

## V2I 통신 기반의 효과적인 차량 대기행렬 길이 추정 기법

최종화\*, 안상현°, 이승재\*

## Effective Vehicle Queue Length Estimation Mechanism Based on V2I Communications

Jonghwa Choi\*, Sanghyun Ahn°, Seungjae Lee\*

## 요약

단속류 환경의 교통 신호제어기(road side unit, RSU)가 효과적으로 신호 제어를 하기 위해서는 차량 대기행렬 길이에 대한 정보를 필요로 하며, 이를 위해서 차량 감지가 선행되어야 한다. 대표적인 차량 감지 기법으로는 비디오 이미지를 분석하는 것과 도로에 설치된 차량 감지용 센서를 이용하는 방법이 있으나, 이들의 경우 차량 정보를 얻는 데 있어서 많은 계산량과 센서 설치 비용을 필요로 한다. 따라서 본 연구에서는 V2I(vehicle to infrastructure) 통신을 이용해서 교통 신호제어기가 적은 오버헤드로 차량 정보를 획득하는 방법을 제안한다. 제안 방식에서는 두 인접 교차로 간의 도로 구간을 섹터들로 나눈 후 각 섹터의 대표 차량만 교통 신호제어기에게 차량 정보 메시지를 보내게 함으로써 V2I 통신량을 감소시킨다. 제안 방식의 성능을 평가하기 위하여 SUMO와 OMNet++을 기반으로 한 차량 네트워크 시뮬레이션 프레임워크인 Veins를 이용해서 전송된 메시지 수 및 추정된 차량 대기행렬 길이의 정확도를 측정하였다.

**Key Words** : V2I, Vehicle queue length, Vehicle detection, Traffic signal control, Road side unit

## ABSTRACT

In the intermittent traffic environment, vehicle detection has to be performed first for the acquisition of vehicle queue length information in order for a traffic signal controller (road side unit, RSU) to effectively control traffic signals. Some of the representative vehicle detection mechanisms are the vehicle detection mechanism based on video image analysis and that based on sensors installed on the road. However, these mechanisms require high amount of computation and high sensor installation cost. Therefore, in this paper, we propose a mechanism in which a traffic signal controller uses the vehicle to infrastructure (V2I) communications to acquire vehicular information with less overhead. In the proposed mechanism, a road segment between two neighboring intersections is divided into a number of sectors in each of which only one vehicle sends a vehicle information message to the traffic signal controller, resulting in less V2I communications. For the performance evaluation of the proposed mechanism, we used the vehicle network simulation framework Veins based on SUMO and OMNet++, and measured the number of transmitted messages and the accuracy of the estimated vehicle queue length.

\* 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. NRF-2015R1A2A2A04 005646).

• First Author : (ORCID:0000-0001-7345-6517)University of Seoul, Department of Computer Science and Engineering, David13@uos.ac.kr, 학생회원

° Corresponding Author : (ORCID:0000-0001-7640-4480)University of Seoul, Department of Computer Science and Engineering, ahn@uos.ac.kr, 종신회원

\* (ORCID:0000-0001-9081-2835)University of Seoul, Department of Transportation Engineering, sjlee@uos.ac.kr

논문번호 : KICS2017-12-386, Received December 18, 2017; Revised February 13, 2018; Accepted April 18, 2018

## I. 서론

단속류 환경에서 교통 흐름을 원활하게 유지하기 위해서는 신호 교차로에서 신호 제어가 적절하게 수행되어야 한다. 교통 신호는 신호 제어 시스템을 사용해서 제어되며, 신호 제어 변수로는 신호 제어 주기, 현시 순서, 현시율, 오프셋 등이 있다. 신호제어 변수의 값은 교통 상황 변수의 값을 기반으로 결정되며, 이와 관련해서 도로 비디오 이미지<sup>[1,2]</sup> 또는 도로에 설치된 무선 센서 네트워크로부터의 정보<sup>[3,4]</sup>가 사용될 수 있다. 그러나 이들 방법의 경우 많은 계산량을 필요로 하는 문제점이 있다.

최근 들어 차량 간(Vehicle to Vehicle, V2V) 통신 또는 차량과 RSU(Roadside Unit) 간(Vehicle to Infrastructure, V2I) 통신 등 V2X(Vehicle to Everything) 통신 기술이 발전하면서 많은 관심을 받고 있다<sup>[5]</sup>. 따라서 본 논문에서는 RSU 역할을 하는 교통 신호 제어기와 차량 간 통신을 이용해서 교통 신호 제어기가 차량에 대한 정보를 수집할 수 있게 하며, 이를 기반으로 교통 상황 변수 중 하나인 차량 대기행렬<sup>[6,7]</sup> 길이를 도로 섹터(sector) 기반으로 효과적으로 구하는 기법을 제안한다.

본 논문의 구성은 다음과 같다. 2절에서는 관련 연구에 대해서 기술하며, 3절에서는 제안 기법인 섹터 기반의 V2I 통신을 활용한 차량 대기행렬 길이를 구하는 방법에 대해서 설명한다. 4절에서는 제안 기법의 성능 분석을 위한 시뮬레이션 환경 및 결과에 대해서 다루며, 5절에서 결론을 맺는다.

## II. 관련 연구

### 2.1 차량 대기행렬

차량 대기행렬에 대한 정의를 먼저 살펴보면 다음과 같다.

“Highway Capacity Manual”<sup>[8]</sup>에서는 차량 대기행렬을 정지 신호에 의해 정지하고 있는 차량과 정지하고 있는 차량 뒤에 따라오는 일정 속도 이하의 차량들을 차량 대기행렬로 정의하였다.

[7]에서는 신호 제어로 정지한 차량의 대기행렬을 정지 대기행렬(standing queue), 정지 대기행렬의 영향을 받아 속도가 일정 속도 이하로 감소한 차량의 대기행렬을 이동 대기행렬(moving queue)로 정의하였다 (그림 1 참조). 등가 정지 대기행렬(equivalent standing queue)은 정지 대기행렬과 이동 대기행렬의 추가 차량을 모두 포함하는 대기행렬을 의미한다. 본

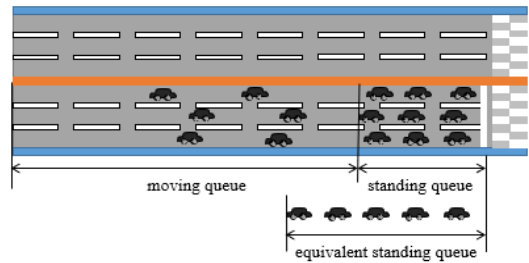


그림 1. 이동 대기행렬, 정지 대기행렬, 등가 정지 대기행렬  
Fig. 1. Standing Queue, moving queue, equivalent standing queue

논문에서의 차량 대기행렬은 [7]의 등가 정지 대기행렬을 의미한다.

### 2.2 차량 감지 기법

차량 대기행렬을 구하기 위해서는 먼저 차량을 감지할 필요가 있으며, 이와 관련해서 도로의 비디오 이미지 정보를 이용하거나 도로에 설치된 센서 네트워크를 통해 차량을 감지한 정보를 이용하는 방법이 있다.

비디오 이미지 데이터로부터 차량을 감지하는 대표적인 방법으로는 프레임 간 비교를 통한 차량 인식 방법이 있다<sup>[1,2]</sup>. 이 방법에서는 차량이 없는 이미지 배경 프레임을 기준으로 차량이 있는 이미지 프레임과 픽셀을 비교해서 객체를 감지한 후 이 객체가 차량인지 검증한다. 이 경우 실시간으로 이미지를 분석하기 때문에 고성능의 컴퓨팅 능력을 필요로 한다.

또 다른 차량 감지 방법으로는 센서를 이용해서 일반 사물과 차량을 구분하는 방법이 있다<sup>[3,4]</sup>. 그러나 이 경우 차량을 구분하기 위해서 교차로마다 센서를 설치하고 유지해야 하는 오버헤드가 있다.

### 2.3 구간 통행 시간 정보 기반 차량 대기행렬 추정 기법

일정 구간동안 수집한 정보를 이용하여 차량 좌표를 추정한 후 대기행렬 길이를 구하는 방법이 제안되었다<sup>[8]</sup>. 이 방법에서는 두 대의 RSU가 한 차량에 대해서 감지한 뒤, 차량을 감지한 시간차를 이용하여 차량의 좌표를 추정한 후 차량 대기행렬 길이를 추정한다.

### 2.4 프로브(Probe) 차량을 활용한 차량 대기행렬 추정 기법

프로브 차량이 교통 신호제어기와 V2I 통신을 통해 정보 교환을 함으로써 차량 대기행렬 길이를 구하는 연구가 진행되었다<sup>[9,10]</sup>. 프로브 차량을 활용하는

방법의 경우 수집된 정보가 부정확하기 때문에 부정확한 정보로부터 효과적으로 차량 대기행렬 길이를 구하기 위한 방법에 대해서 주로 연구되었다.

### III. V2I 통신을 활용한 차량 대기행렬 추정 기법

본 논문에서는 차량과 신호 제어기 간의 V2I 통신 기술을 활용해서 차량 감지를 하고, 이를 기반으로 차량 대기행렬의 길이를 구하는 방법을 제안한다. V2I 통신 기술을 사용하는 경우 교차로마다 센서 네트워크를 설치할 필요가 없으며, 실시간으로 비디오 이미지를 처리하기 위한 고성능의 컴퓨팅 능력을 필요로 하지도 않는다. 본 논문에서는 프로브 차량을 활용하는 방법과 달리 모든 차량이 V2I 통신 기능을 갖추고 있다고 가정하며, 이러한 환경의 문제점인 V2I 통신 오버헤드를 줄이면서도 차량 대기행렬 길이를 효과적으로 추정할 수 있는 방법을 제안한다.

#### 3.1 연구 환경

본 논문에서는 단일 교차로에서 두 인접 교차로 간 도로 구간에서 이동 방향이 같은 차량들을 대상으로 차량 대기행렬의 길이를 구한다. 차량들은 단일 교차로에서 교통 신호제어기의 신호에 맞춰 출발, 정지, 운행을 한다. 차량 대기행렬의 길이를 구하는 시점은 교통 신호제어기에서 정지 신호를 보낸 직후이다.

통신 환경은 차량 간 통신 표준인 IEEE WAVE(Wireless Access in Vehicular Environments) 표준<sup>[11]</sup>을 따른다. 물리 계층은 IEEE 802.11p<sup>[12]</sup> 표준을 기반으로 하며, MAC 계층은 IEEE 802.11p의 백 오프 경쟁 윈도우 기반의 경쟁 해결 방식을 대신해서 본 논문에서 제안하는 거리에 따른 전송 우선순위 방식을 사용한다. 차량과 교통 신호제어기 간에는 WSM(Wave Short Message) 형식에 맞춰 정보 교환이 이루어진다.

#### 3.2 교통 신호 제어기로의 차량 정보 메시지 전송 시점

본 논문에서는 차량이 교통 신호제어기로 전송하는 메시지를 차량 정보(Vehicle Information, VI) 메시지라고 하며, VI 메시지에는 차량 ID, 이동 방향, 위치, 속도에 대한 정보가 포함된다. 신호등이 적색(정지) 신호이고 주행 속도가 급격하게 감소하면 차량은 VI 메시지를 교통 신호제어기로 전송한다. 차량의 VI 메시지 전송 시점은 차량 위치에 따라 결정된다. 평균 차량 길이와 평균 차간 거리를 기반으로 해당 차량이

정지선으로부터 몇 번째 차량인지 정한 후 그에 따라 전송 시점을 정한다. 즉, 정지선으로부터의 거리에 비례해서 전송 시점을 늦춘다(멀리 떨어져 있을수록 늦게 전송한다). 이렇게 함으로써 VI 메시지 전송에 따른 충돌 가능성을 낮출 수 있다.

예를 들어, 그림 2에서 교통 신호제어기가 정지 신호를 전송한 시점이  $t$ 이고, 단위 지연시간이  $\tau$ 이고, 평균 차량 길이와 평균 차간 거리의 합이 3m인 경우, 정지선으로부터 5m 떨어져 있으면 2번째 차량인 것으로 인지하고  $(t+1 \times \tau)$  시점에 VI 메시지를 전송하며, 정지선으로부터 10m 떨어져 있으면 4번째 차량인 것으로 인지하고  $(t+3 \times \tau)$  시점에 VI 메시지를 전송한다.

GPS 기반 위치 정보의 부정확성 등으로 인해 정확한 전송 시점을 결정하는 것이 어려울 수 있으며, 이로 인해 전송되는 VI 메시지 수가 영향을 받을 수 있다. 그러나 차량 대기행렬 길이를 추정하는 데 있어서 오차를 발생시키는 요인인 위치 정보의 부정확성은 문제가 될 수 있으며, 이러한 위치 정보의 부정확성은 프로브 차량을 이용한 방식뿐만 아니라 모든 GPS 기반 위치 정보를 사용하는 방식들에서 문제가 된다. GPS 기반 위치 정보의 부정확성 문제는 별개의 큰 이슈로서, 본 논문에서는 이 부분에 대해서 다루지 않는다.

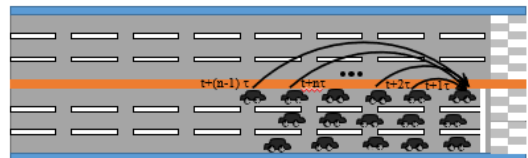


그림 2. 거리 기반 vehicle information 메시지 전송 충돌 감소  
Fig. 2. Collision reduction of vehicle information message transmissions based on distance

#### 3.3 섹터 기반의 차량 정보 메시지 전송 방법

정지선으로부터의 거리에 따라 전송 시점을 늦추더라도 동시에 2대 이상의 차량이 VI 메시지를 전송할 수 있다. 따라서 VI 메시지의 충돌 가능성을 줄이기 위해서 두 인접한 교차로 사이의 도로 구간을 일정 간격의 섹터들로 나눈 후, 섹터에 속한 차량에서만 VI 메시지를 전송하게 한다. 즉, 대기행렬에 속한 모든 차량이 VI 메시지를 전송하는 것이 아니라 대기행렬의 일부분인 섹터에 속한 차량만이 VI 메시지를 전송하도록 함으로써 충돌 가능성을 낮춘다. 차량은 도로

시작점의 위치정보와 자신의 위치정보, 섹터 너비, 섹터 간 거리에 대한 정보를 갖고 있다. 이를 통해 차량은 자신이 섹터에 속해 있는지와 어느 섹터에 속해 있는지를 알 수 있다.

다수의 차량이 동일 섹터에 존재하는 경우 먼저 도착한 차량이 교통 신호제어기에 VI 메시지를 전송하면, 이 메시지를 수신한 같은 섹터에 속하게 되는 다른 차량들은 VI 메시지를 전송하지 않는다. 즉, 차량 대기행렬에 속한 차량이 어떤 섹터에 속하게 되고 해당 섹터 내 다른 차량으로부터 VI 메시지를 수신한 적이 없으면 VI 메시지를 브로드캐스트하며, 그렇지 않으면 VI 메시지를 송신하지 않는다. 따라서 한 섹터당 한 차량만 VI 메시지를 전송하게 된다. 즉, 섹터 수만큼의 VI 메시지가 교통 신호제어기로 전송되며, 따라서 메시지 전송에 따른 충돌 가능성이 줄어들게 된다. 그림 3은 섹터 기반의 VI 메시지 전송 환경을 보여준다. 빗금으로 표시된 영역이 섹터이며, 섹터의 너비는  $w$ 이고 섹터 간 거리는  $d$ 이다.

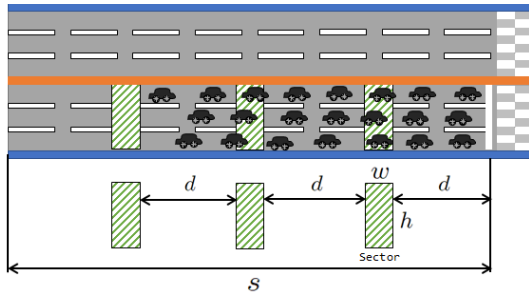


그림 3. 너비  $w$ , 섹터 간 거리  $d$ 인 섹터들  
Fig. 3. Sectors with width  $w$  and inter-sector distance  $d$

### 3.4 섹터 결정

두 인접 교차로 사이 도로 구간의 섹터 수, 섹터 간 거리에 따라 차량 대기행렬 길이의 정확도가 영향을 받는다. 섹터 수가 클수록 정확도가 높아질 가능성은 높지만, 전송되는 VI 메시지 수가 증가하며 그에 따른 메시지 충돌 가능성도 함께 증가한다. 최적의 섹터를 정하기 위해 본 논문에서는 섹터 간 거리와 섹터 크기에 따른 정확도와 네트워크 트래픽의 상관관계에 대해서 알아본다.

먼저 섹터의 개수( $noSec$ )는 교차로 간 거리( $s$ )와 섹터의 너비( $w$ ), 섹터 간 거리( $d$ )에 의해 정해진다. 섹터는 정지선으로부터 섹터간 거리와 섹터 너비의 합만큼 떨어진 곳에 차례대로 배치된다 (그림 3 참조).

$$noSec = s / (w + d) \quad (1)$$

### 3.5 차량 대기행렬 길이 계산

차량 대기행렬 길이는 차량으로부터 수신한 VI 메시지를 기반으로 교통 신호제어기에서 계산된다. VI 메시지는 녹색(직진)신호를 주기 전까지 수집하며 차량의 위치 정보로부터 차량 대기행렬의 길이를 구한다. 차량 대기행렬 길이( $estimatedVehicleLenQ$ )는 최대 대기행렬 길이( $maxLenQ$ )와 최소 대기행렬 길이( $minLenQ$ )의 평균값으로부터 구해진다.

$$maxLenQ = MAX_{i,j} (\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}) + 2d \quad (2)$$

$$minLenQ = MAX_{i,j} (\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}) + d \quad (3)$$

$$estimatedLenQ = \frac{maxLenQ + minLenQ}{2} \quad (4)$$

$$estimatedVehicleLenQ = \frac{estimatedLenQ}{VehicleLen + VehicleDist} \quad (5)$$

여기서  $i$ 와  $j$ 는 RSU인 교통 신호제어기로 VI 메시지를 전송한 차량이다.  $x_i$ 와  $y_i$ 는 차량  $i$ 의 위치의  $x$  좌표와  $y$  좌표 값이다.  $VehicleLen$ 은 차량의 평균 길이이며,  $VehicleDist$ 는 두 인접 차량 간 평균 거리이다. 수식 (2)~(4)는 교통 신호제어기가 2개 이상의 VI 메시지를 수신했을 때만 적용 가능하며, VI 메시지를 1개만 수신한 경우는 다음 수식을 사용해서 차량 대기행렬의 길이를 추정한다. 이때 VI 메시지를 전송한 차량의 위치는  $(x_s, y_s)$ 이며, 도로 구간의 시작 위치는  $(x_s, y_s)$ 이다.

$$estimatedLenQ = \sqrt{(x_i - x_s)^2 + (y_i - y_s)^2} + \frac{d}{2} \quad (6)$$

$$estimatedVehicleLenQ = \frac{estimatedLenQ}{VehicleLen + VehicleDist} \quad (7)$$

그림 4는 섹터 기반으로 대기행렬 길이 ( $estimatedLenQ$ )를 구하는 예를 보여준다. 첫 번째, 두 번째, 세 번째 섹터에 속한 차량들인  $a, b, c$ 로부터 VI 메시지를 수신한 RSU는 이들 간 거리 중 최댓값인  $\sqrt{(x_a - x_c)^2 + (y_a - y_c)^2}$  에  $d$ 를 더해서  $minLenQ$ 를 계산하고  $\sqrt{(x_a - x_c)^2 + (y_a - y_c)^2}$  에  $2d$ 를 더해서  $maxLenQ$ 를 계산한 후  $minLenQ$ 와  $maxLenQ$ 의 평균값인  $estimatedLenQ$ 를 계산한다. 이와 같이 VI 메시지를 전송한 차량 간 거리와  $d$ 를 기반으로  $estimatedLenQ$ 를 계산하는 이유는 교통 신호제어

기가 교차로마다 존재하지 않는 상황에 대비해서 해당 도로 구간의 정지선을 기준으로  $estimatedLenQ$ 를 계산하기 때문이다. 단, 정지선과 가장 가까운 첫 번째 섹터보다 대기행렬이 짧은 경우에는  $estimatedLenQ$ 를  $\frac{d}{2}$ 로 한다.

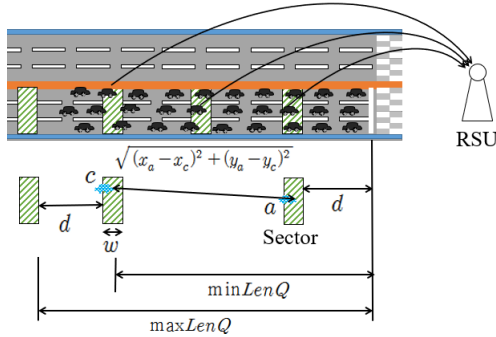


그림 4. 너비  $w$ , 섹터 간 거리  $d$ 인 섹터들을 기반으로 한 대기행렬 길이 추정  
Fig. 4. Queue length estimation based on sectors with width  $w$  and inter-sector distance  $d$

#### IV. 성능 평가

제안 방식의 성능을 평가하기 위해서 교통 시뮬레이터 SUMO<sup>[13]</sup>와 네트워크 시뮬레이터 OMNet++<sup>[14]</sup>를 기반으로 한 차량 네트워크 시뮬레이션 프레임워크인 Veins<sup>[15]</sup>를 이용해서 시뮬레이션을 수행하였다. 제안 방식(그림 5~7의 Proposed)을, 정지 신호에 의해 감속한 모든 차량이 교통 신호제어기로 VI 메시지를 전송하는 방식(그림 5~7의 General)과 비교하였다. 이들 방식 모두에 있어서 정지 대기행렬에 의해 속도가 1m/s 이하로 감소한 차량이 이동 대기행렬에 속하는 것으로 하였다. 단일 교차로 환경에서 두 인접 교차로 사이의 도로 길이는 500m로 설정하였으며, 제안 방식의 경우 섹터 너비  $w$ 와 섹터 간 거리  $d$ 를 모두 10m로 설정하였다.

차량 포화도(degree of saturation, 주어진 도로 구간에 최대 존재할 수 있는 차량 수 대비 배치된 차량 수의 백분율)에 따라 측정된 차량 대기행렬 길이와

표 1. 차량 포화도에 따른 시나리오  
Table 1. Scenarios according to the degree of saturation

시나리오	평가 내용
1	차량 포화도 30%(16대)에서 성능 평가
2	차량 포화도 50%(34대)에서 성능 평가
3	차량 포화도 70%(42대)에서 성능 평가

제안 방식에 의해 추정된 차량 대기행렬 길이를 표 1의 차량 포화도 시나리오에 따라 분석하였다.

차량 대기행렬 길이 값의 정확도를 평가하기 위해서 산술 평균(Arithmetic Mean, AM)과 평균 절대 비율 오차(Mean Absolute Percent Error, MAPE)를 사용하였다.  $A_i$ 가  $i$ 번째 측정값,  $F_i$ 가  $i$ 번째 추정값,  $n$ 이 총 측정 횟수라고 할 때, 평균 절대 편차는 예측 오차 척도로서 각 측정값과 추정값의 오차 평균이다.

$$AM = \frac{\sum |A_i - F_i|}{n} \quad (8)$$

평균 절대 비율 오차는 수요의 크기가 달라질 때 유용한 예측 오차 측정 방법이다.

$$MAPE(\%) = \frac{\sum \left| \frac{A_i - F_i}{A_i} \right|}{n} \times 100 \quad (9)$$

표 2~4에서 알 수 있듯이 차량 포화도가 높을수록 오차율이 작아지며, 보다 정확하게 차량 대기행렬 길이를 추정할 수 있다.

표 2. 시나리오 1 : 차량 포화도 30% ( $w=10m, d=10m$ )  
Table 2. Scenario 1 : degree of saturation 30% ( $w=10m, d=10m$ )

횟수 (시드)	주기 당 차량 수	대기행렬 길이(m)		AM (m)	MAPE (%)
		측정값	추정값		
1(53542)	16	113.3	110.8	2.5	2.21
2(45732)		113.3	110.8		
3(47853)		113.3	110.8		
4(33434)		113.3	110.8		
5(23425)		113.3	110.8		

표 3. 시나리오 2 : 차량 포화도 50% ( $w=10m, d=10m$ )  
Table 3. Scenario 2 : degree of saturation 50% ( $w=10m, d=10m$ )

횟수 (시드)	주기 당 차량 수	대기행렬 길이(m)		AM (m)	MAPE (%)
		측정값	추정값		
1(53542)	34	253.9	247.4	6.92	2.73
2(45732)		253.9	246.9		
3(47853)		253.9	247.0		
4(33434)		254.3	247.0		
5(23425)		253.9	247.0		

표 4. 시나리오 3 : 차량 포화도 70% ( $w=10m, d=10m$ )  
 Table 4. Scenario 3 : degree of saturation 70% ( $w=10m, d=10m$ )

횟수 (시드)	주기 당 차량 수	대기행렬 길이(m)		AM (m)	MAPE (%)
		측정값	추정값		
1(53542)	42	352.7	345.4	3.3	0.94
2(45732)		352.2	345.4		
3(47853)		352.2	351.1		
4(33434)		352.2	351.2		
5(23425)		351.5	351.2		

그림 5~7은 각 방식에 의해 전송된 VI 메시지 별 종단간 지연시간(end-to-end delay)에 대한 측정치를 나타낸 것이다. 이들 그래프에서 알 수 있듯이 제안 방식의 경우 General 방식에 비해 적은 수의 메시지가 전송되었다. 총 전송 메시지 수 측면에서, 시나리오 1의 경우, 제안 방식은 메시지 9개, General 방식은 메시지 26개를 전송하였다. 시나리오 2의 경우, 제안 방식은 메시지 17개, General 방식은 메시지 50개를 전송하였다. 시나리오 3의 경우, 제안 방식은 메시지 26개, General 방식은 메시지 68개를 전송하였다. 즉, 제안 방식이 General 방식에 비해 1/3 정도의 메시지 전송을 한 것을 알 수 있다. 평균 VI 메시지 전송 지연 시간 측면에서, 시나리오 1의 경우 제안 방식은 0.238ms, General 방식의 경우 0.239ms이고, 시나리오 2의 경우 제안 방식과 General 방식 모두 0.24ms이며, 시나리오 3의 경우 제안 방식은 0.241ms, General 방식은 0.241ms로 거의 비슷했다.

섹터 너비  $w$ 와 섹터 간 거리  $d$ 가 제안 방식의 성능에 미치는 영향을 확인하기 위해 시나리오 2 환경에서 다양한  $w$ 와  $d$  값에 대한 성능 분석을 수행하였다. 표 5와 6을 통해  $w$ 와  $d$  값이 증가함에 따라 차량

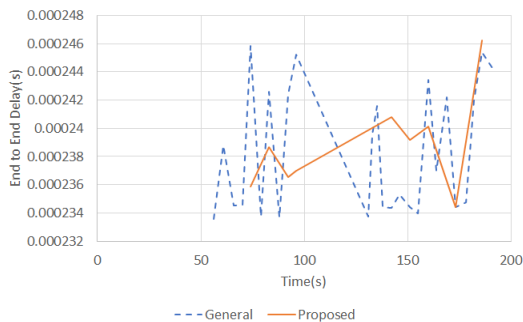


그림 5. 시나리오 1 : 종단간 지연 대 시뮬레이션 시간 ( $w=10m, d=10m$ )  
 Fig. 5. Scenario 1 : End-to-end delay vs. simulation time ( $w=10m, d=10m$ )

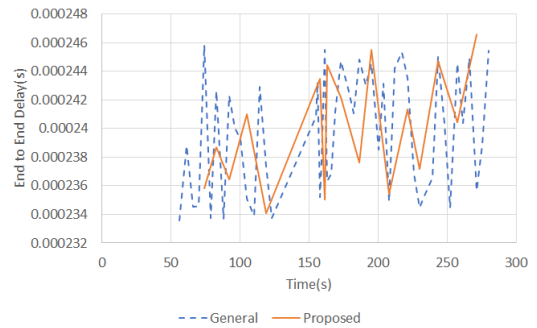


그림 6. 시나리오 2 : 종단간 지연 대 시뮬레이션 시간 ( $w=10m, d=10m$ )  
 Fig. 6. Scenario 2 : End-to-end delay vs. simulation time ( $w=10m, d=10m$ )

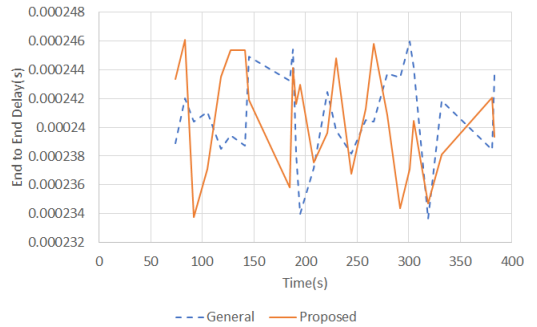


그림 7. 시나리오 3 : 종단간 지연 대 시뮬레이션 시간 ( $w=10m, d=10m$ )  
 Fig. 7. Scenario 3 : End-to-end delay vs. simulation time ( $w=10m, d=10m$ )

대기행렬 길이 추정값의 정확도가 전반적으로 낮아짐을 확인할 수 있다.

표 5. 시나리오 2 : 차량 포화도 50% ( $w=10m$ )  
 Table 5. Scenario 2 : degree of saturation 50% ( $w=10m$ )

횟수	대기행렬 길이(m)			
	측정값	추정값		
		$d=10m$	$d=20m$	$d=30m$
1(53542)	253.9	247.4	247.1	247.2
2(45732)	253.9	246.9	246.7	246.7
3(47853)	253.9	247	247	247
4(33434)	254.3	247	246.9	247
5(23425)	253.9	247	246.9	247
AM(m)		6.92	7.06	7
MAPE(%)		2.73	2.78	2.76

표 6. 시나리오 2 : 차량 포화도 50% ( $d=10m$ )  
 Table 6. Scenario 2 : degree of saturation 50% ( $d=10m$ )

횟수	대기행렬 길이(m)			
	측정값	추정값		
		$w=10m$	$w=20m$	$w=30m$
1(53542)	253.9	247.4	247.2	243.4
2(45732)	253.9	246.9	246.7	242.9
3(47853)	253.9	247	245.8	242
4(33434)	254.3	247	246	242.2
5(23425)	253.9	247	246	243
AM(m)		6.92	7.64	11.28
MAPE(%)		2.73	3	4.44

### V. 결론

차량 대기행렬 길이를 계산하기 위해 기존에는 도로 비디오 이미지를 처리하거나 또는 도로에 설치된 차량 감지용 센서 네트워크를 이용하였다. 그러나 이들 방법의 경우 실시간 이미지 처리를 위한 고성능 컴퓨팅이나 센서 설치 오버헤드가 수반되는 문제가 있다. 본 논문에서는 교통 신호제어기와 차량 간의 V2I 통신 기술을 활용해서 차량 대기행렬 길이를 구하는 방법을 제안했으며, 특히 도로 구간을 섹터로 나눠서 섹터에 속한 차량만 교통 신호제어기로 전송하도록 함으로써 메시지 전송량을 감소시켰다. 제안 방식의 성능 평가를 위해서 SUMO와 OMNet++ 기반의 Veins 시뮬레이션 플랫폼 상에서 시뮬레이션을 수행했으며, 차량 포화도 측면에서 제안 방식의 차량 대기행렬 길이 추정치에 대한 정확도를 평가하였다. 또한 차량 대기행렬에 속한 모든 차량들이 교통 신호제어기로 메시지를 전송하는 기법과 본 제안 방식을 전송 메시지 수 측면에서 성능 비교를 수행함으로써 제안 방식의 성능 우수성을 입증하였다.

### References

[1] A. Lipton, H. Fujiiyoshi, and R. Patil, "Moving target classification and tracking from real-time video," *IEEE WACV*, pp. 8-14, 1998.  
 [2] N. Seenouvang, U. Watchareeruetai, and C. Nuthong, "A computer vision based vehicle detection and counting system," *Int. Conf.*

*Knowledge and Smart Technology (KST)*, 2016.  
 [3] M. Duarte and Y. H. Hu, "Vehicle classification in distributed sensor networks," *J. Parall. and Distrib. Comput.*, 2004.  
 [4] N. Ahmed, M. Rutten, T. Bessell, S. S. Kanhere, N. Gordon, and S. Jha, "Detection and tracking using particle-filter-based wireless sensor networks," *IEEE Trans. Mob. Comput.*, 2010.  
 [5] S. K. Bohi and P. M. Khilar, "Vehicular communication: A survey," *IET Networks*, 2013.  
 [6] Highway Capacity Manual, *Transportation Research Board of the National Academies of Science*, US, 2000.  
 [7] H. X. Liu, W. Ma, X. Wu, and H. Hu, "Real-time estimation of arterial travel time under congested condition," *Transportmetrica*, 2009.  
 [8] M. H. Lee, Y. C. Kim, and Y. Jeong, "Queue length based real-time traffic signal control methodology using sectional travel time information," *J. Korea Inst. Intell. Transp. Syst.*, 2014.  
 [9] K. Tiaprasert, Y. Zhang, B. Wang, and X. Zeng, "Queue length estimation using connected vehicle technology for adaptive signal control," *IEEE Trans. Intell. Transp. Syst.*, 2015.  
 [10] G. Comert, "Simple analytical models for estimating the queue lengths from probe Vvchicles at traffic signals," *Transp. Res. Part B: Methodological*, 2013.  
 [11] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," in *Proc. IEEE*, 2011.  
 [12] IEEE 802.11p Part 11: *Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, Amendment 6: Wireless access in vehicular environments*, 2010.  
 [13] D. Krajzewicz, M. Behrisch, L. Bieker and J. Erdmann, SUMO Home Page(2017), Retrieved Nov. 3, 2017, <http://sumo.sourceforge.net>  
 [14] A. Varga, OMNeT++ Home Page(2017),

Retrieved Nov. 3, 2017, from <http://www.omnetpp.org>.

- [15] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Trans. Mob. Computing*, 2011.

**최 종 화 (Jonghwa Choi)**



2016년 8월 : 서울시립대학교 컴  
퓨터과학부 졸업  
2016년 9월~현재 : 서울시립대  
학교 컴퓨터과학과 석사과정  
<관심분야> 인터넷 프로토콜,  
VANET, MANET, WSN

**안 상 현 (Sanghyun Ahn)**



1986년 2월 : 서울대학교 컴퓨  
터공학과 졸업  
1988년 2월 : 서울대학교 컴퓨  
터공학과 석사  
1993년 12월 : 미네소타대학교  
컴퓨터학과 박사  
1988년 1월~1989년 8월 : (주)  
데이콤 MHS개발본부 사원  
1994년 3월~1998년 2월 : 세종대학교 컴퓨터학과 교수  
1998년 3월~현재 : 서울시립대학교 컴퓨터과학부 교수  
<관심분야> 인터넷 프로토콜, VANET, MANET,  
WSN

**이 승 재 (Seungjae Lee)**



1988년 2월 : 인하대학교 산업  
공학과 졸업  
1990년 2월 : 서울대학교 환경  
대학원 도시계획 석사  
1995년 4월 : 런던대학 토목 및  
환경공학과 박사  
1995년 5월~1995년 10월 : 런  
던대학 통계학과 연구교수  
1995년 10월~1996년 9월 : 한국교통연구원 책임연구원  
1996년 10월~현재 : 서울시립대학교 교통공학과 교수  
<관심분야> 교통계획, 교통모형, ITS