Crowdsourcing Phase and Timing of Pre-Timed Traffic Signals in the Presence of Queues: Algorithms and Back-End System Architecture

Seyed Alireza Fayazi and Ardalan Vahidi

***Abstract*—This paper describes a crowdsourcing-based system for phase and timing estimation of pre-timed traffic signals. The input crowd is a real-time feed of sparse and low-frequency probe vehicle data, and the output is an estimated collection of Signal Phase and Timing (SPaT) information. The estimations could be ultimately fed into a connected vehicle's driver assistant application. Different from the authors' previous work, the approach described in this paper ensures the accuracy of the SPaT estimations even in the presence of queues. This was achieved by investigating the probe data influenced by the heavy traffic and the delay in queues. This paper is also a sequel to the authors' previous work as it provides an in-depth overview of the crowdsourcing algorithms and their back-end implementation. The accuracy of the crowdsourcing algorithm is also experimentally evaluated for a selection of pre-timed traffic lights in San Francisco, CA, USA, by utilizing a real-time data feed of San Francisco's public buses as an example data source.**

***Index Terms*—Traffic lights, signal timing estimation, queueing delay estimation, probe vehicles, crowdsourcing, traffic information mining.**

I. Introduction

CONNECTED vehicle environment enables the vehicles equipped with computing and wireless communication devices to receive Signal Phase and Timing (SPaT) information of traffic lights. Many in-vehicle applications have the potential to benefit from Signal Phase and Timing (SPaT) data in order to achieve better fuel efficiency, emission control, and safety features [1]–[4]. The Velocity Advisory Systems [1], [2], and Start/Stop systems [3] are such applications with fuel efficiency benefits reported in [5]–[7]. Also a Collision Avoidance System [4] can benefit from SPaT information in order to foresee potential signal violations at signalized intersections. In addition to in-vehicle applications, there are also many arterial performance measurement methods that use SPaT as their input [8], [9].

The main challenge in providing SPaT to aforementioned in-vehicle applications and arterial performance measurement methods is in finding an inexpensive and reliable data source. Because direct access to signal timings and real-time state of lights is not available for all the traffic lights across a country, a complementary solution is needed in situations where the phase and timing information is not available directly from a city's Traffic Management Center (TMC).

An increasing interest in obtaining SPaT is obvious in recent years. In [10] the probability of a light being green is found over a planning horizon but by assuming that the baseline timings and schedules are available. Using the cameras of windshield-mounted mobile phones, Koukoumidis *et al.* in [11] present a speed advisory service that detects the current phase of signals; however, the mobile nodes need a database from TMC for the settings of fixed-time signals. In [12], Zhu *et al.* use the maximum a posteriori (MAP) estimation and an optimization algorithm to estimate the state of a traffic light. The system's capability in estimating the next phase-change of a signal (which is needed for most in-vehicle applications) was not specifically addressed by the authors in [12].

The goal of this paper is to obtain deterministic knowledge of SPaT information that is applicable to pre-timed signals using only low-frequency probe data. As an example data source, this paper uses the public feed of the San Francisco's GPS-enabled buses provided by NextBus Incorporated [13]. This input probe data includes the GPS coordinates, and velocities of the public buses at timestamps. The feasibility of gathering traffic signal data from these crowdsourced probe vehicle data is demonstrated in authors' previous work [14]. The obtained SPaT is ultimately fed into the in-vehicle applications in situations where no data is available via TMC.

To the best of authors' knowledge, to date, three works by Kerper *et al.* [15], Cheng *et al.* [16], and Chuang *et al.* [17] are deterministic approaches related to our proposed approach, although, those require high frequency probe data sets. The simulation results in [15], [16] are based on the assumption that the penetration level is high, and full velocity profiles of vehicles are available via high-frequency probe data. However, the results in this paper are achieved under low penetration of probe data: probe vehicles are buses that arrive on average once every 5–10 minutes during day times.

Chuang *et al.* in [17] use smartphones installed in vehicles to collect velocity profiles at a sampling rate of 1 Hz. In [17], the crowdsourcing part is directly implemented on smartphones which decreases the number of reporting events; however the impact on smartphone battery usage was not addressed in [17]. Although the probe data stream used in this paper is not sent from smartphones of individual contributors, we expect that

Manuscript received February 22, 2015; revised June 27, 2015 and August 18, 2015; accepted October 10, 2015. Date of publication November 10, 2015; date of current version February 26, 2016. This work was supported by a research award from the BMW Information Technology Research Center, Greenville, SC, USA. The Associate Editor for this paper was Q. Zhang.

The authors are with the Department of Mechanical Engineering, Clemson University, Clemson, SC 29634 USA (e-mail: sfayazi@g.clemson.edu; avahidi@clemson.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2015.2494045

the battery usage of a smartphone implementation be minimal because not only no crowdsourcing application needs to run on the smartphones, but also we have demonstrated that our crowdsourced-based engine only needs a low-frequency data stream given that each probe vehicle (or smartphone) sends an update only sporadically (every 200 m or 90 seconds).

Furthermore, the aforementioned studies are only applicable to signals that their timings are fixed during the day. Nevertheless, the SPaT estimation proposed in this paper also operate on pre-timed signals that may have different timings for day segments such as A.M.-peak, P.M.-peak, and off-peak.

The SPaT estimation in authors' previous work [14] did not consider the influence of queue delay, and it was based on filtering out the vehicle passes that appeared to be influenced by heavy traffic and long queues. This is why the position in queue was not considered in [14]. However, a considerable number of vehicle passes occur in heavy traffic and excluding them will negatively influence accuracy of SPaT estimation algorithms. In [14] many of the movements during heavy traffic period were filtered, reducing the number of available data points, causing estimation errors during heavy traffic conditions. This is why the approach of [14] is suitable for intersections that have light traffic condition with occasional short periods of heavy traffic throughout the day.

This paper first explains the architecture of the computational back end that processes the incoming crowdsourced data, estimates SpaT information and broadcast it to subscribing vehicles. The general crowdsourcing mechanisms are presented in Section III, and the detailed crowdsourcing methodologies are explained with estimation results in Section IV. Section V describes how SPaT estimations are affected by idling periods in queue. Finally, more ground truth verifications are provided to evaluate the SPaT estimations, and the queue dissipation formulations.

II. Back-End Architecture

If traffic signals of a city are connected to TMC, then access to their real time state may be granted either directly from local and federal entities, or indirectly through third party data providers. Nevertheless, what we are proposing in this manuscript as a crowdsourced-based SPaT estimator is a complementary solution in situations where timing information is not available directly from a city's Traffic Management Center. The implemented system, as shown in Fig. 1, is actually capable of receiving SPaT directly from TMC as well as from a crowdsourcing server.

The input data source of the crowdsourced-based SPaT estimation is probe vehicle data which can be gathered from vehicles of any kind reporting at least their GPS coordinate and velocity at a timestamp, as long as the location privacy of contributing vehicles is preserved. This input data feed is collected by the Crowdsourcing Server, as shown in Fig. 1. The data is then recorded in a SQL database so that the same server can access it to estimate a collection of traffic signal phase and timing information (SPaT) including cycle length, phase length, green-initiation (start-of-green), and signal schedule changes. The traffic signal information of each phase of each intersection (Intersection–Phase pair) is then accessed by a Cloud-based Server, specially configured for the infrastructure-to-vehicle (I2V) communications via wireless cellular networks, such as 4G/LTE. In fact, the Cloud-based Server allows for fluctuations in number of connected vehicles requesting SPaT information. A Web Server is also set up for initialization, maintenance and ground-truth verification purposes.

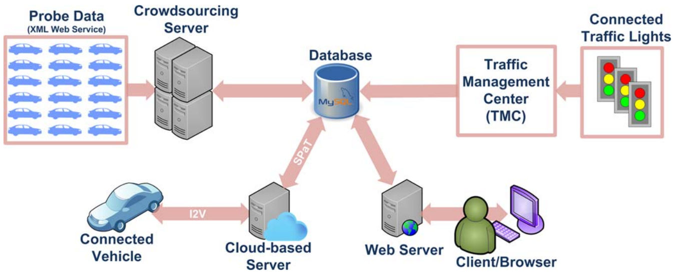


Fig. 1. The overall hardware architecture of the back-end system.

III. Crowdsourcing

The output of the crowdsourcing engine consists of two parts: First is traffic signal baseline timing that includes cycle time, phase lengths (red and green intervals), and signal offset changes. Second is phase-change (sync) data, that is green-initiation or start-of-green. Here we explain the general crowdsourcing mechanisms that the Crowdsourcing Server uses to predict and estimate this collection of traffic signal information, as shown in Fig. 2. After being initialized, the Crowdsourcing Server goes through three processes which are separated by dashed lines in Fig. 2:

- Data Collection, in which probe vehicle data is continuously collected and stored in the MySQL database.
- High-Frequency Process (Phase-Change Prediction), in which only the green-initiations are predicted with high frequency. In fact, the green-initiation prediction is the process of predicting the next transition to green; and because of the clock drift of a traffic signal throughout a day, the next green-initiations should be continuously predicted based on the most recent probe data. In our application, every time the execution of this process cycle begins, the most updated probe data collected during the last few hours is first retrieved from the MySQL database, as shown in Fig. 2. After preprocessing this data, the green-initiations of each Intersection–Phase are predicted and finally stored in the MySQL database.
- Low-Frequency Process (Baseline Timing Estimation), in which the traffic signal baseline timings including cycle time, red and green intervals, as well as signal schedule changes are estimated. Based on the assumption that the penetration level is low, this process is executed with very low frequency (once per month in our application). The drawback is that when a traffic signal is re-timed, it takes a fairly long time to have an accurate baseline timing estimation. This is because a large number of probe vehicle passes needs to be recorded after re-timing

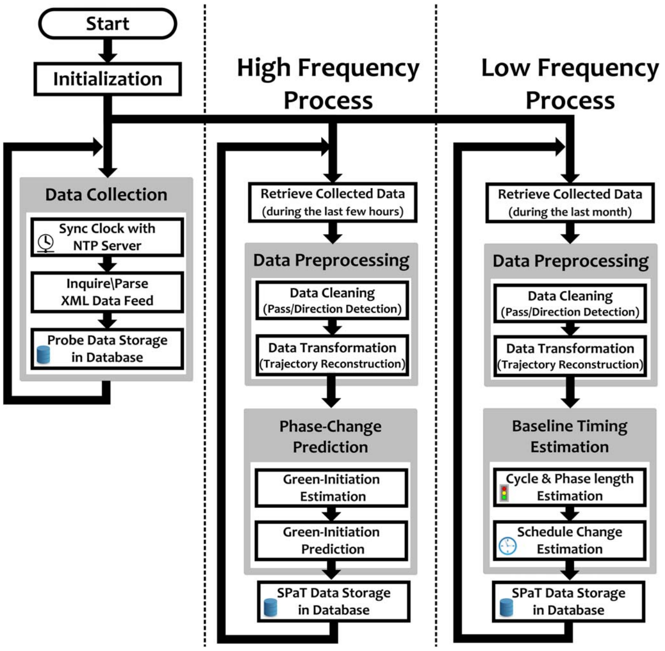

Start
Initialization
High Frequency Process
Low Frequency Process
Data Collection
Sync Clock with NTP Server
Inquire\Parse XML Data Feed
Probe Data Storage in Database
Retrieve Collected Data (during the last few hours)
Data Preprocessing
Data Cleaning (Pass/Direction Detection)
Data Transformation (Trajectory Reconstruction)
Phase-Change Prediction
Green-Initiation Estimation
Green-Initiation Prediction
SPaT Data Storage in Database
Retrieve Collected Data (during the last month)
Data Preprocessing
Data Cleaning (Pass/Direction Detection)
Data Transformation (Trajectory Reconstruction)
Baseline Timing Estimation
Cycle & Phase length Estimation
Schedule Change Estimation
SPaT Data Storage in Database

Fig. 2. The functional architecture of the Crowdsourcing Server.

takes place. However, the traffic signals are re-timed infrequently. Also, as more contributors share their location in the future, a large number of probe vehicle passes can be collected in a considerably shorter time.

IV. Methodologies

This section presents the algorithms applied in the aforementioned crowdsourcing processes. First, the basic steps, named in Fig. 2 as Initialization, Data Collection, and Data Preprocessing, are described. Finally, all the algorithms involved in the SPaT estimations and predictions are explained along with results.

A. Initialization

The Crowdsourcing Server initializes its objects and variables once it's fired up. The initial values such as the geometry of the desired intersection–phase pairs should be predefined by either the web user or the server administrator. In fact, the coordinates of three points should be defined by the user in order to define and initialize every Intersection–Phase pair. The three-point definition method covers all the possible movements at intersections. This definition includes one point on the upstream, one point on the downstream, and a middle point at the intersection center. As an example, Fig. 3(a) and (b) shows through movement and left turn definitions respectively, although using the three-point definition, other movements can also be defined in different directions and in intersections with different geometries. It should be emphasized that a right-turn is usually permitted during the through movement, and an intersection is less likely to have a protected right turn. However, if needed, a protected right turn can be covered by the proposed estimation process only by defining three points. Note that with additional algorithms, not described in this paper, it would be possible to automatically crowdsource the geometry instead of manually defining it.

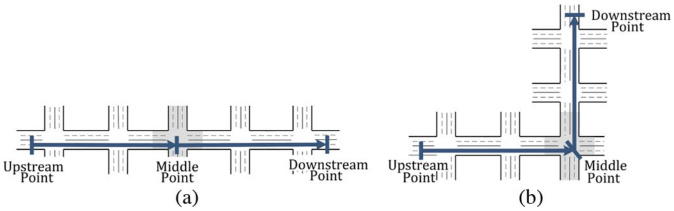


Fig. 3. Three-point definition of an intersection–phase. (a) Through movement. (b) Left turn. (The intended intersection is shown in shaded color.)

B. Data Collection

1) Data Feed: A public feed of bus location and velocity data in the city of San Francisco is used here to crowdsource the traffic signal information. The feed is provided by NextBus Incorporated through eXtensible Markup Language or XML [18] which can be accessed using URLs with parameters specified in the query string [13]. Each vehicle (bus) sends a probe update every 200 meters approximately or 90 seconds, whichever comes first [14]. The communication load imposed by this data feed is minimal given that the penetration level is also low. As higher frequency and larger number of probe reports becomes available in the future from more contributors, more accurate estimates of parameters of traffic signals can be obtained. However, beyond a point, the accuracy improvement ceases to be worth the extra communication costs [19].

As shown in Fig. 2, the Data Collection process periodically inquires the XML feed data of each route. It is crucial to set this process in such a way that its clock is automatically synchronized to a Network Time Protocol (NTP) server; in this work the clock is synchronized with the NIST time server [20] every 10 minutes.

2) Location Data Privacy: Although collecting location data from public buses eliminates the privacy concerns, privacy protection precautions need to be implemented if location data is provided to our system by individuals through their smartphones for example. Based on the success of some location-based applications it is possible to conclude that at least some people would be willing to automatically share their locations with third parties [21]. However, privacy breach should be prevented against all location samples stored in databases. It means that not only all location samples must be anonymized but also it must not be possible for any intruder to reconstruct individual traces.

What we store in our database as location data are the vehicle passages over the desired intersections. Each passage actually consists of few probe reports sent within the three-point definition as previously shown in Fig. 3. The following measures makes it almost impossible for an intruder to reconstruct full traces by looking for correlations between the recorded passages: 1) We do not store vehicle identification numbers on our server. 2) Only passages over arterial roads are recorded. No location data is recorded on other areas such as highways. 3) Passages over intersections without a traffic signal are not recorded. 4) No location data is recorded if a passage cannot be fitted into any of predefined desired trajectories. Even if they fit then at most three probe reports are recorded: Two reports sent before and after the stop-bar of the intended intersection; and one report sent while waiting in a queue, if available.

Even in a worst-case scenario that a probe vehicle's trace from origin to destination happens entirely in an urban area and includes successive signalized intersections and surprisingly all her individual traces at intersections fits to our desired trajectories, only the origin and destination intersections can be identified (not the accurate positions of origin\destination). Another privacy precaution that can be added to hide the sensitive places that drivers have visited is to avoid crowdsourcing SPaT information for the intersections that are near specific privacy sensitive locations (see [22] for a similar approach on virtual trip lines).

C. Data Preprocessing

The first step in data preprocessing is Data Cleaning which consists of identifying the useful probe data to be mined. The second step prepares the probe data for possible use in SPaT estimations; this step is named Data Transformation and actually transforms the single probe updates to travel trajectories which are desirable for SPaT estimation purposes. These two steps can be seen in Fig. 2 and are described as follows:

1) Data Cleaning: The Data Cleaning in this manuscript refers to the process of identifying the desired probe data out of the collected probe data. This process consists of: (1) identifying the probe vehicle reports that are within the three-point definition, (2) detecting and separating each pass of each vehicle, and finally (3) discarding the vehicle passes not on the desired direction.

As described in Section IV-A, each three-point definition includes upstream and downstream parts. Here, we are interested in identifying the probe vehicle reports that have happened within either the upstream or downstream part. Fig. 4 shows the upstream part as an example; where d_{upstream} is the distance of a probe report to the upstream point, d_{middle} is the distance of the same probe report to the middle point, and L_{upstream} is the length of the upstream part. It is obvious that if the condition $L_{\text{upstream}} = d_{\text{upstream}} + d_{\text{middle}}$ is satisfied then it can be concluded that the vehicle had sent the location report exactly on the straight line between the upstream and middle points. However, it is less likely for the vehicle to be exactly on the straight line between the two points, especially on wide streets. Because of this and the inevitable error in GPS position reports, the following conditions are verified instead:

$$
\begin{aligned}
d_{\text{upstream}} + d_{\text{middle}} &< L_{\text{upstream}} + \Delta L_1 \\
d_{\text{upstream}} &< L_{\text{upstream}} + \Delta L_2 \\
d_{\text{middle}} &< L_{\text{upstream}} + \Delta L_2
\end{aligned} \tag{1}
$$

where ΔL_1 and ΔL_2 are values added to account for the street width as well as for the errors in the probe vehicle reports. A similar approach is used to verify whether a probe report is within the downstream part or not.

Finally, the distinct passes of vehicles are detectable due to the fact that each probe vehicle report is labeled with a random vehicle ID number. And the direction of a distinct vehicle pass is detectable by inspecting the distance between the corresponding probe location reports and the upstream point of the intended Intersection–Phase.

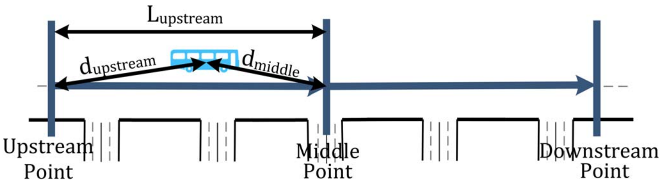
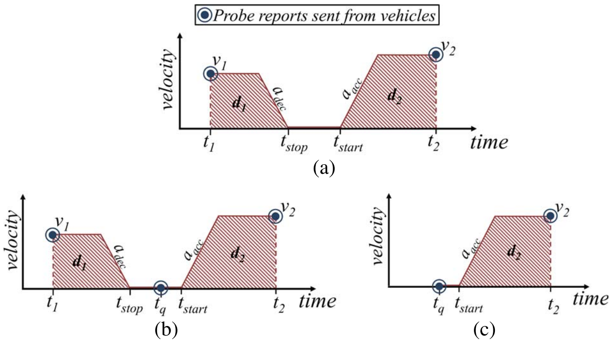


Fig. 4. Identifying if a probe vehicle report is sent around the intended Intersection–Phase located at the middle point. A similar approach is repeated for the downstream part of the Intersection–Phase.



(a)

(b)

(c)

Fig. 5. The probe reports fitted to the desired velocity-vs-time trajectories. (a) Full trajectory with a stop at red but not influenced by delay in queue. (b) Full trajectory with a stop at red and with a probe report sent in queue. (c) Partial trajectory with a stop at red and with a probe report sent in queue.

As a note on the values assigned for ΔL_1 and ΔL_2, a high precision (very small values for ΔL_1 and ΔL_2) may cause some missing data. For example, a probe vehicle report in a wide street, far from the two upstream and middle points may be ignored as it does not satisfy all the aforementioned conditions of (1). On the other hand, a low precision (higher values for ΔL_1 and ΔL_2) may cause some duplicate data. For example, a probe vehicle report close to the middle point of Fig. 4 may be considered as a report in both upstream and downstream parts. As a result, it is better to avoid a high precision and then discard the duplicate detected reports. As an example, for a street of width ≈9 meters, the values of $\Delta L_1 = 9$ meters and $\Delta L_2 = 2$ meters were assigned.

2) Data Transformation: There should be sufficient probe data points in an identified vehicle pass for SPaT estimations. But because the utilized probe data is sparse and the consecutive data points of each pass are far away from each other, we need to approximate a vehicle trajectory between each two probe reports. This is actually a data transformation process where low frequency probe data are transformed and consolidated into vehicle trajectories.

The most beneficial trajectory for our purposes is the trajectory which includes a stop at red signal. Fig. 5(a)–(c) demonstrates the reconstructed velocity-time trajectories that include such a stop; and they may be used to estimate the green-initiation, the red interval, the cycle time, and perhaps more. There is also statistical patterns in travel trajectories with no stop and with constant acceleration [14].

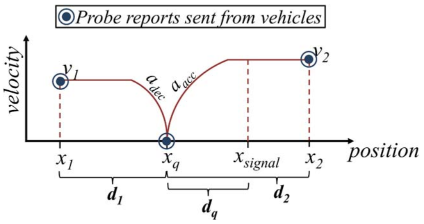


Fig. 6. Equivalent velocity-vs-position trajectory. (applicable to Fig. 5(b) and also to Fig. 5(a) and (c) with modifications).

The Data Transformation process first checks the consistency of the previously identified probe vehicle passes with the trajectories shown in Fig. 5. If successful then the process records the estimated trajectory to be used later by SPaT estimation processes.

Each identified vehicle pass is a series of three-tuple $[v_i, t_i, x_i]$ shown by filled circle points in trajectories of Fig. 5; where v_i is the reported velocity, t_i is the timestamp of the report, and x_i is the distance between each location report and the upstream point of the Intersection–Phase calculated using Haversine Formula [23]. The points with index $i = 1$ are the reports sent within the three-point area and right before the stop-bar of the intended intersection; and the points with index $i = 2$ are the reports sent within the three-point area and right after the stop-bar of the intended intersection. The reports sent while waiting in a queue, if available, are also denoted with index q. Fig. 5(b) and (c) have both a probe report sent in queue with approximately zero reported velocity.

Each reconstructed travel trajectory consists of a vector of $[\text{velocity}, \text{time}, \text{position}, a_{\text{acc}}, a_{\text{dec}}]$ where a_{acc} and a_{dec} are the average acceleration and deceleration as denoted in Fig. 5. We use deceleration and acceleration of 2.2 m/s^2 and 1.0 m/s^2 respectively as obtained in [14].

The equivalent velocity-position trajectory is demonstrated in Fig. 6. The variable x_q, if available, is the reported position of the vehicle while waiting in queue and x_{signal} is the location of light or more specifically stop-bar. The variables $d_1 = x_q - x_1$ and $d_2 = x_2 - x_q$ are areas under the velocity-time curve, and d_q is *position in queue*. It should be emphasized that *position in queue* is different from the term *queue length* used in literature such as in [24].

The major extractable information from reconstructed trajectories is the time that a waiting vehicle starts moving at green (t_{start}) and the time that a moving vehicle comes to a stop at red (t_{stop}) estimated as follows:

$$t_{\text{start}} = t_2 - \max\left\{\frac{d_2}{v_2} - \frac{v_2}{2a_{\text{acc}}}, 0\right\} - \frac{v_2}{a_{\text{acc}}} \tag{2}$$

$$t_{\text{stop}} = t_1 + \max\left\{\frac{d_1}{v_1} - \frac{v_1}{2a_{\text{dec}}}, 0\right\} + \frac{v_1}{a_{\text{dec}}} \tag{3}$$

where the function max(.) in (2) decides whether the estimated trajectories of Fig. 5(a)–(c) include a constant velocity movement after the vehicle accelerates at green-initiation, and the function max(.) in (3) decides whether the estimated trajectories of Fig. 5(a) and (b) include a constant velocity movement before the vehicle comes to a stop.

TABLE I
COUNTS OF PASSES DURING ONE YEAR AT FOUR INTERSECTIONS

	Total Identified passes	Pass counts fittable to Fig. 5 (a)*	Pass counts fittable to Fig. 5 (b)	Pass counts fittable to Fig. 5 (c)
Lombard Intersection	49361	4476 (9.07%)	428 (0.87%)	5295 (10.73%)
Green Intersection	49343	1532 (3.10%)	103 (0.21%)	897 (1.82%)
Vallejo Intersection	49325	4468 (9.06%)	221 (0.45%)	1553 (3.15%)
Broadway Intersection	46938	3225 (6.87%)	153 (0.33%)	5543 (11.81%)

* passes captured by [14].

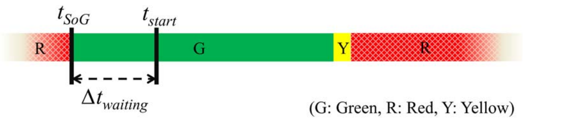


Fig. 7. $\Delta t_{\text{waiting}}$ is the waiting time for moving after the green-initiation or Start-of-Green (t_{SoG}).

Table I reports the percentage of total vehicle passes in a year which are fittable into the desired trajectories. It can be seen that by not ignoring the probe data that are influenced by queue delay, a larger number of the vehicle passes will be available to the SPaT estimation algorithms.

It should be emphasized that if a green wave is already implemented along an arterial road then the successive signals switch to green light as a backward-propagating wave. This has significantly decreased the percentage of the vehicles that their passage fits into the desired trajectories with stop at red.

D. SPaT Estimation and Prediction

The goal of this section is to estimate signal timing considering the probable delay in queue. The methods described here cover the Phase-Change Estimation\Prediction, and Baseline Timing Estimation executed within the Crowdsourcing Server (shown in Fig. 2).

1) Green-Initiation Estimation: When a traffic light changes to green, drivers should wait for the queue in front to move before they can start moving. As shown in Fig. 7, this waiting time is denoted by $\Delta t_{\text{waiting}}$ in this paper.

As a result, based on estimates of $\Delta t_{\text{waiting}}$ and t_{start}, the green-initiation (t_{SoG}) can be estimated as:

$$t_{\text{SoG}} = t_{\text{start}} - \Delta t_{\text{waiting}}. \tag{4}$$

However, depending on which travel trajectory the probe data can be fitted to, there are two approaches to estimate green-initiation:

- First approach uses the probe data that appears to be less influenced by heavy traffic and delay in queue. For this reason only the probe reports fittable to the trajectory of Fig. 5(a) with high upstream velocity (v_1) are selected to estimate the time t_{start}. Because it is assumed that there is no long queue in front, an average value of $\Delta t_{\text{waiting}}$ for

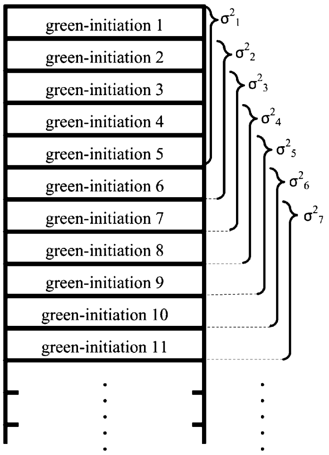


Fig. 8. Extracting the variance (σ^2) trajectory from the green-initiations.

the first few vehicles in queue is used in green-initiation estimation (6 seconds in our application for transit buses).

- Second approach uses the probe reports that reveal the position of the probe vehicle in queue. This actually includes the reports that are fittable to the trajectories of Fig. 5(b) and (c) where at least one probe report sent while in queue is available. Knowing the position of the vehicle in queue, the queue waiting time $(\Delta t_{\text{waiting}})$ can be estimated using the queue clearance time model explained later in Section V. This approach is expected to be more accurate than the first one in persistent heavy traffic conditions, mainly because actual position in queue is available.

2) Signal Schedule Change Estimation: An offset (t_{offset}) is usually added to the timings during the rush hour schedule. The approach described below estimates not only the time that the schedule of a traffic light changes but also the offset that is added to the timings.

First, the green-initiations estimated using the second approach of previous subsection are sorted based on the weekday and the time of the day. Then, the variance of the average (σ^2) of few consecutive green-initiations (e.g. 5 green-initiations) is calculated; and according to Fig. 8 this process is repeated for the next consecutive green-initiations till the whole list of green-initiations of each weekday is covered. By putting the calculated variances together, a trajectory is constructed which demonstrates the change of the variance with respect to time of each week-day (see Fig. 9 for a sample plot of the variance trajectory).

The spikes in the variance plot of Fig. 9 actually show the times that a timing schedule is changed. The more probe data is used, the sharper the spikes are. As a result, the probe data collected of almost ten months is used in depicting Fig. 9, although one month of collected probe data is also enough to have detectable schedule change spikes. The variance trajectories of [14] include some extra and misleading large spikes due to heavier traffic in the middle of rush hour; however, considering the influence of queue waiting time on SPaT estimation, all the spikes in this paper are solely the results of the schedule change.

The value of the offset (t_{offset}) imposed on the timings at schedule change can be extracted by comparing the green-initiation estimations before and after the schedule change.

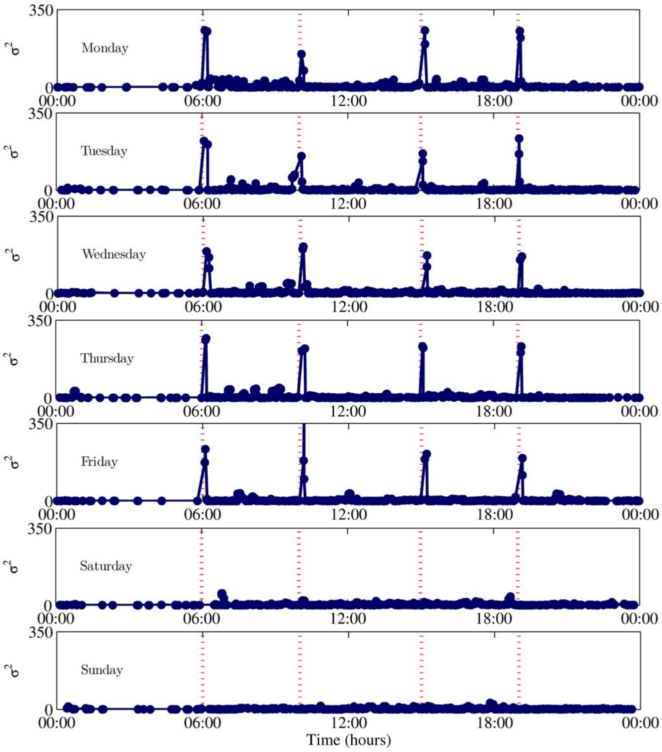


Fig. 9. The variance of estimated green-initiations for Lombard intersection. The actual schedule-changes happen at the dashed vertical lines which are comparable to the jumps in the trajectories.

3) Green-Initiation Prediction: The green-initiation prediction is the process of predicting the next transition to green. Because of the clock drift throughout a day, the next green-initiations should be continuously predicted using the most recent reconstructed trajectories of vehicles that accelerate at green. As a result, a moving average of the most recent estimated green-initiations $(\bar{t}_{\text{SoG}})$ is used to predict the next transition to green. However, three adjustments should be conducted before averaging the estimated green-initiations:

- First, the green-initiation estimations should be adjusted by the estimated offset if they happened during the rush hour schedule change. This is done by simply adding t_{offset} to the estimated t_{SoG}. In this way, all the green-initiation estimations are synchronized to a same time reference.
- Second, the adjusted green-initiation estimations are mapped to one cycle interval before being averaged.
- Third, a filter is used to filter out the outliers and wrong estimations. The filter works based on the fact that the smaller the variance of $\bar{t}_{\text{SoG}}$ average is, the more accurate prediction is expected. As a result, having n latest estimated green-initiations, we propose to calculate $\bar{t}_{\text{SoG}}$ using all possible combinations of $k \leq n$ samples and select the one that produces the minimum variance. In mathematical language, let t_i be the latest green-initiation estimation which is adjusted and mapped based on the aforementioned steps. And let S be the set of n-recent green-initiation estimations as:

$$S = \{t_{i-n+1}, t_{i-n+2}, \ldots, t_{i-1}, t_i\} \quad (5)$$

Let s be the set of k-subsets of S so that s has $\binom{n}{k}$ elements as follows:

$$|s| = {}_nC_k = \text{k} - \text{combinations of set } S \qquad s \subseteq S,\ k \in \mathbb{Z} \tag{6}$$

As a result, s can be partitioned into unique subsets of $s_1, s_2, \ldots, s_{|s|}$ where:

$$s_j = \{(t_{s1}, t_{s2}, \ldots, t_{sk}) \in S | t_{i-n+1} \leq t_{s1} < t_{s2} < \cdots < t_{sk} \leq t_i\} \tag{7}$$

Finally, one of the subsets s_j $(j = 1, 2, \ldots, |s|)$ with minimum variance of average is chosen and its average value is used to predict the next transition to green.

One of the outliers that we expect the filter to eliminate from set S before averaging is any green-initiation estimated from the stop-and-go event that has no correlation with traffic signal changes, for example stopping for passengers at bus stop. The delay in bus stops needs to be studied in depth and remains for future work. Nevertheless, according to the proposed filter design, we expect the filter to eliminate the bus stop correlated stop-and-go events if they lack the cyclic periodicity which can be found in traffic light correlated stop-and-go events. The results on green-initiation prediction are demonstrated later in Section VI-A.

4) Red Split Estimation: The duration of red "observed" by a particular vehicle can be calculated using the trajectories of Fig. 5(a) and (b) as:

$$t_{\text{red}} = t_{\text{SoG}} - \left(t_{\text{stop}} - \frac{v_1}{a_{\text{dec}}}\right) \tag{8}$$

where v_1/a_{dec} is the deceleration time after a driver detects the signal is red. The trajectory of Fig. 5(c) is also used in calculating the observed red interval by verifying (9) if $t_{\text{SoG}} > t_q$; where t_q is the timestamp of the zero-velocity probe data sent while waiting in queue, i.e.,

$$t_{\text{red}} = t_{\text{SoG}} - t_q \tag{9}$$

Fig. 10 shows scatter plots of t_{red} calculated using the aforementioned equations for four intersections. It is expected that the maximum of the aggregated calculations would be actually an upper bound estimate to duration of the actual red phase. Please note that the red split scatter plots at Vallejo and Broadway intersections shown in Fig. 10 are spread out compared to that of Lombard an Green Intersections. One reason for this could be the bus stop right after Vallejo intersection and the two bus stops before and after the Broadway intersection. There is no bus stop around Lombard and Green.

5) Red-Probability Estimation: This section demonstrates how to extract the probability distribution of red signal by aggregating the reports that are sent from the vehicles waiting in queue at red. For this purpose, the timestamps of the zero-velocity reports that has been sent while waiting in queue are collected [more specifically the t_q timestamps of Fig. 5(b) and (c)]. Nevertheless, the t_q timestamps do not necessarily denote the times at which the signal is red; and the condition of $t_q < \bar{t}_{\text{SoG}}$ should be verified before collecting t_q as a sample corresponding to red signal.

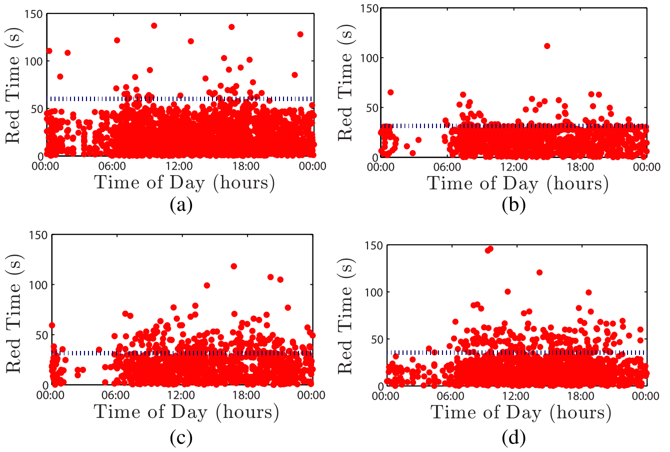


Fig. 10. The red time observed by vehicles throughout the day of ten months at intersections along VanNess St. (the actual intervals were available through city timing cards and are shown by dashed horizontal lines). (a) Lombard intersection. (b) Green intersection. (c) Vallejo intersection. (d) Broadway intersection.

In order to synchronize all the collected t_q timestamps to a same time reference, the estimated offset (t_{offset}) is added to the timestamps that has occurred during the rush hour schedule change. Before aggregating the adjusted timestamps, all the timestamps should be mapped to one cycle interval by (10) where C is the cycle time estimated by the method explained in [14], i.e.,

$$t_{\text{q,mapped}} = t_q - \text{round}\left(\frac{t_q}{C}\right) C. \tag{10}$$

Equation (10) maps all timestamps of t_q onto a reference interval of $[0, C]$ in Unix-Time. These mapped timestamps are all aggregated and can be plotted in polar histograms such as Fig. 11. As shown in this figure, the interval $[0, C]$ can be mapped to an interval of $[0, 2\pi]$ because of the cyclic periodicity. The longer each triangle of histograms is, the more red samples it includes. The shaded portions of the cycle time are the actual red intervals which are depicted according to the city timing cards and the ground truth observations. The histograms represent the probability distributions of red intervals which have much stronger concentration of mapped red-samples in the shaded portions (actual red intervals). However, moving counter clockwise on the polar plots, a short time span can be identified at the beginning of red phase with a very few red samples. This makes sense, because there are no stopped vehicles at the instant that light changes to red. Even if there are, then it is more probable to receive a zero-velocity report from them anytime later waiting at red phase than at the beginning of the red. The method proposed here to extract theses distributions completes the method of our previous work in extracting the probability of green in [14].

V. Queue Waiting Time

As explained in Section IV-D1, the key feature of the SPaT estimator is inclusion of an estimate of the wait time in queue

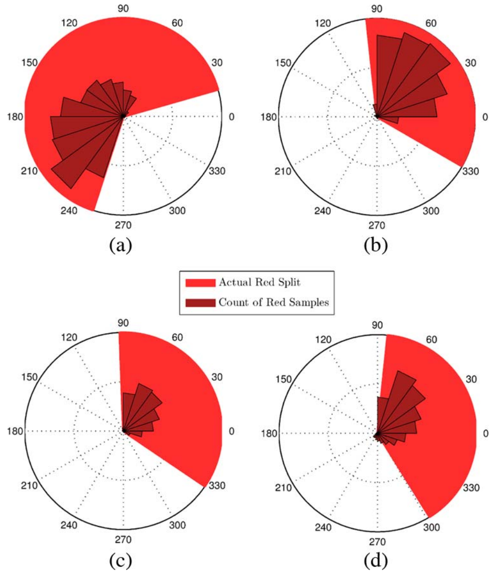


Fig. 11. Polar histogram of the red signal timestamps compared to the actual red splits (the data was collected for ten months at four intersections along Van Ness street).

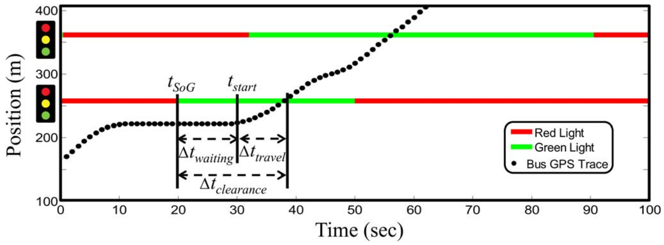


Fig. 12. The time–space diagram of a probe vehicle (bus) waiting for moving after the start of the green signal; the clearance time is the time from $t_{\rm SoG}$ to the instant at which the probe vehicle passes the stop-bar.

after green-initiation. The following subsections describe: how to formulate this waiting time based on the expected queue clearance time, how to find an estimate of the queue clearance time via a queue discharge model, and how to estimate the unknown parameters of the model.

A. Waiting Time Formulation

Let's assume that a probe vehicle is the Nth vehicle waiting in a queue at red, as shown in the time–space diagram of Fig. 12. Then, the clearance time, denoted by $\Delta t_{\rm clearance}$, is the discharge time that it takes all of the N waiting vehicles to pass and leave the stop-bar after the start of the green signal. However, as it is plotted in Fig. 12, the clearance time of N queued vehicles consists of two parts: the waiting interval that it takes the Nth vehicle to start moving after green-initiation ($\Delta t_{\rm waiting}$) plus the interval that it takes that vehicle to travel all the way up to the stop-bar and cross the stop-bar ($\Delta t_{\rm travel}$). As a result, the following formulation is proposed here to estimate the waiting time in queue:

$$\Delta t_{\rm waiting} = \Delta t_{\rm clearance} - \Delta t_{\rm travel}. \tag{11}$$

With the queue clearance time and the estimated travel time in hand, then it is quite straightforward to compute the waiting time and finally the green-initiation ($t_{\rm SoG}$). This is demonstrated in Fig. 13 where the expected $\Delta t_{\rm clearance}$ is derived from a model proposed in the following subsection; and $\Delta t_{\rm travel}$ is estimated[1] using Fig. 6.

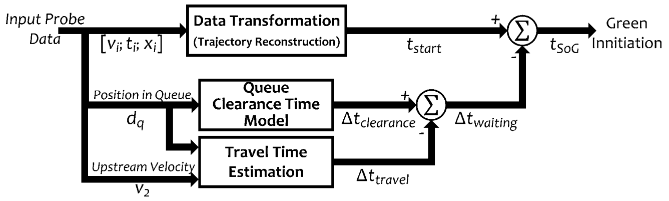


Fig. 13. Green-initiation estimation knowing the position of the probe vehicle in queue.

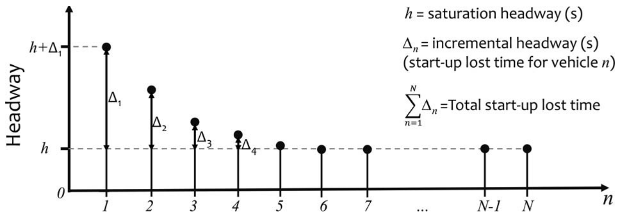


Fig. 14. Average discharge headway.

B. A Model for Queue Clearance Time

The clearance time of queued vehicles can be represented by the summation of discharge headways. Fig. 14 shows the average discharge headway, according to the specifications given in [25]. The Headway$(n = 1)$ is the interval between green-initiation and the time that rear wheels of first vehicle cross the stop-bar, the Headway$(n = 2)$ is the interval between the first vehicle and the second vehicle leaving the stop-bar, and so on. As a result, the summation of headways, as introduced in [26] and as given in (14), equates to the time from green-initiation to the instant at which the Nth vehicle of the queue crosses the stop line, i.e.,

$$\Delta t_{\rm clearance} = \sum_{n=1}^{N} {\rm Headway}(n) = hN + \sum_{n=1}^{N} \Delta_n. \tag{14}$$

Due to the start-up reaction and acceleration, the headways for the first few vehicles are greater than saturation headway h and are shown as $h + \Delta_n$ in Fig. 14 where Δ_n is the incremental headway for the nth vehicle [25]. In this paper, the incremental headways are assumed to decrease exponentially

[1] The $\Delta t_{\rm travel}$ is the expected travel time between $x_{\rm signal}$ and x_q:

$$\Delta t_{\rm travel} = \max\left\{\frac{d_q}{v_2} - \frac{v_2}{2a_{\rm acc}}, 0\right\} + \frac{v_s}{a_{\rm acc}} \tag{12}$$

where v_s is the velocity at stop-bar ($x_{\rm signal}$) and can be a value equal or lower than v_2 as follows:

$$v_s = \sqrt{2a_{\rm acc} \times \min\left\{d_q, \frac{v_2^2}{2a_{\rm acc}}\right\}} \tag{13}$$

with the position in queue. As a result, an empirical formulation for the queue clearance time is achieved by rephrasing (14) as:

$$\Delta t_{\text{clearance}} = hN + \Delta_1 \sum_{n=1}^{N} e^{-(n-1)}. \tag{15}$$

According to Highway Capacity Manual [25] and also [27], the effect of the start-up reaction and acceleration will be dissipated after the fourth vehicle (please see Fig. 14). This means that the steady headway h (saturation flow) will be reached after the fourth vehicle (this is true only if the headway compression, as studied in [28], is ignored). As a result, we should assume that incremental headways decrease in such a way that their value reaches ≈0 after the fourth vehicle. Although one may assume a linear decreasing function, an exponentially decreasing function is used as given in (15) because the incremental headway decreases more rapidly as the vehicle position in queue increases.

C. Parameter Estimation

The clearance time model provided in (15) is in fact a linear combination of saturation headway (h) and the first incremental headway (Δ_1). As a result, Multiple Linear Regression model (MLR) can be used to estimate these parameters. However, there are two challenges:

First, the regression variable is the vehicle position number in queue (N) which is not available. However, it can be estimated by (16), where L_v is the average distance that a vehicle occupies (20 ft), and $\lfloor . \rfloor$ is the flooring function, i.e.,

$$N = \left\lfloor \frac{d_q}{L_v} \right\rfloor + 1. \tag{16}$$

Second, gathering enough observational data on queue clearance time is time consuming. For this reason, an approach is proposed here which provides enough samples of queue clearance time without the need of gathering them locally at intersections. This is achieved using (17) where the green-initiation timestamp of a sample Intersection–Phase $(t_{\text{SoG,observed}})$ was locally collected from direct observation, and t_{start} is estimated based on the reconstructed trajectories. Multiple of C seconds is also included in (17) because $t_{\text{SoG,observed}}$ is a locally collected green-initiation and might be days before or after the estimated t_{start}, i.e.,

$$\begin{aligned} \Delta t_{\text{waiting}} &= t_{\text{start}} - t_{\text{SoG,observed}} \pm kC \quad k \in \mathbb{Z} \\ \Delta t_{\text{clearance}} &= \Delta t_{\text{waiting}} + \Delta t_{\text{travel}}. \end{aligned} \tag{17}$$

The clearance time model in (15) is then verified to fit the aforementioned data, as shown in Fig. 15. However, this data shown in Fig. 15, represented by the blue circles, do not seem to be symmetrically distributed. The queue clearance data is skewed most probably because there are many real world factors, such as lane blockage and downstream queue spillback, that would prolong the time needed for a queue to dissipate. On the other hand, usually there is no factor that could possibly make the queue clearance time shorter than its expected value. This explains why data is skewed to the right and not to the left. As a result, in order to reduce the influence of the unwanted events such as lane blockage, not all the data shown in Fig. 15 was used for fitting purposes. For each one meter increment in distance to stop-bar, we had labeled the data points that were more than 1.0 times the inter-quartile range above the 75th percentiles as outliers. These outlier points were removed from data and are cross-marked in Fig. 15. Please note that the clock drift could also be a reason that the queue clearance calculated by (17) is spread out.

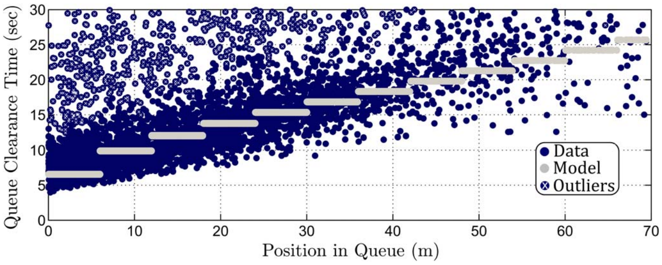


Fig. 15. The queue clearance time model fit to data.

The estimated regression coefficients of this curve fit are $h = 1.47$ s and $\Delta_1 = 5.08$ s for the through movement which are consistent with the measurements in literature [29], [30]. However, the first incremental headway Δ_1 looks slightly greater than expected because of the low acceleration of buses compared to conventional passenger vehicles, and also because of our slightly different definition of headway which considers the rear wheels crossing the stop-bar instead of the front wheels. As an empirical verification of the estimated parameters, assume there is no queue in front of the vehicle then the model estimates the clearance time to be equal to $h + \Delta_1 = 6.6$ s. This yields a waiting time (somewhat akin to first driver's reaction time) of about 1.6 s–2.6 s considering that it takes a vehicle in front of queue about $\Delta t_{\text{travel}} = 4\text{ s} - 5\text{ s}$ to completely pass the stop-bar. This is consistent with our observations in street and also with results in [30].

It must be emphasized that the aforementioned curve fitting was conducted only to get an idea of the queue clearance parameters values, and as it is verified in Section VI-B, it is not necessary to repeat the process for every intersection–phase. However, the results are only applicable to through movement, and similar parameter estimation should be repeated for left turn or shared left/through lanes.

VI. Ground Truth Verification

A. Verification of Green-Initiation

In order to verify the accuracy of green-initiation predictions, we collected the actual green-initiations locally at a sample intersection. These time samples were actually collected by a computer program that would log the time whenever the observer pressed a key at the change of red to green. The program was synchronized to the NIST time server [20] and was used to record the actual green-initiations between hours of 2 P.M. and 10 P.M.. This period of the day was selected so that the proposed green-initiation estimator could be evaluated during the evening rush hour traffic.

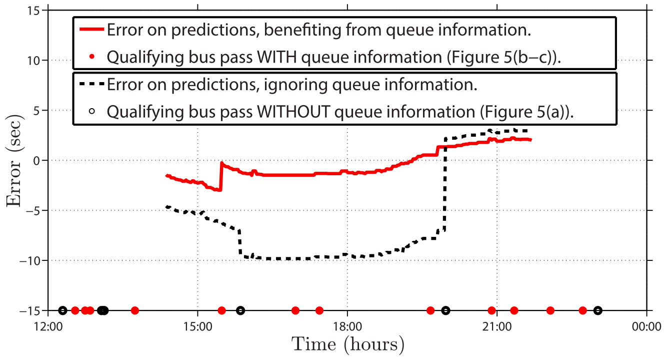


Fig. 16. The error between the predicted and actual green-initiations (southbound phase at Lombard intersection as recorded on April 25, 2013).

Concurrent with the aforementioned ground truth data collection, the green-initiations were also predicted by crowdsourcing the probe data sent from the public buses passing over the same intersection. The error between these predicted green-initiations and the collected actual green-initiations is shown in Fig. 16. The error shown in solid red line is the error of the estimation method when only using data from the probe vehicles that stop at red, send a report while waiting in queue, and leave the intersection at green. These vehicle passes should fit Fig. 5(b) or (c) though. The error shown in dashed black line is the error of the estimation approach of previous work [14] that only took into account those probe vehicles that stopped at red, and left the intersection at green without sending any report while in queue. Because the position in queue is not available in this case, these vehicle passes should not be influenced by queue delay and should fit Fig. 5(a) with high upstream velocity.

As it was expected in Section IV-D1, Fig. 16 demonstrates that during the persistent heavy traffic conditions, the probe reports that include the position in queue, result in more accurate phase-change predictions compared to the reports that do not reveal any queue information.

Please note that the jumps in error plots in Fig. 16 correspond to the times when new qualifying bus passes occurred in this particular scenario. These times are shown with filled or open circles depending on the trajectories that the corresponding passes are fitted to. Also the drift in plotted error in between the passes is due to the actual drift of the signal clock.

B. *Verification of Queue Formulations*

Three ground truth data collection sessions were arranged at 10 intersections along Van Ness street, San Francisco. One of the colleagues physically sat in buses and recorded the trajectory with a GPS tracking device at high frequency. In this way, GPS location and velocity data was collected at the frequency of 1 Hz while traveling on the transit buses. Fig. 12 shows a sample collected high-frequency GPS trace plotted over time–space diagram, wherein the timing of the lights are plotted using the baseline timings given in the city timing cards and the locally collected green-initiation timestamps.

We first searched the aforementioned plotted GPS traces for the stops at red at any of the 10 intersections along Van Ness street. Knowing the stop-bar positions of the intersections, the plots reveal the instant at which the buses pass the stop-bar at green phases. Furthermore, the corresponding time–velocity diagrams (not shown in Fig. 12) reveal the instant at which the waiting buses in queues start moving at green (t_{start}). Using the aforementioned extracted timestamps, the bus actual travel time, and also the queue waiting and clearance intervals are extracted as shown in Fig. 12. These values are tabulated in Table II as Ground Truth (G.T.).

The estimations of the queue waiting and clearance times and the travel time are also given in Table II denoted by Estimations (Est.) which are calculated by applying the proposed queue formulations in Section V to the collected ground truth data. The Root Mean Square Error (RMSE) between the estimations and the collected ground truth data, observed at 10 intersections, was 2.68, 1.37, and 1.98 seconds for $\Delta t_{\text{clearance}}$, Δt_{travel}, and $\Delta t_{\text{waiting}}$ respectively which are accurate enough for the application in this manuscript. The box plot of the errors are also given in Table II.

The correlation between the estimated values and the observed values of Table II is shown in Fig. 17 for queue waiting time. In the same figure, the proposed technique of waiting time estimation is compared and found to be consistent to two other formulations used in related works: First, the formulations proposed by Akçelik *et al.* [29] to estimate the queue departure response time for through closely-spaced intersection sites (used by Kerper *et al.* [15]). Second, the queue discharge shockwave speed formulations proposed by Lighthill [31] (according to the way it is used by Cheng *et al.* [16], Chuang *et al.* [17], and also [32]).

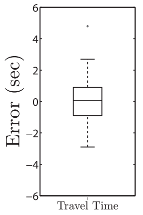
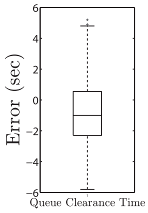
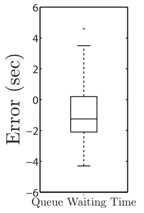
In addition to RMSE, Fig. 17 also provides the coefficient of determination R^2. The R^2 value of our proposed estimation technique is closer to 1.0 which indicates slightly better correlation with the observed values. This is achieved mainly due to taking account of incremental headways for the first few vehicles in queue as well as the downstream velocity into our proposed queue dissipation formulations. It should be emphasized that, although our obtained values for root mean square error (RMSE) and coefficient of determination (R^2) indicate more accurate estimations and better correlation with the observed values, the fact that our estimation method needs the downstream velocity (v_2) as extra information makes it difficult to conclusively claim that our queue dissipation formulation is better than the other two formulations mentioned above.

VII. Conclusion

A complementary approach to estimating traffic Signal Phase and Timing (SPaT) from probe data is proposed in this manuscript for pre-timed traffic signals. The input probe data stream is a low-frequency bus data feed; and the results are achieved under low penetration of probe data: the accuracy of the crowdsourcing algorithms are experimentally evaluated for a selection of intersections in San Francisco, CA where the bus passages are infrequent and happen every 5–10 min on average.

In case of a low-frequency data source, less challenge is expected in the estimation procedure if we only use the probe data that can be fitted into the predefined desired trajectories.

TABLE II
COMPARISON BETWEEN THE COLLECTED GROUND TRUTH (G.T.) DATA AND ESTIMATIONS (EST.) OF TRAVEL TIME, AND QUEUE CLEARANCE AND WAITING TIMES FOR ALL THE INTERSECTIONS COMBINED

Position in Queue (m)	Travel Time (sec) Δt_{travel}		Clearance Time (sec) $\Delta t_{clearance}$		Waiting Time (sec) $\Delta t_{waiting}$	
	G.T.	Est.	G.T.	Est.	G.T.	Est.
0.0	3	3.9	6.0	6.6	3.0	2.7
0.0	5	3.9	6.0	6.6	1.0	2.7
0.0	4	3.9	6.0	6.6	2.0	2.7
0.0	4	3.9	7.0	6.6	3.0	2.7
2.7	6	4.5	10.0	6.6	4.0	2.0
4.9	5	5.0	9.0	6.6	4.0	1.6
7.2	4	5.4	7.0	9.9	3.0	4.5
8.0	8	6.5	13.0	9.9	5.0	3.5
8.9	7	5.7	9.0	9.9	2.0	4.2
9.0	6	5.7	10.0	9.9	4.0	4.2
9.8	5	5.9	8.0	9.9	3.0	4.0
10.1	6	6.0	10.0	9.9	4.0	4.0
10.6	8	6.0	10.0	9.9	2.0	3.9
11.2	5	6.5	8.0	9.9	3.0	3.4
12.8	7	6.4	10.0	12.1	3.0	5.7
13.1	6	6.8	10.0	12.1	4.0	5.3
14.1	5	5.6	7.0	12.1	2.0	5.4
15.2	6	6.7	7.0	12.1	1.0	5.3
15.2	6	6.7	9.0	12.1	3.0	5.3
15.8	5	6.8	8.0	12.1	3.0	5.2
16.1	6	6.9	10.0	12.1	4.0	5.2
16.5	7	6.9	12.0	12.1	5.0	5.1
16.9	9	7.0	17.0	12.1	8.0	5.1
17.0	7	7.0	11.0	12.1	4.0	5.1
17.1	6	7.2	8.0	12.1	2.0	4.9
17.4	8	7.1	13.0	12.1	5.0	5.0
17.6	6	7.1	9.0	12.1	3.0	5.0
19.0	8	7.3	12.0	13.8	4.0	6.5
19.4	5	7.3	15.0	13.8	10.0	6.5
19.5	5	7.9	9.0	13.8	4.0	5.8
20.0	15	10.2	19.0	13.8	4.0	3.6
20.4	7	7.5	12.0	13.8	5.0	6.3
21.5	7	7.6	12.0	13.8	5.0	6.2
21.8	5	7.7	8.0	13.8	3.0	6.1
22.2	11	8.6	18.0	13.8	7.0	5.2
23.5	7	8.1	11.0	13.8	4.0	5.7
23.6	9	7.9	16.0	13.8	7.0	5.9
24.7	9	8.8	14.0	15.3	5.0	6.6
26.2	7	8.5	13.0	15.3	6.0	6.9
27.0	8	8.3	13.0	15.3	5.0	7.0
28.2	7	8.4	12.0	15.3	5.0	6.9
28.8	9	8.9	13.0	15.3	4.0	6.4
31.2	8	9.4	13.0	16.9	5.0	7.5
35.8	11	10.1	17.0	16.9	6.0	6.8
36.1	11	9.9	20.0	18.3	9.0	8.5
37.6	11	9.9	18.0	18.3	7.0	8.4
40.0	11	10.5	19.0	18.3	8.0	7.9
41.8	13	12.9	21.0	18.3	8.0	5.5
42.0	11	10.2	18.0	19.8	7.0	9.6
42.4	10	10.4	19.0	19.8	9.0	9.4
44.1	13	13.2	18.0	19.8	5.0	6.6
46.7	12	11.0	19.0	19.8	7.0	8.8
52.8	17	14.3	22.0	21.3	5.0	7.3
54.0	16	17.0	23.0	22.8	7.0	5.8
67.0	13	12.4	24.0	25.7	11.0	13.3
75.0	14	13.8	32.0	27.2	18.0	13.4
Box Plot						

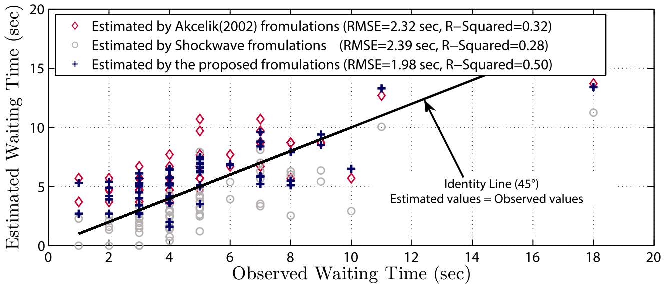


Fig. 17. Estimated versus observed queue waiting time.

Also if we identify and remove the probe data that appear to be influenced by heavy traffic then the more complex queue formulations are not needed in estimations. However, these measures eliminate a large portion of data; and any SPaT estimation method that filters out huge amount of data is subject to error. This is mainly due to the signal clock drift throughout a day that makes it crucial to have recent SPaT.

In this manuscript, it is shown that adding the trajectories that have been influenced by queue delay allows us to access a larger portion of data. This is the reason that the phase-change estimation results remain accurate even during heavy traffic. Also as it was expected, more accurate baseline timing estimations are achieved if we use the trajectories that include at least one report sent while waiting in queue. Obviously, this has been achieved at the price of more complex queue formulations in crowdsourcing algorithms.

In summary, the results presented in this paper can be categorized as: i) The improvements in SPaT estimation, compared to [14], by investigating the probe data influenced by the heavy traffic and the delay in queues. More specifically, the improvements are in *Signal Schedule Change Estimation* and *Green-Initiation Estimation*. ii) Extra SPaT information (*Red-Probability Estimation*) which is hidden in probe reports sent from stopped vehicles waiting in queues. iii) The verifications to evaluate the proposed queue dissipation formulations. In addition to these results, the back-end implementation of some algorithms which could not fit in [14] are explained in this paper that hopefully paves the way for future endeavors in this area.

ACKNOWLEDGMENT

The authors are thankful for the support provided by Mr. Andre Luckow from BMW ITRC. They also thank Mr. Nainfeng Wan from Clemson University, who traveled and collected the sample travel trajectories of public buses in the city of San Francisco. The authors would like to extend their special acknowledgments to Mr. Andreas Winckler from BMW Group Technology Office, USA in Mountain View, California, and Mr. Grant Mahler from Clemson University, who implemented the in-vehicle application.

REFERENCES

[1] B. Asadi and A. Vahidi, "Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 3, pp. 707–714, May 2011.

[2] M. Li, K. Boriboonsomsin, G. Wu, W. Zhang, and M. Barth, "Traffic energy and emission reductions at signalized intersections: A study of the benefits of advanced driver information," *Int. J. Intell. Transp. Syst. Res.*, vol. 7, no. 1, pp. 49–58, 2009.

[3] K. P. Sanketh, S. Subbarao, and K. A. Jolapara, "I2v and v2v communication based VANET to optimize fuel consumption at traffic signals," in *Proc. 13th Int. IEEE ITSC*, Madeira Island, Portugal, 2010, pp. 1251–1255.

[4] M. Maile and L. Delgrossi, "Cooperative intersection collision avoidance system for violations (CICAS-V) for avoidance of violation-based intersection crashes," in *Proc. 21st Int. Conf. ESV*, Stuttgart, Germany, 2009, pp. 1–95.

[5] A. Weber and A. Winckler, "Advanced traffic signal control algorithms, appendix A: Exploratory advanced research project: BMW final report," Caltrans Div. Res., Innov. Syst. Inf., San Diego, CA, USA, Tech. Rep. CA13-2157-B, Sep. 2013.

[6] Audi Travolution Project. [Online]. Available: https://www.audi-mediaservices.com

[7] S. A. Fayazi, S. Farhangi, and B. Asaei, "Fuel consumption and emission reduction of a mild hybrid vehicle," in *Proc. 34th IEEE IECON/IECON*, Orlando, FL, USA, 2008, pp. 216–221.

[8] A. Skabardonis and N. Geroliminis, "Real-time estimation of travel times on signalized arterials," in *Proc. 16th Int. Symp. Transp. Traffic Theory*, College Park, MD, USA, 2005, pp. 387–406.

[9] H. Liu, X. Wu, W. Ma, and H. Hu, "Real-time queue length estimation for congested signalized intersections," *Transp. Res. C, Emerging technol.*, vol. 17, no. 4, pp. 412–427, Aug. 2009.

[10] G. Mahler and A. Vahidi, "An optimal velocity-planning scheme for vehicle energy efficiency through probabilistic prediction of traffic-signal timing," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 6, pp. 2516–2523, Dec. 2014.

[11] E. Koukoumidis, L.-S. Peh, and M. Martonosi, "SignalGuru: Leveraging mobile phones for collaborative traffic signal schedule advisory," in *Proc. MobiSys*, 2011, pp. 127–140.

[12] Y. Zhu, X. Liu, M. Li, and Q. Zhang, "POVA: Traffic light sensing with probe vehicles," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 7, pp. 1390–1400, Jul. 2013.

[13] [Online]. Available: http://www.nextbus.com/

[14] S. A. Fayazi, A. Vahidi, G. Mahler, and A. Winckler, "Traffic signal phase and timing estimation from low-frequency transit bus data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 19–28, Feb. 2015.

[15] M. Kerper, C. Wewetzer, A. Sasse, and M. Mauve, "Learning traffic light phase schedules from velocity profiles in the cloud," in *Proc. 5th Int. Conf. NTMS*, Istanbul, Turkey, 2012, pp. 1–5.

[16] Y. Cheng, X. Qin, J. Jin, and B. Ran, "An exploratory shockwave approach for signalized intersection performance measurements using probe trajectories," in *Proc. Transp. Res. Board 89th Annu. Meet.*, Washington, DC, USA, 2010, pp. 1–23.

[17] Y. Chuang, C. Yi, Y. Tseng, C. Nian, and C. Ching, "Discovering phase timing information of traffic light systems by stop–go shockwaves," *IEEE Trans. Mobile Comp.*, vol. 14, no. 1, pp. 58–71, Jan. 2015.

[18] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau, "Extensible Markup Language (xml) 1.0," W3C Recommendation, Cambridge, MA, USA, 2008. [Online]. Available: http://www.w3.org/TR/REC-xml/

[19] K. K. Sanwal and J. Walrand, "Vehicles as probes," California Partners Adv. Transit Highways (PATH), Richmond, CA, USA, 1995.

[20] Official United States Time by National Institute of Standards and Technology. [Online]. Available: http://nist.time.gov/

[21] J. Krumm, "A survey of computational location privacy," *Pers. Ubiquitous Comput.*, vol. 13, no. 6, pp. 391–399, 2009.

[22] B. Hoh *et al.*, "Enhancing privacy and accuracy in probe vehicle-based traffic monitoring via virtual trip lines," *IEEE Trans. Mobile Comput.*, vol. 11, no. 5, pp. 849–864, May 2012.

[23] R. W. Sinnott, "Virtues of the haversine," *Sky Telesc.*, vol. 68, p. 158, 1984.

[24] N. Rouphail, A. Tarko, and J. Li, "Traffic flow at signalized intersections: Revised monograph on traffic flow theory, Federal Highway Admin., United States Dept. Transp. (U.S. DOT), Washington, DC, USA, Tech. Rep., 2005. [Online]. Available: http://www.fhwa.dot.gov/publications/research/operations/tft/chap9.pdf

[25] Highway Capacity Manual, Transp. Res. Board, Nat. Res. Council, Washington, DC, USA, 2000.

[26] R. P. Roess, E. S. Prassas, and W. R. McShane, *Traffic Engineering*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2011.

[27] J. Niittymäki and M. Pursula, "Saturation flows at signal-group-controlled traffic signals," *J. Transp. Res. Board, Transp. Res. Rec.*, vol. 1572, pp. 24–32, 1997.

[28] F. B. Lin and D. Thomas, "Headway compression during queue discharge at signalized intersections," *J. Transp. Res. Board Transp. Res. Rec.*, vol. 1920, pp. 81–85, 2005.

[29] R. Akçelik and M. Besley, "Queue discharge flow and speed models for signalised intersections," in *Proc. 15th Int. Symp. Transp. Traffic Theory Transp. Traffic Theory 21st Century*, Adelaide, Australia, 2002, pp. 1–20.

[30] M. S. Chaudhry and P. Ranjitkar, "Delay estimation at signalized intersections with variable queue discharge rate," *J. Eastern Asia Soc. Transp. Studies*, vol. 10, pp. 1764–1775, 2013.

[31] M. J. Lighthill and G. B. Whitham, "On kinematic waves. II. A theory of traffic flow on long crowded roads," *Proc. Roy. Soc. Lond. A, Math., Phys. Eng. Sci.*, vol. 229, no. 1178, pp. 317–345, May 1955.

[32] P. Hao, X. Ban, and J. W. Yu, "Kinematic equation-based vehicle queue location estimation method for signalized intersections using mobile sensor data," *J. Intell. Transp. Syst., Technol., Plan., Oper.*, vol. 19, no. 3, pp. 256–272, Jul. 2015.

Seyed Alireza Fayazi received the B.Sc. degree in electrical engineering from K. N. Toosi University of Technology, Tehran, Iran, and the M.Sc. degree in electrical engineering from University of Tehran, Tehran. He is currently working toward the Ph.D. degree in mechanical engineering with Clemson University, Clemson, SC, USA. He is part of a research team at the BMW Information Technology Research Center, Greenville, SC. In 2012–2013, he was a Visiting Researcher with University of California, Berkeley, CA, USA, and was also a Visiting Researcher at the BMW Group Technology Office, Mountain View, CA. Before joining Clemson University, he was a Research Engineer at Kerman Tablo Corporation for three years, where he worked on discrete control systems and digital control for embedded applications.

Ardalan Vahidi received the B.S. and M.Sc. degrees in civil engineering from Sharif University, Tehran, Iran, in 1996 and 1998, respectively; the M.Sc. degree in transportation safety from George Washington University, Washington, DC, USA, in 2002; and the Ph.D. degree in mechanical engineering from University of Michigan, Ann Arbor, MI, USA, in 2005. He is currently an Associate Professor with the Department of Mechanical Engineering, Clemson University, Clemson, SC, USA. He has been a Visiting Scholar with University of California, Berkeley, and a Visiting Researcher at the BMW Group Technology Office USA in 2012–2013. His research interests include control of vehicular and energy systems, and connected vehicle technologies.