Toward Mitigating Phantom Jam Using Vehicle-to-Vehicle Communication

Myounggyu Won, Member, IEEE, Taejoon Park, Member, IEEE, and Sang H. Son, Fellow, IEEE

Abstract—Traffic jams often occur without any obvious reasons such as traffic accidents, roadwork, or closed lanes. Under moderate to high traffic density, minor perturbations to traffic flow (e.g., a strong braking motion) are easily amplified into a wave of stop-and-go traffic. This is known as a phantom jam. In this paper, we aim to mitigate phantom jams leveraging the three-phase traffic theory and vehicle-to-vehicle (V2V) communication. More specifically, an efficient phantom jam control protocol is proposed in which a fuzzy inference system is integrated with a V2V-based phantom jam detection algorithm to effectively capture the dynamics of traffic jams. Per-lane speed difference under traffic congestion is taken into account in the protocol design, so that a phantom jam is controlled separately for each lane, improving the performance of the proposed protocol. We implemented the protocol in the Jist/SWAN traffic simulator. Simulations with artificially generated traffic data and real-world traffic data collected from vehicle loop detectors on Interstate 880, California, USA, demonstrate that our approach has by up to 9% and 4.9% smaller average travel times (at penetration rates of 10%) compared with a state-of-the-art approach, respectively.

Index Terms—Intelligent transportation systems, phantom jams, three-phase traffic theory, traffic jams, vehicle-to-vehicle (V2V) communication.

I. INTRODUCTION

Traffic jams are a serious social problem in many countries. In the United States, more than 5.5 billion hours have been wasted on highways in 2011, which are equivalent to 2.9 billion gallons of fuel that cost more than $121 billion [1]. The vehicle numbers of European countries have sharply increased while the highway capacity has not been commensurate with the rapid growth pace [2]. Traffic jams occur everyday, and many times without apparent reasons such as traffic accidents, roadworks, or closed lanes. Specifically, when traffic density is high enough, even a small perturbation (e.g., a strong braking motion) to traffic flow may result in a traffic jam [3]. This kind of a traffic jam is known as a phantom jam [4]. Since Treiterer et al. [5] demonstrated the existence of phantom jams through aerial photographs, its emergence has been experimentally confirmed [3] and theoretically modeled [6].

Numerous approaches based on Intelligent Transportation Systems (ITS) have been proposed to mitigate the impact of phantom jams [7]–[9]. Recently, Forster et al. designed a cooperative advanced driver assistance system (CADAS) that recommends velocity reduction to redistribute the traffic such that the propagation of phantom jams is prevented [10]. Knorr et al. designed a traffic jam control protocol that effectively alleviates phantom jams with marginal penetration rates [11], [12]. Compared with Foster et al.’s work, a key idea of their work is to perform traffic flow assessment using vehicle-to-vehicle (V2V) communication to determine whether a recommendation of vehicle speed (or a gap to the preceding vehicle) adjustment needs to be provided. The average speed of preceding vehicles, computed based upon data received from other vehicles using V2V communication, are used for traffic flow assessment. More specifically, depending on the average speed, drivers are provided with a “driving advisory” that allows them to keep a certain headway distance to the preceding vehicle in order to reduce the impact of over-deacceleration of a driver that may lead to formation of phantom jams. This kind of driving advisory can be provided in various ways, e.g., for autonomous vehicles, it can be automatic speed adjustment, and for human-driven vehicles, it can be provided with a display device, voice message, or vibration to alert the driver not to drive too close to the preceding vehicle.

In this article, we address three significant issues with regard to mitigating phantom jams. First, we note that a traffic jam is a dynamic phenomenon that continually changes its degree (measured as the average travel time in Sections V and VI) depending on dynamic factors such as random driving behavior and road conditions. In other words, a traffic jam is not a binary event that either occurs or not. As such, determining a traffic jam based on a binary decision making process, e.g., depending on whether the average speed of preceding vehicles is smaller than a given threshold, potentially leads to inefficient traffic jam control. In the proposed phantom jam control protocol, we integrate a fuzzy inference system (FIS) into a traffic-jam-control algorithm to effectively manage the dynamics of a traffic jam and to provide more effective driving advisory to drivers.
Next, we note that there exists speed variance between lanes especially under congestion [13]. Empirical measurements performed on a highway in Japan [14], [15] showed that strong correlation between lanes exist, and the fast lane is often more severely congested than the slow lane, which is referred to as reverse-lane usage. In fact, recent research has shown that a critical factor that causes safety problems on highways is the speed variance between lanes rather than the high vehicle speed [16]. Speed difference between lanes is fully accounted for in the design of the proposed phantom-jam control protocol in which traffic flow assessment based on the fuzzy inference system is separately performed for each lane providing effective driving advisory to drivers.

Finally, real-world traffic data obtained from loop detectors on Interstate 880, CA, United States, was incorporated into our simulation to evaluate the performance of the proposed protocol in a realistic simulation environment. More specifically, according to the real-world data, a highway segment was generated; vehicle traffic was generated and injected at certain locations; and the proposed protocol was applied to evaluate the performance of the protocol. It is demonstrated that the proposed protocol reduces the average travel time by up to 4.9% (at penetration rates of 10%) in comparison with a state-of-the-art protocol [11], [12] for the real traffic data, and by up to 9% for simulated environments. The contributions of this article are summarized as follows:

- A fuzzy inference system is designed and integrated into a phantom jam control mechanism to effectively capture the dynamics of a traffic jam and to provide effective driving advisory.
- A “lane-level” traffic analysis is performed and integrated with the proposed protocol design to improve its performance.
- Real-world traffic data was obtained and applied to the proposed protocol to verify the effectiveness of the proposed protocol.
- Real-world issues including the impacts of driver compliance rates and wireless communication channel quality are studied.

The rest of the paper is organized as follows. In Section II, we review previous approaches designed to reduce phantom jams. We then present a review on the three-phase traffic theory, connected vehicle (CV) technology, and fuzzy logic in Section III. Section IV describes the details of the proposed protocol. Simulations are performed and the results are presented in Section V. We then conduct experiments with real-world traffic data in Section VI. Finally the paper concludes with Section VII.

II. RELATED WORK

There is an ample body of research to develop traffic congestion control mechanisms that deal with traffic dynamics. An example is adaptive traffic signal control mechanisms [17]–[19]. A multi-objective controller was designed to optimize multiple objectives such as trip waiting time, total trip time, and junction waiting time [20], [21]. To represent the road dynamics, a stochastic/statistical method was developed [22], [23]. However, in contrast to regular traffic congestion, traffic jams often appear without apparent reasons. When traffic density is high, minor disruptions to traffic flow may cause a chain reaction of stop-and-go traffic. These traffic jams are known as phantom jams [5], Sugiyama et al. experimentally demonstrated the emergence of phantom jams on a circuit [3]. Some studies showed that phantom traffic jams can be avoided by suppressing such disruptions by controlling vehicle speed or distance to the preceding vehicle [24]. For example, Adaptive Cruise Control (ACC) was used to maintain a constant gap to the preceding vehicle, mitigating phantom jams [25]. Kesting et al. demonstrated that, with more than 25% of vehicles equipped with ACC on a highway, traffic jams can be effectively avoided [26]. Cooperative adaptive cruise control (CACC) integrated V2V communication into ACC to provide more information about surrounding vehicles, thereby enabling more efficient control of the vehicle headway distance [27], [28]. Forster et al. proposed a Cooperative Advanced Driver Assistance System (CADAS) that took into account the density gradient between two communicating vehicles to recommend the optimal vehicle velocity in order to eliminate upstream shockwave formation [10], [29]. A vehicle control framework was developed to adjust acceleration of a vehicle and timing to move to the next lane by predicting the dynamic behavior of surrounding vehicles using V2V communication [30]. The framework was improved by integrating the optimal velocity model (OVM) [31] and by considering the behavior of the following vehicle [32].

An interesting characteristic of many previous approaches is that traffic information, used to control vehicle speed (or distance to the preceding vehicle), is collected from a limited number of surrounding vehicles. For example, CADAS determines the emergence of a traffic jam (the “slowdown event”) based on the speed of a vehicle (i.e., the speed being smaller than a threshold), failing to account for the overall traffic situation [10]. The model predictive control (MPC)-based approaches also considered only a few vehicles in the proximity for controlling the speed of a vehicle [30].

As connected vehicle (CV) technology enables vehicles to collect traffic information from any vehicles on a highway, new approaches have been introduced [11], [12], [33], [34]. Kerner et al. [33] proposed an algorithm that advises a larger space gap to the preceding vehicle when a traffic breakdown is detected to reduce traffic jams. In particular, one notable contribution is their test-bed for wireless V2V communications, which enables the evaluation of their proposed algorithm for various characteristics of wireless communications between vehicles. Fekete et al. [34] proposed a distributed and self-regulated protocol to reduce traffic jams which provides adaptable distributed strategies against traffic jams based on local data clouds formed among vehicles. Their proposed solution guarantees good performance with relatively large penetration rates of more than 60%. Knorr and Schreckenberg [11] designed an approach that effectively stabilizes traffic flow and reduces a traffic jam with penetration rates of as low as only 5%. In their recent work [12], they also factored in human limitations such as reaction time for adapting driving behavior.

We have developed a phantom-jam control protocol that effectively accounts for traffic dynamics and speed variance between
In this article, we focus on this metastable state disturbance to traffic flow with medium-to-high traffic density, demonstrated that this transition occurs due to an internal mechanism in the vicinity of bottlenecks [42]. In particular, it was empirically shown that sufficient speed variance exists between lanes, the proposed protocol offered “lane-level” traffic jam control, i.e., the control algorithm was separately applied to each lane, improving the efficiency of the control algorithm. This article improves our previous work in the following aspects: incorporation of real-world traffic data, analysis of driver compliance rates on the performance of the proposed protocol, and analysis of the impact of the V2V wireless communication channel quality.

III. BACKGROUND

A. Three-Phase Traffic Theory

According to the three-phase traffic theory, traffic flow on a highway is characterized by space-time transitions between three phases: free flow (F), synchronized flow (S), and moving jam (J) [37]-[40] (we, however, acknowledge that there are different interpretations of traffic jams, e.g., suggesting more than three phases to reflect the complexity of traffic flow [41]).

In the F phase, traffic density is low, allowing vehicles to accelerate/decelerate arbitrarily as illustrated in Fig. 1. The traffic flow becomes unstable as traffic density increases. With sufficiently high traffic density, free traffic flow can be easily transitioned to congested traffic, starting the S phase. This transition is referred to as breakdown phenomenon [37], [42]. Kerner remarks that this transition (F → S) occurs mostly in the vicinity of bottlenecks [42]. In particular, it was empirically demonstrated that this transition occurs due to an ‘internal disturbance’ to traffic flow with medium-to-high traffic density, or congested traffic at a different road location downstream [43]. In this article, we focus on this metastable state with medium-to-high traffic density between the F and S phases (Fig. 1). In this state, a traffic jam may arise even by a minor perturbation, which is known as a phantom jam [12].

Within the region of the synchronized flow, self-compression (also called as the pinch region), i.e., significantly high local traffic density, is observed. In the J phase, this pinch region starts to propagate upstream. Kerner showed that the propagation of the pinch region can be reduced by imposing strong vehicle speed adaptation [44]. In this study, we design a V2V-based approach to reduce the chance of traffic perturbation by providing driving advisory to drivers to allow them to maintain a certain headway distance to their preceding vehicles [45].

B. Fuzzy Logic

In this section, we briefly review fuzzy logic to help readers better understand our contributions. For more details on fuzzy logic, please refer to [46].

1) Fuzzy Set: A crisp set defines membership and non-membership of an element, in the form of a membership function \( \mu_A(x) \), i.e., given a crisp set \( A \), \( \mu_A(x) = 1 \) if \( x \in A \) and \( \mu_A(x) = 0 \) if \( x \notin A \). In contrast, a fuzzy set extends a crisp set by introducing partial membership, i.e., the membership function \( \mu_A(x) \) of a fuzzy set \( A \) takes values in the interval \([0,1]\); that is, a fuzzy set can be written as a set of ordered pairs: \( A = \{ (x, \mu_A(x)) | x \in U \} \), where \( U \) is a universe of discourse.

2) Fuzzy Inference System: A fuzzy inference system (FIS) is basically a mapping of a vector of input data into a scalar output. Fig. 2 depicts a diagram of a general FIS. The FIS is composed of fuzzifier, rule base, aggregator, and defuzzifier. The fuzzifier maps a vector input of scalar values into a fuzzy set. The rule base has linguistic rules called fuzzy rules that are often defined by domain experts. More specifically, let \( T(x) = \{ T_1, T_2, \ldots, T_n \} \) be the term set for an input vector \( x = \{ x_1, x_2, \ldots, x_n \} \) (e.g., if \( x_1 \) is a temperature, \( T(x_1) \) is “cold”, “hot”). A fuzzy rule is in the form of a simple if-then rule (e.g., “if \( x \) is \( T \), then \( y \) is \( T' \)”). More formally, denote fuzzy rules by \( R = (R_1, R_2, \ldots, R_n) \). The \( i \)-th fuzzy rule can then be written as:

\[
R_i = \text{if} \quad (x_1 \text{ is } T_{x_1}, \ldots, x_p \text{ is } T_{x_p}) \quad \text{then} \quad (y_1 \text{ is } T_{y_1}, \ldots, y_q \text{ is } T_{y_q}).
\]

Each rule yields a single number which is the firing strength of the rule. When multiple antecedents are present in a rule, fuzzy operation (e.g., min or product) is applied to find a single firing strength. Now given a set of output fuzzy sets with corresponding firing strength, the aggregator combines the output fuzzy sets into a single set. The defuzzifier then translates the output fuzzy sets into a crisp number, which is the final output of FIS.
traffic breakdown would likely to occur [12]; more specifically, it is defined that the critical segment is an area where messages, vehicle $i$ at time $t$ as shown in Fig. 3. Upon receiving the beacon messages, vehicle $i$ calculates the average speed $\overline{v}(t)$ of all vehicles ahead. If $\overline{v}(t)$ is smaller than a predefined threshold $T_v$, more than two consecutive beacon intervals $I$ (i.e., $\overline{v}(t-I) < T_v$ and $\overline{v}(t) < T_v$), the vehicle $i$ is said to be in the “critical road segment”. It is defined that the critical segment is an area where traffic breakdown would likely to occur [12]; more specifically, the location of the upstream boundary of the critical segment, $c_a$ is marked as $x_p(t) + \gamma$, where $\gamma$ is the communication range of a vehicle; and the time when the critical segment is created, denoted by $c_t$, is marked as the current time $t$ (Fig. 3). The information on the critical segment, i.e., $c_t$ and $c_a$, is included in the next scheduled beacon message and broadcasted.

Once a critical segment is identified, vehicles start to adapt their speed if the two conditions are met: first, the critical segment was recently created, and second, the vehicles are geographically close enough to the critical segment. Formally, if the following two conditions are satisfied:

$$t - c_t < T_h \text{ and } 0 < c_a - x(t) < T_s,$$

where $T_h$ and $T_i$ are the spatial and temporal thresholds, vehicles are provided with driving advisory to maintain a certain headway distance to their preceding vehicles. Note that these thresholds were experimentally selected in [12] (i.e., $T_s = 3$ km and $T_i = 3$ s), but we do not rely on this threshold-based binary decision making process to provide more effective driving advisory (Section IV).

Knorr et al. [12] adopted the Comfortable Driving Model (CDM) [47], which is based on the Nagel-Schreckenberg’s cellular automation [48], to model advised driving behavior for their simulations. In this model, a probability $p$ is introduced to determine the probability of braking, which is defined as follows.

$$p \left\{ \begin{array}{ll}
    p_b & \text{if } b_{n+1} = \text{true AND } t_h < t_s \\
    p_0 & \text{if } v_n = 0 \text{ AND NOT } (b_{n+1} = \text{true AND } t_h < t_s) \\
    p_d & \text{otherwise,}
\end{array} \right.$$
More specifically, we fuzzify the parameters used to determine a traffic jam, i.e., $v(t)$, $t - c_1$, and $c_1 - x(t)$ into membership functions denoted by $\mu_c$, $\mu_1$, and $\mu_{s_t}$ respectively; instead of comparing the parameter values with predefined thresholds and making a binary decision, we enable the proposed system to evaluate the dynamics of a traffic jam by inputting the fuzzified parameters into our fuzzy inference system, which outputs varying driving advisory depending on traffic dynamics.

A diagram that illustrates an overview of the proposed protocol is shown in Fig. 4. Traffic data is periodically collected separately for each lane; the data are plugged into the fuzzy inference system (FIS), which then provides an appropriate driving advisory. In the following subsections, we provide the details of our fuzzy inference system design.

**B. Fuzzy Inference System**

In this section, we detail the design of our fuzzy inference system that captures the dynamics of traffic jams and provides varying driving advisory. In particular, the details are presented using an example focusing on general highways. Note that this example-based design is intended to show the advantages of our fuzzy-logic-based approach and we acknowledge that the membership function design may vary depending on highway conditions.

The design of a membership function for the average speed, $\mu_v$, is described. We classify the average speed as the boundary value between FAST and SLOW. For the new parameters can be communicated with vehicles using vehicle-to-infrastructure (V2I) communication to allow vehicles to dynamically update their membership functions as they drive on different highway segments.

Based on the speed limit report, we adopt the average speed limit of 120 km/h (75 mph) as the maximum possible speed for our membership function. Fig. 5(a) depicts the resulting membership function $\mu_v$. Note that the membership function for $\mu_v$ can be redesigned to account for variable speed limits that are widely used in U.S. highways. For example, different membership functions can be pre-computed depending on variable speed limits. An appropriate membership function is then selected in real time by using the current vehicle location and the corresponding speed limit of the vehicle location. It is also possible to use the percentage of speed limit as an indicator to classify the average speed of preceding vehicles.

To design a membership function for distance to a critical segment, denoted by $\mu_c$, we represent the distance using three linguistic terms {CLOSE, MEDIUM, FAR}. We use the threshold $T_c = 3$ km to differentiate the CLOSE and FAR distances [12]. In particular, a system parameter $T_c$ is used to define the MEDIUM distance. This system parameter can be adjusted for different highway conditions. The resulting membership function is depicted in Fig. 5(b).

The membership function for elapsed time after emergence of a critical segment, denoted by $\mu_s$, is designed similarly to $\mu_v$. The elapsed-time is described using three linguistic terms {CURRENT, MEDIUM, OLD}. A boundary value for differentiating CURRENT and OLD is defined based on the threshold $T_t = 30$ s [12]. A system parameter $T_t$ is added to the threshold $T_t$ to define the MEDIUM elapsed time. Fig. 5(c) displays the membership function.

These three membership functions constitute the fuzzifier of the proposed fuzzy inference system. Crisp input values for the average speed, the distance, and the elapsed time are input to the fuzzifier and are transformed into corresponding fuzzified values. Next, the rule base of the fuzzy inference system determines the level of traffic jam and the appropriate driving advisory based on the fuzzified values. The rule base is tabulated in Table I. In particular, the driving advisory is represented using terms {STRONG, MEDIUM, WEAK}. For example, if the fuzzifier provides fuzzified input values of SLOW average speed, CLOSE distance, and CURRENT elapsed time, STRONG driving advisory is recommended. The fuzzified output, e.g., STRONG driving advisory, in turn is translated into a crisp value based on a membership function for driving advisory shown in Fig. 5(d). As explained in Section III-C, the degree of driving advisory is controlled by a parameter $p_c$ for simulation purposes. More specifically, a smaller $p_c$ means a smaller probability of braking that can be interpreted as stronger prevention of the driver's abrupt braking motion.

A value of 0.8 for $p_c$ is used for differentiating STRONG and WEAK driving advisory according to [12]. It is worth noting that we adopt empirically selected parameters of previous work [12] for illustrative purpose because accurately calibrating the parameters require information about the characteristics of an individual highway segment, e.g., history of traffic patterns, relevant traffic regulations, length of highway segments, etc. The new parameters can be communicated with vehicles using vehicle-to-infrastructure (V2I) communication to allow vehicles to dynamically update their membership functions as they drive on different highway segments.
To determine the firing strength, we use the commonly-used MIN method, and for the aggregation method, we use the MAX method. There are various methods for defuzzification such as maximum-decomposition, center of maxima, and height of defuzzification. We refer readers to [46] for details on the defuzzification process. We use the centroid method because it finds the “balance” point of the solution while other methods are limited to a narrow spectrum of applications. In the centroid method, the defuzzifier uses the center of gravity $y'$ of $\mu_p$ as the output of the fuzzy inference system. Formally, in a continuous fuzzy set, the center of gravity is computed as:

$$y' = \frac{\int_y y \mu_p(y) \, dy}{\int \mu_p(y) \, dy}.$$

### C. Real-Time Update

Input values used to determine the degree of a traffic jam for each lane $l$, i.e., $v(t)^l$, $t - c_l$, and $c_l - x(t)$ must be updated as soon as a new beacon message is received in order to provide an up-to-date driving advisory. However, a challenge is that providing new driving advisory every time a beacon message is received incurs very high computational overhead. A possible solution is to update the driving advisory periodically, e.g., every 10 seconds. However, this periodic update-based method may miss an important event that requires new driving advisory; it may also result in unnecessary update, i.e., updating driving advisory even though none of $v(t)^l$, $t - c_l$, or $c_l - x(t)$ has been significantly changed.

To address this challenge, in our proposed protocol, a threshold-based update method is designed. More specifically, the moving averages of $v(t)^l$, $t - c_l$, and $c_l - x(t)$ are monitored and compared with previous critical values, i.e., the most recent input values used for determining the degree of a traffic jam. If the difference between one of the input values and the critical value is greater than a system parameter $\delta$, all input values are plugged into the fuzzy inference system, and new driving advisory is calculated. Fig. 6 illustrates the “update points” for the input value of the average speed, where driving advisory must be updated. In Section V, we evaluate the effect of the system parameter $\delta$ on the performance of our protocol (focusing on the input value $v(t)^i$).

### D. Lane Change

When a vehicle changes lanes from $l$ to $k$, driving advisory for the vehicle must also be immediately changed. To achieve this, we allow vehicles to maintain not only the information for the current lane but also the information for other lanes, i.e., keeping $v(t)^l$, $t - c_l$, and $c_l - x(t)$ up-to-date for all lanes $l$. Consequently, when a vehicle changes lanes from $l$ to $k$, the vehicle is allowed to input immediately $v(t)^k$, $t - c_k$, and $c_k - x(t)$ into the fuzzy inference system and provides new driving advisory to drivers.

### V. Simulation Results

The performance of the proposed protocol was evaluated in comparison with a state-of-the-art traffic jam control algorithm based on V2V [12]. We used a traffic simulator JiST/SWANS.
[54], [55] with extension by Ibrahim and Weigle [56] and Kilot [57]. More specifically, discrete events according to the cellular automaton model [48] and the lane change model [47] for freeway traffic were handled by the Java in Simulation Time (JiST), while the Scalable Wireless Ad Hoc Network Simulator (SWANS) was incorporated to simulate the networking aspects between vehicles (V2V).

### A. Simulation Setup

A two-lane highway segment was used which spans 18 km with a 300 m-long on-ramp located about 1.5 km away from the downstream boundary. The on-ramp was fed with vehicles at a rate of 450 vehicles/hour/lane, and the main road was fed with vehicles at varying rates as shown in Fig. 7 to artificially create a traffic jam. Simulation settings for highway traffic conditions and wireless V2V communication are summarized in Tables II and III, respectively. In particular, to enable V2V communication, Dedicated Short Range Communications (DSRC) [58] was used with a Nakagami-m [59] physical layer model. After running simulations for 6.5 hours with these simulation settings, a phantom jam was observed as illustrated in Fig. 8 (the color bar represents vehicle speed in km/h). A lane change model was also implemented according to [47], consequently generating dynamically changing speed difference between lanes as can be seen in Fig. 9. In these simulations, the average travel time was used as a performance metric, which is the average of travel times of all injected vehicles.

### TABLE II

**Simulation Setup: Traffic**

<table>
<thead>
<tr>
<th>Lane</th>
<th>2 lanes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>18 km (12,000 cells)</td>
</tr>
<tr>
<td>Vehicle speed (car)</td>
<td>34 m/s</td>
</tr>
<tr>
<td>Vehicle speed (truck)</td>
<td>26 m/s</td>
</tr>
<tr>
<td>Vehicle length (car)</td>
<td>4 m</td>
</tr>
<tr>
<td>Vehicle length (truck)</td>
<td>12 m</td>
</tr>
<tr>
<td>Portion of trucks</td>
<td>10%</td>
</tr>
<tr>
<td>Operation time</td>
<td>6.5 hours</td>
</tr>
</tbody>
</table>

### TABLE III

**Simulation Setup: Network**

<table>
<thead>
<tr>
<th>Pathloss</th>
<th>Nakagami-m</th>
</tr>
</thead>
<tbody>
<tr>
<td>TX power</td>
<td>17 dBm</td>
</tr>
<tr>
<td>Antenna gain</td>
<td>0 dBm</td>
</tr>
<tr>
<td>Sensitivity threshold</td>
<td>-90 dBm</td>
</tr>
<tr>
<td>Reception threshold</td>
<td>-81 dBm</td>
</tr>
<tr>
<td>MAC Protocol</td>
<td>DSRC</td>
</tr>
<tr>
<td>Beacon interval</td>
<td>240 ms + random(0,10) ms</td>
</tr>
<tr>
<td>Beacon size</td>
<td>500 Bytes</td>
</tr>
</tbody>
</table>

### B. Effect of Penetration Rates

We measured the average travel time by varying penetration rates and compared with a state-of-the-art phantom jam control protocol (i.e., denoted by Knorr et al. in all figures). Penetration rates define a percentage of vehicles that are equipped with DSRC communication modules capable of running the proposed protocol. The results are shown in Fig. 10. As penetration rates increased, the average travel times of both protocols decreased. These results indicate that traffic jams can be more effectively reduced if more vehicles are equipped with a V2V-based traffic jam control protocol. A notable observation was that even with small penetration rates of 5% to 10%, the average travel time was significantly reduced. Overall, travel time reduction of 17.19% on average was attained at penetration rates of 40%. According to leading U.K. transportation economists, the elasticity of traffic volumes with respect to travel time is 0.5 in the short run and 1.0 over the long run [60]. This means that the average travel time reduction of 17.19% can be translated into significantly increased traffic volumes, i.e., by up to 17.19% over the long run.
Another observation was that the standard deviations of both protocols decreased as more vehicles were equipped with the protocols. These results demonstrate that as the penetration rates increase, randomness stemming from the dynamics of a traffic jam is more effectively controlled. We also noted that our proposed protocol had the smaller average travel time compared with Knorr et al.: it effectively reduced the average travel time by up to 9% in comparison with the state-of-the-art protocol [11], [12].

C. Effect of $\delta$

We investigated the effect of the parameter $\delta$. The system parameter $\delta$ defines the threshold of the input variables for the fuzzy inference system [i.e., individually defined for $v(t)$, $(t - c_i)$, and $(c_i - z(t))$]. More specifically, the values of the input variables are provided to the fuzzy inference system only if they are greater than $\delta$, thereby reducing the overhead caused by frequent provision of driving advisory. In particular, in this simulation, we focus on the parameter $\delta$ defined for the average speed of preceding vehicle [i.e., $(t(t))^1$] to understand the effect of the parameter $\delta$. More precisely, we used penetration rates of 20% and varied the parameter $\delta$ defined for the average speed. The results depicted in Fig. 11 reveal that as the parameter $\delta$ increased, the average travel time increased. A significant increase of the standard deviation was also observed. These results indicate that with higher $\delta$ the dynamics of a traffic jam was not effectively captured, and driving advisory was not provided in a timely manner. On the other hand, with regard to computational resource savings in percentage, even with the small increase of $\delta$, significant savings were observed. An interesting observation was that the computational savings for larger $\delta$ (i.e., $> 0.5$) was marginal. This result can be interpreted that with $\delta > 0.5$, nearly all input values are already smaller than $\delta$ thus not running our fuzzy inference engine and making not much change in the savings. Overall, these results indicate that the parameter $\delta$ needs to be selected such that a good tradeoff between the average travel time and resource savings is achieved.

D. Effect of Lane Change

The lane change behavior of vehicles affect the performance of traffic jam control protocols. In this simulation, we investigated the effect of a lane change by varying the lane change ratio defined in a lane change model [47]. More specifically, we simulated aggressive lane change behavior by adjusting the safety gap (in cells) [47] for changing lanes, i.e., the larger the gap, the more aggressive lane change behavior is exhibited. Fig. 12 depicts the average travel time for different safety gaps. As shown, no significant difference between the average travel time was found. These results indicate that our approach of applying updated driving advisory immediately after a lane change has properly suppressed performance degradation caused by lane changes.

VI. EXPERIMENTS WITH REAL TRAFFIC DATA

In the previous section, we used artificially generated traffic data to create a traffic jam and to measure the average travel time. In this section, we adopted real traffic data collected on February 8th 2008 between 10:00 A.M. and 6:00 P.M. on Interstate 880, CA, United States [61]. For this experiment, the average travel time was used as the main performance metric.
Fig. 13. Interstate 880, California, USA.

Fig. 14. Travel time for Interstate 880, California, USA.

Fig. 15. Vehicle flow per vehicle density over time.

Fig. 16. Traffic flow and vehicle density over time.

Fig. 17. Travel time and vehicle density over time.

A. Experimental Setup

A highway segment (Interstate 880, CA) with the length of 18.12 km was considered as depicted in Fig. 13. Twenty-five detectors were used to collect vehicle flow information every 30 sec. In particular, we considered vehicle in-flow at three major intersections denoted by blue arrows in Fig. 13, as well as vehicle injection at the starting point, denoted by the green arrow. According to the data, a traffic jam occurred in the afternoon as shown in Fig. 14 that depicts the travel time as a function of time. The high travel time measured in the morning between 10:00 A.M. and 11:00 A.M. was due to a traffic accident, which we do not consider in this experiment; we focus on a spontaneous traffic jam that emerged in the afternoon, thereby accounting for the traffic between 1:00 P.M. and 4:00 P.M. It was observed that a relatively large volume of traffic was injected at the intersection with CA92, a red-shaded region. It is easy to note that the major cause of the traffic jam is the bottleneck due to the busy ramp. However, we note that what causes a traffic jam is, more specifically, a bottleneck in combination with a perturbation in the traffic flow [62]. Thus, what we focus on is to minimize the perturbation in traffic flow especially when the traffic density is in the meta-stable state, i.e., in the medium-to-high traffic density, thereby improving the overall travel time. In fact, this meta-stable state was observed in the fundamental diagram for this real-world data set (Fig. 15); more specifically, as the vehicle density increased, the traffic flow reached its capacity at around 3PM, after which the traffic flow decreased.

In this experiment, we demonstrated that the proposed protocol effectively reduces a traffic jam when applied to real traffic data. More specifically, we created a highway segment of the same length with that of Interstate 880 and injected vehicles at the four locations according to the real traffic data (Fig. 13) as input to our simulation model. We then applied our phantom-jam control mechanism to evaluate its performance with the real traffic data.

B. Average Travel Time

We observed the metastate of traffic flow in Fig. 15. Now we are interested to find what will happen if we apply our traffic jam control mechanism. Interestingly, the decrease in the traffic flow between 3:00 P.M. and 5:00 P.M. was significantly improved when our control mechanism was applied (Fig. 16). Given this observation, we quantify this improvement by measuring the average travel time. We repeated the measurements 20 times with penetration rates of 10%. We then compared the average travel time of our approach with Knorr et al.’s. The results are depicted in Fig. 17. When no traffic jam control algorithm was used, more than 90% of the measured average travel time were greater than 900 sec, while both traffic jam control protocols effectively reduced the traffic jam even with small penetration rates of 10%. More specifically, our protocol reduced the average travel time by 11.69% in comparison with the scenario where no traffic jam control algorithm was applied. This reduction of the average travel time is relatively smaller than the results presented in Section V-B. The reason is that in simulations we used idealized conditions while in this experiment real-world data was used. Compared with Knorr et al.’s solution, for this real traffic data, the average travel time was decreased by up to 4.9%.
C. Effect of Driver's Compliance Rates

One of the greatest concerns about deploying the proposed protocol is driver’s compliance rates to this technology. In this section, we investigate the effect of driver compliance rates on the performance of the proposed protocol using real traffic data. Peeta and Ramos analyzed and developed models of driver behavior to traffic information provision, i.e., they studied the decision making process of drivers under real-time information provision [63]. According to their work and many other studies, driver compliance rates vary depending on various factors such as gender, familiarity with the road, weather, day/night-time driving, and so on. In this experiment, we defined driver compliance rates as a probability $p_{\text{comp}}$, varied this probability, and measured the average travel time. Whether or not a driver follows the driving advisory was randomly and uniformly selected based on this probability. We fixed penetration rates to 10%. Fig. 18 shows the results. As shown, when we compared the results for the compliance rates of 80% with that of 100%, relatively small performance degradation (i.e., a decrease of 2.62%) was observed. However, when the compliance rates were decreased to 30%, distinctive performance degradation was observed. Another observation was that even with small compliance rates, our protocol achieved a noticeable improvement in terms of the average travel time compared with a situation where no traffic jam control algorithm was used.

D. Effect of Communication Failure

The vehicle-to-vehicle communication delay affects the performance of the proposed protocol because delayed DSRC messages contain outdated traffic information that potentially leads to inaccurate driving advisory. More specifically, delayed messages result in distorted input values to the fuzzy inference system that outputs driving advisory that does not precisely represents the current traffic situation. A more severe problem occurs, however, when messages are lost. Considering the typical DSRC message interval of 100 ms, message delay of more than 100 ms including the buffering and message propagation delay is incurred for each lost message. This huge delay that significantly affects the system performance is the main concern of this section. In this section, we attempt to analyze the effects of this communication failure.

Bai et al. analyzed the reliability of DSRC communication [64]. They demonstrated that depending on the vehicle-to-vehicle distance, packet loss rates varied between 0% ∼ 20%. Based on this result, we used the uniform probability packet loss model for incoming packets, and varied the packet loss probability from 0.0 to 0.2. More specifically, an incoming packet was dropped according to a randomly selected probability between 0% and 20%. The results are depicted in Fig. 19. When there was no traffic jam, i.e., around 12:00 P.M. and 12:30 P.M., the impact of packet loss was minimal, because our traffic jam control was not triggered yet. However, as the traffic started to get congested, the impact of packet loss rates became greater, increasing the gap between the two curves for packet loss rate of 0% and packet loss rate of 20%.

VII. CONCLUSION

In this article, we have presented a solution to mitigate phantom jams. A fuzzy inference system was designed and integrated into the traffic jam control to cope with the dynamics of traffic jams. A lane-level traffic jam control was designed to further
improve the performance. Practical issues (e.g., the effects of packet loss rates and driver’s compliance rates) related to applying this solution were studied using real-world traffic data. A possible future direction is to improve the adaptability of the proposed system by allowing the system to automatically rebuild its membership functions for FIS depending on the dynamics of highways. The complexities and uncertainties of these road dynamics can be effectively handled and represented using multi-objective optimization techniques [20], [21] or stochastic/statistical methods [22], [23], and can be communicated with vehicles via vehicle-to-infrastructure (V2I) communication.

ACKNOWLEDGMENT

The authors would like to thank the authors of [12] for generously providing their simulation code.

REFERENCES


Myounggyu Won (M’13) received the Ph.D. degree in computer science from Texas A&M University, College Station, TX, USA, in 2013. He has been a Postdoctoral Researcher with the Department of Information and Communication Engineering, Daegu Gyeongbuk Institute of Science and Technology, Daegu, South Korea. He is currently an Assistant Professor with the Department of Electrical and Computer Science, South Dakota State University, Brookings, SD, USA. His research interests include wireless mobile systems, vehicular ad hoc networks, intelligent transportation systems, and wireless sensor networks.

Dr. Won was a recipient of the Graduate Research Excellence Award from the Department of Computer Science and Engineering, Texas A&M University, in 2012.

Taejoon Park (S’04–M’05) received the B.S. degree (summa cum laude) in electrical engineering from Hongik University, Seoul, South Korea, in 1992; the M.S. degree in electrical engineering from Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 1994; and the Ph.D. degree in electrical engineering and computer science from the University of Michigan, Ann Arbor, MI, USA, in 2005. He is currently a Professor with the Department of Robotics Engineering, Hanyang University, Seoul. He is also the Head Director of the Research and Planning Division. Prior to joining Hanyang University, he was an Associate Professor with Daegu Gyeongbuk Institute of Science and Technology, Daejeon, South Korea, from February 2011 to February 2015; an Assistant Professor with Korea Aerospace University, Goyang, South Korea, from September 2008 to February 2011; a Principal Research Engineer with Samsung Electronics, Suwon, Korea, from April 2005 to April 2008; and a Research Engineer with LG Electronics, Seoul, Korea, from February 1994 to June 2000 (promoted to Senior Research Engineer in 2000). He has authored or coauthored over 130 papers and patents, including essential patents for the DVD standard, four of which were cited over 100 times. His research interests are in Internet of Things; cyberphysical systems; smart manufacturing; and their applications to robots, vehicles, and factories.

Sang H. Son (M’85–SM’98–F’13) received the B.S. degree in electronics engineering from Seoul National University, Seoul, South Korea; the M.S. degree from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea; and the Ph.D. degree in computer science from University of Maryland, College Park, MD, USA.

He is currently the Dean of the Graduate School and the Director of the Cyber Physical Systems Global Center, Daegu Gyeongbuk Institute of Science and Technology, South Korea. He has been a Professor in the Department of Computer Science, University of Virginia, and a WCU Chair Professor with Sogang University, Seoul. He has been a Visiting Professor with KAIST, City University of Hong Kong, Hong Kong; Ecole Centrale de Lille, Paris, France; Linkoping University, Linkoping, Sweden; and University of Skovde, Skovde, Sweden. His research interests include cyberphysical systems, real-time and embedded systems, database and data services, and wireless sensor networks. He has authored or coauthored over 300 papers and edited/authored four books in these areas. His research has been funded by the Korean government, the National Research Foundation, the U.S. National Science Foundation, the Defense Advanced Research Projects Agency, the Office of Naval Research, the Department of Energy, the National Security Agency, and IBM.

Prof. Son is a member of the Korean Academy of Science and Technology and The National Academy of Engineering of Korea. He has served as the Chair of the IEEE Technical Committee on Real-Time Systems from 2007 to 2008. He is an Associate Editor for ACM Transactions on Cyber-Physical Systems (T-CPSS) and has been on the editorial board of IEEE TRANSACTIONS ON SYSTEMS, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, and Real-Time Systems Journal. He is a Founding Member of the ACM/IEEE Cyber-Physical Systems (CPS) Week and is a member of the steering committee for Real-Time Computing Systems and Applications, CPS Week, and Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (SEUS). He was a recipient of the Outstanding Contribution Award from the CPS Week in 2012.