Themis: A Participatory Navigation System for Balanced Traffic Routing

Ruilin Liu*, Hongzhang Liu*, Daehan Kwak*, Yong Xiang†, Cristian Borcea‡, Badri Nath*, and Liviu Iftode*
†Dept. of Computer Science and Technology, Tsinghua University, Beijing, China
xyong@tsinghua.edu.cn
‡Dept. of Computer Science, New Jersey Institute of Technology, Newark, NJ, USA
borcea@njit.edu
*Dept. of Computer Science, Rutgers University, Piscataway, NJ, USA
{rl475, antony26, kwakno1, badri, iftode}@cs.rutgers.edu

***Abstract*—Navigators based on real-time traffic achieve suboptimal results since, in face of congestion, they greedily shift drivers to currently light-traffic roads and cause new traffic jams. This paper presents Themis, a participatory system navigating drivers in a balanced way. By analyzing time-stamped position reports and route decisions collected from the Themis application, the Themis server estimates both the current traffic rhythm and future traffic distributions. According to the estimated travel time and a popularity score computed using the learned information, Themis coordinates traffic between alternatives and proactively alleviates congestions. Themis has been implemented and its performance has been evaluated at different penetration rates based on real data. Experiments using data from 26,000 taxis demonstrate that Themis reduces both traffic congestions and average travel time at various penetration rates as low as 7%.**

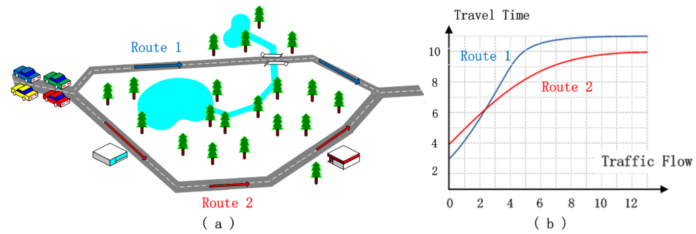

Fig. 1. Four cars are about to choose route 1 or route 2 to go from left to right in (a). Due to no traffic, the latest travel perceptions are three minutes and four minutes for two alternatives, respectively. However, if each car greedily takes route 1, given the travel-delay model of the two routes in (b), the actual travel time for route 1 will be nine minutes, three times of the estimated value.

I. INTRODUCTION

The development of mobile devices and mobile communication has lead to a great prosperity of navigation applications. Modern drivers equipped with GPS-enabled devices not only digest the traffic information but also work as traffic information providers. Google Maps and Waze apply the location and event reports collected from smartphone users to compute the estimated time of arrival (ETA) of the routes. Driving experiences and fuel consumptions are also shared in novel systems [1], [2] to help users' route choices. According to Ericsson ConsumerLab, 29% of smartphone users in the U.S. use Google Maps or other smartphone navigation apps during morning commute in 2011 [3]. Given the similar number of dedicated navigation devices [4], the penetration rate of dynamic navigation users is now considerable.

Current navigators greedily route drivers to the fastest path based on the periodically updated traffic condition. The high penetration of these navigators potentially incurs the Braess's paradox [5] caused by the coupled route choices. For instance, in Fig. 1, greedy routing leads all of the four vehicles to the same route and causes congestions because vehicles need time to reach the bottle neck of the planned route, while subsequent drivers may have made their decisions before the influence of previously routed cars is reflected on the traffic condition. Agent based simulation has demonstrated that the average travel time increases when half of the drivers follow the dynamic fastest path based on real-time traffic [6]. Roughgarden [7] also elaborates on the suboptimal global situation caused by the greedy routing without coordination.

A few routing algorithms have been proposed to overcome the drawback of greedy routing [8]–[13]. These algorithms, called cooperative routing, plan routes based on anticipated traffic volume (ATV) and corresponding predicted travel time (PTT) by assuming previously routed cars follow their suggested routes. For example, in Fig. 1, two cars may anticipate the future congestion in route 1 and take route 2 instead, even if route 2 has longer ETA based on the real-time traffic. In our prior work [14], we also presented a cooperative routing algorithm, EBkSP, to route traffic based on both ETA and the popularity of the candidate routes.

Despite plenty of algorithmic studies, the difficulty to build a cooperative routing service lies in two aspects: 1) The real-time acquisition of different necessary information supporting cooperative routing decisions. These kinds of information need to be mined in one system, as they are usually mutually influential (e.g., current ETA influences the traffic movement expectation and accordingly impacts on the ATV, which in turn determines the PTT). 2) The sensitivity analysis of the possible impact of cooperative routing at different system penetration rates. The analysis should be made based on real data due to the potential impact of unique traffic pattern and the road network of each city. This paper addresses both aspects.

In this paper, we present a participatory navigation system, Themis, which utilizes the data, such as location samples and route choice decisions, collected from road vehicles to estimate the traffic speed as well as the future traffic flow at road segment level (§III). In addition, a balanced routing algorithm is implemented to provide real-time cooperative routing services.

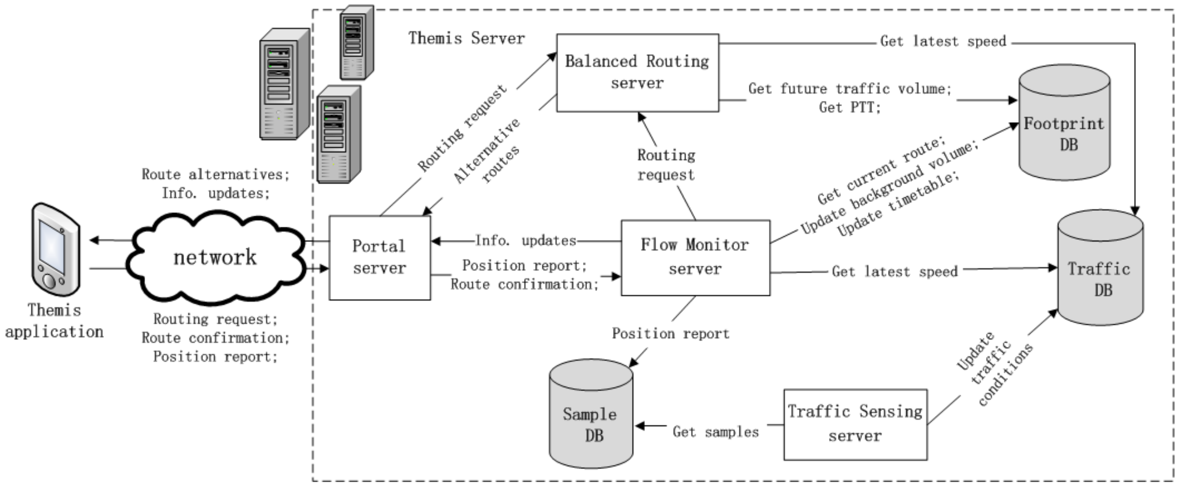

Fig. 2. Themis Service Architecture

More importantly, we present a method to investigate how the performance of Themis scales at different penetration rates by meaningfully expanding the real data collected from probe vehicles (§V). We apply the method to the trajectory data from over 26,000 taxis and demonstrate that Themis outperforms greedy navigation systems. We find that the benefits of Themis emerge even if the penetration rate is as low as 7%.

II. RELATED WORK

Several algorithms were designed to solve the cooperative routing problem, and they can be divided into three categories.

The first category of algorithms focus on user equilibrium, which computes the PTT of road segments and plans the fastest path based on PTT. Yamashita et al. [8] used Greenshield's model [15] to relate PTT with ATV and designed the Passage Weight heuristic to generate the contribution of each planned path towards ATV. [9] used a similar model to relate PTT and ATV except that they assumed the traffic volume to be stochastic and determined by both historical traffic and previously assigned traffic. [12] proposed to compute a few alternatives based on real-time traffic and then route the car to the path with the shortest PTT based on encounter prediction.

Other studies aim to achieve social optimum, whose objective function is to minimize not the individual travel time but the average travel time of a group of users, e.g., users using a proposed system or all drivers in a city. Bosch et al. [13] proposed to handle a routing request by searching a path minimizing the total PTT of all previous assigned drivers. Lim et al. [10] proposed to compute a few route candidates based on real-time traffic and investigate the mutually timing influence of users' route choices based on BPR flow-delay model [16]. They assign a group of drivers to the combination of paths that optimize the total travel time. This algorithm was evaluated using the taxi trajectory data in Singapore [17].

Finally, [14] proposed several heuristics to plan or to choose from first-k shortest paths (KSPs) based on previously assigned traffic. The basic workflow of these approaches is similar to aforementioned categories, except that the criteria of choosing routes are not PTT but some heuristic functions. For example, EBkSP algorithm computes the KSPs according to real-time traffic and chooses the route with the least popularity to balanced the traffic volume distribution. The popularity is defined based on both current traffic condition and ATV.

In this paper, we address the challenges of implementing these algorithms in real life. We present a participatory system, using up-to-present data collected from cars to determine the information, based on which cooperative routing algorithms make decisions (i.e., real-time traffic and PTT or ATV). Compared with the taxi data evaluation in [17], our evaluation method investigates the performance of cooperative navigation at different penetration rates by meaningfully expanding the trajectory data.

III. PARTICIPATORY NAVIGATION SYSTEM

The Themis application is assumed to be installed in a mobile device, which is equipped with GPS and wireless communication module, such as DSRC or cellular module, to connect with the Themis server. The route computation is carried out in the Themis server using a cooperative routing algorithm. While the route suggestions are consumed by Themis application to provide users with turn-by-turn directions, the application also updates route confirmation and time-stamped position to help deal with subsequential navigation requests.

A. System Architecture

As illustrated in Fig. 2, Themis consists of five executable entities: the Themis application in mobile devices, the Portal server, the Traffic Sensing server, the Flow Monitor server, and the Cooperative Routing server. While logically centralized, each server can be implemented in a distributed fashion to provide scalability. The Themis application presents driving directions, uploads time-stamped position reports, and allows the driver to select alternative routes. The Portal server ensures the interaction between the Themis server and Themis applications and performs request dispatching and load balancing. The Traffic Sensing server estimates the segment travel time. The Flow Monitor server supervises cars' movement along each planned route and estimates the future traffic that are scheduled to travel through each road segment. In addition, the Flow Monitor server is responsible to update the information (e.g., ETA) for users and propose a rerouting request if a detour is detected. The Cooperative Routing server computes the route candidate(s) based on real-time speed, PTT, and ATV.

The information that the Cooperative Routing server relies on is stored in two databases. The Traffic database stores the static road map, the traffic-delay model, and the latest ETA of each road segment; it is updated by the Traffic Sensing server. The Footprint database maintains the routes being taken by drivers and its status, such as the timetables labelling the ETA to each road segment included in the confirmed routes. It also stores the short-term predictions of traffic flow on each road segment (i.e., how many cars will go through a road segment in the future) based on the routes being taken and latest traffic condition. The Footprint database is updated by the Flow Monitor server. The Sample database is only used to cache the position reports.

Navigation Process. When a user issues a new navigation request, the Themis application contacts the Portal server with the origin-destination information. The Portal server forwards the routing request to the Cooperative Routing server, where route candidates are calculated using data from two databases. The routing results are returned to the Themis application by the Portal server to generate alternative route previews. The user chooses one of the alternatives, and the Themis application translates the selected route into turn-by-turn directions. A confirmation of the selected route is meanwhile sent back to the Flow Monitor server to update the Footprint database.

Position Report Process. While driving, the Themis application periodically reports the time-stamped position (also called samples) to the Portal server, which is then sent to an available Flow Monitor server. The Flow Monitor server stores each position report in the Sample database, waiting for the process of the Traffic Sensing server. Meanwhile, if the position is successfully matched to the previously confirmed route, the server will return the user its latest ETA and earned score (see §III.D) and update the timetable belonging to this route in the Footprint database; if it is detected as a detour, the Flow Monitor server will end the current confirmed route and issue a new routing request to the Cooperative Routing server.

B. *Participatory Traffic Sensing*

After collecting and preprocessing the samples from multiple cars (including time-stamped location and car ID) during an interval, the Traffic Sensing server takes the following steps to estimate the travel time on each road segment.

1) Map Matching: During the map matching process, samples are matched to the road map (i.e., to the most likely position on road segments), and the possible episode routes linking consecutive samples from each car are also inferred. Fig. 3 shows how the episode routes are constructed.

Themis map matching is based on the Hidden Markov Map Matching (HMMM) method in [18], which considers both the distance to nearby roads and the context of each sample. For example, although sample R2 in Fig. 3 is closer to Lexington Ave, it should be matched to East 37th St because it is unlikely for a driver to travel from R1 to R3 through Lexington Ave.

In [18], the context information used to compute the transition probability in HMMM is the length of the episode

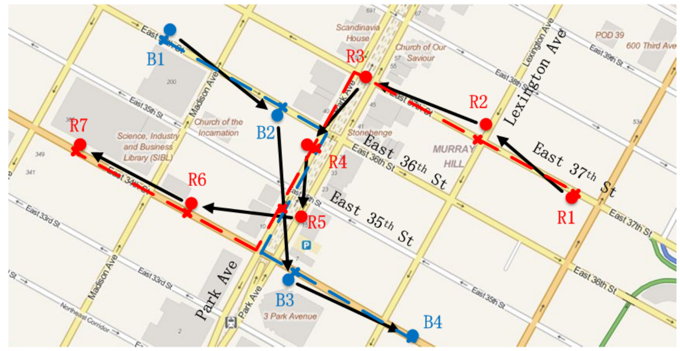

Fig. 3. An example of participatory traffic sensing. Red and blue dots represent the trajectories from two different cars in the same interval. Their movement directions are illustrated using arrows. After map matching process, the samples in each trajectory are matched to points along the roads shown with crosses. Meanwhile, the inferred episode routes that cover the matched points (shown in dash lines) are used for travel time allocation and aggregation.

routes connecting adjacent samples. Themis enhances it by defining a Weighted Route Length (WRL) based transition possibility. The motivation for this change is the observation that people tend to take main roads instead of lower-level (i.e., smaller) roads even if the length of low-level roads is shorter. Suppose p_i is the i-th possible episode route between two matching samples, then:

$$WRL(p_i) = \sum_{e \in p_i} len_e * weight_{l(e)}, \qquad (1)$$

where len_e is the length of the road segment e and $weight_{l(e)}$ is the weight associated with the level of the road segment e. The HMMM method essentially computes the probability of several episode route candidates, each of which is a function with the weights as its parameters. As a result, the $weight$ of each road level can be learned from real data (i.e., training data). In our case, we set the objective function as maximizing the sum of the probability differences between the ground truth path and all other paths and trained the value of $weight$ using over 15,000 manually matched samples.

2) Travel Time Allocation and Aggregation: The episode routes inferred through map matching may consist of multiple road segments and even partial road segments. Travel time allocation divides the travel time observed on an episode route to the road segments covered by this route using the estimated travel time in the previous interval. For example, the travel time from B2 to B3 in Fig. 3 is distributed to the two fully covered road segments and the two partially covered road segments.

Given an episode route p_i and the travel time observation τ_i, the travel time allocation process computes a travel time estimation for each road segment covered by p_i, defined as $R_{p_i}(e_{i,1}, e_{i,2}, ..., e_{i,n})$. For a road segment $e_{i,j}$ partially covered by p_i, we define $\rho_{i,j}$ as the fraction of covered length out of the total length of $e_{i,j}$. We denote the travel time estimation on road segment $e_{i,j}$ in previous interval (i.e., interval $n-1$) as $\bar{t}^{\,n-1}_{i,j}$. The travel time on road segment $e_{i,j}$ estimated from episode route p_i, denoted as $\tau^n_{i,j}$, is computed as follows:

$$\tau^n_{i,j} = \frac{\bar{t}^{\,n-1}_{i,j}}{\sum_{e_{i,k} \in R_{p_i}} \rho_{i,k} * \bar{t}^{\,n-1}_{i,k}} * \tau_i \qquad (2)$$

Suppose $p(p_1, p_2, ..., p_n)$ is the collection of episode routes covering a specific road segment within interval n. The aggregation process utilizes the time estimations for this segment drawn from each $p_i \in p$ and aggregates them into one travel time expectation value. As shown in Fig. 3, by allocating travel time of episode route (R3, R4), episode route (R4, R5), and episode route (B2, B3), respectively, we have three travel time estimations for the road segment on Park Ave from East 36th St to East 35th St. The estimations are aggregated to get $\bar{t}\,_e^n$, the travel time estimation of edge e in interval n:

$$\bar{t}\,_e^n = \frac{\sum\limits_{p_i \in p, e_{i,j}=e} \tau_{i,j}^n * \rho_{i,j}/\tau_i}{\sum\limits_{p_i \in p, e_{i,j}=e} \rho_{i,j}/\tau_i} \tag{3}$$

The aggregation process smoothes the influence by the non-traffic factors, such as the influence of driving style. Equation (3) essentially calculates the weighted average value of individual estimations, which biases the estimation in favor of the episode routes with longer coverage and the episode routes with higher sampling rate.

C. Flow Monitor

The basic function of the Flow Monitor server is to estimate ATVs on road segments such that PTTs could be inferred based on given traffic-delay models. Since the penetration rate of Themis is not expected to be 100%, the ATV includes both the controlled traffic and the background traffic.

The controlled traffic represents the cars that use Themis to plan paths and navigate to destinations. This part of traffic are easy to count. Once a route is confirmed by the Themis application, the Flow Monitor server translates the planned path into a timetable, in which the ETA of each road segment in the path is sequentially estimated starting from the current location using the latest travel time estimation. The timetable is updated when a new location report or route confirmation is received. Based on the timetable, the controlled traffic volume can be estimated given a road segment and a time-stamp.

The method used to estimate the background traffic in Themis is based on [19], which proposed a real-time technique to estimate the unexposed traffic volume based on the baseline roads (i.e., the roads where the ground truth volume can be measured using sensors, such as inductive loop). This algorithm has been proven to have good performances given accurately estimated real-time speed and traffic volume generated by a large number of natural probes. In Themis, the estimation of real-time speed is discussed in §III.B. However, we cannot directly use the volume computed from the controlled traffic to infer the background traffic because the paths of controlled traffic are not "natural" (as expected by the method in [19]) but influenced by the cooperative navigation system. Currently, we assume the natural path to be the fastest path and update the timetable as if the car moves at the latest estimated speed along the path. Although there may be exceptions where users would not take the fastest path in the absence of Themis, we leave more accurate natural path inference for future work.

D. Cooperative Routing Algorithm

Two routing algorithms are implemented in Themis routing server, the Dijkstra fastest path based on the real-time travel time estimation, and the balanced routing algorithm based on EBkSP algorithm [14]. According to the popularity determined in EBkSP, we define the score of an alternative route π_i as:

$$score_{\pi_i} = \frac{ETA_{avg}}{pop(\pi_i)}, \tag{4}$$

where ETA_{avg} is the average ETA for all the alternatives and $pop(\pi_i)$ is the popularity of route π_i. Intuitively, a higher score is associated with the route with lower popularity. Including ETA_{avg} in the equation does not influence the route choice. We add this term only to scale the magnitude of scores.

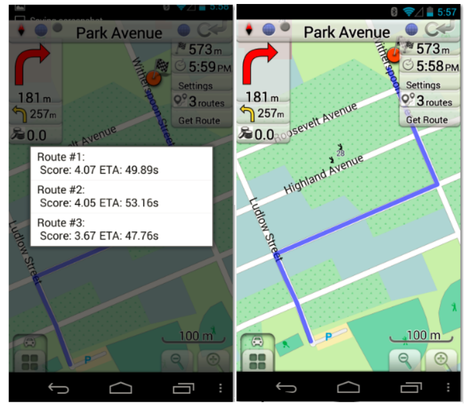

Fig. 4. Themis application interface. The left screenshot shows three possible routes ordered by their scores and ETAs. By choosing among three alternatives, a user enters the navigation mode (right screenshot), which provides turn-by-turn voice assistance. During navigation mode, the upper left part of the screen displays the driving directions for the next two steps. The score that a user has earned so far for taking the current route is shown below the driving directions. The labels in the upper right part show the distance to the destination, estimated arrival time, and other route choices.

IV. IMPLEMENTATION

We implemented the Themis application on Galaxy Nexus (Android 4.3) based on an open-source GPS navigator, OsmAnd [20]. Themis application is implemented as a plugin, together with a customized map layer, such that it can be easily switched on or off. We inherit the interfaces of turn-by-turn driving assistance provided by OsmAnd and move the route calculation to the Themis server. The user interface of Themis is shown in Fig. 4. Themis computes three alternative routes based on real-time traffic condition and associates each route candidate with a score and its ETA. During the navigation process, the application listens to position change events and routing requests from the user. The former triggers a potentially new position update request, and the latter directly starts generating a routing request. These requests are then sent to the Portal server via a JSON Interface.

The implementation of the Themis server combines several open-source softwares or services to provide the cooperative routing service. As the source of static map data, Themis imports OpenStreetMap (OSM) [21] into PostgreSQL [22] database. The traffic information estimations are also stored in the database as the properties of each road segment. Based on the open-source routing library pgRouting [23], the two algorithms presented in §III.D are implemented.

V. EVALUATION

The Themis system has been tested and validated through neighborhood-scale field studies with several cars. For the lack of space, we do not present those results in this paper. Instead, in this section, we analyze the performances of Themis in synthetic scenarios modeling a city-scale deployment and address the following questions:

1) How accurate is Themis's participatory sensing in estimating the traffic characteristics?

2) How does balanced routing compare with greedy routing in terms of traffic distribution and average travel time?

3) How do the traffic distribution and travel time vary with the system penetration rate and the total traffic amount?

A. Experimental Methodology

1) Dataset: We use GPS trajectories from a 26,000 fleet of taxis in three consecutive Tuesdays starting from April 6, 2010 in Beijing, which amount to approximately 58,000,000 valid data points. Each sample contains taxi ID, timestamp, latitude, longitude, and the passenger status. They are sampled at intervals between 30 seconds and five minutes. To derive the total traffic, we use two other datasets: the daily percentage of taxi traffic out of total traffic on 3981 main road segments, and the half-hour variation of taxi traffic out of total traffic. By checking both the covered area and the taxi penetration rate, we decided to carry out our experiment using the data in a rectangle area covering the Beijing second ring area, which is about 65 square kilometers. In this area, our taxi data account for almost 7% of the total traffic.

2) Modeling Trips and Penetration Rates: The first problem we solved is how to extract routing requests from the dataset and manipulate them to get different penetration rates. First, we identify passenger-on-board (POB) trips. The time-stamped location where the passenger status of the taxi changes to POB is recognized as the origin of a POB trip. The time-stamped location where the POB status is reset to a different status is considered as the destination. Since the number of POB trips is limited, especially during late-night hours, we also include non-passenger (NP) trips, which are those trips happening between POB trips. We limit the max duration of NP trips to five minutes to avoid the influence of vacant taxis' wandering driving. The routing requests generated in this way could be used to evaluate Themis when the penetration rate is no more than the original percentage of the taxis in traffic.

For the sensitivity analysis at higher penetration rates, we built a threefold and a sixfold routing request set. The threefold routing request set is generated by proposing each original routing request three times and adding a random delay within five minutes between each of them. The sixfold routing request set is generated likewise. Therefore, the penetration rate can be increased to approximately 20% and 40%, respectively.

Note that the global penetration rate is carefully bounded by 40% so that, on any road segment, the traffic incurred by the amplified routing request set does not exceed the total traffic estimated using the original routing request set (will present in §V.A.3). Different penetration rates only change the proportion of the controlled traffic but do not influence any global traffic features such as traffic volumes and average speeds. The essence of amplifying request set is that we extract some background traffic in consecutive road segments and merge them to generate a new trip based on real previous trips. There are other alternatives to adjust the penetration rate, for example, model the traffic demand based on the original routing request set and then generate amplified request set based on traffic demand model. Due to the complexity, we leave it as future work.

3) Background Traffic Estimation: At each penetration rate, the background traffic volume equals to the original total traffic volume minus the controlled traffic determined by different routing request sets (origin, threefold, or sixfold) and the original traffic condition. Based on the method from [19] (also described in §III.C) and our taxi ratio statistical data, we can derive the background traffic volume if we know the movement of the controlled traffic on the road segment.

Given any routing request set with original routes, the movement of the controlled traffic can be simulated at the original average speed, which can be estimated by applying our traffic sensing algorithm (§III.B) to the taxi trajectory dataset. After each 15-min interval, we directly estimate the average segment speed if there are at least two taxi trajectories passing through the segment during the interval. For the road segments that do not have direct estimations, we derive the estimated average speed using the speed of their joint road segments. As discussed in §V.A.2, the estimated speed when the penetration rate is 7% can also be used for the other two penetration rates (i.e., 20% and 40%). We call the speed estimated using participatory traffic sensing as sensed speed.

4) Learning the Traffic-Delay Model: If every cars takes its original route, the average travel speed on each road segment would be the original sensed speed. However, to evaluate the travel time of the trips after rerouting, this original sensed speed is not useable as reroutings dramatically change the traffic distribution. Therefore, we use a traffic-delay model [16] to infer the travel time of a road segment based on the traffic volume on it in a given time interval, which is determined as described in §V.A.3. The speed or travel time derived based on the traffic-delay model is called inferred speed or inferred travel time. The function used for the traffic-delay model is:

$$t_e(f_e) = T_e^0(1 + \alpha(\frac{f_e}{C_e})^{\beta}) \tag{5}$$

T_e^0 is the free-flow speed. C_e is the capacity of the road segment e and f_e is the average traffic volume. $t_e(f_e)$ is the inferred speed based on the traffic volume f_e.

Using a similar method as [17], we learned the parameters α and β for 7,411 road segments from the total of 11,450 road segments. These segments were used because they have good direct travel time estimation. Fortunately, these models cover most main roads. During the evaluation, we only route traffic over road segments with traffic-delay model such that we are able to measure travel time changes. For the comparisons between the actual travel time of the original routes and the inferred travel time using traffic-delay models, we also only use the routes fully covered by the traffic-delay models.

5) Traffic Movement Simulation: Given the route and the average speeds in each interval, we simply simulate the trip by

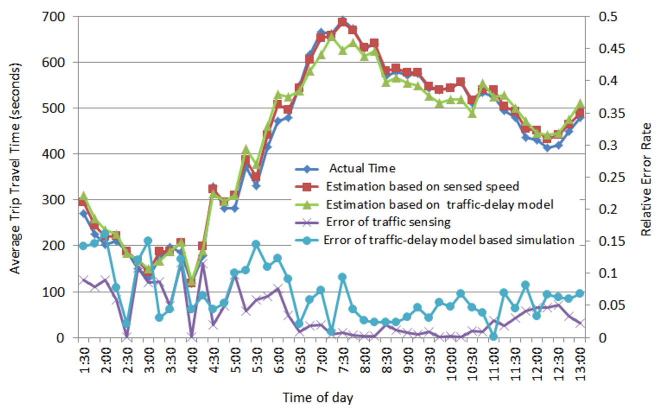

Fig. 5. The accuracy of trip travel time estimation.

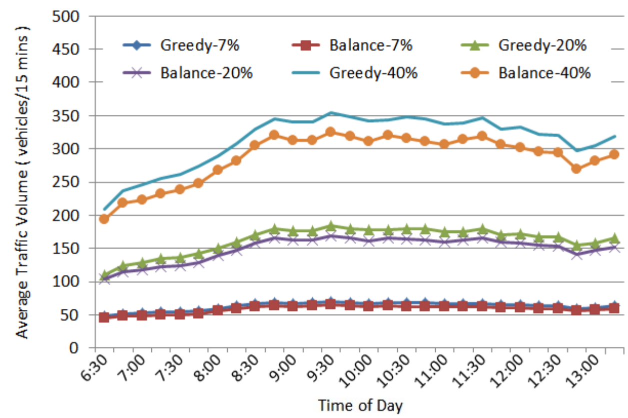

Fig. 6. Traffic volume comparison at different penetration rates.

assuming a car travels through each road segment of the route using the average speed of the current interval. As we know the starting time of the trip, we can compute the starting time of each road segment step-by-step. When the time reaches a new interval, we use the average speeds for this new interval.

One difficult issue is how to simulate the rerouted trips. After rerouting, the movement simulation must be done using the speeds inferred based on the traffic-delay model. However, the traffic-delay model needs the traffic volumes to derive speeds. This is a chicken-and-egg problem because the traffic volume is determined by the movement of the traffic within the 15-min interval. To solve it, we use an iterative method to compute the speeds within each interval. In the first iteration, we use the sensed speeds to simulate the traffic movement. This step will end with new inferred speeds on the road segments. In the next iteration, we used the inferred speeds from the first iteration to update the speeds in the same interval. The process goes on iteratively until no speed changes on any road segment. Fortunately, the process converges very quickly because small speed changes do not influence the traffic volume distribution significantly.

B. Evaluation of Travel Time Estimations

The accuracy of either sensed speed or inferred speed is evaluated by a comparison between the average ground truth trip travel time and the average simulated trip travel time based on corresponding speed estimations. To get the sensed speed, we apply the participatory traffic sensing algorithm (§III.B) to our taxi trajectory dataset. For the inferred speed, we apply the method in §V.A.5 to the 7% penetration routing requests set based on the learned traffic-delay model.

For the trips used for evaluation, we choose 100 random trips from the original request set in each 15-minute interval from 1:00 to 13:00 on April 6, 2010. We choose this period as it contains both peak hours and low-traffic hours. In addition, the number of taxis in operation changes greatly during this period so that we can analyze the sensitivity of Themis participatory traffic sensing. At 4:00, only 7647 taxis (30% in our dataset) are running, while over 95% taxis are operated between 9:00 and 9:30. For sensed speed evaluation, we use "leave-one-out" validation by not including the estimations from the car under evaluation into the aggregation step. This validation is not done for the inferred speed evaluation, because the traffic-delay models are built using the data from three days, not specifically optimized for our evaluation set.

Fig. 5 shows that the actual time and the simulated travel time computed based on sensed speed are closely matched, which means that the accuracy of our participatory sensing is high. During the period from 6:30 to 10:30, the relative error is less than 3%. The highest error comes during late-night hours when the participants are extremely sparse. However, even in this case, the relative error is below 12%. The result shows that our participatory sensing algorithm has good accuracy and robustness with different numbers of participants.

The travel time simulated using inferred speed also matches the ground truth well. During the period from 6:30 to 13:00, the relative error is less than 10%. Similar as participatory sensing, the inferred speed also has a higher error rate during late night, which is less than 15%. These results prove that our traffic-delay model is acceptable. In order to get the accurate evaluation of the balanced navigation system, we only carry out the following evaluation experiments between 6:30 and 13:00. Another reason to abandon the period between 1:00 and 6:30 is that during this period the traffic load is too low to generate traffic congestions.

C. Evaluation of Traffic Distribution

The average traffic volume is used as a global measure of congestion in the road network and is computed over all the road segments that are traversed by at least one car. The higher the traffic volume, the less distributed the traffic; consequently, it is more likely to experience congestion in the network.

Fig. 6 presents the comparison of the navigated traffic volume between two implemented routing algorithms in Themis (i.e., balanced routing and greedy routing, see §III.C) at different penetration rates. The balanced routing results in a lower traffic volume at each interval for each penetration rate. These results demonstrate that balanced routing distributes the traffic better than the greedy routing. Furthermore, the traffic volume is reduced more substantially for higher penetration rates. We also observe that, during the experimental period, the relative traffic volume reduction is steady at each penetration rate regardless of congestion levels. Another interesting finding is that as the penetration rate increases, the relative traffic

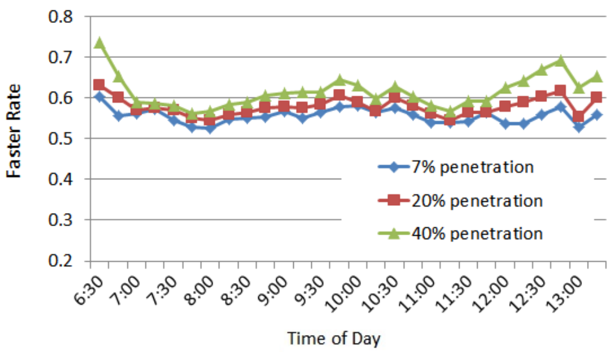

Fig. 7. Comparison of FRs at different penetration rates.

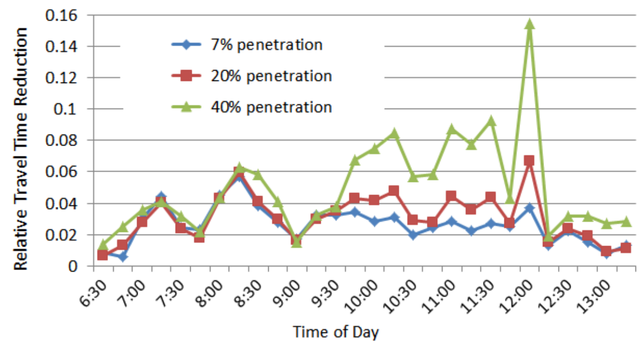

Fig. 8. Comparison of RTTRs at different penetration rates.

volume reduction tends to increase more slowly. This implies that as the penetration rate rises above a certain threshold, the balanced routing's function to reduce global traffic congestion will linearly increase with the penetration rate.

D. Evaluation of Trip Travel Time

One danger of balanced routing is that it could lead to longer trips when distributing traffic to unpopular routes. We defined two criteria to compare the performances of balanced routing and greedy routing in terms of trip travel time:

$$FR = \frac{Number(A's\ travel\ time < B's\ travel\ time)}{Number(all\ the\ routing\ requests)} \quad (6)$$

$$RTTR = \frac{A's\ travel\ time - B's\ travel\ time}{B's\ travel\ time} \quad (7)$$

Faster Rate (FR) reflects how much percent of the routes suggested by A are faster than those suggested by B. Relative Travel Time Reduction (RTTR) reflects to what extent the routes suggested by A are faster than B's. In both definitions, A refers to Themis balanced routing while B is the greedy routing (i.e., fastest path based on real-time traffic sensing).

Fig. 7 shows that over 50% of routes suggested by the balanced routing cost shorter travel time than the greedy routing at any interval for any penetration rate in our experiment. Specifically, the average FR for 7%, 20%, and 40% penetration are 55%, 58%, and 61%, respectively. The results demonstrate that the balanced routing provides users with higher chance to achieve shorter travel time. As penetration rate increases, FR also rises, which means the more users adopt Themis system, the higher chance users gain to reduce their trip travel time. Another finding is that even at 7% penetration rate, Themis users could still expect higher chance to save their travel time, which could be a motivation to change the greedy routing behavior during even the bootstrapping stage of Themis.

Fig. 8 shows that the balanced routing reduced the average trip travel time substantially, e.g., as much as 15% at 40% penetration rate. Moreover, during the experimental period, the higher the penetration is, the more travel time is saved. Another significant trend of travel time reduction is that RTTR is determined by both the traffic density and the penetration rate. During moderate-traffic hours (10:00 to 12:00), Themis averagely reduces travel time by 8.2% for 40% penetration, 4.0% for 20% penetration, and 2.7% for 7% penetration. These results illustrate that the RTTR achieved by Themis over greedy routing is substantial and increases with the penetration rate for moderate traffic. In addition, the relative travel time reduction increases faster when penetration rate rises from 20% to 40%. Therefore, better RTTR can be expected when the penetration keeps rising beyond 40%. Unfortunately, our dataset does not allow us to prove it.

During morning commute (6:30 to 9:30) and lunch time (12:30 to 13:00), though the balanced routing still leads to lower average travel time, it is less significant than that in moderate-traffic hours. Moreover, the penetration rate does not have much influence during these intervals. One reason for this result is that these periods contain rush hours in Beijing when most road segments might have been crowded and there would not be many better options to choose even for Themis balanced routing algorithm. Another reason is that the number of taxis in operation is small at 6:30 and gradually increases to normal level until 9:00, which implies the actual penetration during this period can be lower than daily average value.

VI. DISCUSSION AND FUTURE WORK

Scalability. The Themis system is designed to work in real time. The location samples are processed every interval to update travel speed estimations. The flow-estimation algorithm re-estimates the traffic volume when cars change their travel plans. The cooperative routing module does the route planning once receiving a new routing request in real time. Basically, each car adopting Themis system increases the workload of the system, which makes the scalability issue critical.

Fortunately, the algorithm inside each of three aforementioned components can be parallelized. The participatory traffic sensing is an ideal "map-reduce model" where map-matching and travel time allocation of each car are a "map" process and travel time aggregation is the "reduce" process. The flow estimation is in the unit of road segment, such that the map could be partitioned and processed in parallel. In addition, each component is isolated with its interfaces defined as shared database. For instance, the cooperative routing algorithm executes merely on the data from traffic database and footprint database. Therefore, the scalability of the balanced routing system mainly relies on the complexity of the routing algorithm. In our evaluation, we used a popularity based method to check the sensitivity of the system. When the penetration rate is 40%, the average computation time of routing is less than 0.8 second on a commodity computer, which implies that a city-scale deployment is not completely intractable.

Balanced Routing Algorithms. The Themis system implemented EBkSP algorithm as a first step and we used this algorithm as an example to compare the performance of the balanced routing and the greedy routing. As discussed in §II, most cooperative routing algorithms require similar information as EBkSP. As a result, they can also be implemented into Themis easily and evaluated using the same method as §V.

By choosing the least popular route, EBkSP increases the entropy of the sub traffic system containing only the planned alternative routes, and consequently, increases the entropy of the whole system, which relates to the system-wide degree of balance. Based on this heuristic, EBkSP is lightweight and performs well. However, during our field study, we found it difficult to translate the meaning of each score into a simple metric understood by drivers, which our test drivers believed was critical to incentivize their adoption of the system. In addition, the optimality of EBkSP is not proven even if 100% penetration rate is assumed. We plan to investigate more human acceptable solutions to solve the balanced routing problem. For example, the additional delay incurred to other drivers could be defined as a more meaningful score to minimize the global travel time. Moreover, frequent taxi trajectories could be used as alternatives to incorporate taxi drivers' intelligence.

Bootstrapping the system. During the initial deployment of Themis, there might not be enough users to sense the city-scale traffic condition or collect plenty of routing requests. However, the participatory traffic sensing algorithm in §III.B and the background traffic estimation algorithm in §III.C can also be used in conjunction with external datasets. For example, many taxi companies sell their real-time trajectory data.

Since Themis reduces the global traffic volume and travel time, it could be useful to prevent congestions and reduce pollution. The government may want to incentivize drivers to participate by rewarding users who contribute to a better traffic ecosystem. For instance, users who usually take high-scored route may receive discount for their vehicle registration.

VII. CONCLUSION

This paper presents a navigation system, Themis, which supports cooperative routing algorithms by estimating their required information using the data collected by drivers who participate. Themis integrates a balanced routing algorithm and has been built as an Android application together with a server to demonstrate its feasibility. We also present a method to evaluate Themis system at different penetration rates using real trajectory dataset. City-scale synthetic evaluations using real data from 26,000 taxis demonstrate that balanced routing can reduce the average travel time in the road network and alleviate congestions. The benefit of balanced routing over greedy routing emerges even if the penetration rate is as low as 7%. In addition, the performance variations at different penetration rates consolidate the hypothesis that cooperative routing saves more if the penetration rate is higher.

ACKNOWLEDGMENT

We thank the anonymous reviewers for their valuable and insightful comments. This work was supported in part by the National Science Foundation grants CNS-1111811 and CNS-1409523, and by a Google Research Award.

REFERENCES

[1] W. Sha, D. Kwak, B. Nath, and L. Iftode, "Social vehicle navigation: integrating shared driving experience into vehicle navigation," in *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications*. ACM, 2013, p. 16.

[2] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher, "GreenGPS: a participatory sensing fuel-efficient maps application," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 2010, pp. 151–164.

[3] Ericsson ConsumerLab, "From apps to everyday situations - an ericsson consumer insight study," Tech. Rep., 2011.

[4] A. Malm, "Mobile navigation services and devices - 6th edition," Berg Insight, Tech. Rep., 2013.

[5] J. D. Murchland, "Braess's paradox of traffic flow," *Transportation Research*, vol. 4, no. 4, pp. 391–394, 1970.

[6] D. Buscema, M. Ignaccolo, G. Inturri, A. Pluchino, A. Rapisarda, C. Santoro, and S. Tudisco, "The impact of real time information on transport network routing through intelligent agent-based simulation," in *Science and Technology for Humanity (TIC-STH), 2009 IEEE Toronto International Conference*. IEEE, 2009, pp. 72–77.

[7] T. Roughgarden, *Selfish routing and the price of anarchy*. MIT press, 2005.

[8] T. Yamashita, K. Izumi, and K. Kurumatani, "Car navigation with route information sharing for improvement of traffic efficiency," in *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, Oct 2004, pp. 465–470.

[9] D. Wilkie, J. P. van den Berg, M. C. Lin, and D. Manocha, "Self-aware traffic route planning." in *AAAI*, 2011.

[10] S. Lim and D. Rus, "Stochastic distributed multi-agent planning and applications to traffic," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 2873–2879.

[11] R. Kanamori, J. Takahashi, and T. Ito, "Evaluation of anticipatory stigmergy strategies for traffic management," in *VNC*, 2012, pp. 33–39.

[12] P.-J. He, K.-F. Ssu, and Y.-Y. Lin, "Sharing trajectories of autonomous driving vehicles to achieve time-efficient path navigation," in *VNC*, 2013, pp. 119–126.

[13] A. van den Bosch, B. van Arem, M. Mahmod, and J. Misener, "Reducing time delays on congested road networks using social navigation," in *Integrated and Sustainable Transportation System (FISTS), 2011 IEEE Forum on*. IEEE, 2011, pp. 26–31.

[14] J. Pan, M. A. Khan, I. S. Popa, K. Zeitouni, and C. Borcea, "Proactive vehicle re-routing strategies for congestion avoidance," in *Distributed Computing in Sensor Systems (DCOSS), 2012 IEEE 8th International Conference on*. IEEE, 2012, pp. 265–272.

[15] B. Greenshields, J. Bibbins, W. Channing, and H. Miller, "A study of traffic capacity," in *Highway research board proceedings*, vol. 1935. National Research Council (USA), Highway Research Board, 1935.

[16] Board of Public Roads, "Traffic assignment manual for application with a large, high speed computer," Tech. Rep., 1964.

[17] J. Aslam, S. Lim, and D. Rus, "Congestion-aware traffic routing system using sensor data," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*. IEEE, 2012, pp. 1006–1013.

[18] P. Newson and J. Krumm, "Hidden markov map matching through noise and sparseness," in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2009, pp. 336–343.

[19] J. Aslam, S. Lim, X. Pan, and D. Rus, "City-scale traffic estimation from a roving sensor network," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. ACM, 2012, pp. 141–154.

[20] "OsmAnd Maps and Navigation," http://osmand.net/.

[21] "OpenStreetMap," http://www.openstreetmap.org/.

[22] "PostgreSQL," http://www.postgresql.org/.

[23] "pgRouting," http://pgrouting.org/.