuri, A. R. Pullalarevu, and K. Karlapalem, "Multi agent simulation of unorganized traffic," in *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*. ACM, 2002, pp. 176–183.

[13] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of sumo–simulation of urban mobility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, 2012.

[14] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, "Traci: an interface for coupling road traffic and network simulators," in *Proceedings of the 11th communications and networking simulation symposium*. ACM, 2008, pp. 155–163.

[15] R. Vlad, C. Morel, J.-Y. Morel, and S. Vlad, "A learning real-time routing system for emergency vehicles," in *IEEE International Conference on Automation, Quality and Testing, Robotics, 2008. AQTR 2008.*, vol. 3. IEEE, 2008, pp. 390–395.

[16] H. K. Gedawy, M. Dias, and K. Harras, "Dynamic path planning and traffic light coordination for emergency vehicle routing," *Comp. Sci. Dept., Carnegie Mellon Univ., Pittsburgh, Pennsylvania, USA*, 2008.

[17] C. A. Jordan and M. Cetin, "Signal preemption strategy for emergency vehicles using vehicle to infrastructure communication," in *Transportation Research Board 94th Annual Meeting*, no. 15-5723, 2015.

[18] A. Buchenscheit, F. Schaub, F. Kargl, and M. Weber, "A vanet-based emergency vehicle warning system," in *Vehicular Networking Conference (VNC), 2009 IEEE*. IEEE, 2009, pp. 1–8.

[19] M. Rahman, M. Chowdhury, Y. Xie, and Y. He, "Review of microscopic lane-changing models and future research opportunities," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, no. 4, pp. 1942–1956, 2013.

[20] J. Erdmann, "1 lane-changing model in sumo," 2014.

[21] P. Hidas, "Modelling lane changing and merging in microscopic traffic simulation," *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 5, pp. 351–371, 2002.

[22] P. G. Gipps, "A model for the structure of lane-changing decisions," *Transportation Research Part B: Methodological*, vol. 20, no. 5, pp. 403–414, 1986.

[23] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model mobil for car-following models," *Transportation Research Record: Journal of the Transportation Research Board*, 2015.

[24] W. Schakel, V. Knoop, and B. van Arem, "Integrated lane change model with relaxation and synchronization," *Transportation Research Record: Journal of the Transportation Research Board*, 2012.

[25] M. Kamal, S. Taguchi, and T. Yoshimura, "Efficient vehicle driving on multi-lane roads using model predictive control under a connected vehicle environment," in *Intelligent Vehicles Symposium (IV), 2015 IEEE*. IEEE, 2015, pp. 736–741.

[26] C. of New York Department of Transportation, "Real-time traffic speed data," https://data.cityofnewyork.us/Transportation/Real-Time-Traffic-Speed-Data/xsat-x5sa, Oct. 2015.