Real Time Queue Length Estimation for Signalized Intersections
Using Travel Times from Mobile Sensors

Xuegang (Jeff) Ban*
Department of Civil and Environmental Engineering
Rensselaer Polytechnic Institute (RPI)
110 Eighth Street, Room JEC 4034
Troy, NY 12180-3590
Phone: (518) 276-8043 Fax: (518) 276-4833
Email: banx@rpi.edu
*Corresponding Author

Peng Hao
Department of Civil and Environmental Engineering
Rensselaer Polytechnic Institute (RPI)
110 Eighth Street, Room JEC 5107
Troy, NY 12180-3590
Email: haop@rpi.edu

Zhanbo Sun
Department of Civil and Environmental Engineering
Rensselaer Polytechnic Institute (RPI)
110 Eighth Street, Room JEC 5107
Troy, NY 12180-3590
Email: sunz2@rpi.edu

Resubmitted to Transportation Research Part C
January, 2011
ABSTRACT

We study how to estimate real time queue lengths at signalized intersections using intersection travel times collected from mobile traffic sensors. The estimation is based on the observation that critical pattern changes of intersection travel times or delays, such as the *discontinuities* (i.e., sudden and dramatic increases in travel times) and *non-smoothness* (i.e., changes of slopes of travel times), indicate signal timing or queue length changes. By detecting these critical points in intersection travel times or delays, the real time queue length can be re-constructed. We first introduce the concept of Queue Rear No-delay Arrival Time which is related to the non-smoothness of queuing delay patterns and queue length changes. We then show how measured intersection travel times from mobile sensors can be processed to generate sample vehicle queuing delays. Under the *uniform* arrival assumption, the queuing delays reduce linearly within a cycle. The delay pattern can be estimated by a linear fitting method using sample queuing delays. Queue Rear No-delay Arrival Time can then be obtained from the delay pattern, and be used to estimate the maximum and minimum queue lengths of a cycle, based on which the real-time queue length curve can also be constructed. The model and algorithm are tested in a field experiment and in simulation.

**Keywords:** Mobile Traffic Sensors; Global Positioning Systems (GPS); Cellular Phones; Traffic Signals; Traffic Shockwave Theory; Real Time Queue Length Estimation; Real Time Delay Pattern Estimation; Real Time Arterial Performance Measurement
1. Introduction and Motivation

Queue length is one of the most crucial performance measures for signalized intersections (Balke, 2005), which is also critical to signal optimization (Newell, 1965; Chang and Lin, 2000; Mirchandani and Zou, 2007). It has been a long-standing research topic to estimate the average queue length of traffic signals using loop detector data and signal timing information. Many early studies assumed discrete arrivals and integer cycle lengths, and Markov chain or similar statistical analysis techniques were applied to estimate the mean or distribution of queue lengths (Haight, 1959; Newell, 1960; Darroch, 1964; McNeil, 1968). Newell (1965) proposed a scheme to estimate the average queue length of a fixed-time signal by assuming traffic flow and signal timing parameters are continuous variables. Since then, queue length estimation methods can be generally grouped into two categories (Liu et al. 2009): input-output models (Webster, 1958; May, 1975; Akcelik, 1999; Sharma et al., 2007; Vigos et al., 2008) and shockwave models (Lighthill and Whitham, 1955; Richards, 1956; Stephanopoulos and Michalopoulos, 1979; Liu et al., 2009). The former derives queue lengths from cumulative arrivals and departures of an intersection, while the latter looks at how the queue forms and dissipates at the intersection.

Recently, estimating real time arterial performance measures using more fine-grained signal timing and detector data has gained much attention. Skabardonis and Geroliminis (2008) focused on the estimation of intersection queue length using 30-second loop detector data based on the shockwave theory. Their methods can consider both long queue (i.e., queue extends over the location of the upstream detector) and queue spillover (i.e., queue extends to the upstream intersection). This is done by using the flow and occupancy information measured at the detector to identify when long queue or spillover happens and then make appropriate adjustments to queue estimation. Using event-based signal and vehicle detection data (Balke et al. 2005; Smaglik et al. 2007; Liu and Ma, 2009), Liu et al. (2009) showed that “break points”, those that indicate the traffic state change (e.g., from queue discharge to queue clearance), can be identified or approximated by investigating the detailed signal-timing and vehicle actuation data. Using similar shockwave-theory-based methods as those in Skabardonis and Geroliminis (2008), the relationship between signal phase changes and traffic states during the queue forming and discharging processes was identified, from which real time queue length was estimated. The methods were also applied to estimate long queues and to detect queue spillover (Wu et al., 2010). Note that the methods in Skabardonis and Geroliminis (2008) for long queue and queue spillover identification implicitly defined the “break points” when queue extends over the detector or spillover conditions happen, but the concept of break points were not introduced in Skabardonis and Geroliminis (2008) until later in Liu et al. (2009).

Most existing queue estimation methods are based on data from fixed-location sensors (e.g., loop detectors) such as flow, occupancy, gap, and speed. The limited coverage of current arterial detection systems and the challenge to further expand them due to resource limitations have
restrained arterial performance measurement for wide areas. Mobile traffic sensors, including GPS and other types of tracking devices that can trace the movement of individual vehicles, have great potential to fill current arterial performance measurement gaps or provide an alternative way for arterial performance measurement. According to Harris Interactive (2007), “one in six (17%) U.S. adults currently own or use a GPS location device or service” and this trend is growing rapidly. A high penetration of mobile sensors is thus expected in the near future given the fact that most future cellular phones may be equipped with GPS and the penetration of cell phones is nearly 100% in developed countries and nearly 50% in developing countries in 2007 (ITU, 2009). Recently, investigating the feasibility of using mobile sensors for traffic monitoring and state estimation has received attention, with the main focus on freeway traffic (Lu and Skabardonis, 2007; Herrera et al. 2009; Herrera and Bayen, 2009). These methods are based on shockwave theory or Kalman filtering for continuous flow and may not be directly applied to signalized intersections due to the discontinuities of arterial traffic introduced by signal timing.

Another challenge of using mobile sensor data for queue estimation is due to the fundamental difference between mobile data and fixed-location sensor data – mobile sensor data are samples of real traffic flow (unless the penetration is exactly 100% which is highly unlikely in the foreseeable future). They cannot provide “true” occupancy or flow information which is the key input to existing queue estimation methods. Therefore, existing methods cannot be directly applied to mobile data. As a result, previous studies mainly focused on distributions of average queue length using probe data (Comert and Cetin, 2008) or queue estimation from probe trajectories directly (Izadpanah et al., 2009; Cheng et al. 2010). To develop real time queue estimation methods, one needs to resolve two issues: (1) what types of mobile data to use, and (2) what modeling techniques to use based on the (new) mobile data elements.

Ban et al. (2009) showed that sample intersection travel times (i.e., those between an upstream and a downstream location of an intersection) can be used to estimate real time intersection delay patterns. The delay pattern describes the delay an arbitrary vehicle will experience when arriving at the intersection at a given time, which can be further used to estimate arrival volumes or even signal timing information under relatively high penetration. This result lead to an important observation: under certain assumptions (such as uniform arrival), the real time intersection delay pattern is piece-wise linear, and contains discontinuities (i.e., a dramatic increase of delays) and non-smoothness (i.e., significant changes of travel time slopes). The discontinuities correspond to signal timing changes (such as the start of red time), while the non-smoothness is due to traffic state changes (such as a queue is cleared). Similar observation was also reported in Liu et al. (2009) in terms of how queue changes within a cycle.

Motivate by this observation, we develop a reverse modeling process in this paper to estimate real time intersection queue length using travel times collected from mobile sensors. The method

---

1 For example, the New York City has over 95% of its total 12,225 traffic signals as pre-timed and non-detector is deployed at these intersections (according to a private communication with traffic engineers of New York State Department of Transportation (NYCDOT)).
contains two major steps. The first step is to estimate real time intersection delay pattern based on sample intersection travel times. The second step is to use the estimated delay pattern to identify critical points of when queue is maximized, minimized, or cleared within a cycle, which enables the construction of the real time queue length. Such an approach represents a quite different thinking process of existing intersection modeling methods that use volume or occupancy to estimate queue length and travel times.

We first introduce the concepts of Queue Rear No-delay Arrival Time (QRNAT) and Queue Front No-delay Arrival Time (QFNAT), and show how they relate to queue length changes based on shockwave theory. This helps us develop models to calculate the maximum and minimum queue lengths of a cycle using QRNAT. To estimate QRNAT, we first propose models to calculate the queuing delay (defined in this paper as the total delay with oversaturation delay (if any) excluded) if QRNAT and signal timing parameters are known. The models show that the queuing delay linearly decreases within a cycle and the critical point of the delay pattern corresponds to QRNAT. This motivates us to (1) process the total measured vehicle delays available directly from mobile sensors to calculate sample queuing delays; and (2) estimate the queuing delay patterns from sample queuing delays by linear fitting. Knowing the queuing delay pattern enables us to obtain QRNATs, which can then be used to estimate the minimum and maximum queue lengths, and further to construct the real time queue length curve. The models and algorithms are tested in field experiment and simulation.

It turns out that collecting intersection travel time or delay samples instead of detailed vehicle trajectories can also lead to improved privacy protection, if the data collection and sampling schemes are properly designed (Hoh et al., 2008). This helps address the privacy issue when mobile sensors are widely available and the public is more concerned with privacy (Krumm, 2008; Hoh et al. 2007). In particular, we refer to the upstream and downstream locations of an intersection for travel time collection as Virtual Trip lines (VTL) due to the fact that they are virtual and do not require deployment of any physical infrastructure. For detailed definitions of VTL and the data that can be collected, one can refer to Hoh et al. (2008), Herrera et al. (2009), and Ban et al. (2009). We note here that previously Demers et al. (2006) applied a similar concept, called monument (He et al. 2002), to collect travel times from mobile sensors to study the path choice behavior of drivers. Using only VTL travel times will also make it possible to apply the proposed scheme in this paper to arterial travel time collection system based on wireless sensors (Kwong et al., 2009) and related vehicle re-identification techniques such as Bluetooth MAC address matching (Wasson et al., 2008).

The rest of this paper is organized as follows. We introduce in Section 2 the assumptions made and terminologies used in this paper. Section 3 presents the concept of QRNAT and QFNAT, and how they relate to the maximum and minimum queue lengths in a cycle. We discuss in Section 4 how QRNAT can be used to compute queuing delay. Section 5 presents an algorithm to estimate QRNAT and then the real time queue length curve. The algorithms are then tested using field and simulation data in Section 6, followed by concluding remarks in Section 7.
2. Preliminaries

We present in this section the assumptions to derive the queue estimation model. The definitions of some terminologies and a simplified queue forming/discharging process then follow.

2.1 Assumptions

To simplify our discussion, we impose the following assumptions to the queue estimation problem. Some of these assumptions can be relaxed as we show later in the Conclusion section.

(1) Uniform arrivals within one cycle which can be considered as the average arrival rate of the cycle. Note that the arrival rates may vary from cycle to cycle under this assumption.
(2) We focus on an approach of a given intersection and signal timing information is known.
(3) Vehicle acceleration and deceleration are ignored.
(4) No queue spills back to the upstream VTL. This may be considered as the “short queue” estimation problem (Liu et al. 2009). However since the VTL here is virtual, one can deploy the upstream VTL relatively far from the stop line for long queue cases. The VTL locations may also be changed dynamically to respond to actual traffic conditions. Thus this assumption is less stringent compared with methods using loop detector data. Since VTL cannot be deployed beyond the entrance of the link, this assumption also implies that queue does not spillover to the upstream intersection.
(5) A vehicle may not be cleared within one cycle (in this case oversaturation happens), but it must be cleared in the next cycle. This assumption does not impose limitations on the number of consecutive oversaturated cycles; it does impose limitations on the level of congestion, i.e., no vehicle will wait for more than 2 cycles to get through an intersection.
(6) Relatively high penetration so that the sample intersection travel times for at least two queued vehicles can be obtained within each cycle. Such scenario is valid when mobile sensor data are widely available in the future.

2.2 Terminology definitions

- Normal condition and over-saturation condition
  If the queue can be cleared completely during the green phase of a cycle, the cycle is defined as normal. If the queue cannot be fully discharged within a cycle and a residual queue is formed, the cycle is in an over-saturation condition.

- Queuing delay and over-saturation delay
  Queuing delay is defined as the summation of signal waiting time and delay due to the presence of a queue. It is thus the summation of single vehicle delay (including the signal waiting time) and the delay due to queues as defined in Skabardonis and Geroliminis (2005). Over-saturation delay is defined as the additional delay caused when the arrival rate exceeds the service rate at the traffic signal. As we show later, oversaturation delay is always equal to the duration of an appropriate red time. Vehicle total delay is the sum of queuing delay and over-saturation delay.
- **Real time queue length**
  
  Real time queue length is defined as the *number of vehicles* in the queue of an approach of an intersection at any given time. Note that this is called queue size in Liu et al. (2009). Here we define a queued vehicle based on the delay the vehicle experienced when passing the intersection, as discussed in Section 6.1.

### 2.3 Simplified queue forming and discharging process

Based on the above assumptions, how queues form and discharge at a signalized intersection can be approximated as shown in Fig. 1(b) for normal conditions and 1(c) for over-saturation conditions. The construction of these two cases assumes a triangular fundamental diagram for an approach of the intersection as shown in Fig. 1(a). The fundamental diagram can be determined by three parameters: free flow speed $v_f$, jam density $k_j$, and capacity $q_m$. In addition, $k_m$ is the critical density, $w$ is the shockwave speed, and $(q_a, k_a)$ is a given traffic state.

[Place Fig. 1 about here]

In Fig. 1(b) and 1(c), $T_r^n$ and $T_g^n$ indicate the start time of the effective red and effective green of the $n^{th}$ cycle, respectively. $T_b^n$ and $T_f^n$ denote the Queue Rear No-delay Arrival Time (QRNAT) and the Queue Front No-delay Arrival Time (QFNAT) of the $n^{th}$ cycle, which are defined later in Section 3.1. $L_{max}^n$ and $L_{min}^n$ are the maximum and minimum queue length in terms of distance of the $n^{th}$ cycle as defined in Section 3.2. $T_{max}^n$ and $T_{min}^n$ are times when the maximum and minimum queue lengths happen, which are also defined in Section 3.2. VTL1 indicates the upstream location for collecting travel times with the understanding that the downstream location (VTL2) is deployed appropriately (not shown in the figure). In addition, define $v_1^n$ the queuing forming wave, $v_2^n$ the queue discharging wave, $v_3^n$ the departure wave, and $v_4^n$ the residual queue forming wave. As shown in Skabardonis and Geroliminis (2005), and Liu et al. (2009), the speeds of the four waves can be calculated as:

$$v_1^n = \frac{0-q_a^n}{k_j-k_a^n} = \frac{q_a^n}{k_j-k_a^n}.$$  \hspace{1cm} (1)

$$v_2^n = \frac{q_m-0}{k_m-k_j} = \frac{q_m}{k_j-k_m} = w.$$ \hspace{1cm} (2)

$$v_3^n = \frac{q_m-q_a^n}{k_m-k_a^n} = v_f.$$ \hspace{1cm} (3)

$$v_4^n = \frac{0-q_m}{k_j-k_m} = v_2^n = w.$$ \hspace{1cm} (4)

Here $q_a^n$ and $k_a^n$ are the average volume and density of the arrival flow during the $n$-th cycle. Hereafter in this paper, $v_2^n$, $v_3^n$, $v_4^n$ will be replaced by $w$, $v_f$, $w$, respectively.

### 3. Queue Rear and Queue Front No-Delay Arrival Times

#### 3.1 Queue Rear and Queue Front No-Delay Arrival Times
Instead of using the actual time when a vehicle arrives at the intersection (defined as when the vehicle arrives at the stop line), we define the No-delay Arrival Time (NAT) of a vehicle, which can be calculated by summing the free flow travel time from VTL1 to the stop line and the actual time when the vehicle passes VTL1. The NAT indicates the estimated time that a vehicle would have arrived at the intersection if no delay had been experienced. In this paper, the “arrival time” of a vehicle to the intersection refers to the NAT of the vehicle unless specified otherwise.

The Queue Rear No-delay Arrival Time (QRNAT) of a cycle is defined as the NAT of the queue rear of the cycle, i.e., the time when the queue would have been fully discharged if there had been enough green time in the cycle. Similarly, the Queue Front No-delay Arrival Time (QFNAT) of a cycle is defined as the NAT of the queue front of the cycle. As shown in Fig. 1(b) and 1(c), the actual values of the QRNAT (denoted as $T_b^n$ for the n-th cycle) and QFNAT (denoted as $T_f^n$ for the n-th cycle) of a cycle depend on the traffic condition of the cycle. For a normal cycle, since queue can be fully discharged, QRNAT is within the green time of the cycle. In Fig. 1(b), the QRNAT is shown by the intersection of the wave $v_3$ and the stop line. For an oversaturated cycle, since the queue cannot be fully discharged, QRNAT is after the next cycle starts. In Fig. 1(c), QRNAT is represented by extending wave $v_3$ as a (dashed) straight line to intersect with the stop line. On the other hand, the QFNAT is more closely related to whether a beginning residual queue exists at the start of a cycle. If no beginning residual queue exists, QFNAT in general depends on when the first vehicle arrives in red time. In this paper, as we assume uniform arrival, QFNAT is assumed to be the start of the cycle (i.e., the red time) when there is no beginning residual queue. If a beginning residual queue does exist however, QFNAT is before the start of the current cycle. In Fig. 1(c), for example, the QFNAT of cycle n+1 in Fig. 1(c), denoted as $T_f^{n+1}$, is indicated as the intersection of the extension of trajectory $c$ (the vehicle that is the head of the beginning residual queue of cycle n+1) and the stop line. QRNAT and QFNAT play an important role in characterizing the real time delay pattern and the queue length of a signalized intersection which will be discussed in more detail in Section 4.

In this paper, vehicles “arriving within a cycle” are defined based on their NATs. That is, a vehicle whose NAT is between the start of the red time and the end of the green time (or the start of the next red time) of a cycle is considered as a vehicle arriving at the intersection within the cycle. Under this definition, oversaturation delay can occur in two ways. If a cycle is oversaturated and a vehicle is in the beginning residual queue of the next cycle, the vehicle will experience an oversaturation delay which equals to the duration of the red time of the next cycle. In Fig. 1(c), vehicles between trajectory $c$ and $d$ represent those vehicles ($d$ is the vehicle whose NAT is $T_f^{n+1}$, i.e., the last vehicle arriving at the nth cycle). If a vehicle arrives at the very beginning of a cycle with a beginning residual queue, the vehicle will experience an oversaturation delay which equals to the duration of the red time of the current cycle. In Fig. 1(c), vehicles between trajectory $d$ and $e$ represent those vehicles. As a result, vehicles between $c$ and $e$ are over-saturated vehicles which will all experience an oversaturation delay equal to the duration of the red time of cycle n+1.
3.2 Minimum and Maximum Queue Lengths of A Cycle

QRNAT and QFNAT are closely related to how queue forms and discharges at a signalized intersection. As shown in Fig. 1(b) and 1(c), based on the triangle formed by shockwave \(v_2\), \(v_3\), and the stop line (we need to extend \(v_3\) to reach the stop line at \(T_{b}^{n}\) for the oversaturation case), we can derive the equation

\[
\frac{L_{\max}^{n}}{v_2^n} + \frac{L_{\max}^{n}}{v_3^n} = T_{b}^{n} - T_{g}^{n},
\]

(5)

where \(L_{\max}^{n}\) is the maximum queue length in terms of distance of the \(n^{th}\) cycle, which can be expressed as a function of \(T_{b}^{n}\) from (5)

\[
L_{\max}^{n} = (T_{b}^{n} - T_{g}^{n}) \frac{v_2^n v_3^n}{v_2^n + v_3^n} = (T_{b}^{n} - T_{g}^{n}) \frac{w v_f}{w + v_f}.
\]

(6)

For the oversaturated case, the minimum queue length (in distance) of the \(n^{th}\) cycle (also the beginning residual queue for the next cycle) \(L_{\min}^{n}\) can be derived in the same way

\[
L_{\min}^{n} = (T_{b}^{n} - T_{r}^{n+1}) \frac{w v_f}{w + v_f}.
\]

(7)

We can also get the time when queue is maximized:

\[
T_{\max}^{n} = \frac{L_{\max}^{n}}{v_4^n} + T_{g}^{n} = (T_{b}^{n} - T_{g}^{n}) \frac{v_f}{w + v_f} + T_{g}^{n}.
\]

(8)

And for an oversaturated case, the time that a minimum queue length happens is:

\[
T_{\min}^{n} = \frac{L_{\min}^{n}}{v_2^n} + T_{r}^{n+1} = (T_{b}^{n} - T_{r}^{n+1}) \frac{v_f}{w + v_f} + T_{r}^{n+1}.
\]

(9)

Note that equations (5) and (6) are derived based on the discharge triangle in Fig. 1(b) and 1(c); equation (7) is derived based on the residual queue forming triangle. Therefore, they should hold even when the arrival flow pattern is not uniform.

Based on the definition of QRNAT, all of the queued vehicles could be fully discharged from \(T_{g}^{n}\) to \(T_{b}^{n}\) at a saturation flow rate. Therefore the maximum queue length (in terms of number of vehicles) of the \(n^{th}\) cycle is

\[
Q_{\max}^{n} = (T_{b}^{n} - T_{g}^{n}) q_m
\]

(10)

Similarly, the minimum queue length is

\[
Q_{\min}^{n} = (T_{b}^{n} - T_{r}^{n+1}) q_m
\]

(11)

Thus, if QRNAT can be estimated accurately, equations (5) – (11) show how the maximum queue length (and minimum queue length in oversaturated cases) of a cycle can be estimated,
together with when the queue is maximized (or minimized). In addition, queue is cleared at QRNAT within a normal cycle. The identification of these critical points will enable us to construct the real time queue length pattern since the change of queue length between any two critical points is assumed as linear due to the uniform arrival assumption (refer to Fig. 1(b) and 1(c)). The process to estimate QRNAT and QFNAT will be discussed in the next section.

4. Real Time Intersection Queuing Delay Pattern

The real-time intersection delay pattern depicts the real time delay an arbitrary vehicle will experience when arriving at the intersection at a given time. It is a continuous approximation of intersection delays which in practice are just discrete samples of individual vehicles. Ban et al. (2009) showed that the real time intersection delay pattern can be estimated using sample intersection delays by detecting the discontinuities of the pattern followed by line fitting. We focus on the real-time intersection queuing delay pattern here which is the intersection delay pattern by excluding the oversaturation delay.

In Fig. 2(a), a signalized intersection is depicted to the left, with VTL1 and VTL2 denoting the upstream and downstream VTLs. The travel time or delay is measured between VTL1 and VTL2. The bold triangles and trapezoids illustrate how queues form and discharge, similar to those in Fig. 1. The dashed lines are vehicle trajectories. Based on the assumptions in Section 2, the acceleration and deceleration of vehicles are ignored and vehicles cannot wait for more than two cycles at an intersection. Then given the possible over-saturation condition for either the previous cycle or the current cycle or both, there are four possible cases of a cycle as illustrated as Cycle A – D in Fig. 2. The four cycles have the following characteristics:

- **Cycle A**: Queue is fully discharged in both the previous and current cycles. Vehicles in this case do not have any over-saturation delay.
- **Cycle B**: No beginning residual queue exists, but queue cannot be fully discharged in the current cycle. The rear part of the queue (e.g., vehicles after trajectory d) has an over-saturation delay. The over-saturation delay equals to the red time of the next cycle.
- **Cycle C**: Queue cannot be fully discharged in either the previous cycle or the current cycle. Only vehicles arriving during the middle of the cycle may avoid over-saturation delay (e.g., vehicles between e and g). The over-saturation delay for the front part of the queue is the red time of the current cycle; for the rear part, it is the red time of the next cycle.
- **Cycle D**: Queue in the current cycle consists of a residual queue from last cycle and vehicles arriving at the beginning of the current cycle. All the queued vehicles are discharged within the current cycle. The front part of the queue (e.g., vehicles between g and h) experiences an over-saturation delay, which equals to the red time of the current cycle.

Cycle A and D are under normal conditions as we defined earlier; Cycle B and C are under oversaturated conditions. Clearly, this is mainly based on whether the queue of a cycle can be fully
discharged or not. On the other hand, the beginning residual queue refers to whether a queue exists at the \textit{beginning} of a cycle.

**[Place Fig. 2 about here]**

Vehicle trajectories in Fig. 2(a) indicate when travel time (or delay) \textit{discontinuities} or \textit{non-smoothness} occur. The piecewise linear curve at the bottom of Fig. 2(a) represents the delay pattern vs. the time when a vehicle passes VTL1 (i.e., the actual time instead of in NAT). \textit{Discontinuities} occur when signal timing changes. For example, trajectory \(c\) indicates that when the red time for Cycle B starts, the delay will experience a significant increase (i.e., the duration of the red time of Cycle B). For a cycle with beginning residual queue such as Cycle C, the start of the red time is “felt” earlier as shown by trajectory \(d\). \textit{Non-smoothness}, on the other hand, happens when traffic state changes. For example, trajectory \(b\) indicates that the queue is cleared and the delay will level off until the end of the cycle, resulting in a “kink” in the delay pattern. Trajectory \(e\) indicates that after the vehicles that experienced over-saturation delays are cleared, a “kink” in delay will also exist.

Fig. 2(b) further shows the delay pattern in NAT which is a translate of the pattern in Fig. 2(a) to the right by the free flow travel time from VTL1 to the stop line. Fig. 2(c) depicts the queuing delay pattern in NAT. We can observe that:

- In Fig. 2(b), the \textit{discontinuities} in delay pattern happen at QFNAT. It is the same as the start of the red for cycles with no beginning residual queue. For cycles with beginning residual queue, QFNAT is earlier than the start of the red time, indicating the red is felt earlier by vehicles. The non-smoothness happens at QRNAT when either the over-saturated vehicles or the entire queue is cleared. Also, the oversaturation vehicles are those whose NATs are between QFNAT of the next cycle \(T_{f}^{n+1}\) and QRNAT of the current cycle \(T_{b}^{n}\).

- As shown in Fig. 2(c), after the oversaturation delay is excluded, discontinuities at those QFNATs that are not the same as the start of red times will disappear. Here we define the queuing delay pattern of the \(n\)th cycle starts at \(T_{b}^{n-1}\) (or \(T_{f}^{n}\)) if the \((n-1)\)th cycle is oversaturated (or if in normal condition) and ends at \(T_{b}^{n}\) (or \(T_{f}^{n+1}\)) if the current cycle is oversaturated (or if in normal condition). Then the queuing delay pattern consists of a straight line with a constant reduction rate. This line represents the delays of \textit{queued} vehicles. The pattern may also contain a horizontal line representing zero delay of \textit{free-flowing} vehicles in normal cycles. The constant reduction rate is due to the fact that the delay reduction rates for vehicles, e.g., those between \(c\) and \(d\) and \(d\) and \(e\), are the same.

The reason we are focusing on the queuing delay pattern is that it only relates to QRNAT and the start of red time (assumed to be known). The sample total delay can be measured directly from the VTLs; the queuing delay can also be derived from the total delay by a simple transformation (see Section 5 below). This can help estimate the queuing delay pattern of a cycle using mobile sensor data which can be used to calculate QRNAT. In the following, we show in Equations (12) – (15) for the four cycles in Fig. 2 respectively, how the queuing delay pattern can be expressed in terms of QRNAT and the start of the red time.
Here $q(t)$ is the queuing delay for a vehicle arriving at the intersection at time $t$ and $r_n$ is the length of the red time of the $n^{th}$ cycle. Summarizing all the above equations, the queuing delay pattern can be expressed as follows:

$$q(t) = \begin{cases} 
\frac{r_n(t^n_b - t)}{T^n_b - T^n_r}, & \text{if } T^n_r \leq t \leq T^n_b \\
0, & \text{if } T^n_b \leq t \leq T^n_{b+1} 
\end{cases}$$  \hspace{1cm} (12.1)$$

$$q(t) = \frac{r_n(T^n_b - t)}{T^n_b - T^n_r}, \quad \text{for } T^n_r \leq t \leq T^n_b$$  \hspace{1cm} (13)$$

$$q(t) = \frac{r_n(T^n_b - t)}{T^n_b - T^{n-1}_b}, \quad \text{for } T^{n-1}_b \leq t \leq T^n_b$$  \hspace{1cm} (14)$$

$$q(t) = \begin{cases} 
\frac{r_n(t^n_b - t)}{T^n_b - T^n_r}, & \text{if } T^{n-1}_b \leq t \leq T^n_b \\
0, & \text{if } T^n_b \leq t \leq T^n_{b+1} 
\end{cases}$$  \hspace{1cm} (15.1)$$

$$q(t) = \begin{cases} 
\frac{r_n(T^n_b - t)}{T^n_b - T^{n-1}_b}, & \text{if } T^n_r \leq T^{n-1}_b \text{ and } T^{n-1}_b \leq t \leq T^n_b \\
0, & \text{otherwise} 
\end{cases}$$  \hspace{1cm} (15.2)$$

Here (16.1) and (16.2) are for a cycle without and with a beginning residual queue respectively. Equation (16.3) is for the case with zero queuing delay. Equation (16) indicates that if the QRNAT and signal red times are known, one can use (16) to calculate the queuing delay pattern. One important feature of (16) is that the queuing delay varies linearly with time $t$ once signal timing and QRNATs are known. This enables us to use linear fitting methods to estimate queuing delay patterns via delay samples. Fig. 2(c) and equation (16) also indicate that in order to properly estimate the linear queuing delay pattern within a cycle, we need travel time samples from at least two queued vehicles. Free-flowing vehicles, whose delays are on the horizontal part of the pattern, generally do not help the estimation of the queuing delay pattern curve.

### 5. Real Time Queue Length Estimation

In the previous sections we show how the queuing delay pattern can be calculated given QRNAT and the timing of signal red times. What we can obtain directly from mobile sensors however are the measured travel times or total delays of an intersection. In this section, we show that the measured total delays can be processed (see Section 5.1) to obtain the samples of queuing delays. The resulting queuing delay samples can be further used to estimate the queuing delay pattern curve via a cycle breaking plus line fitting process (Section 5.2) which was discussed in detail in Ban et al. (2009). As shown in Fig. 2(c) and equation (16), QRNAT is when the zero queuing
delay first happens within a cycle. This enables us to obtain QRNAT using the estimated queuing delay pattern curve. The QRNAT can then be used to estimate the maximum and minimum queue lengths during each cycle based on equations (5) – (11). This constitutes the three major steps for estimating real time queue length: calculating queuing delay from measured total delay, estimating queuing delay pattern by linear fitting and obtaining QRNAT from the estimated queuing delay pattern, and calculating the minimum (or zero) and maximum queue lengths using QRNAT. These three steps represent a reverse process of the standard intersection performance measurement approaches based on fixed-location sensors. That is, in standard methods, the input is volume or occupancy and the output is travel time (or delay) or queue lengths; in the method using mobile sensors, travel time or delay is the input, while queue length (or even traffic volume as shown in Ban et al. (2009)) is the output.

5.1 Calculating Queuing Delay from Measured Total Delay

As shown in Section 4, the total delay of an oversaturated vehicle can be larger than the red time of the next cycle (e.g., I in Fig. 2(b)) or the red time of the current cycle (e.g., II in Fig. 2(b)). However the reverse may not be true: while a vehicle whose delay is larger than the red time of its current cycle always indicates that the current cycle is oversaturated, a vehicle whose delay is larger than the next red time does not always indicate that the current cycle is oversaturated. This is due to the variation of red times across different cycles. Based on this observation, we propose a two-step approach to detect whether over-saturation happens in this paper. First, when a vehicle arrives (in NAT) in one cycle, but departs from the stop line during the green time of the next cycle, it is considered as an over-saturated vehicle. Fig. 2(b) shows a sample I when this happens. Vehicle I should have passed the stop line in Cycle B if the cycle is normal, but had to wait for the next green time because of over-saturation. The condition can be expressed as:

\[ d(t_I) + t_I \geq T_r^{n+1} + r_{n+1}, \text{ for } T_r^n \leq t_I \leq T_r^{n+1}. \]  

(17)

Here \( t_I \) is the arrival time (in NAT) of vehicle I, \( d(t_I) \) is its total delay, and \( r_{n+1} \) is the red time of the next cycle. The left hand side of (17) represents the departure time of vehicle I which is within the green time of the next cycle (notice \( T_g^{n+1} = T_r^{n+1} + r_{n+1} \)). Also \( T_r^n \leq t_I \leq T_r^{n+1} \) indicates that vehicle I arrives during the n-th cycle.

For vehicles that arrive at the very beginning of a cycle that starts with a beginning residual queue, e.g., vehicle II in Fig. 2(b), their over-saturation delay is the duration of the red time of the current cycle (i.e., Cycle C in Fig. 2(b). This condition can be expressed as:

\[ d(t_{II}) > r_n, \text{ for } T_r^n \leq t_{II} \leq T_r^{n+1} \text{ and } d(t_{II}) + t_{II} \leq T_r^{n+1}. \]  

(18)

Here \( t_{II} \) is the arrival time (in NAT) of vehicle II, \( d(t_{II}) \) is its total delay, and \( d(t_{II}) + t_{II} \) represents the departure time of vehicle II which is within the green time of the current cycle. To obtain queuing delay samples, all measured delays will be checked against (17) and (18). For those that satisfy either of them, their measured delays will be subtracted by a proper red time.
5.2 Estimation of the QRNAT

After we obtain the queuing delay samples, the next step is to estimate the queuing delay pattern using those samples. According to equation (16), the queuing delay pattern linearly reduces with time \( t \) during each cycle and may or may not level off depending on the cycle is normal or oversaturated. The estimation can thus be done by a linear fitting approach, similar to that in Ban et al. (2009). Notice that in Ban et al. (2009), an algorithm is developed to estimate the total delay pattern from sample total delays which is more complicated than the curve of the queuing delay pattern. In this sense, the line fitting in this paper is easier to conduct compared with that in Ban et al. (2009). Details of the linear fitting are thus omitted here. The pattern is illustrated using piece wise linear curves in Fig. 2(c). The figure also shows that when \( q(t) \) becomes zero the first time, the arrival time will be the QRNAT. In other words, QRNATs can be obtained at the points where the queuing delay pattern curve intersects with the horizontal line for zero queuing delay. Knowing the QRNAT, one can determine whether a cycle is oversaturated or not: \( T_b^n \leq T_r^{n+1} \) indicates normal condition and \( T_b^n > T_r^{n+1} \) indicates over-saturation. In other words, QRNAT can help detect the traffic condition of each cycle (see examples in Section 6).

5.3 Estimation of Minimum and Maximum Queue Lengths

Substitute the estimated QRNAT, denoted as \( \tilde{T}_b^n \), into equation (10), the estimated maximum queue length can be calculated. For oversaturated cycles, the minimum queue length can also be calculated by substituting \( \tilde{T}_b^n \) into equation (11). For normal cycles, the zero queue length happens first at the estimated QRNAT(\( \tilde{T}_b^n \)). In summary, the real time queue length estimation algorithm can be listed as follows:

Real Time Queue Length Estimation Algorithm

Step 1. Initialization. Collect VTL travel times and process them into NATs and total intersection delays.

Step 2. Queuing delay calculation. Check measured total delays against equations (17) and (18); Subtract a proper red time when any of the conditions is satisfied. The resulting delays are queuing delays.

Step 3. Queuing delay pattern curve estimation. Perform the cycle breaking and line fitting algorithm using the queuing delays obtained in Step 2. This will produce all QRNATs by finding the intersections of the queuing delay pattern with the horizontal line representing zero queuing delay.

Step 4. Maximum and minimum queue lengths estimation. Substitute the estimated QRNAT into equations (11) to calculate maximum queue length for each cycle. The minimum queue length can also be calculated using (12) for oversaturated cycles or zero queue length happens at the estimated QRNATs for normal cycles.

Step 5. Construction of real time queue length. Since the change of queue length between any two critical points (i.e., maximum or minimum or zero queue lengths) is assumed to be linear, the real time queue length pattern can be constructed by connecting any two consecutive critical points using straight lines.
6. Numerical Experiments

6.1 Test Sites

The models and algorithms proposed in this paper were tested in field experiments and microscopic traffic simulation. The field experiment was conducted in Albany, NY area during a PM rush hour in November 2009. A dedicated group of 9 drivers repeatedly drove over the intersection of Route-4 vs. Jordan Rd by making a left turn as shown in Fig. 3. This intersection is actuated, coordinated and somewhat isolated: on Route-4 (the main road), the next signalized intersection is about 1 mile from both the north (at Williams Rd) and the south (at Blooming Grove Dr); on Jordan Rd (the minor road), there is also no nearby signalized intersection. The driving time of the loop is about 2-4 minutes. Each vehicle was equipped with a GPS logger so that the travel times of equipped vehicles between VTL1 and VTL2 can be obtained, together with the second by second trajectory of each vehicle. This resulted in 126 travel time measurements. At the roadside, the signal timing of the left turn was recorded together with the traffic volume and queue length information. During the experiment period, the measured total traffic flow was 441. This means that the penetration rate of the equipped vehicles was about 30% (126 out of 441). Two video cameras were also deployed at VTL1 and VTL2 respectively. From the video data, we were able to detect the times vehicles arriving at VTL1 and leaving VTL2 for each of the 441 left-turning vehicles. The travel times of these 441 vehicles were thus obtained, which are the ground-truth to validate the model results. The intersection was under mild congestion for the PM peak hour. During the experiment period, data for about 60 cycles were collected. It was observed that some vehicles waited for 2 cycles to make the left turn (i.e., over saturation happened) but no vehicle waited for more than 2 cycles. The parameters used by the queue estimation model are shown in Table 1.

[Place Table 1 about here]

[Place Fig. 3 about here]

[Place Fig. 4 about here]

The experimental site is for a mild congested intersection. To further test the models and algorithms on more congested intersections, we select one intersection in simulation. Fig. 4 shows the simulation network in City of Fresno, California. We selected the left-turn from McKinley Ave to N Blackstone Ave as our study site. The selected intersection is actuated and uncoordinated. The simulation network was developed in Paramics as part of the Corridor Management Plan Demonstration project (CCIT, 2006). The simulation model was well calibrated against field collected data, following the micro-simulation development and calibration guideline developed by FHWA (Dowling et al., 2004). The reader can refer to Liu and Jabari (2008) for details on how the simulation model was developed and calibrated.
Delay-based Definition of Queued Vehicles. Determining whether a vehicle is queued is not as straightforward as it first sounds, mainly due to the existence of the “moving queue.” A moving queue consists of vehicles that are affected (i.e., slowed down) but not fully stopped by the (standing) queue. The determination of a moving queue usually has to be done manually based on engineering judgment and thus may vary depending on specific observers. As shown in Liu et al. (2009), vehicles in a moving queue are those that join the queue after the rear of the original (standing) queue starts to move. As they are affected by the queue, they should be considered as queued vehicles. Although they do not make any contribution to the maximum queue length in terms of distance, they do contribute to the maximum queue length in terms of the number of queued vehicles. A common feature of vehicles in a standing queue or moving queue is that they are affected by the queue, i.e., they are delayed.

In this paper we define queued vehicles as those that are delayed at the intersection. For this purpose, we define two measures: the minimum traversal time (MTT) between VTL1 and VTL2, and a threshold, $\Delta T$. MTT describes the travel time that a vehicle needs to traverse the intersection from VTL1 to VTL2 in free flowing condition, i.e., not affected by the queue or signal timing. MTT is not simply the distance between VTL1 and VTL2 divided by the free flow speed, for two reasons. First, vehicles may slow down when passing an intersection. Fig. A1 in Appendix A depicts the speeds of non-queued vehicles (determined via observing the detailed vehicle trajectories) from the Next Generation Simulation (NGSIM) data. We can see that the speeds are generally lower than the speed limit (i.e., free flow speed) of the arterial road which is 35 mph (Cambridge Systematics, 2007). Secondly, for left turn non-queued vehicles, they need to slow down even more to make the left turn. Fig. A2 in Appendix A shows the speeds of non-queued left turn vehicles at the experimental site in Fig. 4 measured from GPS loggers. The figure clearly shows the slow-down of vehicles (from more than 30mph to about 15 mph) in order to make the left turn. For both cases, using simply the speed limit is not reasonable. For this reason, we obtain MTT of an intersection via field observations. If we collect the travel times of $n$ non-queued vehicles, denoted as $t_{ti}$ for $i=1, \ldots, n$, MTT can be defined as their average travel time, i.e., $MTT = \frac{\sum t_{ti}}{n}$. The threshold $\Delta T$ can then be used to define queued vehicles.

That is, for the $j$-th vehicle, if its travel time $tt_j \geq MTT + \Delta T$, the $j$-th vehicle is a queued vehicle; otherwise it is not. Fig. A3 in Appendix A illustrates the threshold $\Delta T$ and how it can be used to define queued vehicles. The threshold is to account for possibly different driving behaviors of different drivers (i.e., some are slower and some are faster even there is no queue).

In this paper, we use $\Delta T = 3$ seconds for all case studies. We also tested for different values of $\Delta T$ and found that the results are not sensitive for $\Delta T = 2 - 5$ seconds. Table 1 shows the MTT values for the field test site and the simulation site, together with the saturation flow rate and jam density. We recognize here that all these parameters may need to be calibrated at each location in order to produce better results, which however were not done in the current research.

6.2 Results of the Field Experiment
Fig. 5(a) depicts the calculated queuing delays (the asterisks) by applying the method in Section 5.1 and the estimated queuing pattern curve (the piecewise linear curve) for the field experiment, using travel times obtained from video data. This represents a 100% penetration. We can see that the queuing delay decreases approximately linearly within each cycle and the observed delays overlay well with the estimated pattern. Fig. 5(b) shows a “zoom-in” version of Fig. 5(a) for about 17 minutes from 16:33 to 16:50. In the figure, vertical dashed lines represent start of red times (also start of cycles), while vertical dotted lines represent start of green times. There are 16 cycles for this time window. Two of them (i.e., the 6th and 10th cycles) have only one queued vehicles for which the estimation algorithms cannot be applied due to insufficient samples (at least 2 queued vehicles are needed, see Fig. 2(c)). This is indicated using triangles in Fig. 5(b). For the entire one hour (i.e., the one shown in Fig. 5(a)), about 18% of all cycles have insufficient samples (i.e., 0 or 1 queued vehicles). From the figure, we can detect whether a cycle (say the n-th cycle) is over-saturated by comparing the time the delay first becomes zero (i.e., \( T^n_r \)) and the start time of the next cycle (i.e., \( T^{n+1}_r \)). In particular, \( T^n_r > T^{n+1}_r \) indicates the n-th cycle is over-saturated. Fig. 5(b) shows one cycle when this happens. In Fig. 5(a), the two circles indicate over-saturated cycles, which are both confirmed via the video.

Fig. 6(a) and 6(b) show similar figures as compared to Fig. 5(a) and 5(b) respectively, but using travel times obtained from GPS loggers. This represents the case when mobile sensor penetration is 30%. We can see in Fig. 6(a) that, if there are sufficient number of samples on queued vehicles (2 or more), the delay pattern can be estimated, which still exhibits the same characteristics by decreasing within a cycle. The queuing delay samples also match fairly well with the estimated pattern. However, there are more cycles that have insufficient samples of queued vehicles (0 or 1) under the 30% penetration. For those cycles, queue delay pattern curve cannot be estimated. In Fig. 6(a), about half of the cycles have zero or one sample and thus the algorithm cannot be applied. In Fig. 6(b), there are 4 cycles that have 0 sample (indicated using squares) and 4 cycles that have 1 sample (indicated using triangles). This shows that as penetration decreases, the estimation performance also degrades mainly because more cycles have insufficient samples. Fig. 6(b) illustrates how over-saturation is detected using the GPS logger data. The circle in Fig. 6(a) indicates the detected over-saturated cycle using GPS logger data from 16:33 to 16:50. The other oversaturated cycle however cannot be detected due to lack of samples.

[Place Fig. 5 about here]

[Place Fig. 6 about here]

Fig. 7(a) depicts the comparison between the observed queue lengths (the solid line with ‘.’ signs) and the estimated queue lengths by the proposed algorithm (the dashed line) for the field test (100% penetration). We can see that the estimated queue length can capture well the pattern changes of the observed queue length curve. In Liu et al. (2009), the Mean Absolute Percent Error (MAPE) of the maximum queue lengths of all cycles was used to evaluate the performance of the queue estimation algorithm. In this paper we use both the Mean Absolute Error (MAE) and MAPE of the maximum queue lengths of all cycles for fair comparisons between congested and uncongested situations. First, MAE is defined as:
\[ MAE = \frac{\sum_{i=1}^{N} |\Delta x_i|}{N}. \] (19)

where \( \Delta x_i \) is the error term defined as:

\[ \Delta x_i = \bar{x} - x^*. \] (20)

Here \( \bar{x} \) is the estimated value and \( x^* \) is the true value. The MAPE can then be defined as:

\[ MAPE = \frac{\sum_{i=1}^{N} |\Delta x_i|}{\sum_{i=1}^{N} x^*}. \] (21)

Notice the difference between MAE and MAPE is that the latter is a ratio of the former and the maximum queue length (\( x^* \)). Since for congested intersections, \( x^* \) tends to be larger than that for uncongested intersections, MAE is less impacted by the congestion level of an intersection as we will show later. The MAE and MAPE calculation in this paper only considers cycles that the estimation can be performed (i.e., for cycles with 2 or more samples of queued vehicles).

A zoom-in version of Fig. 7(a) is shown in Fig. 7(b) from 16:33 to 16:50. It shows the noticeable discrepancies between the observed and estimated queue length curves especially in terms of the actual shapes of the curves. The estimated queue length curve is piece-wise linear, while the actual queue length curve is more complex. This is mainly due to the uniform arrival assumption applied in the estimation which did not happen in the field as can be seen from the observed queue curve (i.e., it is not a straight line). For the same reason, the performance of the queue estimation algorithm developed in this paper is not very satisfactory, as indicated by the MAEs (1 – 2 vehicles) and MAPEs (20-30%) in Table 1. In section 6.4, we provide more discussions on this and propose some initial thoughts on how this may be resolved.

Fig. 8(a) and 8(b) show respectively the queue length curves estimated by using GPS logger data from 16:33 to 16:50. The MAE is nearly 2 vehicles and the MAPE is about 30% as shown in Table 1. Similar to the queuing delay pattern in Fig. 6(b), there are eight cycles that the estimation algorithm cannot be applied due to insufficient samples in Fig. 8(b). In Fig. 7(b), however, there are only two such cycles.

[Place Fig. 7 about here]

[Place Fig. 8 about here]

By comparing Fig. 5 and Fig. 6, as well as Fig. 7 and Fig. 8, we can observe that (1) as penetration decreases from 100% to 30%, there are more cycles that the algorithms developed in this paper cannot be applied properly due to insufficient travel time samples of queued vehicles (this can be seen more clearly later in Fig. 9(b)); (2) the estimation results of those cycles that have sufficient samples also degrade but not very dramatically; and (3) for those cycles with insufficient samples (zero or one sample), they are most likely the cycles with fewer queued vehicles. This is true especially when the penetration rate is high. For example, comparing Fig.
6(b) with Fig. 5(b), we can see that the four zero-sample cycles in Fig. 6(b) (i.e., the 2nd, 5th, 6th, 10th cycles as indicated using squares) have only 4, 3, 1, 1 queued vehicles respectively. The four one-sample cycles (i.e., the 7th, 8th, 11th, 12th cycles as indicated using triangles) have 8, 4, 3, 4 queued vehicles respectively. The other eight cycles with more than 2 samples typically have more than 5 queued vehicles except the 13th cycle. We also see that even for a given penetration, the actual queued vehicles and numbers of observed samples in different cycles may vary significantly. This shows both the fluctuation of vehicle arrivals across cycles (due to the nature of the real world traffic flow) and the randomness of sampling under a given penetration.

Fig. 9(a) depicts the MAEs of maximum queue lengths for different penetration rates (from 20% to 100% using 5% as the increment and 50 runs for each case) for the field test. The asterisks for each penetration show the MAEs for all 50 random draws (except the 100% penetration which has only 1). The solid line depicts the average MAE for each rate, which shows a decreasing trend as the penetration rate increases. The figure shows that if the penetration is 30%, the proposed algorithm can produce estimation of intersection queue lengths with average MAE about 1.7 (vary from nearly 1.2 to about 2.0). When the penetration rate increases to 100%, the MAE is about 1.2, showing that the MAE decreases slightly with penetration rates. Since, as shown before, the MAE is calculated only for cycles with sufficient samples so that the algorithm can be successfully applied, the results imply that with more samples in one cycle, the performance of the queue estimation algorithm does get improved. The improvement however is not dramatic. Notice also that the variation of the performance (i.e., the spread of the asterisks under a given penetration) is much smaller when the penetration rate is higher.

Fig. 9(b) shows the success rate, defined as the percentage of cycles that the proposed algorithms can be successfully applied (i.e., cycles that have 2 or more samples of queued vehicles), vs. the penetration rate. This figure is produced in a similar fashion as Fig. 9(a). We can see that when the penetration increases, the success rate increases monotonically. In particular, if the penetration is 30% (e.g., the field experiment scenario), the queue length estimation can be performed for about 40% of the cycles (varying from 30% - 50%); as the penetration increases to 60%, this percentage increases to about nearly 70% (varying from 55% - 75%); under 100% penetration, there are still about 18% cycles that the proposed algorithm cannot be applied due to insufficient samples. This indicates that penetration rates impact significantly the success rate of the proposed algorithms in this paper. In order for the algorithms to work, a significant penetration (larger than 30%) is needed, while 100% penetration is not necessary.

The above findings on how MAE and success rate change with respect to the penetration rate may be site-specific in terms of the actual values; however we believe that the general trend should also hold for other sites. For example, we tested the algorithm using NGSIM data at the Peachtree St in Atlanta, Georgia (Cambridge Systematics, 2007) and obtained similar results. Since the NGSIM data also experienced mild congestion, similar to the field experiment, detailed results are omitted here. We show in the next section results of some simulation data.

[Place Fig. 9 about here]
6.3 Results of the Simulation

We ran the simulation model from 3:30 pm to 6:30 pm and present the results for the most congested period of 4:20 – 5:10 pm. The actual signal timing parameters and the travel times of all vehicles between VTL1 and VTL2 were recorded (see Fig. 4(b) and 4(c)). We also manually recorded the queuing information for each cycle.

Fig. 10(a) first depicts the queuing delay pattern from 4:20 pm to 5:10 pm. There are 33 cycles within this 50-minutes period and 19 of them are oversaturated, indicating that this intersection is very congested for this period of time. The oversaturated cycles are indicated using circles in Fig. 10(a). The proposed algorithm can correctly identify 17 out of the 19 oversaturated cycles (solid circles in Fig. 10(a)). The remaining 2 oversaturated cycles (dashed circles in Fig. 10(a)) only had one oversaturated vehicle, which represent the boundary case between non-oversaturated and oversaturated cycles. We can also see that the measured queuing delays in Fig. 10(a), i.e., the asterisks, follow well the estimated linear queuing delay pattern curves. Fig. 10(b) shows a “zoom-in” version of Fig. 10(a) from roughly 4:20 – 4:40 pm, similar to Fig. 5(b), which includes 13 cycles. The figure shows more clearly how the oversaturated cycles are identified.

Fig. 11(a) and 11(b) show, respectively, the real time queue length of the intersection and a “zoom-in” version for the 13 cycles from 4:20 pm to 4:40 pm. We can see that the estimated queue length curve can track well the actual queue length curve, but the actual shapes of how queue length varies do not match very well. Fig. 12(a) and 12(b) further depict how the MAE and the success rate of queue length estimation vary under different penetration rates for the simulation data. As the penetration increases from 20% to 100%, the MAE decreases from 1.9 (varies from 1.1 to 2.5) to about 1.2. Also the success rate increases from nearly 60% (varies from 35% to 72%) to 100%.

Obviously, under the same penetration rate, we can usually obtain more samples of queued vehicles when the intersection is more congested because it has larger volumes and more vehicles are queued. For example, the 16 cycles in Fig. 5(b) for the field test have 98 totally queued vehicles respectively. The 13 cycles in Fig. 10(b) for the simulation have 122 total queued vehicles respectively. In average, the simulation data have 9.4 queued vehicles per cycles, which is 1.5 times of that for the field test cycles (6.1 in average). Comparing the results for the field test and the simulation data, we can see that the general trends of how the MAEs and success rates change with the penetration rate are quite similar. The MAEs does not change much between the field test (less congested intersection) and the simulation (more congested intersection), indicating that once the number of samples are sufficient to apply the algorithm,
the performance of the estimation only improves slightly with more samples. However, with the sample sizes increases (e.g., for more congested intersections), the success rate increases significantly as shown by Fig. 9(b) and 12(b) because more cycles will have more than 2 samples of queued vehicles. The comparisons show that the models and algorithms presented in this paper generally perform better for heavily congested intersections than less congested ones. This finding may also depend on the differences between simulation and real world traffic and other factors, and will need to be verified in real world traffic especially on heavily congested intersections. To further test the observation, we calculated the MAE and success rate for the simulation intersection for one less congested period (3:30 – 4:30 pm). Due to space limitations, the details are omitted here. In summary, as the penetration increases from 20% to 100%, the MAE decreases from 1.9 (varies from 1.2 to 2.7) to about 1.7. Also the success rate increases from nearly 45% (varies from 25% to 60%) to 90%. Therefore, the MAE varies in a similar fashion as the congested period while the success rate is significantly smaller (over 10%).

In summary, both the penetration rate and traffic conditions can impact the performance of the queue estimation algorithm presented in this paper. As penetration increases or the intersection gets more congested, the success rate will increase significantly since more samples will be obtained for each cycle. The MAE will also reduce (improved) but not significantly, in the range of 1 – 2 vehicles. Notice that if MAPE is considered, since there will be more queued vehicles under congested situations, MAPE tends to be smaller under the same penetration for congested periods than less congested periods. This can be seen clearly in Fig. 13 which shows the MAPEs for the field test, and the congested and uncongested periods of the simulation. Notice that the MAPE may not decrease monotonically, e.g., the MAPE of uncongested period of simulation, as shown in Fig. 13.

[Place Fig. 13 about here]

6.4 Discussions

We discuss in this subsection the limitations of the models and algorithms developed in this paper, and present some initial thoughts on how to address them. The most restrictive assumption in this research is the uniform arrival assumption, which as shown in subsection 6.2 is not really true in reality. To clearly show this, we look at Fig. 5(b) from 16:33 to 16:50. There are 16 cycles for this time window. Fig. 14 shows the distribution of the time headways for each cycle (based on 100% penetration video data). In the figure, the vertical lines separate cycles. Between any two consecutive lines, the histogram of headways is shown for a cycle using column charts. Each column (could be blank) represents a 5-second interval. The y-axis represents the percentage of vehicles with headways within a specific interval. For example, the first cycle has about 55% vehicles with headways less than 5 second, 18% vehicles with headways from 5-10 seconds, and 27% vehicles with headways from 10-15 seconds. Due to space limitations, the unit for the x-axis is not shown. Clearly, the 6th cycle is approximately uniformly distributed but the other cycles are not.
A closer look at Fig. 5(b) reveals that, there are actually four cases regarding vehicle arrival patterns: arrivals Spread Out (SO) during the entire cycle especially the red time (i.e., cycles 2,3,4,7,8,9,11), No-Arrival at the Beginning (NAB) of the cycle (i.e., cycles 14, 15), No-Arrival at the Middle (NAM) of the cycle (i.e., cycles 1,5,16), and No-Arrival at the Rear (NAR) of the cycle (i.e., cycles 5,12,13,15). SO is exclusive with respect to any of the NAB/NAM/NAR, while one of the latter three does not exclude the other. Each cycle in Fig. 5(b) is assigned with one or multiple of these patterns in Table 2, which also gives the estimated and observed maximum queue lengths for each cycle under the 100% penetration (i.e., for Fig. 5(b)). A note for the 5th cycle is that the maximum queue length for this cycle happens at the very beginning of the cycle, containing the residual queue (in total 5 vehicles). This results in 8 observed queuing vehicles by considering the 3 newly arrived vehicles during the red time.

[Place Fig. 14 about here]

[Place Table 2 about here]

We can see from Table 2 that most SO patterns have small relative errors (most less than 2) calculated via equation (19). This indicates that although SO does not necessarily mean uniform arrival, it seems a fairly reasonable approximation to uniform arrival when estimating real time queue lengths. The other three patterns however may lead to relatively high errors (highlighted in Table 2 in grey). This empirically shows that the NAB/NAM/NAR arrival patterns are particularly problematic if the uniform arrival assumption is applied. To further illustrate how the estimation errors distribute, we show in Fig. 15 the histogram of the errors for all cycles of the field test. It shows that most cycles have errors between -1.5 to +1.5 (75%) and nearly 55% cycles have errors between -1 to +1.

The problems of NAB/NAM/NAR will become more severe if the penetration is not 100%. For example, for the 30% penetration case in Fig. 6(b), the original SO cases (under 100% penetration) may become NAB (e.g., cycle 3), NAM (e.g., cycle 4), or NAR (e.g., cycle 9), or a combination of them. A non-SO case may also become other types of non-SO cases. For example, cycle 13 is NAR under the 100% penetration, which however becomes NAB and NAR under the 30% penetration. The most critical issue, if the penetration is less than 100%, is that one cannot tell if an NAB/NAM/NAR case accurately reflects the actual vehicle arrival patterns or merely due to the fact that the samples do not represent the actual arrival pattern.

[Place Fig. 15 about here]

To relax the uniform arrival assumption, one needs to investigate in detail the arrival pattern especially how platoon forms and dissipates (Geroliminis and Skabardonis, 2005). This is particularly critical for closely spaced intersections. Notice that if the arrival pattern is not uniform within a cycle, discontinuities cannot happen in the cycle, which are related to signal timing changes only. However, non-smoothness may occur, e.g., two platoon arrivals with no-arrival in between. In this case, the specific characteristics of the non-smoothness under these
conditions can be investigated to reveal their connections to actual traffic state changes. Such investigations may help construct more accurate real-time delay and queue length estimation when the arrival pattern is not uniform. Research in this direction will be pursued in the future.

Another way to partially address the non-uniform arrival issue is to study the queue discharging process. For a given sample in a cycle, it may be possible to estimate the position of the corresponding vehicle in the queue of the cycle by investigating the discharging process of the queue and the acceleration/deceleration of the vehicle. This will provide a lower bound of the maximum queue length and can be used to correct the estimation results produced by the methods in this paper. As a result, as long as a cycle has at least one sample of queued vehicles (e.g., the four cycles indicated using triangles in Fig. 5(b) and 6(b)), an estimated queue length can be generated. This will improve both the estimation accuracy and the success rate. The authors are now investigating specific schemes and algorithms for this and preliminary results can be found in Hao and Ban (2010).

7. Conclusions

In this paper, we proposed methods to estimate real-time queue lengths at signalized intersections using sample travel times from mobile traffic sensors. Travel times were used as model input instead of detailed trajectories for privacy protection purposes. The method consists of three major components: processing of the raw sample vehicle (total) delays to the queuing delays, estimation of the queuing delay pattern using sample queuing delays and estimation of the queue rear no-delay times (QRNATs), and the calculation of the maximum/minimum queue lengths and the construction of the real-time queue length curve. The key concept of the method is QRNATs, which relates closely to the non-smoothness of queuing delay patterns and queue length changes. The focus on queuing delay pattern also makes the estimation of the delay curve much simpler. The unique feature of the method is that instead of using traffic volume or occupancy as the input (as most modeling techniques based on fixed-location sensors would do), the input of the model is sample travel times, which can be obtained directly from mobile sensors. The proposed method thus presents a reverse thinking process compared to standard methods, which the authors think may be useful for traffic modeling using mobile data. The model and algorithm were tested using field experiment and simulation data, with their limitations discussed.

The current model is based on rather restrictive assumptions as listed in Section 2. As a result, many limitations exist and the most problematic one on uniform arrival is discussed in Section 6.4. Below we summarize other limitations based on our understanding, which also point to future research directions to further improve the methods presented in this paper:

(1) The assumption that signal timing information is available was commonly used by many previous queue estimation models. Sample travel times from mobile sensors can actually be used to estimate cycle by cycle signal timing information (such as cycle length and the duration of effective red time for each cycle). The authors are investigating this issue and
results will be presented in subsequent papers. Vehicle accelerations and decelerations can be considered, e.g., by the method in Liu et al. (2009). The assumption that queue never passes the upstream VTL may be relaxed by looking at changes of delay patterns in case some delays are lost due to queue spillover. The methods presented in Skabardonis and Geroliminis (2008) and Wu et al. (2010) may be applied in this case. The assumption that a vehicle must be cleared in two cycles is mainly for the ease of discussion. From the vehicle delay and signal timing information, the number of cycles a vehicle had to wait before it was released can be obtained. This information can be used for queuing delay pattern and queue length estimation, in a way similar to that discussed in Section 5.1. Therefore the methods presented in this paper can be easily extended to cases when vehicles have to wait for more than 2 cycles to be cleared. We do not regard the high penetration assumption as a major limitation, although it does prevent us from applying the proposed model directly to the field at the moment. Rather it is the scenario we target on when penetration of mobile sensor data become large in the future.

(2) We only assumed the availability of travel times (or delays) in this paper for model development. It may be worthwhile investigating what additional data can be retrieved from mobile sensors, which can help resolve some of the critical issues (like uniform arrival assumption) without severely impact privacy at the same time.

(3) In many cases, aggregated loop detector data are available as well. Therefore, exploring the most effective way to fuse fixed-location and mobile sensor data is an important future research topic.

(4) Last but not the least, the proposed method was only tested at isolated intersections and in microscopic traffic simulation. To further validate the model, testing on more diversified arterial intersections (isolated, closely spaced, coordinated, etc.) will be crucial, especially on heavily congested intersections.

With the advent of emerging technologies and transportation applications (such as IntelliDrive (2010) and traffic crowdsourcing (Inrix, 2010)), estimating accurate and reliable real time performance measures for signalized intersections and arterial networks using mobile sensors (possibly with fusion of existing fixed location sensor data) is a new and challenging area, which calls for innovative modeling techniques to solve newly emerging critical issues (such as privacy). The proposed method is just the first step towards this direction by focusing on real time intersection queue length estimation using privacy preserving mobile sensor data. We hope that this will open up discussions for developing more robust models and algorithms for real time arterial performance measurements using mobile sensors, as well as for other critical transportation applications.

Acknowledgement

The authors would like to thank the two anonymous referees for their insightful comments and helpful suggestions on an earlier version of the paper. This research is partially sponsored by the
California Department of Transportation (Caltrans) via a grant to the California Center for Innovative Transportation (CCIT) of the University of California, Berkeley, and the National Science Foundation Grant CMMI-1031452. We appreciate Dr. Alexandre Bayen and Ryan Herring for their support and discussions. The authors also thank the New York State Department of Transportation (NYSDOT) for supporting the field tests in the Albany, NY area.

References


California Center for Innovative Transportation (CCIT), 2006. Final report of task order 3: corridor management plan demonstration. California Center for Innovative Transportation, University of California, Berkeley.


Herrera, J.C., Bayen, A., 2009. Incorporation of Lagrangian measurements in freeway traffic state estimation. Accepted by Transportation Research Part B.


Newell, G.F., 1965. Approximation methods for queues with application to the fixed-cycle
prediction of delay and maximum queue length at a signalized intersection. Transportation
Research Record, 2035, 88–96.
Journal of Intelligent Transportation Systems. 12 (2), 64–74.
collection for generating actuated controller performance measures. Transportation Research
Record, 2035, 97-106.
dynamics at signalized intersections. Transportation Research, 13A.
signalized links. Transportation Research, Part C 16 (1), 18–35.
access control address matching. ITE Journal, 78(6), 20-23.
Webster, F.V., 1958. Traffic signal settings. Road Research Laboratory Technical Paper No. 39,
HMSO, London.
resolution traffic signal data. Transportation Research, Part C 18(4), 626-638.
Appendix A: Definition of Queued Vehicles

[Place Fig. A1 about here]

[Place Fig. A2 about here]

[Place Fig. A3 about here]
List of Figure Captions

Fig. 1(a). Fundamental diagram
Fig. 1(b). Shockwaves propagation under normal
Fig. 1(c). Shockwaves propagation under over-saturation
Fig. 2. Intersection delay and queuing delay patterns
Fig. 3. The field experiment site
Fig. 4(a). Simulation network
Fig. 4(b). Study intersection
Fig. 4(c). Study intersection in Paramics
Fig. 5(a). Queuing delay pattern for field experiment (100% penetration)
Fig. 5(b). Queuing delay pattern for field experiment for 16:33-16:50 (100% penetration)
Fig. 6(a). Queuing delay pattern for field experiment (30% penetration)
Fig. 6(b). Queuing delay pattern for field experiment for 16:33-16:50 (30% penetration)
Fig. 7(a). Real time queue length for field experiment (100% penetration)
Fig. 7(b). Real time queue length for field experiment for 16:33-16:50 (100% penetration)
Fig. 8(a). Real time queue length for field experiment (30% penetration)
Fig. 8(b). Real time queue length for field experiment from 16:33-16:50 (30% penetration)
Fig. 9(a). MAE of queue length vs. penetration rate (field test)
Fig. 9(b). Success rate vs. penetration rate (field test)
Fig. 10(a). Queuing delay pattern for simulation data (100% penetration)
Fig. 10(b). Queuing delay pattern for simulation data for 4:20-4:40 pm (100% penetration)
Fig. 11(a). Real time queue length for simulation data (100% penetration)
Fig. 11(b). Real time queue length for simulation data for 4:20-4:40 pm (100% penetration)
Fig. 12(a). MAE of queue length vs. penetration rate (simulation)
Fig. 12(b). Success rate vs. penetration rate (simulation)
Fig. 13. MAPE vs. penetration rate for field test, congested and uncongested periods of simulation
Fig. 14. Distributions of headways within a cycle (16:33 – 16:50)
Fig. 15. Percentage error for field experiment
Fig. A1. Speeds of non-queued vehicles for a through movement

Fig. A2. Speeds of non-queued vehicles for a left turn movement

Fig. A3. Illustration of $\Delta T$
List of Table Captions

Table 1 Characteristics of Test Sites and Algorithm Performances

Table 2 Cycle-by-cycle maximum queue length estimation (16:33 – 16:50)
Fig. 1(a). Shockwaves propagation under normal

Fig. 1(b). Shockwaves propagation under normal

Fig. 1(c). Shockwaves propagation under over-saturation

Fig. 2(a). Fundamental diagram
Fig. 2. Intersection delay and queuing delay patterns
Fig. 3. The field experiment site
Fig. 4(a). Simulation network

Fig. 4(b). Study intersection

Fig. 4(c). Study intersection in Paramics
Fig. 5(a). Queuing delay pattern for field experiment (100% penetration)

Fig. 5(b). Queuing delay pattern for field experiment for 16:33-16:50 (100% penetration)
Fig. 6(a). Queuing delay pattern for field experiment (30% penetration)

Fig. 6(b). Queuing delay pattern for field experiment for 16:33-16:50 (30% penetration)
Fig. 7(a). Real time queue length for field experiment (100% penetration)

Fig. 7(b). Real time queue length for field experiment for 16:33-16:50 (100% penetration)
Fig. 8(a). Real time queue length for field experiment (30% penetration)

Fig. 8(b). Real time queue length for field experiment from 16:33-16:50 (30% penetration)
Fig. 9(a). MAE of queue length vs. penetration rate (field test)

Fig. 9(b). Success rate vs. penetration rate (field test)
Fig. 10(a). Queuing delay pattern for simulation data (100% penetration)

Fig. 10(b). Queuing delay pattern for simulation data for 4:20-4:40 pm (100% penetration)
Fig. 11(a). Real time queue length for simulation data (100% penetration)

Fig. 11(b). Real time queue length for simulation data for 4:20-4:40 pm (100% penetration)
Fig. 12(a). MAE of queue length vs. penetration rate (simulation)

Fig. 12(b). Success rate vs. penetration rate (simulation)
Fig. 13. MAPE vs. penetration rate for field test, congested and uncongested periods of simulation
Fig. 14. Distributions of headways within a cycle (16:33 – 16:50)
Fig. 15. Percentage error for field experiment
Fig. A1 Speeds of non-queued vehicles for a through movement
Fig. A2 Speeds of non-queued vehicles for a left turn movement
Fig. A3 Illustration of $\Delta T$
<table>
<thead>
<tr>
<th>ID</th>
<th>Test Site</th>
<th>Distance Btwn.VTLs (Ft)</th>
<th>Minimum Traversal Time (sec)</th>
<th>Sat. flow (vphpl)</th>
<th>Jam density (vpmpl)</th>
<th>MAE (MAPE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Field Test (30%)</td>
<td>860</td>
<td>19.60</td>
<td>1800</td>
<td>200</td>
<td>1.92 (30.3%)</td>
</tr>
<tr>
<td>2</td>
<td>Field Test (100%)</td>
<td>860</td>
<td>19.60</td>
<td>1800</td>
<td>200</td>
<td>1.17 (21.9%)</td>
</tr>
<tr>
<td>3</td>
<td>Simulation (30%)</td>
<td>671</td>
<td>13.60</td>
<td>1800</td>
<td>200</td>
<td>1.68 (16.6%)</td>
</tr>
<tr>
<td>4</td>
<td>Simulation (100%)</td>
<td>671</td>
<td>13.60</td>
<td>1800</td>
<td>200</td>
<td>1.15 (14.2%)</td>
</tr>
<tr>
<td>ID</td>
<td>Estimated Queue Length</td>
<td>Observed Queue Length</td>
<td>Relative Error</td>
<td>Arrival Pattern</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>------------------------</td>
<td>-----------------------</td>
<td>----------------</td>
<td>----------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>6.5</td>
<td>9</td>
<td>-2.5</td>
<td>NAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4.7</td>
<td>4</td>
<td>0.7</td>
<td>SO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>8.9</td>
<td>11</td>
<td>-2.1</td>
<td>SO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>16.5</td>
<td>16</td>
<td>0.5</td>
<td>SO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>8.3</td>
<td>8</td>
<td>0.3</td>
<td>NAM/NAR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>/</td>
<td>1</td>
<td>/</td>
<td>/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7.6</td>
<td>8</td>
<td>-0.4</td>
<td>SO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>3.9</td>
<td>4</td>
<td>-0.1</td>
<td>SO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>11.1</td>
<td>10</td>
<td>1.1</td>
<td>SO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>/</td>
<td>1</td>
<td>/</td>
<td>/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>2.9</td>
<td>3</td>
<td>-0.1</td>
<td>SO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>7.2</td>
<td>4</td>
<td>3.2</td>
<td>NAR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>5.2</td>
<td>3</td>
<td>2.2</td>
<td>NAR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>8.0</td>
<td>6</td>
<td>2.0</td>
<td>NAB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>8.1</td>
<td>10</td>
<td>-1.9</td>
<td>NAB/NAR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>6.4</td>
<td>5</td>
<td>1.4</td>
<td>NAM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>