SAINT+: Self-Adaptive Interactive Navigation Tool+ for Emergency Service Delivery Optimization

Yiwen Shen, *Student Member, IEEE*, Jinho Lee, *Student Member, IEEE*, Hohyeon Jeong, *Student Member, IEEE*, Jaehoon Jeong, *Member, IEEE*, Eunseok Lee, *Member, IEEE*, and David H. C. Du, *Fellow, IEEE*

***Abstract*— This paper proposes an evolved Self-Adaptive Interactive Navigation Tool (SAINT+) to reduce the delivery time of emergency services and to improve navigation efficiency for the vehicles influenced by accidents. To the best of our knowledge, SAINT+ is the first attempt to optimize the delivery of emergency services as well as the navigation routes of vehicles around accident areas. Based on the congestion contribution model of SAINT and aggregated information from vehicles in the vehicular cloud, we propose a virtual path reservation strategy for emergency vehicles to guarantee a fast emergency service delivery. We also develop an accident area protection scheme based on an adjusted congestion contribution matrix and protection zones to evacuate vehicles in the accident area. To further reduce travel delay of neighbor vehicles in the accident area, we also present a dynamic traffic flow control model. Through extensive simulations with a real-world map, SAINT+ outperforms other state-of-the-art schemes for the travel delay of emergency vehicles. In scenarios with a high vehicle density, SAINT+ reduces the travel delay of emergency vehicles by 42.2%.**

***Index Terms*— Navigation, path planning, road emergency service, road accident, vehicular networks, self-adaptive, interactive.**

I. Introduction

ROAD safety is always the highest concern for people. According to the Traffic Safety Fact 2014 [1], published by the U.S. National Highway Traffic Safety Administration (NHTSA) in 2016, more than 32 thousand people were killed, and more than 3.2 million people were injured in the estimated 6 million vehicle traffic crashes in the U.S.. A similar situation happened in European Union countries in 2013, where road crashes took about 26 thousand lives [2]. Road emergency services (e.g., Emergency Medical Service (EMS) and firefighting squad) provide life-and-death rescue services to people who are in road crashes. The delivery time of emergency services is one of the most important aspects to reduce fatalities in road crashes. The delivery time is influenced by many factors (e.g., response time, training level, road traffic and situation on site), and some of the factors can be improved through intensive exercises. However, a real-world and dynamic road traffic mixed with randomness challenges the efficient delivery of emergency services. In addition, since the vicinity of the accident is affected by the road accident, the overall traffic quality tends to be downgraded.

With the advancement of Vehicular Ad Hoc Networks (VANET), vehicles equipped with an On-Board Unit (OBU) can share driving information with each other via Vehicle-to-Vehicle (V2V) communication, and report road conditions to a vehicular traffic center via a Road Side Unit (RSU) [3] or a base station of cellular networks (e.g., Evolved Node B (eNodeB) in 4G-LTE [4], [5]) by a Vehicle-to-Infrastructure (V2I) data forwarding scheme. A Traffic Control Center (TCC) [6], which is a vehicular traffic center, can disseminate driving instructions to vehicles by an Infrastructure-to-Vehicle (I2V) data forwarding scheme. Eventually vehicles and TCC can share information in real-time, which enables vehicles to have a global sense of the traffic situation. Lots of V2I [7], [8] and I2V data forwarding schemes [9]–[11] for VANET have been proposed to improve the efficiency of information sharing among vehicles and RSUs. The merits to install OBUs in vehicles and RSUs in transportation infrastructures, respectively, can justify the installation cost. Meanwhile, public service departments in various nations have been considering to recommend or mandate the installation of OBUs and RSUs. For example, the U.S. Department of Transportation (USDOT) has been evaluating the policy of mandatory installation of OBUs and RSUs, and conducting a series of field tests since 2014. It is expected that a new mandatory regulation for installing V2V and V2X devices in new vehicles can be approved in the U.S. in near future [12].

On the other hand, the technology of cloud computing and big data processing has provided a powerful tool for dealing with huge volumes of data in a cloud server. A cloud server dedicated to traffic monitoring with a big data processing engine can process real-time traffic information

Manuscript received July 13, 2016; revised January 18, 2017 and April 4, 2017; accepted May 28, 2017. This work was supported in part by the National Research Foundation of Korea through the Ministry of Science, ICT, and Future Planning, Basic Science Research Program under Grant 2014006438 and in part by the Next Generation Information Computing Development Program under Grant 2015045358. The Associate Editor for this paper was Z. Ding. *(Corresponding author: Jaehoon Jeong.)*

Y. Shen, J. Lee, H. Jeong, and E. Lee are with the Department of Computer Science and Engineering, Sungkyunkwan University, Seoul 16419, South Korea (e-mail: chrisshen@skku.edu; jinholee@skku.edu; jeonghh89@skku.edu; leees@skku.edu).

J. Jeong is with the Department of Interaction Science, Sungkyunkwan University, Seoul 16419, South Korea (e-mail: pauljeong@skku.edu).

D. H. C. Du is with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: du@cs.umn.edu).

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

Digital Object Identifier 10.1109/TITS.2017.2710881

collected from road networks. Also, the widely equipped GPS module enables vehicles to obtain mobility information such as geographic location, speed and direction. A vehicle can report the information to other vehicles and a TCC. Furthermore, the GPS navigator, either a dedicated one (e.g., Gamin [13] and TomTom [14]) or a smartphone App (e.g., Waze [15] and Navfree [16]), can improve the driving efficiency by suggesting driving routes to a driver.

However, the current GPS road navigators mainly suggest a time-wise shortest path using the Dijkstra algorithm [17] along with a real-time road traffic. Basically, they provide vehicles with a non-interactive navigation service. When the road traffic abruptly shifts, a vehicle may follow an inefficient navigation in a road network. Our previous work, SAINT [18], proposed a new model, called congestion contribution, which estimates the congestion level at each road segment. Through the interaction between vehicles and the traffic cloud, the running vehicles can have a global optimized navigation route. Nevertheless, SAINT lacks a dedicated mechanism to handle the delivery of emergency services, where emergency vehicles may take a big detour on a heavy traffic condition that contradicts the purpose of emergency services. And, the deteriorated traffic near an accident site is not considered by SAINT either.

This paper proposes an evolved Self-Adaptive Interactive Navigation Tool (SAINT+) for the delivery of emergency services and accident area protection. *The goal of this work is to reduce the delivery time of emergency services and the traffic congestion resulting from a road accident*. An accident road segment will be isolated from the road network by intentionally increasing the congestion level corresponding to the road segment. As a result, other vehicles that will use the accident road segment in the near future will be required to reroute so that the influence of the accident can be minimized. Also, SAINT+ protects the route of an emergency vehicle by restricting other vehicles to use the route, which can maximally guarantee the efficiency of the emergency service delivery. Furthermore, because the accident road segment usually causes congestion in the vicinity, SAINT+ creates protection zones for the accident road segment and controls the traffic flow of the zones. The main difference between SAINT+ and SAINT is that SAINT+ identifies and improves an efficiency issue on emergency services delivery in modern intelligent transportation systems having wireless communications between vehicles and infrastructures. To the best of our knowledge, SAINT+ is the first attempt to optimize emergency service delivery and to efficiently reroute the traffic around an accident area in a road network through an interaction between vehicles and traffic cloud.

The main contributions of this paper are as follows:

- **An architecture for emergency service delivery and road accident area protection:** We propose an architecture that provides an optimized delivery of emergency services and road accident protection in an interactive manner.
- **An optimized road emergency service delivery scheme:** We propose an optimized scheme that can minimize the travel time of emergency vehicles via virtual road reservation. The scheme also provides an efficient rerouting for vehicles around an accident area. This strategy dynamically adjusts virtual congestion on the path of an emergency vehicle, and can reduce the traffic influence from the emergency vehicle.
- **A novel dynamic traffic flow control model for accident protection area:** We derive a traffic flow control model based on the congestion contribution model, which can adjust the traffic in the vicinity of an accident area. The dynamic traffic flow control strategy considers inflow and outflow rates to form a traffic indicator. This traffic indicator along with the average congestion contribution of road segments outside the protection zones decides the congestion level inside the protection zones.

The rest of this paper is organized as follows. Section II summarizes and analyzes related works. Section III explains the architecture of SAINT+, assumptions, goals, and challenges. In Section IV, we describe the travel delay model and SAINT navigation system. Section V elaborates the design of emergency service delivery and accident area protection in SAINT+. Section VI describes the working procedure of SAINT+. Section VII demonstrates the effective performance of SAINT+ by comparing it with legacy navigation schemes. Finally, in Section VIII, we conclude this paper along with future work.

II. Related Work

A lot of commercial vehicle dedicated GPS navigators (e.g., Garmin [13], TomTom [14], and iNAVI [19]) or smartphone-App navigators (e.g., Waze [15], Navfree [16], Skobbler [20], and Tmap [21]) use time-wise shortest path algorithms, which utilize traffic statistics or real-time traffic information. These legacy navigators can to some extent provide vehicles with good navigation services when the traffic is not heavy. However, during rush hours in commute areas or an accident area, many vehicles use the same routes at the same time, the time-wise shortest navigation path at a certain moment can not provide those vehicles with a efficient navigation. This is because the legacy navigators do not consider an instant congestion caused by an accident.

Many research results [22]–[25] related to mitigating road traffic congestion have been reported. Wang *et al.* [22] proposed a method called Next Road Rerouting (NRR) to mitigate unexpected urban traffic congestion. NRR reroutes vehicles based on a routing cost function combining occupancy, travel time, distance to destination, and geographic closeness to the congestion. Pan *et al.* [23] proposed a distributed vehicular traffic rerouting system (DIVERT). DIVERT enables vehicles to make rerouting decisions collaboratively, which moves the bulk of rerouting computation from servers to vehicles. Kim *et al.* [24] explored the Markov decision process model to solve a dynamic vehicle routing problem. They utilized a neuro-dynamic programming algorithm to avoid the high dimensionality. Chen and Chang [25] proposed a new traffic control framework. This framework aims at optimizing the throughput of road networks and the travel time of vehicles by considering the joint passing rate and traffic lights at adjacent intersections.

Pan *et al.* [26] proposed five strategies to proactively reroute vehicles for lower travel time (called PVTR). The five strategies are the dynamic shortest path (DSP), A* shortest path with repulsion (AR*), random k shortest path (RkSP), entropy-balanced kSP (EBkSP), and flow-balanced kSP (FBkSP). The five strategies can be categorized into two groups, single shortest path strategies and multiple shortest path strategies. DSP and AR* belong to the first group, and RkSP, EBkSP, and FBkSP belong to the second. DSP is a classical shortest path method that reschedules the routes of vehicles according to the current road traffic. AR* is a modification of the A* search algorithm [27], which lets the heuristic function of A* include a repulsive force $R(x)$ that considers the influence of other vehicles on the optimal selection of alternative paths. RkSP, EBkSP, and FBkSP are all based on the k shortest path algorithm [28]. The difference is that RkSP randomly selects alternative paths, whereas EBkSP exploits an entropy model and FBkSP considers traffic flow, respectively, in order to select alternative paths. Their proposed system compares the above five strategies and selects the best paths for the users. SAINT+ is also based on the Dijkstra algorithm and k shortest path algorithm. However, with a new congestion contribution model, it provides not only a short travel time for general vehicles as PVTR did [26], but also the assurance of accident area protection and fast emergency service delivery.

Wang *et al.* [29] presentd a real-time path (RTP) planning for a transportation system based on a hybrid network combining the cellular network and VANET. A vehicle-traffic server feeds real-time path planning with traffic information collected from RSUs or cellular networks. RTP utilizes the measurement of traffic inflow and outflow metrics, the road capacity, and the virtual queues that maintain the count of the buffered vehicles staying at each intersection, which are classified by different destinations. Especially, it considers recurrent congestion and non-recurrent congestion [29]. When a traffic congestion occurs at a road segment, the vehicles around the congested area will obtain suggestions for their new paths from the vehicle-traffic server through an RSU or a base station. The new suggested paths depend on the selection of the maximum weight derived from the average turning cost for each related candidate intersection. Different from the local metrics of RTP's virtual queues at each intersection along with distance-based turning cost, SAINT+ employs a congestion contribution matrix with a traffic prediction to globally plan the routes of vehicles that are requested to detour due to an accident or congestion. Moreover, SAINT+ is dedicated to the protection of an accident area and emergency service delivery. For an accident area, it constructs protection zones based on the realistic rationale that traffic congestion at an accident area can affect the traffic of the vicinity.

Chen *et al.* [30] proposed a novel emergency vehicle dispatching system that includes automatic emergency resource assignment and a path planning scheme [30]. The proposed system learns traffic condition from traffic history and suggests a lane reservation scheme for emergency vehicles. It utilizes RSU as a dissemination mean to tell other vehicles to adjust their velocity or switch to another lane when an emergency vehicle moves along each road segment. In the path planning scheme of this system, the Dijkstra algorithm is used to schedule the time-wise shortest navigation path of an emergency vehicle. This scheme may fail to perceive future congestion, which is possible that a route based on current traffic conditions or historic traffic statistics may be congested in the near future. Thus, this approach cannot guarantee the efficient delivery of emergency vehicles. SAINT+ plans the routes for emergency vehicles based on a prediction model, and schedules globally optimal routes for other vehicles by using this prediction model. Therefore, SAINT+ can guarantee not only the efficient delivery of emergency vehicles, but also minimize the impact of the road accident on the traffic flow in neighboring areas.

Tian *et al.* [31] explored an algorithm that combines the differential evolution (DE) and the particle swarm optimization (PSO) to schedule rescue vehicles for extinguishing forest fires. The algorithm minimizes both the extinguishing time of fires and the number of dispatched vehicles by the DE and the PSO. They particularly considered the influence of the fire spread speed. Tian *et al.* [32] also proposed multiobjective optimization models for placing vehicle inspection stations. The optimization models minimize the total transportation cost and time of customers subject to the constraints that the customers shall arrive at the inspection locations within a specific time and with a certain cost. The optimization models employ a teaching-learning-based optimization algorithm, and combine it with a multi-objective optimization method. Instead of optimizing the on-site time and the number of dispatched vehicles, and the location of vehicle inspection stations, SAINT+ focuses on optimizing the travel time of emergency vehicles in an urban scenario with background traffic. SAINT+ utilizes the congestion contribution model of SAINT to dynamically control traffic flow of the emergency event area.

Malviya *et al.* [33] proposed two new classes of approximate techniques for dynamic route planning in a continuous query system. The two techniques are K-paths and *proximity measures*, which can speed up the processing of designated routes in a continuous route query environment. Xu *et al.* [34] proposed two efficient route search strategies, called incremental route search (IRS) and hierarchical route search (HRS), in hierarchical dynamic road networks. IRS strategy computes a partial path towards some intermediate destinations, and HRS strategy computes the fastest path on a small graph based on a generated hierarchical road network. Unlike the techniques from Malviya *et al.* and Xu *et al.* using real-time traffic update, SAINT+ reserves paths for emergency vehicles based on the traffic congestion prediction in order to detour vehicles that will use the paths of the emergency vehicles in a short-term time duration. Also, SAINT+ constructs protection zones for a vehicle accident and suggests a dynamic flow control strategy for the accident area.

III. Problem Formulation

This section describes the architecture, assumptions, and design challenge for road emergency service delivery. Based on our previous work [18], we try to improve the efficiency

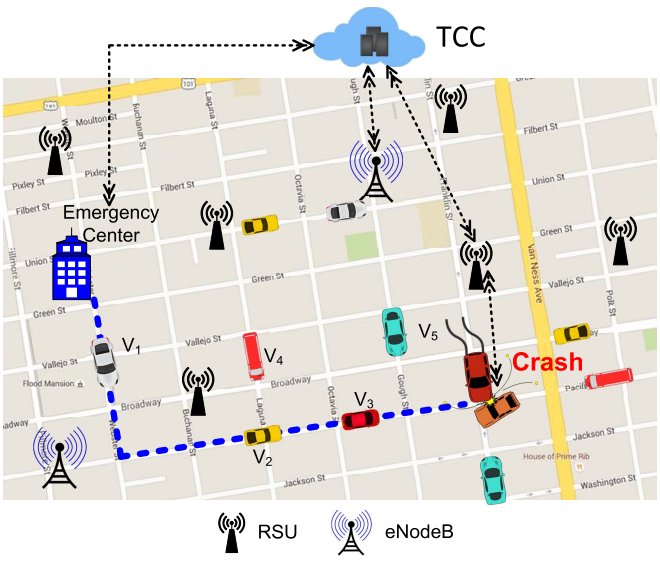


Fig. 1. Architecture and Scenario of Emergency Services Delivery.

of emergency rescue services for a road traffic accident, and maintain the global optimized routes for other vehicles, i.e., guarantee a fast delivery of emergency services and reduce the influence of the traffic accident on the travel time of other vehicles.

A. Road Emergency Service Delivery Architecture

This section describes the architecture of road emergency rescue and the component nodes of the vehicular cloud. Fig. 1 shows the architecture for road emergency service delivery. This architecture includes the following components:

- **Traffic Control Center (TCC)**: A TCC is an urban traffic dedicated management server complex that can include high performance computing clusters, wired and wireless networks, data storage, etc. TCC can collect real-time road traffic information such as average speed, vehicle density, and traffic flow. It has a reliable connection with an emergency center to which a vehicle accident event can be reported. TCC is also in charge of the maintenance of the congestion contribution matrix and calculation of the optimal navigation route.
- **Emergency Center (EC)**: An EC is an urban emergency event response hub that is responsible for the emergency event processing and the dispatch of emergency vehicles. Usually, it is located inside a hospital that can provide emergency medical treatment. EC can receive notification of a road emergency from TCC.
- **Road-Side Unit (RSU)**: An RSU is a wireless access point located at each intersection. RSU communicates with vehicles in vehicluar ad hoc networks via DSRC and is connected to the Internet. In a target urban road network, RSUs are connected with each other via either wired or wireless networks.
- **Evolved Node B (eNodeB)**: eNodeB is a base station in 4G-LTE cellular networks. A subscribed mobile device can use voice, text message and data access services if it is under an eNodeB's coverage. eNodeB is a supplementary component for vehicles to communicate with TCC when they cannot connect with RSUs.
- **Vehicle**: A vehicle is equipped with a GPS navigator, a DSRC communication device [35], [36] and a 4G-LTE communication module [4]. It can communicate with an RSU or an eNodeB to report current mobility information (e.g., speed, location, and acceleration) and the planned navigation route. In this paper, we categorize vehicles into two types: emergency vehicles (EVs) and general vehicles (GVs).

B. Assumptions

This section lists assumptions for SAINT+ as follows:

- A vehicle can communicate with the TCC by either DSRC [35], [36] or 4G-LTE [4]. The default communication mode is via DSRC. If a vehicle cannot communicate with an RSU by DSRC, 4G-LTE communication module can be used to exchange information with the TCC.
- The communication delay between vehicles and the TCC can be ignored because the order of communication delay via DSRC [37] or 4G-LTE [38] is much smaller than the time that a vehicle takes to pass through a road segment. Thus, a vehicle and a TCC can exchange information in time.
- Loop detectors are connected with RSUs at road segments. Loop detectors can count the number of vehicles entering and leaving the road segment, and RSUs can know the road traffic accurately. The traffic information is reported to the TCC for the navigation service of SAINT+.
- Vehicles agree to report their positions and destinations to TCC whenever necessary for SAINT+. The information transmission between vehicles and TCC shall be encrypted, and TCC shall not disclose the drivers' mobility and identity information to any 3rd party for their privacy protection.

C. Scenario and Challenges

We consider a scenario that a car crash happens in an urban area, as shown in Fig. 1. The crashed cars report the accident to the TCC via an RSU or an eNodeB, and TCC informs the nearest EC of the accident. EC dispatches an EV toward the accident site. The blue dash line is a selected path with the minimum travel time. Although this path is the shortest one, along this path there are many other vehicles currently driving (e.g., V_1, V_2, V_3). When the EV is passing, other vehicles may stop or pull up on the road side in order to leave space for the EV. Meanwhile, vehicles on neighboring roads (e.g., V_4 and V_5) may also need to use the path of the EV. The vicinity of the car crash is facing a possible traffic or low-speed driving caused by the accident. As a result, the travel time for vehicles around the accident will be increased.

To improve the delivery efficiency of emergency services and also protect an accident site, we have following challenges:

- Even though the path of an EV is time-wise shortest for the current traffic condition, vehicles on the path of the EV may hinder the penetration of the EV.

- Vehicles that will use the path of the EV may also affect the efficiency of the emergency service delivery.
- The slow traffic flow around the accident site may cause extra traffic to other vehicles.

In the next section, we will explain our solution for the above challenges, which is based on the SAINT navigation scheme.

IV. Travel Delay Prediction and SAINT Navigation

This section explains the modeling of travel delay on both road segments and End-to-End (E2E) travel paths, based on our early work [9], [39], as well as the SAINT navigation scheme [18]. First, a definition is given as follows.

Definition 1 (Road Network Graph): A ***Road Network Graph*** *is a directed graph $G = (V, E)$ for a road map such that V is the set of vertices (i.e., intersections), denoted as $V(G)$, and E is the set of directed edges e_{ij} (i.e., road segments) for $i, j \in V$, denoted as $E(G)$.*

A. Travel Delay on Road Segments and End-to-End Delay

The previous research [9], [39] showed that in a scenario of a light road traffic, the travel delay of a vehicle of a certain length follows a Gamma distribution [40]. The travel delay on a road segment i is defined as d_i, which follows a Gamma distribution, $d_i \sim \Gamma(\kappa_i, \theta_i)$, where κ_i and θ_i are the shape parameter and scale parameter, respectively [41]. κ_i and θ_i can be derived from the mean travel delay μ_i and the variance of travel delay σ_i^2 on road segment i, where μ_i and σ_i^2 can be collected from either a real-time driving report of vehicles or loop detectors on each road segment [42], [43].

Assume that the travel delays of road segments in a path are independent from each other and follow the same Gamma distribution. That is, the collection of all random variables d_i in the travel delay D of this path is *independent and identically distributed* (*i.i.d*). Thus, the mean and variance of the travel delay of this path can be estimated as the cumulative mean and variance of the road segments along the path [9], [39]. Hence, supposing that, from the source position to the destination position, there are n intersections and $n-1$ road segments, the mean $E[D]$ and variance $Var[D]$ of travel delay D of this path can be expressed as:

$$E[D] = \sum_{i=1}^{n-1} E[d_i] = \sum_{i=1}^{n-1} \mu_i, \tag{1}$$

$$Var[D] = \sum_{i=1}^{n-1} Var[d_i] = \sum_{i=1}^{n-1} \sigma_i^2. \tag{2}$$

Therefore, the E2E travel delay D can be modeled as a Gamma distribution, $D \sim \Gamma(\kappa_D, \theta_D)$, where κ_D and θ_D are formulated from $E[D]$ and $Var[D]$ [9].

In practice, various methods can be used to measure the delay of a single road segment or an E2E path, e.g., a camera or loop detectors. As mentioned in Section III-B, each road segment can be installed with loop detectors at its entrance and exit to measure travel delay accurately in real-time. Fig. 2 shows the installation of loop detectors. Assume that for a sample period T, the count of vehicles traveling on a road segment (v_i, v_j) is n, where (v_i, v_j) represents an edge with two vertices v_i and v_j, the time for each vehicle V_i to travel on this road segment is t_{V_i}, and the mean travel delay $d_{(v_i,v_j)}$ of road segment (v_i, v_j) can be calculated as follows:

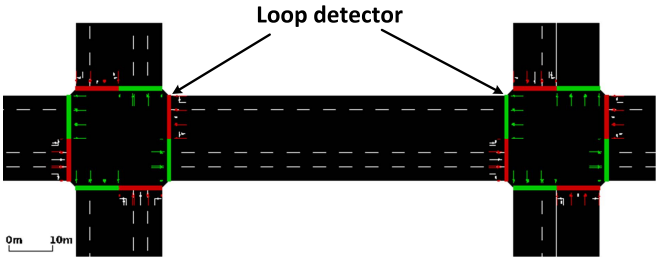


Fig. 2. Installation of Loop Detector.

$$d_{(v_i,v_j)} = \begin{cases} \dfrac{\sum_{i=1}^{n} t_{V_i}}{n} & n > 0, \\ \dfrac{l_{(v_i,v_j)}}{v_L} + \overline{d_{TL}} & n = 0, \end{cases} \tag{3}$$

where $l_{(v_i,v_j)}$ is the length of the road segment (v_i, v_j), v_L is the speed limit of this road segment, and $\overline{d_{TL}}$ is the mean waiting time of traffic light. In (3), two cases are considered. The first case is when there are vehicles driving on a road segment. In this case, $d_{(v_i,v_j)}$ is the mean travel time for all these vehicles. The second case is no vehicle moved on a road segment during the sampling time. In this case, $d_{(v_i,v_j)}$ is the travel time for the road segment by the speed limit plus the mean traffic light waiting time. Note that the time t_{V_i} includes the waiting time for traffic lights at an intersection before the next road segment.

For a vehicle V_i with a route R_{V_i} containing a sequence of intersections: $R_{V_i} = \langle v_1, v_2, \ldots, v_n \rangle$, $v_k \in V(G)$, the E2E delay can be formulated as:

$$D_n^{V_i} = \sum_{k=1}^{n-1} d_{(v_k,v_{k+1})}, \tag{4}$$

where R_{V_i} is an intersection set including all vertices in the route of V_i.

In the next section, we introduce our early work, called the SAINT navigation scheme [18].

B. SAINT Navigation Scheme

This section introduces the SAINT [18] navigation scheme from two aspects: (i) a Congestion Contribution model and (ii) a Congestion Contribution model-based shortest path algorithm.

1) Congestion Contribution Model: Based on the *road segment delay* and *E2E delay* defined in the previous section, each vehicle has a travel delay on each road segment (called link delay) along its route, as well as E2E delay for the whole route. For a particular vehicle V_j, the Congestion Contribution (CC)

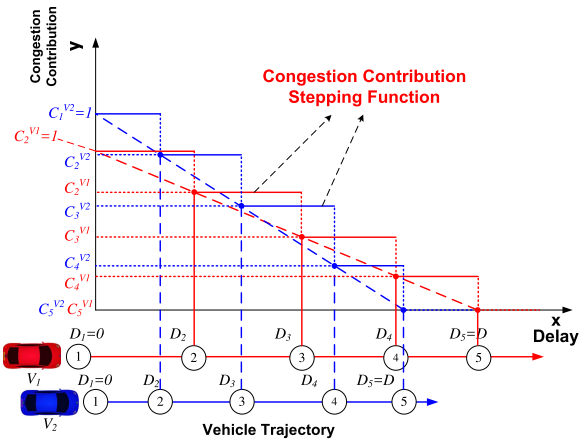


Fig. 3. Example of a Step Function.

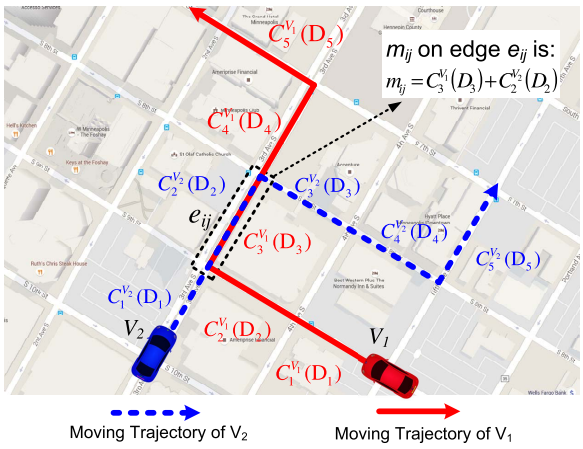


Fig. 4. Example of a Congestion Contribution Matrix.

$c_i^{V_j}$ [18] is modeled as:

$$c_i^{V_j} = 1 - \frac{D_i^{V_j}}{D_n^{V_j}}, \tag{5}$$

where $D_n^{V_j}$ is the E2E delay of a vehicle for a travel path with n vertices, i.e., the travel delay from the source intersection 1 to the destination intersection n, and $D_i^{V_j}$ is the sub-route delay from the source intersection 1 to an intermediate intersection i:

$$D_i^{V_j} = \begin{cases} \sum_{k=1}^{i-1} d_{(v_k, v_{k+1})} & \text{for } i \geq 2, \\ 0 & \text{for } i = 1, \end{cases} \tag{6}$$

where $d_{(v_k, v_{k+1})}$ is the travel delay for a road segment (v_k, v_{k+1}) in the route. Note that $D_1^{V_j}$ is defined as 0 since it is travel delay at the beginning of the route, and the corresponding CC $c_1^{V_j}$ is 1.

The $c_i^{V_j}$ on each road segment of a route is maintained as invariant, so we define the Congestion Contribution Step Function (CCSF) $C_i^{V_j}(x)$ for the sub-route delay x from a vehicle's start position to an intermediate position on its trajectory:

$$C_i^{V_j}(x) = c_i^{V_j} \cdot u(x - D_i^{V_j}), \tag{7}$$

where $u(x - D_i^{V_j})$ is a shifted unit step function defined as:

$$u(x - D_i^{V_j}) \triangleq \begin{cases} 1 & x \geq D_i^{V_j} \\ 0 & x < D_i^{V_j} \end{cases} \quad \text{for } i \in (1, n). \tag{8}$$

For example, Fig. 3 shows the step functions of two vehicles mapped to their routes. Red vehicle V_1 and blue vehicle V_2 have different scales for the CC value on the y-axis. Both the vehicles have a CC value 1 (indicating the injection of one vehicle into a road segment) at the entrance of their 1st road segment in their route, and then the CC value linearly decreases, while on each road segment it is maintained as constant.

For a road network graph G with n vertices (intersections), a Congestion Contribution Matrix (CCM) is defined as:

$$M = \begin{bmatrix} 0 & m_{1,2} & \cdots & \cdots & m_{1,n} \\ m_{2,1} & 0 & & m_{i,j} & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & & m_{n-1,n} \\ m_{n,1} & \cdots & \cdots & m_{n,n-1} & 0 \end{bmatrix}, \tag{9}$$

where $i, j \in V(G)$ and $m_{i,j}$ is the cumulative link congestion contribution of road segment $e_{i,j}$, i.e., the sum of congestion contribution from all vehicles that are passing and will pass through edge $e_{i,j}$. After a vehicle that contributes to the congestion passes through the edge, the corresponding congestion contribution of this vehicle will be subtracted from the corresponding entry in the matrix CCM. Note that the CCM is maintained by TCC in the vehicular cloud and dynamically updated via V2I and I2V communications between vehicles and RSUs. With this CCM, we know that the upper bound of any element in the CCM is the number of vehicles (denoted by N) running in the road network if all N vehicles pass through the same road segment e_{uv}.

As shown in Fig. 4, two vehicles V_1 and V_2 are moving in a target road network and they have different routes leading to different destinations, but there is a common road segment e_{ij} that they will pass through. Assume that at the current time, there are no other vehicles driving in this road network. The congestion contribution m_{ij} of edge e_{ij} in CCM is $m_{ij} = C_3^{V_1}(D_3) + C_2^{V_2}(D_2)$. After they pass edge e_{ij}, the entry m_{ij} of the CCM is updated by subtracting the congestion contribution value $C_3^{V_1}(D_3)$ and $C_2^{V_2}(D_2)$.

In the next section, we introduce the congestion contribution model-based Delay-constrained Shortest Path (DSP) Algorithm.

2) Delay Constrained Shortest Path Algorithm: To explain DSP algorithm, we first introduce the α-increase travel path [18]. The α-increase travel path for a vehicle is a path such that the E2E delay of this path is within $(1 + \alpha)D$, where D is the shortest path travel time on the current traffic condition.

Algorithm 1 Delay-Constrained Shortest Path Algorithm

function DSP(G, u, v, α)
  $P \leftarrow \emptyset$ ▷ P will contain the list of intersections for a selected α-increase shortest path.
  $D_{uv} \leftarrow$ Compute-Dijkstra-Path-E2E-Delay(G, u, v)
  $\hat{D}_{uv} \leftarrow (1+\alpha) \times D_{uv}$
  $K \leftarrow$ Compute-k-Smallest-Congestion-Increase-Paths(G, u, v)
  $k \leftarrow |K|$
  for $i \leftarrow 1$ to k **do**
    $D \leftarrow$ Compute-E2E-Delay(K, i)
    if $D \leq \hat{D}_{uv}$ **then**
      $P \leftarrow$ Get-Path(K, i)
      return P
    end if
  end for
  $P \leftarrow$ Compute-Dijkstra-Path(G, u, v)
  return P
end function

The objective of the DSP algorithm is to find an α-increase travel path based on CCM and Yen's k-shortest-path algorithm [28]. As shown in Algorithm 1, the input of the algorithm is a graph G, source intersection u, destination intersection v, and the increase percent of travel delay α. First, the E2E delay of a time-wise shortest path is computed based on Dijkstra's shortest path algorithm. Then we obtain the α-increase E2E delay $\hat{D}_{uv}$. Based on CCM, the k smallest congestion increase paths are stored in K and the size of K is k. For each path in K, the E2E delay is calculated and put into D. If D is smaller than the E2E delay $\hat{D}_{uv}$ of an α-increase travel path, the corresponding path for the pair of (K, i) is selected. Otherwise, continue to check the next path. If all k smallest congestion increase paths cannot fulfill the condition, a time-wise Dijkstra shortest path is selected.

The idea behind the DSP algorithm is that at the current moment, a constrained detour route is selected to minimize the future congestion in the target road network. Comparing this to the greedy algorithm (e.g., Dijkstra algorithm), DSP intentionally plans a time-wise suboptimal route for vehicles, but the overall travel delay can be significantly reduced. The results in Section VII confirm this idea.

Through this algorithm, a vehicle can find a global optimized path to minimize future possible congestion. In the next section, we will describe the design of the new navigation scheme for emergency service delivery.

V. The Design of SAINT+ Emergency Navigation

This section explains the design of SAINT+ navigation for the delivery of road emergency services from the following perspectives: (i) congestion contribution adaption for an accident road segment and emergency vehicles, (ii) zone-based accident area protection, and (iii) traffic flow-based dynamic congestion contribution adjustment for the protection zones.

As explained in the previous section, the congestion contribution model provides a prediction of the future congestion so that each vehicle can be assigned an optimal route that not only is the constrained detour path but also distributes uniformly the traffic flows of all the vehicles on all possible road segments in a target road network. Based on this model *the goal of the emergency navigation of SAINT+ is to optimize a route for each vehicle when any accident happens in the road network, and allow emergency vehicles to reach the accident site as fast as possible while reducing the traffic congestion influence on other vehicles as much as possible.* For the explanation of SAINT+, we first give several definitions as follows:

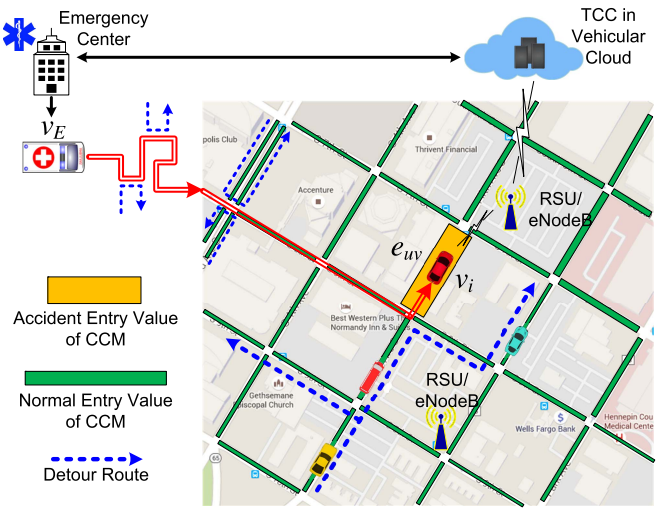


Fig. 5. Accident Road Segment Protection and Emergency Navigation.

Definition 2 (Accident Road Segment): Let an ***Accident Road Segment*** *be an edge $e_{uv} \in E(G)$ where a vehicle accident occurs.*

Definition 3 (Contour Vertices): Let the ***Contour Vertices*** *be a set $S_{CV} \subset V(G)$ that includes all adjacent vertices of the two vertices u and v of the* ***Accident Road Segment*** *e_{uv}, with $u \notin S_{CV}$, $v \notin S_{CV}$.*

Definition 4 (Protection Zone): Let ***Protection Zone*** *be a set $S_{PZ} \subset E(G)$ such that all edges of S_{PZ} are directly jointed to the two vertices u and v of the* ***Accident Road Segment*** *e_{uv} or the* ***Contour Vertices*** *of another* ***Protection Zone****.*

A. DSP-Based Reroute Algorithm for an Accident Road Segment and Emergency Vehicle

Congestion contribution model gives a good prediction for future congestion based on a decreasing step function that utilizes the percentage of travel time on the whole travel course. That is, the farther road segment on the vehicle's trajectory has the lower congestion contribution according to the travel delay of the road segment. Moreover, the CCM maintained by TCC is periodically updated through the interaction with vehicles via V2I and I2V, or a 4G-LTE link, and provides a global optimized route for each vehicle.

1) Emergency Service Delivery Optimization: Now we consider a special case where a vehicle v_i has an accident (e.g., vehicles collision or crash) on a road segment e_{uv} of a target road network, as shown in Fig. 5. This accident vehicle directly (by itself when the DSRC device works well, even after

a serious damage by the accident) or indirectly (by other vehicles or RSU when the DSRC device in the accident vehicle does not function) communicates with a nearby RSU to report the accident, and then RSU forwards this information to TCC in vehicular cloud. In Fig. 5, the green rectangles on road segments represent the congestion contribution values in CCM. A thicker rectangle indicates a higher congestion in the near future. First, TCC delivers the accident information to EC, where an an emergency vehicle is prepared (e.g., ambulance, police car, and fire engine) for this accident. Second, in order to protect the accident road segment e_{uv}, the corresponding entry $m_{u,v}$ in CCM is set to an artificial congestion increase value γ (e.g., 5000) as follows:

$$m'_{u,v} = m_{u,v} + \gamma, \quad \text{for } m_{u,v} \in M \quad \text{and } \gamma > N, \tag{10}$$

where N is the total number of vehicles in the current road network. In Fig. 5, the thick yellow rectangle shows the protection of the accident road segment.

In addition, because TCC has the route information of all SAINT+ clients, by the updated matrix and DSP algorithm, TCC recalculates the new routes for the vehicles that will pass through the accident edge in their current routes, as shown in Algorithm 2. Then, TCC initiates the messages for the reroute request, including the new route, to each vehicle via the I2V protocol. Subsequently, the affected vehicles will follow the new routes from the requests. As shown in Fig. 5, the blue dashed lines are possible detour paths. Any newly joined vehicle will also receive an accident-protected route.

After a short period, an EV v_E will be ready to head to the accident road segment designated by EC. Once the EV moves to the road network, it requests TCC to calculate a fastest route from the current position to the accident spot. TCC replies to it with a calculated route. Also, the corresponding entries $m_{i,j}$ in CCM, including all road segments in the EV's route $R_{EV} = \langle v_1, v_2, \ldots, v_n \rangle$ for $v \in V(G)$, are set to an artificial congestion increase value δ (e.g., 5000) as follows:

$$m'_{v_k,v_{k+1}} = m_{v_k,v_{k+1}} + \delta, \quad \text{for } m_{v_k,v_{k+1}} \in M \quad \text{and } \delta > N, \tag{11}$$

where N is the total number of vehicles in the current road network. TCC collects the information for all vehicles that will pass through any road segment of the EV's route, and then calculates and sends the new routes to these vehicles, as shown in Algorithm 2. These vehicles will avoid the route of the EV so that the EV can get a clear way toward the accident road segment. The blue dashed lines in Fig. 5, near the red path of the EV are the possible detour routes for the vehicles that will pass through the red line.

The aim of Algorithm 2 is to check every vehicle's route as to whether it is overlapped with the accident edge or the route of EV. If so, a new route P replaces the current route R_n in line 8 of Algorithm 2. The inputs of Algorithm 2 are the road graph G, vehicle set N, the route of EV R_{EV}, the accident edge e_{ACC}, and the detour factor α for the DSP algorithm.

Once the accident is handled and the road segment is cleaned, the EV informs TCC that the accident has been handled. Then TCC will recover the CCM to its original level by subtracting the γ and δ, compute the new route for each vehicle and distribute it to each of them. Thereafter, all the vehicles will reroute based on the updated routes for efficient navigation.

Algorithm 2 SAINT+ Reroute Algorithm

```
1: procedure REROUTE(G, N, R_EV, e_ACC, α)
2:     for all n ∈ N do
3:         R_n ← Get-Route(n)
4:         if any e ∈ R_n = e_ACC or any e ∈ R_EV then
5:             s ← Get-Current-Location(n)
6:             d ← Get-Destination(n)
7:             P ← DSP(G, s, d, α)
8:             R_n ← P
9:         end if
10:    end for
11: end procedure
```

2) Zone-Based Accident Area Protection: Since the surrounding areas of the accident road segment are usually affected by the accident, many vehicles may experience congestion, so they slowdown their moving speed, which causes further congestion; this in turn degrades the navigation efficiency. For example, because of a collision, a vehicle driving in the vicinity will take a longer time than in normal traffic conditions. This is the reason we propose *zone-based accident area protection* to lessen the impact of the accident on the traffic flow near the accident area.

In a road network graph G, the edge e_{uv} is an accident road segment, and the 1st **Protection Zone**, $Zone$ 1 (Z_1), consists of all edges jointed to vertices u and v except e_{uv}. Formally, the set of edges S_{Z_1} for the zone Z_1 is defined as follows:

$$S_{Z_1} = \{e : e \in N(e_{uv})\}, \tag{12}$$

where $N(e_{uv})$ is the neighbor edges of the accident edge e_{uv}.

Similarly, the 2nd **Protection Zone**, $Zone$ 2 (Z_2), includes all edges jointed to the **Contour Vertices**, which can be represented as:

$$S_{Z_2} = \{e : e \in N(S_{Z_1}) \; and \; e \notin S_{Z_1} \; and \; e \neq e_{uv}\}, \tag{13}$$

where $N(S_{Z_1})$ is the neighbor edges of all edges in S_{Z_1}.

For instance, Z_1 and Z_2 are formed as illustrated in Fig. 6. As a driving rule in Z_1 and Z_2, all the vehicles try to evacuate from the both areas. The outside vehicles try a detour to avoid moving into the both areas, that is, a detour around these areas.

B. Dynamic Traffic Flow Control for the Protection Zones

In this section, we will explain the design of dynamic congestion contribution adjustment for the protection zones based on traffic flow. Even though the protection zones provide strong protection for the accident road segment and fast delivery of emergency services, other vehicles may have to use long detour paths to avoid using the road segments in protection zones. Especially, in a road network dominated by one-way road segments, the detoured vehicles may be congested due to limited options of detour road segments. Through the observation of traffic flows in protection zones, it

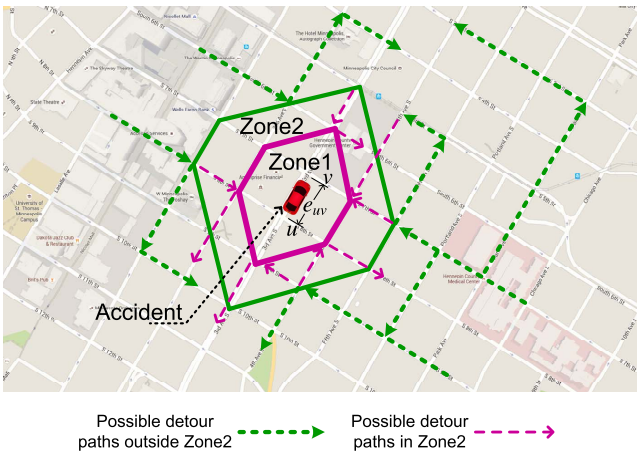


Fig. 6. Zone Protection for the Accident.

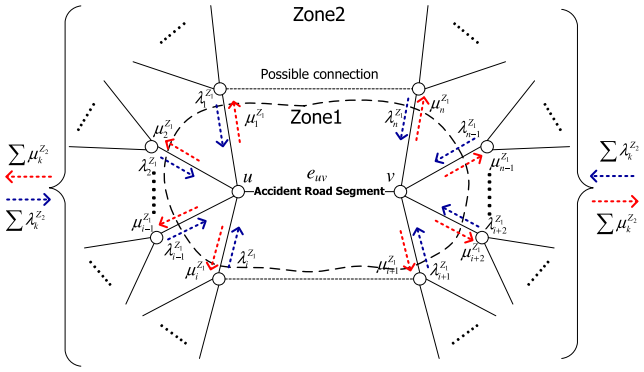


Fig. 7. Inflow and Outflow Rates for the Road Segments in Zone1 and Zone2.

is reasonable to permit some vehicles to use the road segments in the zones when the traffic in this area is relatively light. According to this rationale, we propose a traffic flow control scheme for the zone areas.

Without loss of generality, we consider a road network graph G based on an accident road segment e_{uv} as shown in Fig. 7, where the vertices u and v are the two ends of the edge e_{uv}. Referring to the Definition 4, $Zone1(Z_1)$ and $Zone2(Z_2)$ are formed to protect the accident spot. We assume that the vehicle set $S_V(T_i)$ includes each vehicle during the ith sample time T_i, and the inflow rate $\lambda(T_i)$ of a road segment during the ith sample duration T_i can be expressed as:

$$\lambda(T_i) = \frac{|S_V(T_i)|}{T_i} \quad \text{for } T_i > 0. \tag{14}$$

In order to measure the traffic status inside zones, we define the inflow rate as λ^x, $x \in \{Z_1, Z_2\}$. Then

$$\lambda^x = \sum_{j \in S_x} \lambda_j(T_i) \quad \text{for } x \in \{Z_1,\ Z_2\}. \tag{15}$$

Outflow rate μ^x has a similar calculation process to the inflow rate λ^x for $x \in \{Z_1,\ Z_2\}$. Then we define a traffic indicator I, which is the ratio of inflow rate to outflow rate, for each zone as

$$I_x = \begin{cases} \dfrac{\lambda^x}{\mu^x}, & \lambda^x > 0,\ \mu^x > 0 \\ \lambda^x, & \lambda^x > 0,\ \mu^x = 0 \\ 0, & \lambda^x = 0,\ \mu^x > 0 \\ 1, & \lambda^x = 0,\ \mu^x = 0 \end{cases} \quad \text{for } x \in \{Z_1,\ Z_2\}. \tag{16}$$

Intuitively, if I is larger than 1, this means that the inflow rate is larger than the outflow rate, i.e., more vehicles move into the area. On the contrary, if it is smaller than 1, this means that more vehicles leave the area, and if it is close to 1, the corresponding zone has a balanced traffic flow. For example, as shown in Fig. 7, the inflow rates, $\lambda_1^{Z_1}, \lambda_2^{Z_1}, \cdots, \lambda_n^{Z_1}$, and the outflow rates, $\mu_1^{Z_1}, \mu_2^{Z_1}, \cdots, \mu_n^{Z_1}$, (here we omit the sequence of the sample time T_i) of all the edges in Z_1 are measured. Therefore, for Z_1 with n edges and Z_2 with m edges, the total inflow rate and the total outflow rates are $\lambda^{Z_1} = \sum_{i=1}^{n} \lambda_i^{Z_1}$, $\mu^{Z_1} = \sum_{i=1}^{n} \mu_i^{Z_1}$, and $\lambda^{Z_2} = \sum_{k=1}^{m} \lambda_k^{Z_2}$, $\mu^{Z_2} = \sum_{k=1}^{m} \mu_k^{Z_2}$, respectively. Then, we obtain the corresponding I_{Z_1} and I_{Z_2}.

Recall CCM in equation(9); we know that the CC value m_e on an edge e in CCM is the summation of all CC values from different vehicles accumulated on this edge, which can be described as:

$$m_e = \sum_{k \in S_V} C_j^k(D_j) \quad \text{for } S_V = \{V_1, V_2,\ \ldots\ , V_a\}, \tag{17}$$

where S_V is the set of vehicles that will pass through the edge e, and $C_j^k(D_j)$ is the corresponding value of CCSF for edge e contributed by a vehicle (denoted by k) in the set S_V, and j is the index of the edge e in the travel path of k. So the CC value m_e^x of each edge e for $x \in \{Z_1, Z_2\}$ can be updated via the following model:

$$m_e^{'x} = m_e^x + \overline{C} \times I_x, \quad x \in \{Z_1, Z_2\}, \tag{18}$$

where $\overline{C}$ is the mean CC value of edges not belonging to Z_1 or Z_2, and m_e^x is the CC value of each edge e for $x \in \{Z_1, Z_2\}$.

The rationale for this model relies on the observation of vehicle behaviors changing before and after the construction of zone areas. Before the construction of zones, vehicles could go through zones regardless of the speed and traffic affected by the accident, whereas after the construction of them, the detoured paths for the vehicles avoid the edges of zones by checking CCM, i.e., comparing the CC values of the edges inside the zones to those of the edge outside the zones. This model updates the CC values for edges inside each zone with the mean CC value $\overline{C}$ of all the edges outside the zones and the traffic indicator I_x. When I_x is larger than 1, m_e^x is enlarged by I_x times $\overline{C}$, i.e., m_e^x increases dramatically to the sum of m_e^x and $\overline{C} \times I_x$. On the other hand, when I_x is smaller than 1 but larger than 0, the augment is the fraction of $\overline{C}$, which enables path planning to assign the zone-inside edges to the vehicles with less probability than the zone-outside edges for the zone protection.

So far, we have illustrated the design of SAINT+ for emergency service delivery and accident site protection. In the next section, we show the overall navigation procedure of SAINT+.

VI. SAINT+ Navigation Procedure

This section explains the navigation procedure of emergency service delivery and accident protection along with practical considerations in this procedure.

A. Navigation Procedure

The navigation procedure is as follows:

1) A vehicle functioning as a SAINT+ client follows the same procedure of our previous work SAINT [18] in regular traffic condition (i.e., the non-accident case). By V2I [8] and I2V [9] data delivery schemes or a 4G-LTE link [4], the vehicle sends its navigation request, including its source and destination, to a nearby RSU, and then the RSU forwards the request to the TCC. The TCC calculates a global optimized route for the vehicle based on the congestion contribution matrix, and replies to the vehicle via an RSU. If the vehicle deviates from the notified route, the same procedure can be repeated to obtain a new optimized route from the TCC.
2) Whenever a vehicle using the SAINT+ service has an accident with other vehicles, the accident information will be forwarded to a nearby RSU (or eNodeB). The RSU informs the TCC, and then the congestion contribution of the accident road segment will be updated with a very large value, and the original CC value will be recorded in TCC for future CC recovery.
3) Meanwhile, TCC virtual protection areas (i.e., protection zones) for the accident road segment are constructed around the accident road segment. The congestion contribution of the protection zones will be dynamically adjusted based on the traffic flow control model described in Section V-B.
4) After that, TCC immediately broadcasts a reroute request to all vehicles via RSUs, but only those that will pass through the accident road segment and protection zones in the planned routes will be asked to reroute, i.e., detour from accident road segment and protection zones. Note that the detour route still follows the SAINT [18] procedure to guarantee the global optimized route.
5) After an accident happens, an EV will be dispatched to the accident road segment as soon as possible. With the SAINT+ service, the EV follows the same procedure as SAINT [18] to obtain a global optimized route, but the difference is that the congestion contributions of the road segments along its route to the accident road segment will be set to a very large value to disallow other vehicles to use the road segments along its route, as described in V-A. The original CC values will be recorded in TCC for future CC recovery after the delivery of the EV. Similarly, TCC will broadcast a detour request to all vehicles and only the vehicles that will pass through the road segments of EV will reroute according to the method of SAINT [18].
6) When arriving at the accident road segment and after finishing the rescue and medical care, the EV will initiate a mission accomplished message to TCC via an RSU. TCC will recover the congestion contribution of the accident road segment and the road segments of the protection zones. When starting to move back to the EC from the accident road segment, the EV can use SAINT+ service again to obtain a global optimized route so that the travel time can be reduced.
7) All SAINT+ clients navigate according to the normal SAINT navigation procedure [18].

Note that a vehicle with the origin or the destination or the both that are in the protection zones can follow its current planned path to reach the destination. SAINT+ does not stop a GV moving, and schedules an optimal path for it. If the destination of a GV is in the accident road segment, the GV can move to the destination, but will experience a serious delay due to the influence of the accident.

B. Practical Considerations

To use SAINT+ in real traffic environment, a navigator in a vehicle shall connect to an On-Board Unit (OBU). The navigator can receive a scheduled route for the current travel. The driver first inputs the destination for the travel, and the current position and the destination will be transmitted to TCC via OBU. TCC calculates a global optimal travel path and sends it back to the navigator. The vehicle can start to move. When there is an emergency event and the vehicle will pass the road segment of the emergency event, TCC will multicast the reroute request to the vehicle. The vehicle automatically reports current position without an intervention of the driver. TCC unicasts the new scheduled route to the vehicle. To allow SAINT+ to work well in the real world, there are practical considerations:

- During an emergency service delivery, there are two reroute procedure called in SAINT+. The first is the moment when the accident happened, and the second is the moment when the EV is dispatched. Considering the time between the accident occurrence and the dispatch of an EV, if it is a reasonably short time, the first reroute can be limited to vehicles in the vicinity of the accident road, or the two calls of the reroute procedure can be combined as one call after the EV is dispatched. This reduces the reroute frequency of SAINT+ clients since the frequent reroute may cause an uncomfortable driving experience. This simplification shall be considered on a case-by-case basis, e.g., emergency response time and infrastructure level.
- Normally, to deal with an accident event, several EVs would be dispatched at the same time or different time period. These EVs may include police cars, EMS, firefight team, etc. The proposed scheme in SAINT+ may show an unsatisfactory performance in the delivery of several EVs. Basically each EV's route will be protected, so every new entered EV needs to follow the planned path based on the current CCM, and this results in serious detours for the latest joined EVs. One practical solution is maintaining two CCMs in the TCC such that one is a normal CCM used by the vehicles except EVs, and the other is tailored for EVs by not adding extra CC values for the routes of

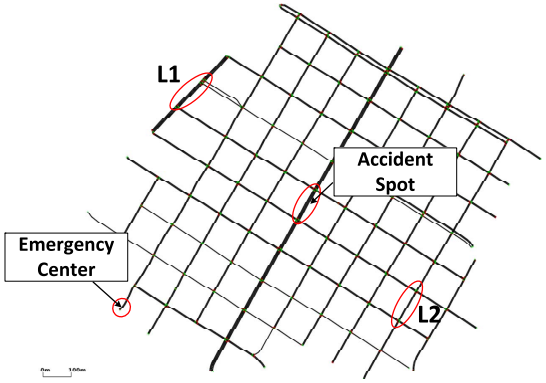


Fig. 8. Road Network in the Downtown Area of Minneapolis for Simulation.

the EVs. In the next section, we evaluate the scenario of multiple EVs to disallow the other vehicles to use the road segments along the routes of the EVs.

So far, we have explained the design and procedure of SAINT+, as well as practical issues. In the next section, we will show performance evaluation of SAINT+ with other state-of-the-art schemes.

VII. Performance Evaluation

We demonstrate the performance through simulations based on SUMO (Simulation of Urban MObility) [44]. We use a real-world map obtained from OpenStreetMap [45], which is the downtown area of Minneapolis, MN, US, as shown in Fig. 8. The information of the map is shown in Table I. We edited the map by *Java OpenStreetMap Editor* (JOSM) [46] and *NETCONVERT* [47] to remove dead ends and modify redundant streets. The emergency center is located at an end of the road on the left bottom of the map. The accident spot is set at a road segment of the center of the map. Simulation settings in SUMO were as follows: the vehicle length is 5 m, the minimal gap between vehicles is 2.5 m, and the deceleration is 6 m/s^2.

The conducted simulations include two scenarios:

- **Consecutive accidents are reached by one emergency vehicle (EV)**. To measure the travel delay of an EV and general vehicles (GVs), every new accident vehicle is injected after 30 s when the previous accident is handled by an EV. 20 vehicles are selected to continuously travel between two fixed points, $L1$ and $L2$, in order to make road traffic exist around the accident spot, as shown in Fig. 8.
- **One accident is reached by ten EVs**. To measure the delivery ratio of the ten EVs, an accident vehicle is only injected once at a fixed simulation time point, and ten EVs are set to be dispatched by the EC.

At the beginning of a simulation, vehicles are sequentially placed on the target road network and the destination of each vehicle is randomly chosen. Once a vehicle reaches its destination, it will disappear, and then a new vehicle will be placed at a random road segment to start a new travel. The emergency response time (i.e., the delay from the moment when the accident is reported to the moment when the EV is dispatched) is set to 60 s. Note that we only measure the performance that EV heads to the accident road segment, the return of EV to EC is not measured in the current simulation.

TABLE I
Description of the Road Network

Road network size	1623 m×1345 m (i.e., 1.00 mile×0.84 miles).
Total number of road segments	227
Total number of intersections	105
Total length of all road segments	24.277 km
Sampling time of the loop detectors	10 s

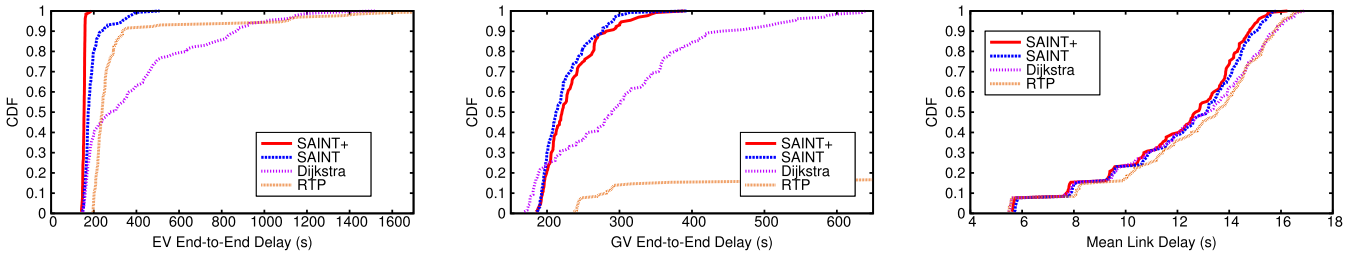
TABLE II
Simulation Configurations

Parameter	Description
Number of vehicles (N)	The number N of vehicles moving within the road network. The default N is 600.
Delay increase threshold (α)	Delay increase threshold in Algorithm. 1. The default α is 0.5 [18].
Path number (k)	The number of shortest paths searched in Algorithm. 1. The default k is 5 [18].
Speed limit (v_{max})	Maximum vehicle speed for road segments. The default is 80 km/h (i.e., 50$mile/h$ or 22.22 $meter/s$).
Vehicle acceleration (a_v)	Vehicle acceleration per unit time for vehicles. The default is $3m/s^2$.
EMS scene time [48]	Average time for an EV's crew to handle an accident vehicle and clear it from the accident road segment. The default is 16.1 minutes (i.e., 961 s) for the street or highway [48].

All vehicles conform to the *car following model* (Krauss *et al.* [49]). All traffic lights in the target road network follow a *static traffic light scheduler* [50]. All road segments are installed with two loop detectors at their entry and exit. GVs do not give way to an EV so that the EV can move fast in the road segment having those GVs in our simulations. Other evaluation settings are as follows:

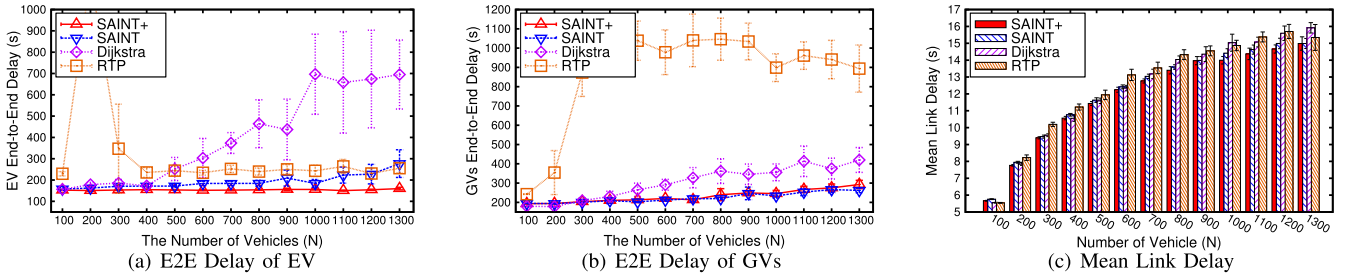
- **Performance Metrics:** (i) *the mean E2E travel delay of EV and GVs*, (ii) *the mean link delay of all vehicles*, (iii) *the mean number of accidents handled*, (iv) *the successful delivery ratio of EVs*, and (v) *the tradeoff of mean E2E delay between EV and GVs*.
- **Baseline:** (i) SAINT [18], (ii) Dijkstra [13]–[17], [19]–[21], and (iii) RTP [29].
- **Parameters:** In the performance evaluation, we investigate the impact of the following parameters: (i) *Vehicular traffic density* N (i.e., the number of vehicles), (ii) *Maximum vehicle speed* v_{max} (i.e., speed limit), and (iii) *Vehicle acceleration* a_v.

The simulation time is set to two hours (i.e., 7200 s) and the simulations are repeated ten times with different seeds. The performance results include an error bar to show the 90% confidence intervals. The default values of the simulation are specified in Table II. The CC increase-values γ and δ mentioned in Section V-A are set to a value larger than the upper bound of the CC value in the matrix CCM. In our simulations, they are all set to 5000. The EMS scene time in Table II is based on real-world data [48], which is the average time for an EV's crew to deal with an emergency event. Note that the general urban speed limit varies from



(a) The CDF of EV E2E Delay (b) The CDF of GVs E2E Delay (c) The CDF of Mean Link Delay

Fig. 9. The Cumulative Distribution Functions of Navigation Delays. (a) The CDF of EV E2E Delay. (b) The CDF of GVs E2E Delay. (c) The CDF of Mean Link Delay.



(a) E2E Delay of EV (b) E2E Delay of GVs (c) Mean Link Delay

Fig. 10. The Navigation Delays. (a) E2E Delay of EV. (b) E2E Delay of GVs. (c) Mean Link Delay.

50 km/h to 80 km/h (i.e., 13.89 $meter/s$ to 22.5 $meter/s$ or 31.07 $mile/h$ to 50 $mile/h$), and the actual vehicle speed is affected by many factors (e.g., traffic light and congestion level). The 80 km/h speed limit for the road segments in the simulation shown in Table II can reflect a general scenario, and we also studied the impact of the speed limit in Section VII-C.

A. The Comparison of Navigation Behavior

The purpose of SAINT+ is to optimize the delivery time of EVs. To show the overall performance comparison, we collected all the data of the simulations for the CDFs of SAINT+, SAINT, Dijkstra, and RTP illustrated in Fig. 9. From Fig. 9(a), we can see that more than 95% of mean emergency delivery times of SAINT+ are less than 160 s, which significantly outperforms the Dijkstra algorithm. Even for SAINT and RTP, SAINT+ maintains its good performance all the time, and when the traffic grows heavy, the advantage increases.

For the mean E2E delay of GVs, as shown in Fig. 9(b), the CDF of SAINT+ is slightly below that of SAINT and still maintains a large gap compared to Dijkstra, and a huge gap compared to RTP. Since the priority of SAINT+ is to guarantee fast delivery of EVs, it sacrifices to some extent the E2E delay of GVs, but this slight sacrifice is worthwhile because the time saved for the EV is related to saving a human life.

Fig. 9(c) shows the mean link delays (i.e., mean travel time of road segments) of all the schemes. The results in the figure suggest that SAINT+ grows faster than other schemes, which confirms the improvement in the E2E delay of the EV. Also, we noticed an interesting curve shape in the figure. As shown in Fig. 9(c), when the mean link delay (x-axis) is smaller than 10 s, the shapes of all the curves have a clear step style, and as the mean link delay increases, the step style becomes vague. The lower link delay (e.g., smaller than 10 s) represents the case of the light traffic condition (i.e., low vehicle density), which is mainly decided by the length of each road segment and traffic lights. When the traffic grows heavy (i.e., high vehicles density), the traffic congestion increases the travel time for each vehicle on each road segment, which causes a longer link delay for each road segment. In addition, the step style shape is softened, shown as part of larger mean link delay (i.e., larger than 10 s at x-axis) in Fig. 9(c).

B. The Impact of Vehicle Number

The results from Fig. 10 show that the E2E delays of the EV and GVs, and the link delays in all the schemes increase according to an increase in the number of vehicle running in the simulation. In Fig. 10(a), the E2E delay of the EV for the other schemes increases dramatically as the the number of running vehicles increases, but SAINT+ keeps it low as a constant-like value almost all of the time. Especially when vehicle density is the highest (e.g., the data 1300 at x-axis), in comparison with SAINT, SAINT+ reduces the mean E2E delay of EV from 277.0 s (SAINT) to 159.9 s (SAINT+), about a 42.2% reduction, and has very small variance (i.e., it is stable). If we consider the worst case at the highest density, compared to SAINT and Dijkstra, the mean E2E delay of EV in SAINT+ is reduced by 175.0 s (SAINT) and 696.7 s (Dijkstra), respectively.

Meanwhile, the E2E delay of GVs in SAINT+ gains very small degradation, as shown in Fig. 10(b). Considering the significant reduction in E2E delay of the EV in SAINT+, the tradeoff here is acceptable (we will show this tradeoff later in the section). That is, again analyzing the data at the highest vehicle density (i.e., vehicle number of 1300), the E2E delay

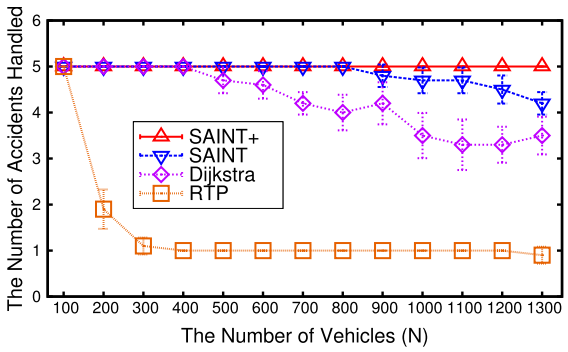


Fig. 11. The Number of Accidents Handled Within Limited Time.

for the GVs for SAINT+ increased from 262.0 s to 292.7 s, only about 30 s added (i.e., 11.2% downgrade), compared with SAINT.

Fig. 10(c) illustrates the change of the mean link delay according to the increase of running vehicles. We found that the mean link delays for all schemes go up when the vehicle density increases. However, when the number of vehicles reaches 700 or above, the difference of the mean link delay among Dijkstra, SAINT, RTP, and SAINT+ is enlarged, which confirms the improvement of SAINT+ in both E2E delay of EV and GVs to Dijkstra in the high density situation. From the observation in the simulation, the mean link delay of SAINT+ is smaller than that of SAINT, and this small gap on each link accounts for the overall reduction of E2E delay for EV.

Fig. 11 shows the number of accidents handled from each scheme within the simulation time. We can see that SAINT+ maintains good performance even though the vehicle density is high, whereas such a number from the other schemes gradually decreases. The improvement from SAINT+ is due to the optimized EV delivery scheme in which the EV's route can be protected, and so the delivery speed is improved. Therefore, within limited time, more accidents can be handled.

C. The Impact of Vehicle Speed Limit

To assess the impact of the vehicle speed limit, we measured the metrics by varying the vehicle's speed limit from 20 km/h to 80 km/h with a step of 5 km/h (i.e., from 5.56 m/s to 22.22 m/s with a step of about 1.39 m/s). Usually, 80 km/h is the highest speed limit in an urban area, which is why it is selected as the upper bound. As displayed in Fig. 12(a), the E2E delay of the EV is constantly below that of the other schemes on the whole x-axis, which shows a promising performance. Comparing to SAINT with a speed limit of 22.22 m/s, the E2E delay of SAINT+ was improved by 13.5%. Also, though Dijkstra has a high variation, SAINT+ and SAINT exhibit a very stable E2E delay regardless of the speed change.

In Fig. 12(b), the E2E delay of GVs decreases as the speed limit of vehicles increases. Although improvement in the E2E delay of the EV is achieved, the E2E delay of GVs of SAINT+ are similar to that of SAINT, and much better than that of Dijkstra. For the mean link delay in Fig. 12(c), it overall decreases according to increases in the speed limit, such that as the vehicle speed limit increases, the travel time spent on each link decreases. Thus, the relative lower mean link delay of SAINT+ verifies its good performance in Fig. 12(a) and Fig. 12(b).

D. The Impact of Vehicle Acceleration

We also examined the impact of vehicle acceleration. Fig. 13 shows the metrics comparison among the schemes under the change of acceleration. Similarly, the E2E delay of EV of SAINT+ is constantly better than those of SAINT and Dijkstra for all test points. There is about a 15% improvement for SAINT+. The E2E delay of the GVs for SAINT+ is similar to that of SAINT when the acceleration is lower than 2.5 m/s^2, and slightly worse than that of SAINT when the acceleration increases. The mean link delays of SAINT+ shown in Fig. 13(c) show a similar pattern with those of the speed limit in Fig. 12(c). SAINT+ in all cases has a short mean link delay than other schemes.

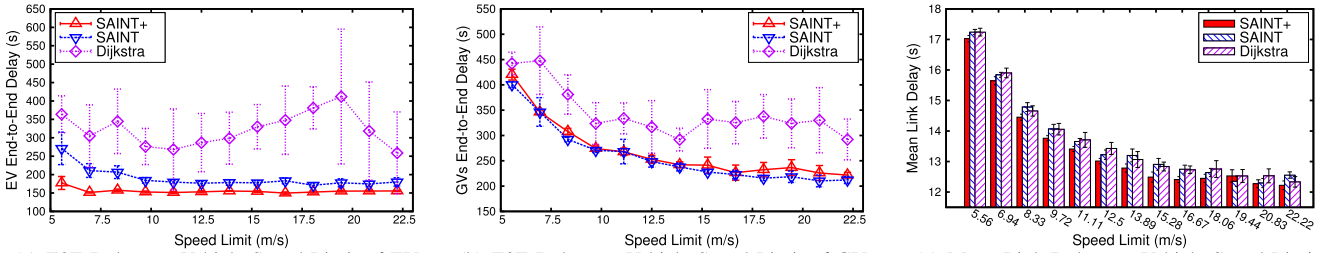
E. The Tradeoff of E2E Delay

In order to show how much SAINT+ improves the mean E2E delay of the EV in comparison with SAINT, as well as the tradeoff of E2E delay between the EV and GVs, we investigated the E2E delay reduction of SAINT+ against vehicle density, speed limit, and acceleration illustrated in Fig. 14. From Fig. 14(a) we can see that the E2E delay reduction of SAINT+ increases as the number of vehicles increases, whereas the E2E delay of the GVs increases moderately. For example, when the number of vehicle is 1200, SAINT+ reduces the mean E2E delay of EV by 30%, but that of the GVs increases by less than 5%.

On the other hand, Figs. 14(b) and 14(c) demonstrate the impact of the vehicle speed limit and acceleration on the tradeoff of the E2E delay of EV. Fig. 14(b) shows that the E2E delay reduction decreases as the speed limit increases. When the speed limit is larger than 12.5 m/s, the E2E delay reduction shows a stable improvement of about 14%. Meanwhile, the E2E delay on the GV side shows a modest downgrade. Fig. 14(c) shows that as the acceleration varies, the E2E delay reduction of EV fluctuates between 10% to 17%, whereas the GVs' E2E delay does not show a distinguishable pattern.

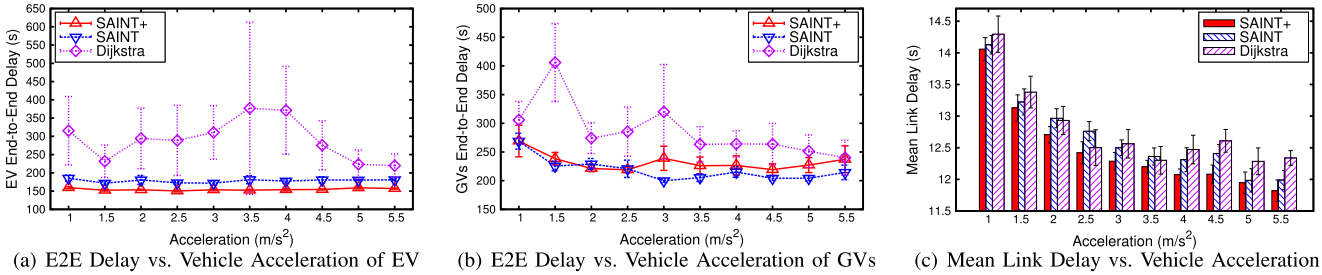
F. The Successful Delivery Ratio of Multiple EVs

To show the robustness of SAINT+ in real-world emergency situations, we conducted simulations with another scenario (the 2nd scenario mentioned at the beginning of this section) to calibrate the successful delivery ratio of EVs by dispatching several EVs to an accident site. As shown in Fig. 15, compared with the other schemes, from a low density to a high density of vehicles, SAINT+ always obtains a 100% successful delivery ratio of EVs. However, SAINT fails to fully deliver all EVs to the accident site when the vehicle density is high. This is because some EVs are jammed on the way to the accident site. Dijkstra and RTP lack a mechanism for a



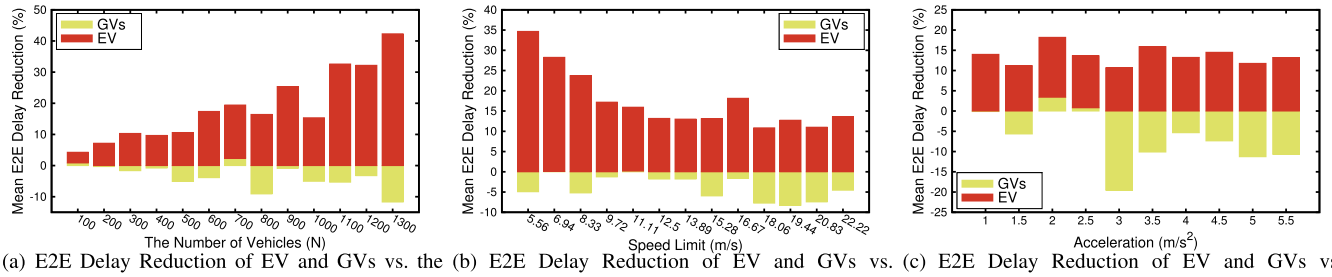
(a) E2E Delay vs. Vehicle Speed Limit of EV (b) E2E Delay vs. Vehicle Speed Limit of GVs (c) Mean Link Delay vs. Vehicle Speed Limit

Fig. 12. The Impact of Vehicle Speed Limit. (a) E2E Delay vs. Vehicle Speed Limit of EV. (b) E2E Delay vs. Vehicle Speed Limit of GVs. (c) Mean Link Delay vs. Vehicle Speed Limit.



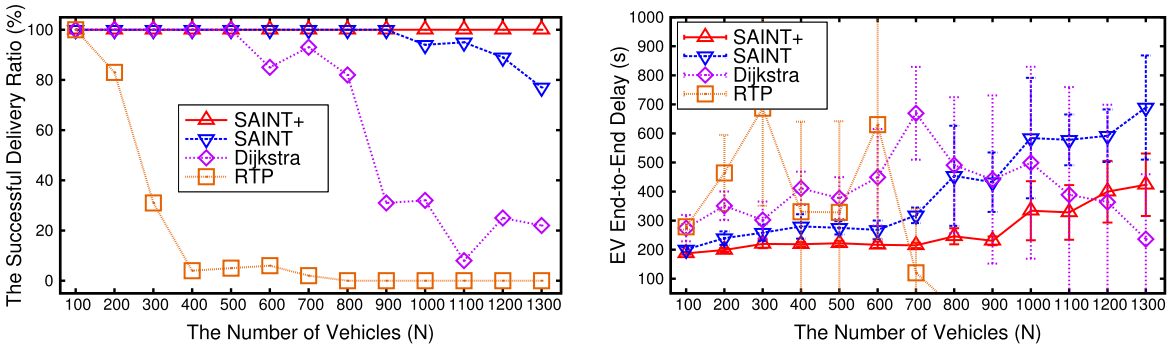
(a) E2E Delay vs. Vehicle Acceleration of EV (b) E2E Delay vs. Vehicle Acceleration of GVs (c) Mean Link Delay vs. Vehicle Acceleration

Fig. 13. The Impact of Vehicle Acceleration. (a) E2E Delay vs. Vehicle Acceleration of EV. (b) E2E Delay vs. Vehicle Acceleration of GVs. (c) Mean Link Delay vs. Vehicle Acceleration.



(a) E2E Delay Reduction of EV and GVs vs. the Number of Vehicles (b) E2E Delay Reduction of EV and GVs vs. Vehicle Speed Limit (c) E2E Delay Reduction of EV and GVs vs. Vehicle Acceleration

Fig. 14. The Tradeoff Between EV and GVs in Terms of E2E Delay of SAINT+ Compared with SAINT. (a) E2E Delay Reduction of EV and GVs vs. the Number of Vehicles. (b) E2E Delay Reduction of EV and GVs vs. Vehicle Speed Limit. (c) E2E Delay Reduction of EV and GVs vs. Vehicle Acceleration.



(a) The Comparison of the Successful Delivery Ratio of EVs (b) E2E Delay of Multiple EVs Delivery

Fig. 15. The Performance of Multiple EVs Delivery. (a) The Comparison of the Successful Delivery Ratio of EVs. (b) E2E Delay of Multiple EVs Delivery.

dedicated reroutes for the EV and accident protection, so the success rate is low when the density of vehicles grows. On the other hand, Fig. 15(b) shows the E2E delay of multiple EVs for the schemes. The performance of SAINT+ is much better than SAINT when the vehicle density increases. For the Dijkstra and RTP, the performance is acceptable when the traffic is light. However, when the density increases, their performance seems to improve because of the lack of data due to the plummeting of successful delivery ratios, as seen in Fig. 15(a). However, the E2E delay can be computed only by the several initial EVs, so the actual performance becomes worse.

VIII. Conclusion

This paper proposes an evolved Self-Adaptive Interactive Navigation Tool (SAINT+) for emergency service delivery and accident area protection. Based on our previous work, SAINT, SAINT+ optimizes the emergency service delivery by intentionally increasing the congestion level of an accident area and the path of an emergency vehicle in an urban district. Other vehicles are detoured around the accident area and the path of the emergency vehicle. Further, the formation of accident protection zones is suggested to protect the accident area and improve the delivery efficiency of the emergency service. In addition, SAINT+ reduces the impact of the accident by leveraging a dynamic traffic flow control model for the accident protection zones. Through the extensive and realistic simulations, the results demonstrate that SAINT+ outperforms other state-of-the-art schemes for the travel delay of the emergency vehicle(s). With the SAINT+ service, the efficiency of the emergency service delivery can be improved, and the impact of the accident upon the other vehicles can be reduced.

For future work, we will study a more accurate waiting time model of the traffic lights, since it is related to the accuracy of the congestion contribution model. We are also interested in an appropriate traffic light control scheme in order to globally schedule traffic light for the emergency service delivery. Moreover, we will investigate a traffic-light-free intersection passing scheme based on the SAINT+ navigation system as another interesting direction for enhancing efficient traffic flow at intersections.

Acknowledgment

The authors thank Dr. Junghyun (Peter) Jun for his valuable comments.

References

[1] National Center for Statistics and Analysis, "Traffic safety facts 2014: A compilation of motor vehicle crash data from the fatality analysis reporting system and the general estimates system," Nat. Highway Traffic Safety Admin, Washington, DC, USA, Tech. Rep. DOT HS 812 261, Mar. 2016.

[2] E. Uhlemann, "Introducing connected vehicles [connected vehicles]," *IEEE Veh. Technol. Mag.*, vol. 10, no. 1, pp. 23–31, Mar. 2015.

[3] A. Abdrabou and W. Zhuang, "Probabilistic delay control and road side unit placement for vehicular ad hoc networks with disrupted connectivity," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 1, pp. 129–139, Jan. 2011.

[4] R. Kirui, "Accessing cloud computing resources over 4G LTE," Dept. Inf. Technol., Helsinki Metropolia Univ. Appl. Sci., Helsinki, Finland, Tech. Rep. 10024 238, May 2014.

[5] C. D. Monfreid, "The LTE network architecture—A comprehensive tutorial," Alcatel-Lucent, Boulogne-Billancourt, France, Tech. Rep., 2017. [Online]. Available: http://www.cse.unt.edu/~rdantu/FALL_2013_WIRELESS_NETWORKS/LTE_Alcatel_White_Paper.pdf

[6] *District 6-0 Traffic Management Center*, accessed 2017. [Online]. Available: http://philadelphia.pahighways.com/philadelphiatcc.html

[7] J. Zhao and G. Cao, "VADD: Vehicle-assisted data delivery in vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 57, no. 3, pp. 1910–1922, May 2008.

[8] J. Jeong, S. Guo, Y. Gu, T. He, and D. H. C. Du, "Trajectory-based data forwarding for light-traffic vehicular ad-hoc networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 5, pp. 743–757, May 2011.

[9] J. Jeong, S. Guo, Y. Gu, T. He, and D. H. C. Du, "Trajectory-based statistical forwarding for multihop infrastructure-to-vehicle data delivery," *IEEE Trans. Mobile Comput.*, vol. 11, no. 10, pp. 1523–1537, Oct. 2012.

[10] J. P. Jeong, T. He, and D. H. C. Du, "TMA: Trajectory-based multi-anycast for multicast data delivery in vehicular networks," *Comput. Netw.*, vol. 57, no. 13, pp. 2549–2563, Sep. 2013.

[11] J. P. Jeong *et al.*, "TPD: Travel prediction-based data forwarding for light-traffic vehicular networks," *Comput. Netw.*, vol. 93, no. P1, pp. 166–182, Dec. 2015.

[12] NHTSA. *U.S. DOT Advances Deployment of Connected Vehicle Technology to Prevent Hundreds of Thousands of Crashes*, accessed 2017. [Online]. Available: https://www.nhtsa.gov/press-releases

[13] Garmin. *Dedicated Navigator*, accessed 2017. [Online]. Available: http://www.garmin.com

[14] TomTom. *Dedicated Navigator*, accessed 2017. [Online]. Available: http://www.tomtom.com

[15] Waze. *Smartphone App for Navigator*, accessed 2017. [Online]. Available: https://www.waze.com

[16] Navfree. *Free GPS Navigation for Android Smartphone*, accessed 2017. [Online]. Available: http://navfree.android.informer.com

[17] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959.

[18] J. Jeong, H. Jeong, E. Lee, T. Oh, and D. H. C. Du, "SAINT: Self-adaptive interactive navigation tool for cloud-based vehicular traffic optimization," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 4053–4067, Jun. 2016.

[19] iNAVI. *iNAVI Navigation System*, accessed 2017. [Online]. Available: http://www.inavi.com

[20] Skobbler. *Smartphone App for GPS Navigation and Maps*, accessed on 2017. [Online]. Available: http://www.skobbler.com/apps/navigation/android

[21] Tmap. *SKT Smartphone Navigator*, accessed 2017. [Online]. Available: http://www.tmap.co.kr/tmap2

[22] S. Wang, S. Djahel, Z. Zhang, and J. McManis, "Next road rerouting: A multiagent system for mitigating unexpected urban traffic congestion," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2888–2899, Oct. 2016.

[23] J. Pan, I. S. Popa, and C. Borcea, "DIVERT: A distributed vehicular traffic re-routing system for congestion avoidance," *IEEE Trans. Mobile Comput.*, vol. 16, no. 1, pp. 58–72, Jan. 2017.

[24] G. Kim, Y. S. Ong, T. Cheong, and P. S. Tan, "Solving the dynamic vehicle routing problem under traffic congestion," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 8, pp. 2367–2380, Aug. 2016.

[25] L.-W. Chen and C.-C. Chang, "Cooperative traffic control with green wave coordination for multiple intersections based on the Internet of vehicles," *IEEE Trans. Syst., Man, Cybern., Syst.*

[26] J. Pan, I. S. Popa, K. Zeitouni, and C. Borcea, "Proactive vehicular traffic rerouting for lower travel time," *IEEE Trans. Veh. Technol.*, vol. 62, no. 8, pp. 3551–3568, Oct. 2013.

[27] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci.*, vol. 4, no. 2, pp. 100–107, Jul. 1968.

[28] J. Y. Yen, "Finding the k shortest loopless paths in a network," *Manage. Sci.*, vol. 17, no. 11, pp. 712–716, Jul. 1971.

[29] M. Wang, H. Shan, R. Lu, R. Zhang, X. Shen, and F. Bai, "Real-time path planning based on hybrid-VANET-enhanced transportation system," *IEEE Trans. Veh. Technol.*, vol. 64, no. 5, pp. 1664–1678, May 2015.

[30] C.-Y. Chen, P.-Y. Chen, and W.-T. Chen, "A novel emergency vehicle dispatching system," in *Proc. IEEE 77th Veh. Technol. Conf. (VTC Spring)*, Jun. 2013, pp. 1–5.

[31] G. Tian, Y. Ren, and M. Zhou, "Dual-objective scheduling of rescue vehicles to distinguish forest fires via differential evolution and particle swarm optimization combined algorithm," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 11, pp. 3009–3021, Nov. 2016.

[32] G. Tian, M. Zhou, P. Li, C. Zhang, and H. Jia, "Multiobjective optimization models for locating vehicle inspection stations subject to stochastic demand, varying velocity and regional constraints," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1978–1987, Jul. 2016.

[33] N. Malviya, S. Madden, and A. Bhattacharya, "A continuous query system for dynamic route planning," in *Proc. IEEE 27th Int. Conf. Data Eng.*, Apr. 2011, pp. 792–803.

[34] J. Xu, Y. Gao, C. Liu, L. Zhao, and Z. Ding, "Efficient route search on hierarchical dynamic road networks," *Distrib. Parallel Databases*, vol. 33, no. 2, pp. 227–252, 2015. [Online]. Available: http://dx.doi.org/10.1007/s10619-014-7146-x

[35] Y. L. Morgan, "Notes on DSRC & WAVE standards suite: Its architecture, design, and characteristics," *IEEE Commun. Surveys Tuts.*, vol. 12, no. 4, pp. 504–518, 4th Quart., 2010.

[36] ETSI. *DSRC Standardization*, accessed 2015. [Online]. Available: http://www.etsi.org/WebSite/Technologies/DSRC.aspx
[37] Y. Yao, L. Rao, X. Liu, and X. Zhou, "Delay analysis and study of IEEE 802.11p based DSRC safety communication in a highway environment," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 1591–1599.
[38] Y. Nagai, L. Zhang, T. Okamawari, and T. Fujii, "Delay performance analysis of LTE in various traffic patterns and radio propagation environments," in *Proc. IEEE 77th Veh. Technol. Conf. (VTC Spring)*, Jun. 2013, pp. 1–5.
[39] J. Jeong and E. Lee, "VCPS: Vehicular cyber-physical systems for smart road services," *J. Korean Inst. Commun. Sci.*, vol. 31, no. 3, pp. 103–116, Feb. 2014.
[40] A. Polus, "A study of travel time and reliability on arterial routes," *Transportation*, vol. 8, no. 2, pp. 141–151, Jun. 1979.
[41] M. DeGroot and M. Schervish, *Probability and Statistics*, 3rd ed. Reading, MA, USA: Addison-Wesley, 2001.
[42] G. Dimitrakopoulos and P. Demestichas, "Intelligent transportation systems," *IEEE Veh. Technol. Mag.*, vol. 5, no. 1, pp. 77–84, Mar. 2010.
[43] Research and Innovative Technology Administration (RITA). *IntelliDrive: Safer, Smarter and Greener*, accessed 2017. [Online]. Available: http://www.its.dot.gov/intellidrive/index.htm
[44] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO–Simulation of urban mobility," *Int. J. Adv. Syst. Meas.*, vol. 5, nos. 3–4, pp. 128–138, Dec. 2012.
[45] M. Haklay and P. Weber, "OpenStreetMap: User-generated street maps," *IEEE Pervasive Comput.*, vol. 7, no. 4, pp. 12–18, Oct. 2008.
[46] D. S. I. Scholz. (2016). *Java OpenStreetMap Editor*. [Online]. Available: https://josm.openstreetmap.de
[47] DLR and Contributors. (2016). *NETCONVERT*. [Online]. Available: http://sumo.dlr.de/wiki/NETCONVERT
[48] The NEMSIS technical assistance center. (2016). *National EMS Information System*. [Online]. Available: http://www.nemsis.org/reportingTools/reports/nationalReports/createARep%ort.html
[49] S. Krauss, P. Wagner, and C. Gawron, "Metastable states in a microscopic model of traffic flow," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 55, pp. 5597–5602, May 1997. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevE.55.5597
[50] SUMO. *Simulation of Urban Mobility*, accessed 2017. [Online]. Available: http://sumo-sim.org

Yiwen (Chris) Shen received the B.S. degree from the Department of Communication Engineering and the M.S. degree from the Department of Mechatronics Engineering, North University of China, in 2009 and 2013, respectively. He is currently working toward the Ph.D. degree with the Department of Computer Science and Engineering, Sungkyunkwan University, South Korea. His research areas include intelligent transportation systems, wireless networks, and VANET. He received the Chinese Government Scholarship from the China Scholarship Council.

Jinho Lee received the B.S. degree from the Department of Mathematics, Sungkyunkwan University, South Korea, in 2014, where he is currently working toward the M.S. degree with the Department of Computer Science and Engineering. His research areas include intelligent transportation systems and VANET.

Hohyeon Jeong received the B.S. and M.S. degrees from the School of Electrical and Computer Engineering, Ajou University, South Korea, in 2012 and 2014, respectively. He is currently working toward the Ph.D. degree with the Department of Computer Science and Engineering, Sungkyunkwan University, South Korea. His research area is self-adaptive software systems.

Jaehoon (Paul) Jeong (M'14) received the B.S. degree from the Department of Information Engineering, Sungkyunkwan University, South Korea, in 1999; the M.S. degree from the School of Computer Science and Engineering, Seoul National University, South Korea, in 2001; and the Ph.D. degree from the Department of Computer Science and Engineering, University of Minnesota, in 2009. He is currently an Assistant Professor with the Department of Software, Sungkyunkwan University. His research areas are vehicular networks, wireless sensor networks, and mobile ad hoc networks. He is a member of the ACM and the IEEE Computer Society. His two data-forwarding schemes (called TBD and TSF) for vehicular networks were selected as spotlight papers in IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS in 2011 and IEEE TRANSACTIONS ON MOBILE COMPUTING in 2012, respectively.

Eunseok Lee received the B.S. degree from the Department of Electronic Engineering, Sungkyunkwan University, South Korea, in 1985, and the M.S. and Ph.D. degrees from the Department of Information Engineering, Tohoku University, Japan, in 1991 and 1988, respectively. He was an Assistant Professor with Tohoku University, Japan. He was a Research Scientist with Mitsubishi Electric Corporation. He is currently a Professor with the Department of Computer Engineering, Sungkyunkwan University. His research areas are self-adaptive software systems, software testing, and autonomic computing.

David H. C. Du received the B.S. degree in mathematics from National Tsing Hua University, Taiwan, in 1974, and the M.S. and Ph.D. degrees in computer science from University of Washington, Seattle, in 1980 and 1981, respectively. He is the Qwest Chair Professor with the Computer Science and Engineering Department, University of Minnesota, Minneapolis. His research interests include cyber security, sensor networks, multimedia computing, storage systems, and high-speed networking.