An Infrastructure-less Vehicle Counting without Disruption

Jie Wu*, Paul Sabatino†, Jennifer Tsan†, and Zhen Jiang*†
*Dept. of Computer and Information Sciences, Temple University
†Dept. of Computer Science, West Chester University
jiewu@temple.edu, {PS772461, JT751802, zjiang}@wcupa.edu

***Abstract*—This paper presents a solution to count all moving vehicles in a target region. This is a large-scale counting that cannot be easily solved without a global view. However, there is no single force that can provide such a global view. To achieve an accurate result without either double- or miscounting, the local counting at each checkpoint is synchronized in our wireless communication by using the information carried by vehicles along the traffic flow. Our analytical and experimental results illustrate the correctness of the proposed scheme in both closed and open road systems - even when the wireless signal is affected by many factors. In this way, we provide an essential support for the resource management in VANETs.**

Keywords*—*Distributed algorithm, inter-vehicle wireless communication, resource management, traffic surveillance.

I. Introduction

The automotive industry is about to enter a period of transformative change, as relevant service support continues to develop. Instantly counting all of a given type of moving vehicles in the target region can provide critical information for us to improve the current road services such as traffic congestion control, disaster evacuation, and vehicle tracing. Our target region can expand as large as an entire state, and cannot have a single coverage at the global view level.

To precisely count the vehicles without any double- or miscounting becomes a multi-site problem. Note that the traffic flows dynamically, and each vehicle has an unpredictable speed, trajectory, and direction. It is very difficult for any single checkpoint to determine its role in miscounting or double-counting at the global view level. A synchronization is needed among all checkpoints.

More importantly, we cannot rely on any traffic pattern, which requires vehicles to change speed or trajectory in order to achieve certain regularity. In the extreme case when our counting scheme is applied to trace suspect vehicles, the target can deliberately drive in an unpredictable manner to avoid being caught, making any counting difficult to implement. Second, any global information, such as VIN, or centralized resource, such as the Internet access (of DMV/NDR databases), can be too expensive to cover the entire region, or becomes unavailable when it does not exist or is damaged. The inconsistent local views among different checkpoints increase the difficulty of our synchronization.

Our solution is motivated by the early work [1] to capture a consistent global status (also called a "snapshot") with the distributed algorithm. In our approach, the counting is initiated at the seed checkpoint(s) for all inbound traffic. The active status (of site u), in one bit on/off information, will be caught by the vehicles passing through via the wireless communication. It will be carried to every adjacent site along the traffic flow in a flooding-like manner. When an idle site v received this information, it can start to count its own inbound traffic. According to the order of their initializations, u is the "predecessor" of v. The status change at v will trigger a new flooding in the neighborhood. When an activated site received the counting status of any adjacent site, as the backwash of its outgoing flooding, it can stop the local counting along such an inbound direction without missing any vehicle. The entire process expands in a cascading manner like a wave. By precisely controlling the frontier wave and the backwash, our miscounting-free approach will not have any double-counting.

In this paper, we synchronize the counting in the entire target area under the "everyone" model, in which each site will apply the same generic process in a fully-distributed manner. The impact of the inconsistence among local views can be mitigated. The counting converges with the ultimate result at the global view level in a cost-effective way - without any infrastructure support at the global view. The contribution of our infrastructure-less counting is fourfold.

1) We first implement the counting scheme in an "everyone" model at each checkpoint in a fully-distributed manner, where only short range surveillance and communication are available.
2) We prove that, by precisely controlling each checkpoint, active or not, in our synchronization, neither mis- nor double-counting will occur.
3) Some extension work is provided so that our approach will still be effective when the overtake, lossy communication, and odd traffic patterns are considered. We also extend our work from the closed system to the open road system. Moreover, we provide an efficient method to gather those local views from the entire system and then to constitute the global result.
4) We develop a simulation. Its results verify the correctness of our approach and show its scalability as a practical solution for the large-scale problem.

The remainder of this paper is organized as follows: Section 2 introduces the target problem and related work. Section 3 provides some preliminary information. Section 4 presents our approach, extended from the closed system under a simple road model to the open system under the realistic road model. Its correctness will be analyzed and then be verified in Section 5 with our experimental results. Lastly, Section 6 concludes this paper and provides ideas for future research.

TABLE I. NOTATIONS

u	monitored intersection / checkpoint
$\{u, v\}$	a road segment with two adjacent intersections u and v
$u \leftarrow v$	the inbound traffic of u along $\{u, v\}$
$u \rightarrow v$	the outbound traffic of u along $\{u, v\}$
$n^i(u)$	set of neighboring checkpoints of u along inbound traffic
$n^o(u)$	set of neighboring checkpoints of u along outbound traffic
$c(u, v)$	local counter of inbound traffic $u \leftarrow v$
$c(u)$	local counter of all inbound traffic at u, $\sum_{v \in n^i(u)} c(u, v)$
$p(u)$	(set of) the predecessor of u whose counting triggers the one at u
$s(u)$	set of the successor(s) of u, i.e., $n^o(u) - p(u)$

II. TARGET PROBLEM AND RELATED WORK

Under our "everyone" model, we catch every vehicle's coming in any direction at each intersection where we install the camera accessories [2, 3]. To protect privacy, any ownership information such as VIN cannot be accessible. Considering the overhead cost and reliability of the entire counting process, our counting process does not necessarily rely on any centralized control or resource such as the Internet. Therefore, only exterior characteristics of the vehicle such as color, brand, and type are used to identify the target vehicle.

However, each local surveillance has a short range of vision and cannot cover the entire road segment from one end to the other. While the monitoring remains until each vehicle comes into the surveillance, some vehicles might have traveled many sites and may have been counted multiple times, i.e., double-counting. Unlike many existing services (e.g., [4, 5]) that have independent job(s) at each site, the statuses of each vehicle, counted or not, are interrelated among different checkpoints. Because the traffic is often unpredictable, the double-counting problem cannot be solved completely by any deployment strategy of checkpoints. Adopting image recognition to avoid double-counting is costly and cannot ensure 100% accuracy, since surveillance videos are taken from different angles and the vehicle's appearance is unpredictable.

In this paper, we provide a complete counting solution by synchronizing all checkpoints with the information received and carried by the vehicles. The recent technical advances of wireless communication in VANETs can ensure the success of each information exchange (e.g., [6]) and the delivery in the expected direction (e.g., [7]). The relative location of each adjacent vehicle in the flow can also be detected by such communication among neighboring vehicles (e.g., [8]). Based on these, our counting will adapt to any dynamic change along the road that is beyond the surveillance from intersections. As a result, the counting is ensured 100% correct.

III. PRELIMINARY AND NETWORK MODEL

Our counting is applied without any disruption to the traffic. Table I summarizes all of the notations used in this paper, which will be explained in the following.

A. Road

In this paper, we first adopt a simple road model, and then extend the work to a more realistic road system. A road segment, denoted by $\{u, v\}$, is a section of road that is separated by two adjacent intersections u and v. Viewed from the angle of u, outbound traffic from u to v and inbound traffic from v to u are denoted by $u \rightarrow v$ and $u \leftarrow v$, respectively. All adjacent intersections that can be reached via the outbound traffic are denoted by $n^o(u)$. All adjacent intersections at the other end of each inbound traffic flow are denoted by $n^i(u)$.

We assume that the road system is connected. To simplify the discussion, we also assume that each road segment is bidirectional (i.e., $n^o(u) = n^i(u)$ at each intersection u) and there is no overtake allowed. Each time, only one vehicle is allowed to enter the intersection and to make the turn. In our extension, the real metropolitan map is used and it allows multiple lanes and overtakes (i.e., traffic is not always FIFO). Some of the road segments can be one-way (i.e., unidirectional). That is, $n^o \neq n^i$. Multiple vehicles are allowed to pass the intersection simultaneously and roundabouts are considered.

B. Vehicles

Each vehicle can change its speed and trajectory in an unpredictable manner. It forms a node of VANETs, and its built-in equipment has sufficient power and capabilities to support the following functionalities:

- a directional communication [6] that can quickly send and receive a short message,
- a coarse-grained collaboration based on the approach in [8] that can detect whether this vehicle has been overtaken by another, and
- a store of checkpoint status (also called the label, i.e., one-bit on/off information) and counting result that can be carried to the next intersection and be forwarded to the corresponding receiver.

C. Checkpoint, communication, and counting

A checkpoint will be set at every intersection. Any vehicle entering the intersection u, say along inbound traffic $u \leftarrow v$, will be identified by its exterior characteristics (e.g., color, brand, or/and type) with a simple image recognition process (e.g., [3]). It can be counted in $c(u, v)$ before joining any outbound traffic flow.

The checkpoint has the same communication and capability as a vehicle node of VANETs, in order to exchange and share the information. In this way, u can obtain any status update of an adjacent checkpoint v and its counting result $c(v)$ from the traffic $u \leftarrow v$.

By recursively accumulating information from each adjacent checkpoint, we can obtain the final result at the global view level at a specific checkpoint, where the data sink is available. This sink can also accept the signal to start the entire counting process. Such a checkpoint is also called the "seed."

In our extensive work, we study the use of multiple seeds in order to speed up our counting process. Additional work is also provided under a more realistic model when the communication is prone to failures by many factors.

IV. THE PROPOSED COUNTING APPROACH

Initially, each checkpoint is inactive. Our counting will start from the seed checkpoint and at the corresponding intersection. After each checkpoint has initiated, all the vehicles passing through can be counted and will form the flow spreading out

Algorithm 1 Counting process, applicable to any intersection u under "everyone" model, in a closed, simple road system

```
Require: Checkpoint $u$ is deployed, with an initial "inactive"
    status, $p(u) = \phi$, and $s(u) = n^o(u)$.
Ensure: Each role of checkpoint $u$, the processing, and its
    local view as the result of vehicle counting.
1: Initialization of an inactive seed checkpoint:
    Activate local counting of each inbound traffic $u \leftarrow v$
    where $v \in n^i(u)$.
2: Neighbor Synchronization from an active checkpoint:
    Upon the frontier wave of counting, label the vehicle first
    joining any outbound traffic to successors $u \rightarrow v \in s(u)$.
3: Propagation to an inactive non-seed checkpoint:
    Set $p(u) = \{v\}$ and $s(u) = n^o(u) - p(u)$, when a
    labeled vehicle enters along $u \leftarrow v$. Then activate the
    local counting of each inbound traffic $u \leftarrow w \in s(u)$.
4: Reaction of an active checkpoint under synchronization:
    Stop the local counting in an inbound direction after a
    labeled vehicle enters the intersection from that direction.
5: Counting of an active checkpoint:
    Count in $c(u, v)$, for any unlabeled vehicle entering via
    the inbound traffic $u \leftarrow v$.
6: $c(u) = \sum_{v \in n^i(u)} c(u, v)$ stabilizes after the local counting
    activated for each inbound direction has ended.
```

along the outbound directions. When any inactive checkpoint encounters such a counted flow, its local counting will be activated and the above flooding will continue. After all checkpoints become active, every vehicle will have a chance to be counted, regardless of its driving route.

Our proposed work focuses on the control at the frontier of the propagation wave in order to avoid any double-counting. We first show how all vehicles can be counted precisely in the closed system under a simple road model (see Alg. 1). Then we introduce a cost-effective method (see Alg. 2) to collect the results that have been distributed to each checkpoint, in order to form the global view at the seed. After that, we will extend the above information constitution and collection.

A. Our approach in a simple road model

The idea is demonstrated in Fig. 1. First, in the initial phase (see Fig. 1 (a)), checkpoint 1 (which is equipped as the only sink and seed) will initiate its local counting of each inbound traffic, say $1 \leftarrow 2$ and $1 \leftarrow 3$. This is determined by $p(u) = \phi$ and $s(u) = n^o(u) = \{2, 3\}$. Respectively, $c(1, 2)$ and $c(1, 3)$ will be updated in phase 5 of Alg. 1.

In the simple road model, there is no overtake allowed, and every information exchange between a vehicle and the checkpoint is successful. After each local counting is initiated, the vehicle first joining an outbound traffic will always be upon the frontier wave of the flooding of counted vehicles. With phase 2 of Alg. 1, such a vehicle, say along $1 \rightarrow 2$ in Fig. 1 (a), will be notified with a label and then will bring this counting status to checkpoint 2.

After the inactive checkpoint $u = 2$ observes such a labeled vehicle approaching from $v = 1$, it can initiate its local counting (see phase 3 of Alg. 1) in order to ensure that each vehicle leaving from it has been counted. Using the same way

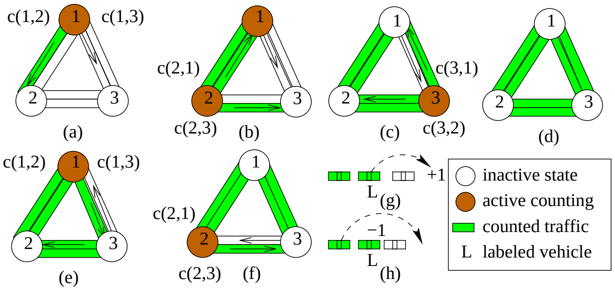


Fig. 1. Counting process in the closed road system with 3 intersections, where "1" is the seed and sink. (a) Initialization from seed(s), (b) propagation to a new checkpoint 2 and the corresponding synchronization, (c) end of local counting at 1 and 2, (d) end of the entire counting at a global snapshot, (e)-(f) possible intermediate configuration during the convergence of counting process, and counting adjustment for (g) overtaking and (h) being overtaken.

of the seed to label the outgoing vehicles, the counting status will propagate until each checkpoint obtains a chance to be active. According to the timing, we record $p(u) = \{v\}$ as the predecessor. Since all vehicles coming along $2 \leftarrow 1$ behind that labeled vehicle have been counted by checkpoint 1, we initiate the local counting for $2 \leftarrow 3$ only where $n^i(u) - p(u) = \{3\}$ (see Fig. 1 (b)).

As shown in Fig. 1 (b), when the active checkpoint $u = 1$ observes such a labeled vehicle approaching along $1 \leftarrow 2$, it will continue to count until that labeled vehicle enters the intersection. Now, all uncounted vehicles that drive along $1 \leftarrow 2$ can be counted by checkpoint 1. Since both 1 and 2 are in counting, no uncounted vehicle can enter $1 \leftarrow 2$. We stop the counting for $1 \leftarrow 2$ (in phase 4 in Alg. 1) to avoid any double-counting (see Fig. 1 (b)).

After receiving the label from each inbound traffic, $1 \leftarrow 2$ in Fig. 1 (b) and $1 \leftarrow 3$ in Fig. 1 (c), the activated local counting at checkpoint 1 can completely stop. The counters $c(1, 2)$ and $c(1, 3)$ stabilize (in phase 6 of Alg. 1).

Finally, each checkpoint ends its counting. We have the ultimate result, which has been distributed in the entire region (see Fig. 1 (d)). In the following Theorem 1, we will prove that this result is 100% correct, without any double- or miscounting. However, the progress of the convergency at each checkpoint is asynchronous; this is shown in Fig. 1 (c), (e), and (f), where each presents a possible intermediate configuration during the convergence from Fig. 1 (b) to (d). The unpredictable nature of traffic flow causes the chaos of the timing of each local counting and increases the difficulty of our synchronization to avoid mis- or double-counting.

Theorem 1 (correctness in a closed system). *Any mis- or double-counting can be avoided in the counting where each checkpoint's counting phase (i.e., phase 5 of Alg. 1) is synchronized under our control in phases 1 to 4 of Alg. 1.*

Proof: We first assume there is a miscounting. Note that a seed checkpoint will apply phase 1 and a non-seed checkpoint will apply phase 3, in order to propagate their counting status. In such a propagation, the successor set can include every inactive adjacent checkpoint. The frontier wave will form a closed line to include all activated checkpoints. If this miscounted vehicle never passed though an active checkpoint to experience the

Algorithm 2 Information collection for the global view where Alg. 1 is applied with a single seed checkpoint.

```
Require: Stable checkpoint $u$ in Alg. 1 phase 6.
Ensure: Global view of vehicle counter.
If $s(u) \neq \phi$, wait for $c(v)$ ($v \in s(u)$) reported from the vehicle coming along $u \leftarrow v$.
Ask a vehicle leaving along $u \rightarrow p(u)$ to carry the result $c(u) + \sum_{v \in s(u)} c(v)$ to $p(u)$.
```

counting phase, it will always travel outside of the counting area. This will lead to a contradiction, because every inactive checkpoint will get a chance to be active. Otherwise, this uncounted vehicle must use the road segment of the predecessor to approach a checkpoint in the counting area. Recursively, we can trace back to the seed checkpoint along such a kind of inbound traffic. This will lead to a contradiction of this uncounted status, because the seed checkpoint will count all inbound traffic. Therefore, we do not have any miscounting.

After that, we consider to avoid any double-counting. When a counted vehicle enters an intersection along the road segment from the predecessor, it will not be counted by the setting of phase 3. Otherwise, it must use the traffic flow from a successor. Then, it cannot surpass the labeled vehicle going along the road segment from that successor (see phase 2), which will stop each counting process ahead (see phase 4). Therefore, the statement is proven. ■

Next, we focus on an effective method for the seed checkpoint (also the sink) to collect stabilized counting results from each site in the entire road system. An easy way is to initiate a broadcast from each checkpoint to the entire system. To reduce the complexity and overhead cost, we simplify the collection process and present it in Alg. 2. Basically, each checkpoint waits for the end of every activated counting for the inbound traffic and obtains a stable result at the local view level (phase 6 in Alg. 1). Then, along the spanning tree built with p-s relation by phase 3 in Alg. 1, each non-seed checkpoint u will accumulate the reports from all successors ($\in s(u)$) to its own, and then report this to the predecessor (i.e., $p(u)$). The final report at the global view level can be obtained at the root seed.

B. Extensive work

Extension to non-FIFO traffic model. In our extended road model, the real metropolitan map is adopted. Multiple lanes are used and each of them allows overtake. The collaboration process in [8] is applied to detect the occurrence of overtake. Note that we just need to confirm such an occurrence of overtake before the labeled vehicle reappears in the surveillance (of the next checkpoint), not exactly when, where, or how long it took. After the labeled vehicle reaches the other end of the road segment, the corresponding counter needs to be adjusted as follows. For the labeled vehicle overtaking an uncounted one, the latter one will miss our counting (in either phase 3 or phase 4), and the counter value must be corrected by an increase of one (i.e., +1 in Fig. 1 (g)). Similarly, for each time the labeled vehicle is overtaken by a counted one from behind, the counter must be decreased by one (i.e., -1 in Fig. 1 (h)).

The vehicle-to-vehicle communication used in the labeling process is implemented by built-in equipment and cannot be manipulated by the driver. Usually, the change of relative location between vehicles is relatively slow, and gives us sufficient time to achieve a reliable detection.

Algorithm 3 Constitution of local views in the closed system, where overtake, multiple lanes, lossy wireless connection, and one-way streets all are considered.

```
Require: The same in Alg. 1.
Ensure: Local view of vehicle counter.
for each checkpoint $u$ deployed do
    Apply Alg. 1.
    In phase 2, set $c(u) = c(u) - 1$ when $u$ misses the contact with the target vehicle in labeling and cannot accomplish the notification process. Then the checkpoint will repeat this trial of labeling process in that direction until the first vehicle confirms the receiving of label.
    Phase 5 counting is extended to a multi-target tracking.
for each labeled vehicle moving along $u \leftarrow v$ do
    Apply collaborative V2V communication to maintain its relative position to each moving vehicle nearby.
    When it overtakes an uncounted vehicle in the front, $c(u) = c(u) + 1$.
    When it is overtaken by a counted vehicle from behind, $c(u) = c(u) - 1$.
```

Extension to multi-target tracking. In this extended model, we also allow multiple vehicles to enter the intersection simultaneously and consider the surveillance of a possible roundabout. With the existing techniques of image recognition (e.g., [3]), we can precisely identify each vehicle passing through or parking around the intersection (or roundabout).

Extension to lossy communication model. In phase 2 of Alg. 1, the label will be initiated, say at u, in order to synchronize the counting at the adjacent checkpoint v. When the information exchange with the outgoing vehicle fails, the initialization of the counting at the inactive checkpoint v will be delayed. This counted vehicle will be double-counted as the flow enters the surveillance of any other active checkpoint. Similarly, when checkpoint v has been activated, the counting of $v \leftarrow u$ must stop upon receiving such a label. The delay will also incur double-counting. To solve the problem, the local counter will be adjusted (decreased by 1) until receiving can be confirmed with the TCP acknowledgment in [6].

Extension for counting along one-way streets. One-way street is designed to direct vehicles to move in one direction, say from u to v. Thus, we have $n^o(v) = n^i(u) = \phi$. It typically results in higher traffic flow as, we don't have to monitor the on-coming traffic in the other direction. But synchronizing checkpoints at both ends u and v often means taking extra work because the information delivery is unidirectional. Alg. 3 shows a complete counting solution with the above extensive considerations. Theorem 2 proves that such counting is 100% correct, without any mis- or double-counting.

Theorem 2 (correctness of extended counting scheme). *The counting with Alg. 3 can avoid any mis- or double-counting.*

Proof: We prove that Alg. 1 can directly be applied to one-way streets. The necessity and sufficiency of other additional parts in Alg. 3 can be seen in the above discussion.

Any outbound traffic from u, if it is used in the predecessor-and-successor relation and initiates the counting at the adjacent checkpoint v, will not be counted at v (because $u = p(v) \notin s(u)$ in phase 3). That is, there is no need for the labeling process along the opposite inbound direction. Otherwise, v will be initiated earlier than u. The labeling process at u (in phase 2) will end the counting at v. This is correct since both u and v are active and there is no uncounted vehicle traveling between them. From the view of both u and v, Alg. 1 does not need to change. Thus, the statement can be proven. ■

Extension for odd traffic pattern. In the above road system, we assume that each counting initiated for an inbound traffic $u \leftarrow v$ will always have a chance to encounter another vehicle carrying a label from v so that the result can stabilize. When our counting is applied to a real road system without such an assumption, we may have a deadlock problem as addressed in [9]. If all vehicles deliberately detour around and avoid entering a road segment in the active status of counting, this counting cannot converge. The corresponding directional road segment is called the "orphan." Such a starvation status also suspends the convergence of the adjacent checkpoints along the spanning tree that is built with the predecessor-and-successor relation, forming a waiting chain. We resort to police patrol because the trajectory of vehicles is unpredictable and cannot be used to build a reliable deadlock-free solution. In our approach, every patrol car can store the on/off statuses of inbound traffic counters of each checkpoint, and can have the same ability as the counted vehicles to share information with a checkpoint via wireless communication.

Theorem 3 (guaranty of the convergence). *The counting with Alg. 3 will converge and not have the deadlock problem when any two adjacent checkpoints can be reached by a patrol car in a finite delay $\tau < \infty$.*

Proof: Without loss of generality, we have two adjacent checkpoints u and v ($v \in s(u)$). After v is visited, u will eventually encounter such a patrol car and fetch the counting status of v. According to the protocol in phase 4 of Alg. 1, the counting of $u \leftarrow v$ stops. Thus, the statement is proven. ■

Based on the above theorem, we design a cycle to cover every checkpoint at least once. The following theorem ensures the existence of such a cycle in our closed road system. In the ideal case, a Hamilton cycle can be used. Otherwise, some checkpoints can be visited multiple times. Every police car will evenly be distributed and drive along such a cycle. The patrol car will not be counted by any checkpoint, but will help to deliver the "stop" signal to adjacent checkpoints. As a result, each checkpoint will eventually stop its counting.

Theorem 4 (success ensured). *In a closed road system with directional road segments where each intersection can be visited by vehicle(s) at least once, we can always find a patrol cycle, but not necessarily a Hamiltonian cycle.*

Proof: A closed road system is a connected network where all vehicle trajectories can be connected as well. Since each intersection can be visited at least once, the patrol path can reach each intersection. Since the closed system does not have any dead end and the entire region is limited, the patrol path can always move forward and eventually reach an intersection that was previously visited. Thus, the statement is proven. ■

Algorithm 4 Information collection for the global view where Alg. 3 is applied.

```
Require: Counting at checkpoint $u$, stable with Alg. 3.
Ensure: Global view of vehicle counter.
Apply phase 1 in Alg. 2.
For each one-way traffic $u \rightarrow v$ in the above phase, $v \in s(u) \wedge v \notin n^i(u)$, wait for $c(v)$ reported from the patrol vehicle coming along a circuitous route from $v$.
Apply phase 2 in Alg. 2.
For each one-way traffic $u \leftarrow p(u)$ in the above phase, $p(u) \notin n^o(u)$, ask every patrol vehicle passing through $u$ to carry the result $c(u) + \sum_{v \in s(u)} c(v)$ to $p(u)$.
```

In Alg. 2, the stabilized counting result will be collected along the traffic back to the predecessor. A one-way road segment will force such a process to take a circuitous route, which adds distance and time. To ensure the success of information collection, we also rely to the police patrol. The chance of using one-way road segment in the predecessor-and-successor relation is so little. Moreover, in the real world, many one-way streets have been upgraded to bi-directional [10]. So, our counting is practical to implement. The revised information collection can be seen in Alg. 4.

Does anyone see that white van? – Extension for counting a specified type. When the surveillance at each intersection can identify certain kinds of vehicles from their exterior characteristics, our counting can provide a complete search for many applications. For instance, the Beltway sniper-attacks took place over the course of three weeks in October 2002 in Washington, D.C., Maryland, and Virginia [11]. Early tips from eyewitnesses included reports of a white box truck with dark lettering. Police across the area and the state of Maryland were pulling over white vans and trucks. Later, evidence shows that had this vehicle search for a "white van" resulted in the shooter being caught, more lives would have been saved.

We do not need to worry about information privacy. No ownership information is used. Vehicle registration information provided by manufacturers in the standard can be used to identify the uncounted vehicle in the type that we are counting.

Extension with multiple seeds. In the above counting, the entire region is equipped with one sink only. When multiple sinks are equipped, a spanning tree of predecessor-and-successor relation will spread out from each sink. When all trees use the same label, a forest will form, making the tree-depth relatively shorter. This can quicken our counting process.

Extension to the open road system. Then, we extend our counting from a closed road system to an open system where there are vehicles in and out along the border continuously. A complete counting in the open system can be defined in Def. 1, as follows.

Definition 1: A complete counting in an open road system is a scheme that can count every vehicle inside the area enclosed by the border, and everyone that travels in and out along the border in any possibility. Such a global status reached in the counting is called a "complete status."

Definition 2: A border of an open road system consists of all intersections that have both traffic flows in and out of the

Algorithm 5 Constitution of local views in the open system.

```
Require: The same in Alg. 1.
Ensure: a complete status in the global view level.
for each checkpoint $u$ deployed do
    Apply Alg. 3.
    In phase 3 of Alg. 3, the counting is activated for not only inbound traffic but also outbound interaction (i.e., $-1$ in counting phase 5).
    In phase 6 of Alg. 3, the result stabilizes after the counting activated for each non-interaction (inbound) traffic ended.
```

system, either inbound or outbound. Specifically, the traffic that starts/ends at an intersection along the border and is connecting the outside of the system is called "interaction."

Our strategy is to activate the counting of interaction along the border (i.e., the surveillance of in-out traffic) while accomplishing the counting inside the area that is enclosed by the border line. As a result, our counting can quickly reach a "complete status," as we proved in Corollary 1. Simply, we apply Alg. 3 to non-interaction traffic. We will initiate each local counting for interaction, either inbound or outbound, as well as other non-interaction inbound traffic at the same checkpoint. Note that those counting processes on the interaction will remain active for any possible vehicle coming in or out at any time. The details can be seen in Alg. 5.

Corollary 1 (convergence in open system). *Initiated from seed(s), Alg. 5 applied in the open road system can eventually reach the complete status.*

Proof: The counting with Alg. 5 will propagate to every checkpoint, as would the one with Alg. 3. The counting along the border for interaction will remain active for any possible vehicle that suddenly comes in or leaves. Based on Theorem 3, other counting processes will stabilize due to the update in phase 6. After that, based on Theorem 2, any vehicle in a non-interaction traffic will eventually encounter the frontier wave and be counted before the counting result stabilizes. Any vehicle newly coming or leaving the system can be observed by the active counting for interaction along the border. Thus, the statement is proven. ■

Corollary 2 (correctness in asynchronous global view). *During the convergence of Alg. 5, there is no mis- or double-counting at any active checkpoint.*

Proof: Our counting is initiated from a limited number of seeds. Before the counting process reaches the final "complete status," a vehicle can escape from any inactive checkpoint along the border. If this vehicle has been counted, it should surpass a labeled vehicle and has updated the information (line 8 in Alg. 3). For any vehicle coming into the counting area via such an inactive checkpoint, it will eventually encounter the frontier wave of our counting and have the information up to date. The rest of the proof is referred to Theorem 2. ■

V. Simulation

We adopt the city map of Manhattan in New York (from OpenStreetMap [12]), which contains many one-way streets. The test focuses on the traffic from Central Park to Madison Square Park in midtown. With different traffic volumes changing from 10% to 100% of the average, the trace data of each vehicle is generated by the simulation SUMO [13]. Multiple lanes and overtakes are considered. The instantaneous velocity is limited to 15 mph. We simulate the surveillance at each intersection (or roundabout) and lossy wireless communication, with a 30% chance of failure. We also test with different numbers of seeds, ranging from 1 to 10, which is randomly selected from the available checkpoints in the system.

We first close the traffic lanes along the border and build a closed system. The checkpoint will repeatedly apply Alg. 3 and Alg. 4, playing its deterministic role locally under our real road model of New York City. Initiated from the seeds, our counting is expected to converge with the stable local views. These results will be collected by the seeds as the data sinks. We verified the correctness of our counting without any mis- or double-counting. We study the performance of information constitution and collection, in terms of the elapsed time.

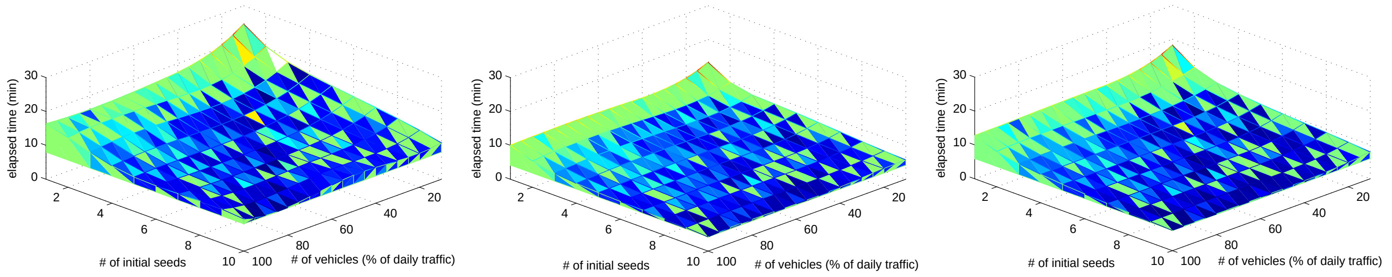
Fig. 2 shows the time needed for each checkpoint to constitute a stable result with Alg. 3. Similarly, Fig 3 shows the time of the whole counting until the data sinks fetch the information with Alg. 4 and form the global view. Fig 4 (c) shows the average time for Alg. 3 to converge when the vehicle velocity limit is lifted from 15 mph to 25 mph [14], a 66.67% speed enhancement. Relatively speaking, the size of the entire region shrinks by 64% with more dense deployment of checkpoints. Compared with the results in Fig. 2 (c), a performance enhancement up to 58% can be seen. Respectively, Fig 5 (c) shows the results until the global view forms at the seed(s) (with both Alg. 3 and Alg. 4) with the same vehicle speedup. Compared with the results in Fig. 3 (c), an enhancement up to 57% can be seen.

Next, by allowing in-out traffic of the interaction, the above closed system will be transformed to an open system. Each checkpoint will apply Alg. 5. Initiated from the same seeds (in the closed system), the counting will reach the "complete status" at the global view level. To fetch such a "complete status" and to verify the correctness of our approach, each checkpoint will apply Alg. 4 to those non-interaction traffic (enclosed by interaction as a closed system) after its local counting stabilizes in Alg. 5.

In Fig. 4 (a), the average time for Alg. 5 to converge is demonstrated. Fig. 4 (b) shows the performance of the same procedure after the same speedup in Fig. 4 (c). Compared with the results in Fig. 4 (a), a 30 $\sim$ 40% enhancement is achieved. Respectively, Fig. 5 (a) and Fig. 5 (b), one with the limit of 15 mph and the other with the limit of 25 mph, demonstrate the total time needed after both Alg. 5 and Alg. 4 converge.

We summarize our observations as follows.

1) The experimental results prove that our counting in both closed and open systems does not have any mis- or double-counting. We also verify the "complete status" after Alg. 5 stabilizes in the open system.
2) The speed of achieving the "complete status" in the open system mainly relies on the convergence speed of counting on non-interaction traffic, which is similar to counting in the closed system. Thus, the difference between Fig. 2 (c) and Fig. 4 (a) is limited.

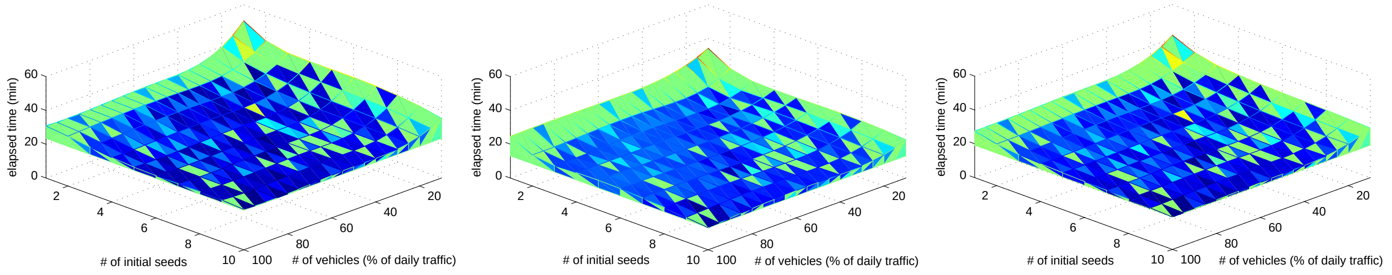


(a) maximum

(b) minimum

(c) average

Fig. 2. Elapsed time (9 ∼ 30 min) of Alg. 3 in the closed system of New York midtown.

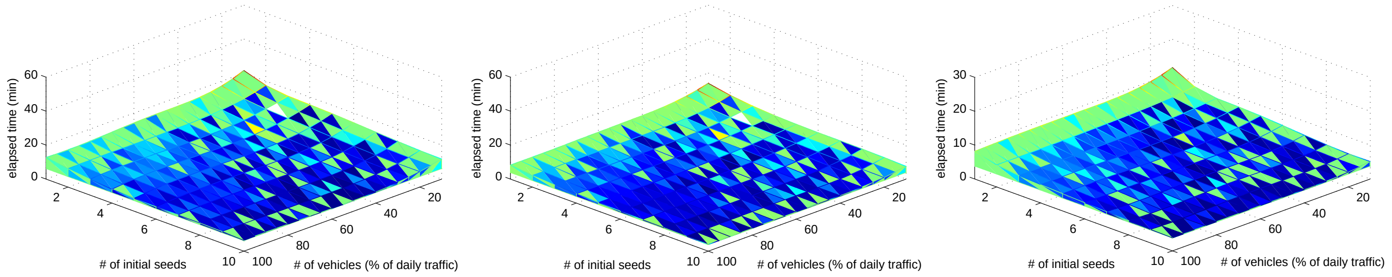


(a) maximum

(b) minimum

(c) average

Fig. 3. Time (20 ∼ 50 min) needed for the seed(s) to obtain the global view after both Alg. 3 and Alg. 4 converge in the closed system of New York midtown.

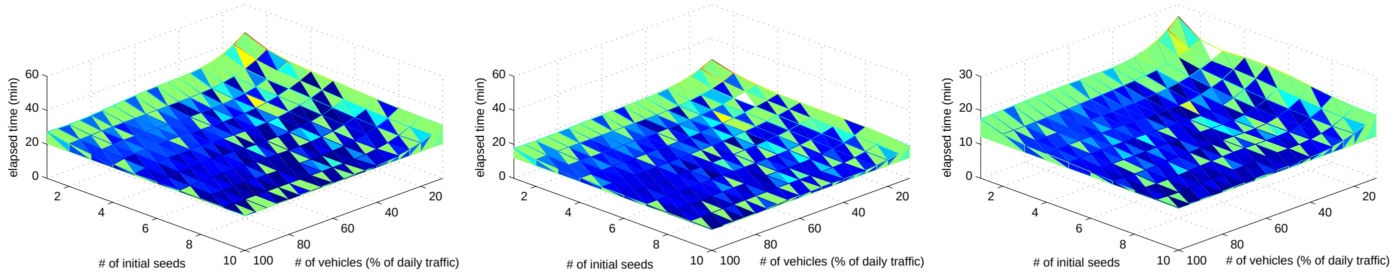


(a) time to reach a "complete status"

(b) time after speed limit lifted to 25 mph

(c) speedup in the closed system for comparison

Fig. 4. (a) Elapsed time (on average) of Alg. 5 in the open system of New York midtown, (b) performance of Alg. 5 (34 ∼ 40% quicker vs. Fig. 4 (a)) after vehicle speed increased by 66%, and (c) performance of Alg. 3 (the version of our counting in the closed system) after the same speed enhancement (up to 58% quicker vs. Fig. 2 (c)).



(a) time for seed(s) to fetch a "complete status"

(b) time after speed limit lifted to 25 mph

(c) speedup in the closed system for comparison

Fig. 5. (a) Elapsed time (on average) for both Alg. 5 and Alg. 4 to converge in the open system of New York midtown, (b) performance (34 ∼ 40% quicker vs. Fig. 5 (a)) after vehicle speed increased by 66%, and (c) performance of Alg. 3 and Alg. 4 (the version of counting & collection in the closed system) after the same speed enhancement (up to 57% quicker vs. Fig. 3 (c)).

3) We noticed the impact of the traffic pattern. Once labeled in the closed system, a vehicle will keep its leading role in the frontier wave and deliver the information without any delay. In the open system, when such a vehicle joins the interaction traffic, the border intersection must wait for another vehicle to resume the information relay, causing a considerable delay when many vehicles are through traffic in New York City. Thus, the performance shown in Fig. 4 (a) and Fig. 5 (a) are lower than those in Fig. 2 (c) and Fig. 3 (c), respectively. But such a difference is in a very limited range.
4) The speed of our counting mainly relies on how quickly the spanning tree with the predecessor-and-successor relation can be built in information constitution as well as in information collection. The average vehicle velocity (or the relative size of counting region) is key. A proportional enhancement can be seen in Fig. 4 (b) and Fig. 5 (b) from the open system, and Fig. 4 (c) and Fig. 5 (c) from the closed system. Those results also demonstrate the scalability of our approach as a solution for the large-scale problem.
5) The time needed in our information constitution (in Fig. 2 and Fig. 4) and information collection (in Fig. 3 and Fig. 5) are in proportion to how quickly a vehicle travels along the diameter of our target region. The counting spreads as a wave. The impact of other factors such as overtake, traffic block, one-way street, lane change, and lossy communication can be mitigated. Due to the high traffic volume, we have enough vehicle appearance to end the local counting and do not need police patrol.
6) The speedup of our counting by increasing the number of seeds is not significant, until the spanning trees initiated by each seed can evenly cover the entire target region. This raises a concern of cost in the collection of the "complete status" in the open system, when each checkpoint along the border is deployed as a global data sink. The delay needed to collect the global snapshot from the border to the seed is still considerable, while such a costly deployment itself cannot significantly speed up our counting process. Our results suggest the cost-effective deployment with only one single sink.

VI. Conclusion

In this paper, an infrastructure-less scheme is provided for precisely counting all vehicles in a target region without any mis- or double-counting. The scheme is applicable for both closed and open road systems. The unique directive is to provide a synchronization without any disruption on vehicle trajectory or support of global infrastructure in order to mitigate the impact of inconsistent local views among different checkpoints. Our approach is implemented under an "everyone" model in a fully-distributed manner, in order to achieve the reliability and scalability. With the counting result, we can have better strategies in resource management. For instance, our result can be used to determine an appropriate level in disaster evacuation, or to evaluate the market need for a new car service. In our future work, we will conduct further studies on the impact of traffic patterns, so that even better solutions can be achieved.

Acknowledgment

This research work was supported in part by NSF grants ECCS 1231461, ECCS 1128209, CNS 1138963, CNS 1065444, and CCF 1028167.

References

[1] K. Chandy and L. Lamport, "Distributed snapshorts: Determining global states of distributed systems," *ACM Transactions on Computer Systems*, vol. 3, no. 1, pp. 63–75, 1985.

[2] S. Kamijo, H. Koo, X. L. K. Fujihira, and M. Sakauchi, "Development and evaluation of real-time video surveillance system on highway based on semantic hierarchy and decision surface," in *Proc. of IEEE International Conference on System, Man and Cybernetics*, 2005.

[3] R. Reulke, S. Bauer, T. Doring, and F. Meysel, "Traffic surveillance using multi-camera detection and multi-target tracking," *Proc. of Image and Vision Computing New Zealand*, pp. 175–180, 2007.

[4] Y. Xu, Y. Wu, J. Xu, D. Ni, and G. Wu, "A queue-length-based detectoin scheme for urban traffic congestion by VANETs," in *Proc. of IEEE NAS*, 2012.

[5] Y. Zhang, J. Zhao, and G. Cao, "Service scheduling of vehicle-roadside data access," *Mobile Network Applications*, vol. 15, no. 1, pp. 83–96, 2010.

[6] C. Huang, H. Krishnan, R. Sengupta, and Y. Fallah, "Implementation and evaluation of scalable vehicle-to-vehicle transmission control protocol," *Proc. of the IEEE Vehicular Networkging Conference*, 2010.

[7] T. Little, A. Agarwal, J. Chau, M. Figueroa, A. Ganick, J. Lobo, T. Rich, and P. Schimitsch, "Directional communication system for short-range vehicular communications," *Proc. of IEEE Vehicular Networking Conference*, 2010.

[8] X. Yang, J. Liu, F. Zhao, and N. Vaidya, "A vehicle-to-vehicle communication protocol for cooperative collision warning," *MOBIQUITOUS*, 2004.

[9] T. Lai and T. Yang, "On distributed snapshots," *Information Processing Letters*, vol. 25, pp. 153–158, 1987.

[10] A. Ehrenhalt, "The return of the two-way street," 2009, information available at http://www.governing.com/column/return-two-way-street.

[11] Wikipedia, "Beltway sniper attacks," information available at http://en.wikipedia.org/wiki/Beltway_sniper_attacks.

[12] M. Haklay and P. Weber, "OpenStreetMap: User-generated street maps," in *Proc. of IEEE Pervasive Computing*, 2008.

[13] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO−simulation of urban mobility: An overview in: SIMUL 2011," *The 3rd International Conference on Advances in System Simulation*, 2011, information available at http://sumo.sourceforge.net.

[14] E. Durkin, "City council bill aims to lower speed limit in much of city to 25 mph," *Daily News City Hall Bureau*, 2013, document available at http://www.nydailynews.com/news/politics/council-bill-aims-speed-limit-20-mph-article-1.1530483.