Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 3727865, 11 pages
http://dx.doi.org/10.1155/2016/3727865

Research Article

Dynamic Vehicular Route Guidance Using Traffic Prediction Information

Kwangsoo Kim,[1] Minseok Kwon,[2] Jaegeun Park,[3] and Yongsoon Eun[3]

[1] *Hanbat National University, Daejeon 34158, Republic of Korea*
[2] *The Rochester Institute of Technology, Rochester, NY, USA*
[3] *DGIST, Daegu 42988, Republic of Korea*

Correspondence should be addressed to Yongsoon Eun; yeun@dgist.ac.kr

Received 10 December 2015; Revised 8 April 2016; Accepted 31 May 2016

Academic Editor: Qixin Wang

We propose a dynamic vehicular routing algorithm with traffic prediction for improved routing performance. The primary idea of our algorithm is to use real-time as well as predictive traffic information provided by a central routing controller. In order to evaluate the performance, we develop a microtraffic simulator that provides road networks created from real maps, routing algorithms, and vehicles that travel from origins to destinations depending on traffic conditions. The performance is evaluated by newly defined metric that reveals travel time distributions more accurately than a commonly used metric of mean travel time. Our simulation results show that our dynamic routing algorithm with prediction outperforms both Static and Dynamic without prediction routing algorithms under various traffic conditions and road configurations. We also include traffic scenarios where not all vehicles comply with our dynamic routing with prediction strategy, and the results suggest that more than half the benefit of the new routing algorithm is realized even when only 30% of the vehicles comply.

1. Introduction

Recent data show that traffic conditions in metropolitan areas continue to worsen with increased wasted hours, extra fuel cost, and travel unreliability, for example, the extra time needed to arrive at destinations [1]. Intelligent Transportation Systems (ITS) attempt to solve this problem by exploiting the advances in information technology, for example, dynamically controlling traffic lights based on traffic conditions and routing vehicles using current and historical traffic information.

Intelligent Transportation System is one of the major applications of the Internet of Things (IoT) technology, which measures traffic conditions using various sensors including cameras, loop detectors, and mobile devices, and collects traffic-related data through communication channels and then shares the data among vehicles on the road and road facilities, for example, traffic lights and ramp meterings. As several commercial platforms supporting the IoT architecture have come into the market [2, 3] and wireless communication technologies have been advanced, a car has become a sensor platform and an essential element of Internet of Vehicles, a representative instantiation of the IoT [4]. The technology development and large-scale employment of the IoT introduce new services and also make the existing services more reliable and more sophisticated. One example is intelligent vehicle navigation service which computes the least travel time path or the most economical path based on real-time traffic information rather than the traditional shortest path.

Specifically, we consider in this paper the scenario where current traffic information is collected by the central vehicular routing system using in-car navigation systems, traffic sensors deployed in the road network, and other applications. The routing system then computes the shortest path for a given origin-destination (OD) pair at the request of an individual vehicle and sends the route to each vehicle. Such a system may reroute the path as updates of traffic conditions continue to become available [5, 6].

For such a central vehicular routing system to operate efficiently, it is critical to incorporate predictive traffic information. Otherwise, traffic routing would suffer from the instability problem; that is, a road segment with little traffic can quickly turn into a heavily congested segment as many vehicles are routed or rerouted through this road segment at the same time. This is because the system computes for each vehicle its shortest path independently using the current traffic condition without taking into account other vehicles routed concurrently. In other words, instability arises primarily because every vehicle responds to the same traffic conditions with lack of knowledge about other vehicles.

In this work, we show that predictive traffic information is important to improve road efficiency. We develop a routing algorithm called Dynamic with prediction that periodically reroutes all vehicles based on current and predictive traffic conditions. The central routing system collects all the routing requests from the vehicles arriving at any nodes (intersections) and randomizes the order of priority of the requests. From the highest priority order to the lowest, the central routing system computes new routes considering the real-time traffic information and anticipatory traffic changes on the links which will be occupied by the vehicles with the higher priorities. We compare Dynamic with prediction with two other routing strategies, namely, Static and Dynamic. In the Static routing, vehicles follow the routes computed initially without changes, and in Dynamic, vehicles are rerouted periodically like Dynamic with prediction, but only using the current traffic conditions.

We develop a microlevel traffic simulator and use it to evaluate the performance of the Dynamic with prediction routing. The simulator consists of a map creator, a traffic generator, a routing controller, a vehicle simulator, and a performance evaluator. Although several microtraffic simulators are available commercially or for research purposes (e.g., VISSIM [7], VISSUM, and CORSIM [8]), they are not suitable to test our routing algorithms mainly due to their rigid routing policy. We use OpenStreetMap [9] to create a realistic map and parse the map to extract the information that we need for simulation.

Mean travel time, which is the average time taken for all vehicles to move from the origin to the destination, is popularly used for performance evaluation [10, 11]. This metric, however, does not reveal travel time variations effectively that may be more important for individual driver experience. For example, mean travel time can be similar in two vastly different cases with respect to travel time distributions. We develop a new metric that captures the travel time distribution of entire vehicles and evaluate the performance based on this metric. Our results indicate that Dynamic with prediction effectively reduces travel time in comparison to the Static and Dynamic routings under a variety of traffic conditions.

The rest of this paper is organized as follows. Section 2 formulates our problem and discusses related work. Section 3 discusses details of our simulator, and Section 4 presents the three routing algorithms that we develop and compare. Section 5 defines metrics for performance evaluation. Section 6 presents results and their analysis followed by conclusions and future work in Section 7.

2. Related Work

Vehicular route guidance deals with the problem of assigning an optimal path to each vehicle from its origin to the destination. The optimality in this problem is defined on several criteria, for example, the shortest path, the shortest time, and the least usage of local paths. The traditional routing algorithms embedded in vehicle navigation systems used only road network features and have evolved to consider real-time traffic information in part thanks to broadcasting networks such as DAB in Europe and DMB in Korea [12, 13]. Recent navigation software runs on mobile platforms and receives route updates periodically from a central telecommunication center over the cellular network [14, 15]. While both real-time traffic information and historical data are used to compute those route updates, the effects of individual routing decisions on overall system stability are not considered and well studied.

In the literature, there have been attempts to design route guidance strategies that effectively find shortest paths for given OD pairs while achieving stability when road networks are large and dynamically change. Claes et al. [10] develop a decentralized routing guidance system for anticipatory vehicle routing in which vehicles are routed based on current as well as forecast traffic information. Their system is modeled after the food foraging mechanism in ant colonies using pheromones. Multiagents are deployed in vehicles, infrastructure, and the central server, and they collect and communicate traffic information to compute the best path for a given OD pair. Most literature on dynamic routing systems did not show the details of how traffic prediction is obtained or how routing algorithm deals with traffic predictions. There are some dynamic routing algorithms receiving a traffic prediction as an input, but the prediction is based on a traffic flow model or traffic history [16, 17].

Most commonly used routing algorithms are Dijkstra's and A^* algorithms. One fundamental problem with them is their prohibitively high time complexity when the number of nodes (intersections) in the network increases. Jagadeesh et al. [18] use hierarchical routing to reduce this time complexity and propose a simple heuristic method that compensates the loss of accuracy in route quality, which is inevitable in hierarchical routing. Motivated by the same problem, Song and Wang [19] also use hierarchical routing but rather focus on scalability by reducing heavy precomputation, storage, and querying costs. They use recently discovered aspects of network topology, specifically hierarchical communities, to decompose the network and to design a heuristic for fast search. While lowering the computational complexity of routing algorithms is important, rerouting as the traffic conditions dynamically change is another critical challenge. There have been several attempts to tackle such a problem using different approaches including dynamic programming, genetic algorithms, and hierarchical routing [20–22]. Kim et al. show that dynamic route determination can be modeled as a Markov deception process and propose procedures for identifying unnecessary traffic data that can be removed for route decision making. Using their approach, we can selectively use only a subset of vast real-time traffic data in route selection; otherwise, dealing with all of the incoming data will be computationally challenging.

Forecasting traffic conditions has been investigated as an important problem in ITS research. Lv et al. and Huang et al. adopt a deep learning approach and architecture [23, 24] to predict traffic flows. Abadi et al. [25] propose a traffic flow prediction algorithm with current demands, historical data, and limited real-time data based on an estimated dynamic origin-destination matrix and simulations. Recent work on traffic flow prediction or forecasting can be found in [23, 26]. To the best of our knowledge, however, the traffic prediction methods used by the work in the literature do provide system-wide optimal route guidance.

3. Traffic Simulator

In this section, we discuss our approach for creating a map, running an experiment, and modeling roads and vehicle movement in the simulator.

3.1. Map Creation. The first step in a simulation is to create a map where a vehicle moves from the source to the destination passing through intersections. We consider the map as a graph $\mathscr{G} = (\mathscr{N}, \mathscr{E})$, where $\mathscr{N}$ is a set of nodes and $\mathscr{E}$ is a set of edges. A node n_i denotes an intersection where traffic light systems are in operation, and an edge e_{ij} denotes a road segment from n_i to n_j in the road network. Our simulator reads the map as an image and converts it into an adjacency linked list, which is a data structure popularly used to represent a graph. A major challenge is to automate this map creation process, so that the user does not need to manually create a complex map, which is tedious and time consuming; for example, the user needs to enter map information manually with CORSIM [8] and VISSIM [7]. We use OpenStreetMap [9] for this purpose. With the help of OpenStreetMap, we can grab an area of interest of the map image and convert that into an XML file. We then parse the XML file filtering unnecessary information (e.g., stores and gas stations) leaving only the information that we need such as intersections and road segments between intersections. We can extract the properties of each link and node including the number of lanes, the distance of a link, and even speed limits. After parsing, our preprocessor converts the simplified XML file into a target adjacency linked list. The preprocessing steps for map creation are summarized in Figure 1.

In our simulations, we use a real map topology instead of artificially made ones, namely, the Greater Rochester area in New York, the United States, as shown in Figure 2.

3.2. Simulator Structure. Our simulator runs on a discrete time basis, which is modeled as time tick. Events occur at each time tick such as vehicles movement, turning at intersections, recomputing their routes, changing their next move, departure from the origins, and arrival at the destinations. Of course, the time tick value that represents the wall clock time is parametrized to be changed according to the time scale in a simulation, for example, 1 tick as 1 second in our experiments. The overall simulator structure is illustrated in Algorithm 1, where $R(t)$ is the set of vehicles on the road at time t and $\mathbb{N}(R(t))$ is the number of elements in $R(t)$.

3.3. Modeling Roads and Vehicle Movement. An intersection and the street between two intersections are modeled as a node and a link in the graph, respectively. A vehicle arriving at the intersection can proceed in three directions; namely, it can turn left, go straight, and turn right. In the real world, the vehicle goes straight or turns left/right following the traffic lights, while no traffic light control exists in the simulator. Each direction maintains a queue where incoming vehicles await their turn. Since no traffic light control is provided, the vehicles in the queues can move to their next road segment at every tick. However, if the next road segment experiences traffic congestion, that is, there is no room for a vehicle to enter, the vehicle needs to wait leading to delay at the intersection. The delay in this case is the time elapsed from when the vehicle arrives at the queue to when the vehicle departs the intersection.

A road segment has three primary properties: distance, the number of lanes, and capacity. The capacity of the road segment is defined as the number of vehicles in transit and can be computed as the product of distance and the number of lanes. Different road segments have different capacities. For example, highways have high capacity mainly due to long distance, road segments in downtown in a metropolitan area have high capacity due to the high number of lanes, and local roads have low capacity due to short distance and the small number of lanes.

Several car-following models exist in the literature [27], and these models can be used to keep track of the position of each vehicle on the road. For simplicity, however, we use a vehicle moving rule instead based on traffic conditions. The velocity of a vehicle v on a road segment is determined by

$$v = \begin{cases} v_{\max} & \rho \le \rho_{\text{th}} \\ \dfrac{v_{\min} - v_{\max}}{1 - \rho_{\text{th}}} \left(\rho - \rho_{\text{th}}\right) + v_{\max} & \text{otherwise,} \end{cases} \quad (1)$$

where $v_{\max}$ and $v_{\min}$ are the speed limit and the minimum speed of the segment, respectively, ρ_{th} is a constant between 0 and 1, and ρ is the ratio between the number of current vehicles and the maximum number of vehicles that the road segment can accommodate denoted by $N_{\max}$, which is in turn computed as

$$N_{\max} = \frac{dL}{l_{\text{car}} + h} \quad (2)$$

in which d is the length of the road segment, L is the number of lanes, l_{car} is the average length of a vehicle, and h is the average headway. An example of a vehicle's velocity is shown in Figure 3.

4. Routing Controller

Using the map, roads, and vehicle movement defined earlier, we design a routing controller in this section. We first discuss how travel time is estimated and then provide details of our route guidance algorithms.

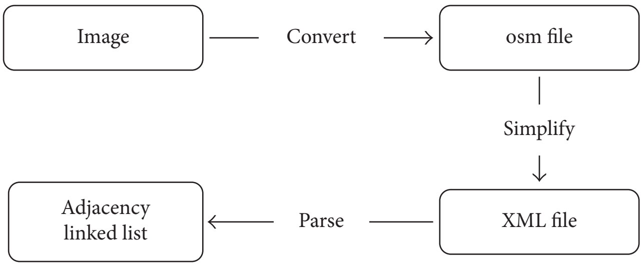


FIGURE 1: Preprocessing steps for map creation.

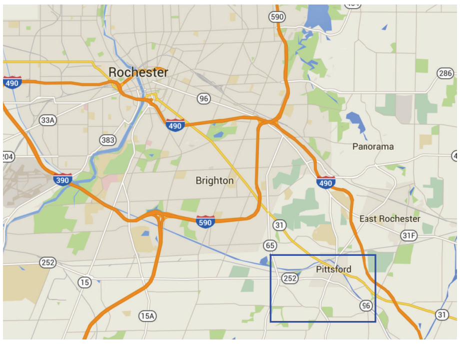


FIGURE 2: A map of the Great Rochester area in New York, the United States.

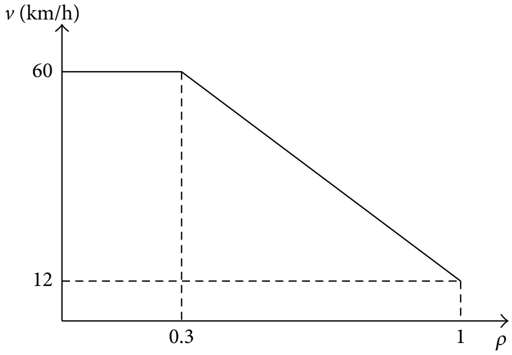


FIGURE 3: Example of a vehicle's velocity curve ($v_{\max} = 60$ km/h, $v_{\min} = 12$ km/h, and $\rho_{\text{th}} = 0.3$).

4.1. Travel Time Estimation. Two types of delay arise as a vehicle travels: (1) delay to travel from one intersection to another and (2) delay waiting at an intersection. The latter is hard to measure because it depends on traffic signals with probabilistic characteristics. In this research, we focus only on delay on road segments (the former, also known as link delay) for route guidance, and the latter will be dealt with in the future work. Specifically, we estimate delay d_{ij} on link e_{ij} as follows:

$$d_{ij} = \frac{l_{ij}}{v_{ij}}, \tag{3}$$

where l_{ij} is the length of e_{ij}, that is, the distance from n_i to n_j, and v_{ij} is the velocity of a vehicle on the link calculated by (1).

4.2. Route Guidance. With this estimated link delay, our route guidance mechanism directs each vehicle to its destination along a high-quality path. We assume that a central server possesses the entire map information including vehicle positions, delays at both intersections, and road segments and computes paths for all vehicles. Before a vehicle leaves from the origin, the vehicle sends its routing request to the server, which in turn computes the shortest path from the origin to the destination. The vehicle may be rerouted in the middle of travel. We use a well-known single-source shortest path algorithm like the Dijkstra or Bellman-Ford algorithms [28] for routing.

We formally define network state $X(t)$ as a column vector of all link delays. Let us consider a road network with three vertices, n_1, n_2, and n_3, and the corresponding edges e_{12}, e_{23}, and e_{13} as shown in Figure 4. Then the state at time t is $X(t) = [d_{12}(t)\ \ d_{21}(t)\ \ d_{23}(t)\ \ d_{32}(t)\ \ d_{13}(t)\ \ d_{31}(t)]^T$, where $d_{ij}(t)$ is the delay from n_i to n_j as a function of time representing the dynamic nature of the traffic network. The three routing strategies that we use and compare for route guidance are described in the following in detail.

(i) Static Routing. For a vehicle with an OD pair generated at time t, we find the shortest path from the origin to the destination using $X(t)$. This vehicle does not change the route until it reaches the destination.

(ii) Dynamic Routing. Unlike static routing, we reroute all vehicles periodically. When a vehicle arrives at any intersections at time t, it is rerouted based on the current traffic conditions which is represented by $X(t)$. This rerouting

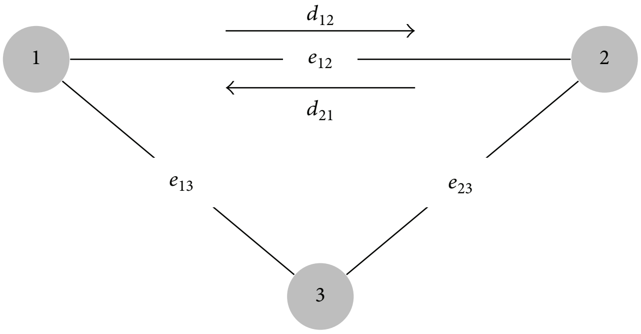


Figure 4: An example of simple road network and notations.

```
$t \leftarrow 0$ & Continue $\leftarrow$ TRUE
empty $R(t)$
$N \leftarrow$ Number of vehicles generated at each tick
while Continue = TRUE do
    if Vehicles to be generated remain then
        Generate $N$ vehicles
        Update $R(t)$ to include generated vehicles
        Assign OD pairs randomly for $N$ vehicles
        Get a route guidance for $N$ vehicles
    end if
    for $i \leftarrow 1, \mathbb{N}(R(t))$ do
        Update position
        if Arrive at the destination then
            Store time $t$ & Remove from $R(t)$
        else
            if Dynamic routing algorithm & Arrive at a node then
                Get a new direction for the next link
            end if
        end if
    end for
    if $R(t)$ is empty then Continue = FALSE
    end if
    $t \leftarrow t + \Delta t$
end while
```

Algorithm 1

repeats at every intersection through the journey until the vehicle reaches its destination.

(iii) Dynamic Routing with Prediction. One potential drawback of dynamic routing is its limited knowledge on future network states since it only utilizes current traffic information when rerouting vehicles. So, all vehicles arriving at the same intersection with the same destination will be guided into same links, which may lead to traffic jam on those links. One way to mitigate this problem is to consider the network states at time $t + k$, $k > 0$, when vehicles are routed at time t. In other words, dynamic routing is extended to incorporate predictive information. Unlike many other researches on dynamic routing algorithm in which traffic conditions are predicted statistically with a long time horizon, for example, 30 minutes or 1 hour, our proposed algorithm corresponds to the case when $k = 1$. Prediction over multiple ticks (which is proportional to the prediction time horizon) is not necessary because we reroute all vehicles at every tick. There is no guarantee that vehicles will travel as predicted after multiple ticks elapse. They will receive new routes calculated using updated traffic condition.

At time t, a priority order is randomly generated for each vehicle arriving at intersections. Let k be the number of vehicles to be rerouted at time t. We define an order function $I(\cdot)$ that maps integers 1 to k to randomly arranged k vehicle IDs. The vehicle with ID $I(1)$ is rerouted first, the vehicle with $I(2)$ is rerouted second, and then the vehicle with $I(3)$ is rerouted. The random function $I(\cdot)$ is different at every time t.

When the first vehicle is rerouted to one of the links connected to the current node, this link will be occupied by the vehicle at time $t + 1$. This prediction is taken into account when the second vehicle is rerouted. After the second

```
Update $X(t)$ & $C(t)$
$k \leftarrow$ the number of vehicles arriving at some nodes
$X_p(1) \leftarrow X(t)$
Create an order function $I$ with integers 1 to $n$ randomly arranged
for $i \leftarrow 1, k$ do
    Get a routing guidance for vehicle with $I(i)$ based on $X_p(i)$
    Update $X_p(i+1)$ including vehicle with $I(i)$'s route
end for
```

Pseudocode 1

rerouting is completed, another change is predicted on the link into which the second vehicle is guided. The change caused by the second vehicle as well as the first vehicle is taken into consideration when the third vehicle is rerouted, and this process continues until all k vehicles are rerouted. This means that the road segments which will be occupied by the already rerouted vehicles are penalized when the next vehicles are rerouted, so the penalized roads have lower chances to be selected by the vehicles rerouted later.

The pseudocode for dynamic routing with prediction is described in Pseudocode 1, where $C(t)$ is the set of all the vehicles arriving at some nodes, $X(t)$ is the state vector representing the current traffic condition, and $X_p(i)$ is a temporary state vector that takes into account the predictive future traffic condition caused by previous rerouted vehicles with $I(n), n = 1, \ldots, i-1$.

5. Performance Metrics

Most intelligent traffic control systems use the average time elapsed from departure to arrival for all vehicles. Specifically, the metric widely used is given by

$$\frac{1}{N}\sum_{i=1}^{N} T_m(i), \tag{4}$$

where N is the number of vehicles and $T_m(i)$ is the measured travel time of the ith vehicle. This measure, however, does not provide traffic conditions that individual vehicles experience accurately. For an example, assume that three vehicles traveled as shown in Table 1. The average travel time for three vehicles is 1 hour. However, Vehicle #1 experienced heavier traffic, compared with Vehicle #2 and Vehicle #3. It can be told that Vehicle #2 took a journey without any traffic interrupts. However, the metric like (4) does not inform such individual experiences.

Motivated by this drawback, we develop a new performance metric that evaluates traffic conditions more accurately. Our metric is defined as

$$\begin{gathered} M(\alpha) = \text{Percentage of the vehicles that satisfy} \\ T_m \le (1+\alpha) T_e, \end{gathered} \tag{5}$$

where T_m is the measured travel time of a vehicle, α is a parameter, and T_e is the expected shortest travel time of the vehicle, that is, the time taken by the vehicle to travel from the origin to its destination along the shortest route without any traffic interruptions.

Table 1: A simple example of vehicle travel time.

Vehicle ID	Estimated shortest travel time	Measured travel time
1	1/6 of an hour	1 hour
2	1 hour	1 hour
3	1/2 hour	1 hour

One interpretation of this metric is that the higher $M(\alpha)$ is, the better the performance is, for a fixed α. For example, if routing algorithm A gives a higher percentage of vehicles whose measured travel time is less than or equal to 120% of the expected shortest travel time than routing algorithm B does, then more vehicles by algorithm A than algorithm B finish their travels in the 20% extended time. Note that the conventional average travel time metric like (4) does not tell about such a distribution of travel time.

Of course, instead of the percentage value of the vehicles satisfying (5), we can compute α as the percentage values change. In such a case, for a fixed percentage value, smaller α values imply better performance. For instance, algorithm A performs better than algorithm B if algorithm A gives 50 vehicles satisfying (5) with $\alpha = 0.1$ and algorithm B gives 50 vehicles with $\alpha = 0.2$ for a total of 100 vehicles. In this case, the vehicles following algorithm A travel in less time than those following algorithm B for the same number of vehicles.

For ease of computation, we define $\beta(i)$ for a vehicle i as

$$\beta(i) = \frac{T_m(i)}{T_e(i)} - 1. \tag{6}$$

With $\beta(i)$, $M(\alpha)$ is now computed as

$$M(\alpha) = \frac{\mathbf{N}(\{i \mid \beta(i) \le \alpha\})}{\text{\# of vehicles routed}} \times 100. \tag{7}$$

In addition, we define

$$\beta_m = \frac{1}{N}\sum_{i=1}^{N} \beta(i). \tag{8}$$

Note that β_m is similar to the average travel time in (4) as large (small) β_m corresponds to large (small) average travel

Table 2: Comparison of $M(\alpha)$ values for the routing strategies when $\alpha = 0.1$, 0.2, 5, and 10 and vehicles are generated at rate = 1, 2, 5, 7, and 10 per sampling time.

Gen. rate	Algorithms	$M(0.1)$	$M(0.2)$	$M(5)$	$M(10)$
1	Static	74.32	98.85	100.00	100.00
	Dynamic	74.32	98.85	100.00	100.00
	Dynamic/prediction	73.01	97.20	99.99	100.00
2	Static	69.76	98.68	100.00	100.00
	Dynamic	69.76	98.68	100.00	100.00
	Dynamic/prediction	68.37	93.83	99.98	100.00
5	Static	52.14	67.59	84.10	95.51
	Dynamic	52.17	67.59	84.10	95.51
	Dynamic/prediction	53.75	74.54	99.19	99.95
7	Static	48.53	66.83	78.19	88.66
	Dynamic	48.60	67.01	78.18	88.64
	Dynamic/prediction	47.48	72.57	93.32	98.60
10	Static	38.52	55.90	75.325	84.65
	Dynamic	35.76	52.94	77.71	86.57
	Dynamic/prediction	34.26	51.25	91.20	96.68

time. Moreover, a large β_m suggests heavy traffic conditions during simulation, about which a simple average travel time does not give any information.

6. Simulation Results

We compare the performance of the routing strategies discussed in Section 4 using the developed traffic simulator. The results are analyzed based on the metrics proposed in Section 5.

6.1. Vehicle Generation Rates. We first examine which routing strategy minimizes $M(\alpha)$ for different vehicle generation rates. For this experiment, we use a part of the Rochester map with 337 intersections and a total of 20,000 vehicles generated. The vehicle generation rate varies from one to 10 every sampling time. The simulation begins when the first vehicle departs the origin and ends when all the vehicles generated arrive at their destinations. The results are summarized in Table 2. As an example, $M(0.1)$ when Dynamic with prediction is used is 53.75% when the vehicle generation rate is 5. This means that 53.75% of the vehicles arrive at the destinations with $\beta(i) \leq 0.1$, that is, less than 110% of the shortest expected travel time. The results show that the values of $M(\alpha)$ decrease in general as the vehicle generation rate becomes high (from 1 to 10), which indicates that more vehicles arrive at their destinations late. This is because more vehicles are added to traffic over a fixed duration as the rate increases.

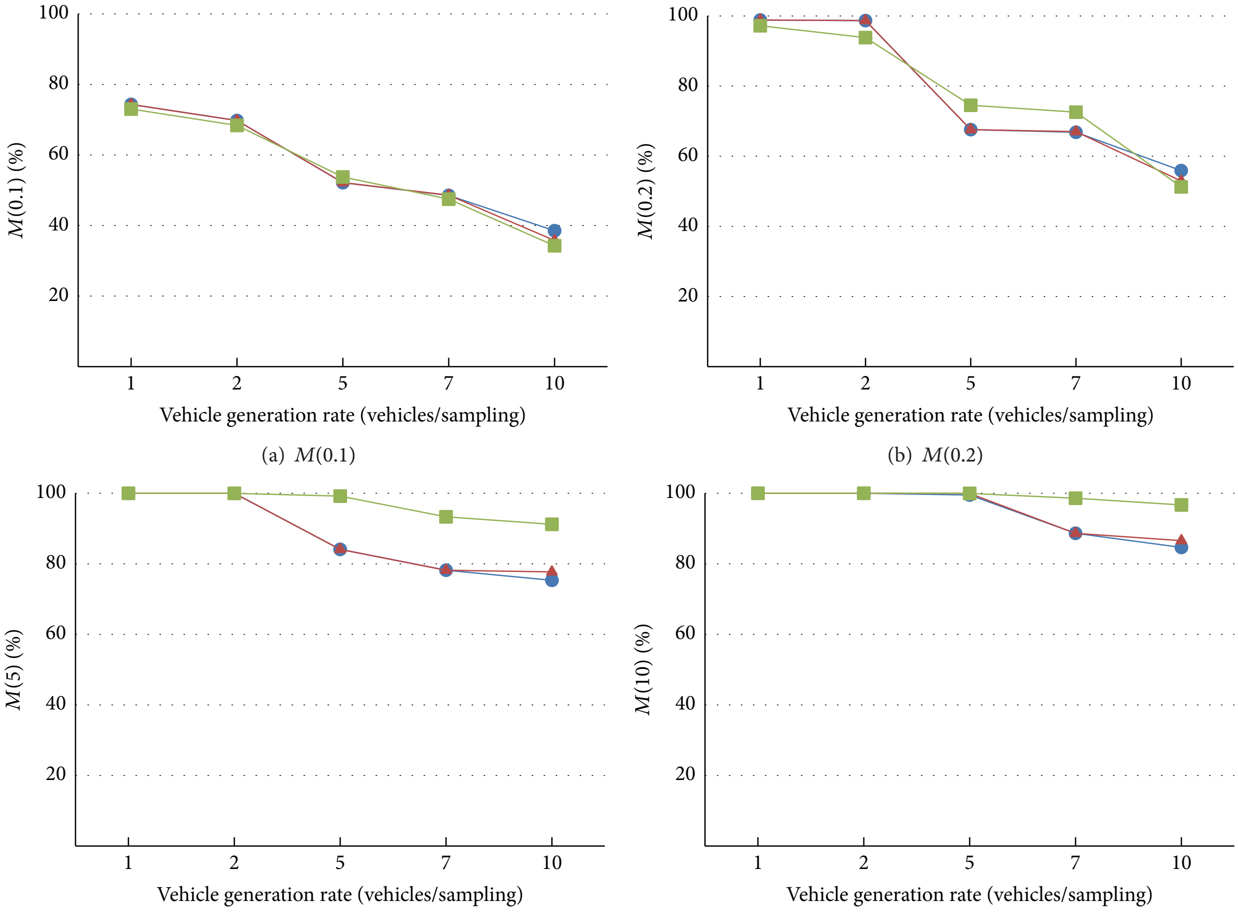
These results are also graphically depicted in Figure 5. Note that more vehicles arrive at their destination close to the shortest travel time with higher $M(\alpha)$ values. In the figure, all the algorithms perform similarly under light traffic conditions regardless of α, whereas Dynamic with prediction outperforms the other two under heavy traffic conditions, for example, when the vehicle generation rate is 7 or 10.

Table 3: β_m for Static, Dynamic, and Dynamic with prediction with different vehicle generation rates.

Generation rate	Algorithms	β_m
1	Static	0.087
	Dynamic	0.087
	Dynamic/prediction	0.102
2	Static	0.090
	Dynamic	0.090
	Dynamic/prediction	0.121
5	Static	1.929
	Dynamic	1.929
	Dynamic/prediction	0.455
7	Static	2.883
	Dynamic	2.888
	Dynamic/prediction	0.913
10	Static	3.707
	Dynamic	3.283
	Dynamic/prediction	1.450

In addition, we compute β_m (similar to average travel time) for the routing strategies and compare them under the same traffic conditions as Table 3. Like $M(\alpha)$, Dynamic with prediction shows the lowest β_m values under heavy traffic conditions, while all the three algorithms behave similarly when not many vehicles are on the road. With higher vehicle generation rates, β_m for the Static and Dynamic algorithms increases more abruptly than that of Dynamic with prediction, as shown in Figure 6. It can be interpreted as Dynamic with prediction results in low average travel time and low $M(\alpha)$ while being more robust to dynamically changing traffic conditions.

6.2. Map Topology. Our routing strategies are also tested with five different maps for more reliable performance evaluation.



(a) $M(0.1)$

(b) $M(0.2)$

(c) $M(5)$

(d) $M(10)$

Figure 5: $M(\alpha)$ under various vehicle generation rates: blue = Static, red = Dynamic, and green = Dynamic with prediction.

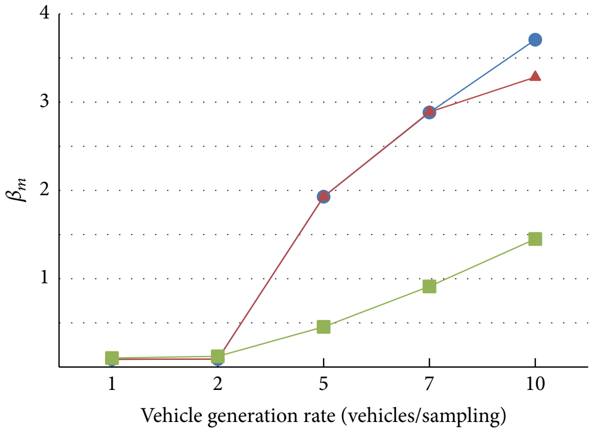
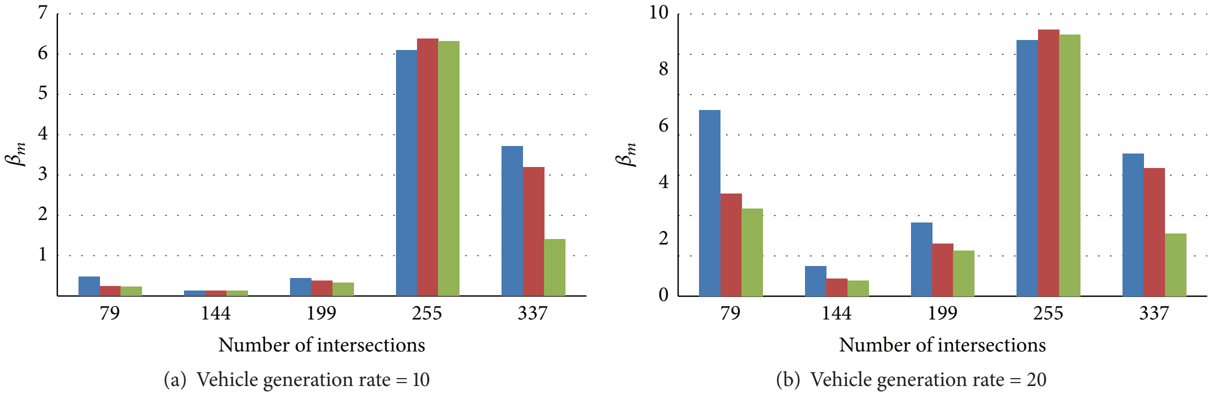


Figure 6: β_m under various vehicle generation rates (blue = Static, red = Dynamic, and green = Dynamic with prediction).

Specifically, the number of intersections in the maps is 79, 144, 199, 255, and 337, in which different areas of the Rochester map are included with various numbers of intersections. In Figure 7, the results (β_m) are plotted in both relatively light and heavy traffic cases, when the vehicle generation rate is 10 in Figure 7(a) and the rate is 20 in Figure 7(b).

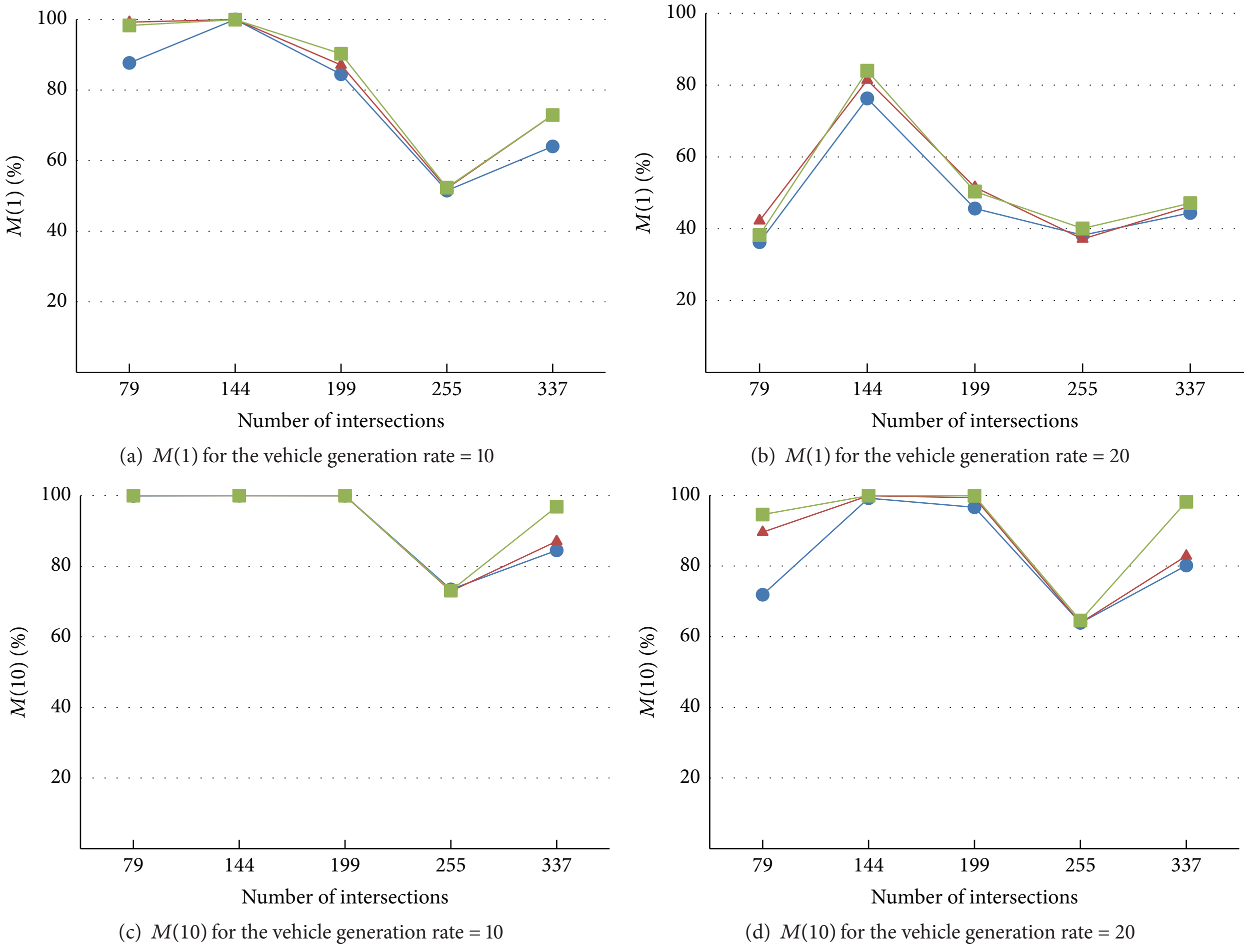
In Figure 7(a), the Dynamic with prediction algorithm exhibits the lowest β_m for all the maps but the number of intersections equals 255. This trend is more pronounced in Figure 7(b) where heavy traffic occurs. We observe that all the algorithms perform poorly with high β_m and Static yields slightly lower β_m than the other two when the number of



(a) Vehicle generation rate = 10

(b) Vehicle generation rate = 20

FIGURE 7: Use of different routing strategies on various road maps (blue = Static, red = Dynamic, and green = Dynamic with prediction).



(a) $M(1)$ for the vehicle generation rate = 10

(b) $M(1)$ for the vehicle generation rate = 20

(c) $M(10)$ for the vehicle generation rate = 10

(d) $M(10)$ for the vehicle generation rate = 20

FIGURE 8: $M(\alpha)$ with different numbers of intersections (blue = Static, red = Dynamic, and green = Dynamic with prediction).

intersections equals 255. Although the reason behind this outlier is not clearly understood, we surmise that certain properties of map topology affect the performance.

The results of $M(1)$ and $M(10)$ are also plotted for the same data set in Figure 8. We see that $M(1)$ and $M(10)$ of the Dynamic with prediction algorithm are higher than the others for all the maps including the number of intersection equal to 255, and this observation is more pronounced in $M(10)$, which is under heavy traffic.

6.3. Coexisting Routing Strategies. In practice, it is unlikely that all the vehicles in a particular region adopt a new route guidance system altogether at the same time, not to mention to convince all the drivers to comply with the

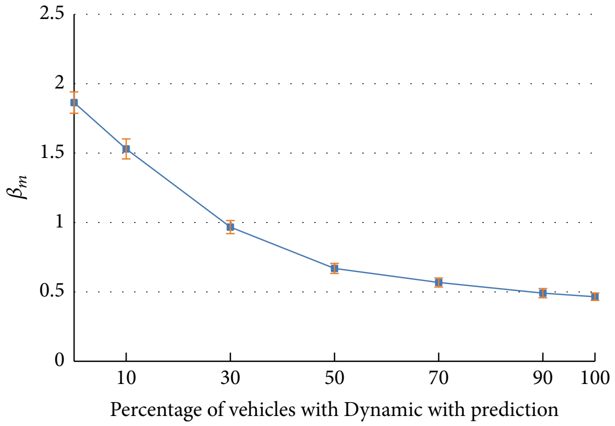


Figure 9: β_m as vehicles following Dynamic with prediction coexist with the ones using the Static routing and their percentage increases from 0% to 100%.

routing guidance. Hence, it is important for our system to be incrementally deployable, not to disrupt the existing system, but to gradually improve the overall performance. To get a glimpse on how well our routing strategies coexist with others, we measure β_m at different compliance levels. We define a compliance level as the percentage of the vehicles that follow a dictated routing guidance. We assume that the rest of the vehicles use the Static algorithm for their routing; that is, these vehicles drive through the best route given at departure and do not comply with rerouting decisions provided by the central routing controller. For this experiment, we use a map of 337 intersections with 20,000 vehicles generated at a rate of 5 vehicles per sampling. Note that other cases show a similar trend.

In Figure 9, the β_m results of vehicles using the Dynamic with prediction routing are illustrated as their compliance level changes from 0% to 100%. The results were obtained by 10 simulations with different random seeds. The case with 0% indicates that all vehicles follow the Static routing algorithm, whereas the case with 100% means that all vehicles comply with Dynamic with prediction. In the figure, β_m is high (a little less than 2) when the compliance level is 0% and monotonically decreases as the level increases to 100% (less than 0.5). More interestingly, β_m drops sharply at around the compliance level of 30% implying that more than half of the travel time reduction already occurs then. Nearly all travel time reduction occurs at the compliance level of 50%. This result demonstrates high potentials of the Dynamic with prediction routing algorithm, as the results imply that the algorithm can be deployed for higher road efficiency even when not all the vehicles are equipped with the new routing guidance and even if not all the drivers comply with the guidance.

7. Conclusions and Future Work

We have demonstrated that traffic routing can benefit from using predictive information as it helps reduce travel time and improve road efficiency based on simulation studies. We propose a traffic routing algorithm that utilizes both current and near-future traffic conditions using already routed vehicles. The performance of this routing algorithm is evaluated via simulations under various traffic conditions including light and heavy traffic and small to large areas of the road network. Our results show that the algorithm outperforms other more conventional ones and also successfully reduces travel time even when not all vehicles comply with the guidance of the algorithm.

This study has some limitations including the assumptions made in the simulator, for example, traffic light control at intersections. A more realistic simulation may be possible with traffic light control, car-following dynamics, lane change rules, and routing algorithms that take into account intersection delays. Such simulations will certainly help assess the proposed algorithm more accurately.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank Kuangwei Zhang for his valuable contribution that helped develop the simulator initially. This research was supported in part by GRL Program (2013K1A1A2A02078326) and by the BSR Program (NRF-2013R1A1A2058304) both through the NRF of Korea, in part by the IITP grant funded by the Korea government (B0101-15-0557), and in part by Hanbat National University in 2011.

References

[1] D. Schrank, B. Eisele, and T. Lomax, "TTI's 2012 urban mobility report," Tech. Rep., Texas A&M Transportation Institute, 2012.

[2] Intel, "Intel® In-Vehicle Solutions: Brief," http://www.intel.com/content/www/us/en/embedded/automotive/in-vehicle-solutions-platform-brief.html.

[3] IBM, "Internet of Things for Automotive," http://www.ibm.com/analytics/us/en/industry/iot-for-automotive/.

[4] M. Gerla, E.-K. Lee, G. Pau, and U. Lee, "Internet of vehicles: from intelligent grid to autonomous cars and vehicular clouds," in *Proceedings of the IEEE World Forum on Internet of Things (WF-IoT '14)*, pp. 241–246, Seoul, Republic of Korea, March 2014.

[5] Google, Google Maps, http://maps.google.com.

[6] LOCNALL Inc, Kimgisa Navigation, http://www.kakao.com/services/67.

[7] PTV Planung Transport, PTV VISSIM, http://vision-traffic.ptvgroup.com/en-uk/products/ptv-vissim/.

[8] McTrans Center, University of Florida, TSIS-CORSIM, http://mctrans.ce.ufl.edu/featured/tsis/.

[9] OpenStreetMap, OpenStreetMap: The Free Wiki World Map, http://www.openstreetmap.org.

[10] R. Claes, T. Holvoet, and D. Weyns, "A decentralized approach for anticipatory vehicle routing using delegate multiagent systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 364–373, 2011.

[11] I. Leontiadis, G. Marfia, D. MacK, G. Pau, C. Mascolo, and M. Gerla, "On the effectiveness of an opportunistic traffic management system for vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1537–1548, 2011.

[12] WorldDAB, Digital Radio in Car, http://www.worlddab.org/technology-rollout/digital-radio-in-car.

[13] Y. H. Jeong and W. W. Kim, "A novel TPEG application for location based service using terrestrial-DMB," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 1, pp. 281–286, 2006.

[14] SK Planet, "T-map," http://www.tmap.co.kr.

[15] Verizon Wireless, VZ Navigator, http://www.verizonwireless.com/wcms/consumer/products/navigator.html.

[16] Z. Liang and Y. Wakahara, "Real-time urban traffic amount prediction models for dynamic route guidance systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, article 85, 2014.

[17] D. Park, H. Kim, C. Lee, and K. Lee, "Location-based dynamic route guidance system of Korea: system design, algorithms and initial results," *KSCE Journal of Civil Engineering*, vol. 14, no. 1, pp. 51–59, 2009.

[18] G. R. Jagadeesh, T. Srikanthan, and K. H. Quek, "Heuristic techniques for accelerating hierarchical routing on road networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 4, pp. 301–308, 2002.

[19] Q. Song and X. Wang, "Efficient routing on large road networks using hierarchical communities," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 132–140, 2011.

[20] M. K. Mainali, K. Shimada, S. Mabu, and K. Hirasawa, "Optimal route of road networks by dynamic programming," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '08)*, pp. 3416–3420, June 2008.

[21] Y. Wang, S. Mabu, Q. Meng, M. K. Mainali, and K. Hirasawa, "Multiple ODs routing algorithm for traffic systems using GA," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '10)*, pp. 1–8, Barcelona, Spain, July 2010.

[22] M. K. Mainali, S. Mabu, and K. Hirasawa, "Hierarchical efficient route planning in road networks," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC '11)*, pp. 2779–2784, Anchorage, Alaska, USA, October 2011.

[23] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[24] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: deep belief networks with multitask learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191–2201, 2014.

[25] A. Abadi, T. Rajabioun, and P. A. Ioannou, "Traffic flow prediction for road transportation networks with limited traffic data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 653–662, 2015.

[26] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: where we are and where we're going," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 3–19, 2014.

[27] M. Treiber and A. Kesting, *Traffic Flow Dynamics: Data, Models and Simulation*, Springer, Berlin, Germany, 2013.

[28] T. H. Cormen, C. E. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, Mass, USA, 3rd edition, 2009.