

# VANET 환경에서 효과적인 데이터 배포를 위한 RSU 협업 스케줄링

## Cooperative RSU Scheduling for Efficient Data Dissemination in VANET Environments

복경수, 홍승완, 차재홍, 임종태, 유재수  
충북대학교 정보통신공학과

Kyoung Soo Bok(ksbok@chungbuk.ac.kr), Seung Wan Hong(hswwhite@nate.com),  
Jae Hog Cha(kakasi1527@nate.com), Jong Tae Lim(jtlim@chungbuk.ac.kr),  
Jaesoo Yoo(yjs@chungbuk.ac.kr)

### 요약

최근 VANET에 대한 연구가 활발하게 진행됨에 따라 보다 편리하고 안정적인 서비스를 차량에 제공할 수 있다. 본 논문에서는 다수 RSU들을 우선망으로 연결하고 RSU들 사이의 협업을 통해 데이터를 전송하는 새로운 스케줄링 기법을 제안한다. 제안하는 기법은 다중 RSU 협업 전략을 통해 안전 데이터와 비안전 데이터를 효율적인 전송하며 데드라인 실패율 및 평균 응답 시간을 감소시킨다. 안전 데이터가 발생할 경우 이전 RSU에서 데이터를 미리 전송하고 데드라인, 수신율을 고려하여 우선 순위를 부여한다. 비안전 데이터는 사용자의 요청에 따라 처리되는 요구 기반 데이터이기 때문에 RSU에서 부하가 발생할 경우 데드라인 실패율이 방지하기 위한 기법을 제공한다. 제안하는 기법의 우수성을 입증하기 위해 차량 수와 속도를 변화시키면서 성능 평가를 수행한 결과 데드라인 실패율과 평균 응답 시간 측면에서 제안하는 기법이 기존 기법보다 모두 우수함을 입증하였다.

■ 중심어 : | VANET | RSU | 스케줄링 | 비안전데이터 | 안전데이터 |

### Abstract

Recently, as researches on VANET have actively progressed, the convenient and reliable services can be provided to the vehicles. In this paper, we propose the collaborative RSU scheduling scheme using multiple RSUs which are connected through a wired backbone. The proposed scheme efficiently disseminates both unsafety data and safety data, and reduces the deadline miss rate and the average response time. When safety data is occurred, it is send to previous RSU in advance and is scheduled by deadline and reception rate. We provide the prevention technology for unsafety data when the workload is occurred in RSU because it is on-demand data processed by user requests. It is shown through performance evaluation that the proposed scheme outperforms the existing scheme in terms of the deadline miss rate and the average response time according to the number of vehicles and the velocities of vehicles.

■ keyword : | VANET | RSU | Scheduling | Safety Data | Unsafety Data |

\* 이 논문은 미래창조과학부 및 정보통신산업진흥원의 IT융합 고급인력과정 지원사업 (NIPA-2013-H0401-13-2011)과 2012

년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임( 2012R1A1A2041898)

접수일자 : 2013년 07월 23일

심사완료일 : 2013년 09월 04일

수정일자 : 2013년 09월 04일

교신저자 : 유재수, e-mail : yjs@chungbuk.ac.kr

## I. 서론

VANET(Vehicular Ad-hoc NETWORKS)은 이동하는 차량들 사이에 무선 통신을 통해 애드혹 네트워크를 구축하고 교통 정보, 파일 공유, 위치 주변 정보 등을 제공한다[1]. VANET은 차량들 사이에 통신 및 차량과 인프라 간의 통신을 통해 운전자의 요구에 적합한 정보를 제공하여 이동 중에도 서비스 만족도를 향상시킬 수 있다. 또한, 차량 충돌 및 도로 교통 정체로 인한 비용을 감소시킬 수 있어 편안하고 안전한 운행을 할 수 있도록 한다[1][2][11][12]. VANET 통신 방식은 V2I(Vehicle To Infrastructure)와 V2V(Vehicle To Vehicle)로 구분된다. V2V 기반의 통신 방식은 차량들 사이에 애드혹 형태의 네트워크를 구축하고 차량들 사이에 데이터를 직접 전송하기 때문에 다수 운전자의 요구에 적합한 정보를 제공하기 위해 많은 통신 비용이 소요된다. 또한, 차량의 빠른 이동으로 인해 안정적으로 정보 전송을 보장하지 못한다. 최근 다수의 운전자에게 필요한 정보를 효율적으로 제공하기 위해 V2I를 이용한 통신 방식들이 제안되었다. V2I에서는 다수의 운전자가 요구한 정보 또는 특정 정보를 다수의 사용자에게 전달하기 위해 RSU(Road Side Units)를 사용한다. RSU는 도로 주변에 설치되어 차량의 요청을 수신하거나 차량에 필요한 정보를 전송하는 기지국 역할을 수행한다. 그러나 RSU의 통신 범위가 제한적이기 때문에 다수의 차량에게 정보를 배포하기 위한 전송 기법이 요구된다. 또한, 제한된 대역폭에서 다수 차량의 요청을 서비스하기 위한 프로토콜이 필요하다[3][4].

RSU는 안전/비안전 데이터를 다수의 차량에게 전송하기 위해 다운로드 서비스를 제공하며 제한된 통신 환경 내에서 차량이 요청한 정보를 최대한 제공하기 위해 데이터 전송 스케줄링을 수행한다[5-7]. 단일 RSU 스케줄링 기법은 RSU의 통신 반경이 작고 RSU와 차량 간의 연결 시간이 짧아 차량들에게 안정적인 데이터를 제공하는데 한계가 있다. 특히 바쁜 시간대나 차량의 밀집도가 높은 핫스팟 도로 구간에서는 데드라인 실패율이 증가한다. 이러한 문제점을 해결하기 위해 다중 RSU 협업을 통한 데이터 전송 기법들이 연구되고 있

다. multiple RSU 기법은 기본적인 RSU의 스케줄링은 DSIN 기법을 사용하여 우선 순위를 할당한다[9]. RSU 상호 간의 위치를 고려하여 낮은 부하를 갖는 RSU에게 요청의 일부를 전송한다. MPO에서는 차량들의 이동을 예측하고 RSU 간의 부하 균형 맞춤과 동시에 다수의 차량에게 많은 데이터를 제공하도록 한다[10]. 차량의 이동을 예측하기 위해 Manhattan mobility model의 이동성 모델을 사용한다. 차량들의 이동을 예측하고 특정 RSU에서 실패할 요청은 다른 RSU에게 전달한다.

기존 다중 RSU 협업 전략 기법에서는 안전/비안전 데이터의 구분이 없이 데이터 전송 스케줄링을 수행한다. 또한, 교통 상황에 따라 차량의 속도와 RSU의 배치가 다르지만 차량의 이동과 무관하게 RSU가 모든 차량의 요청을 수신하여 정확하게 처리할 수 있다고 가정하고 있다. 따라서 실제 차량의 이동 속도 및 방향에 따라 RSU 내에서 데이터 수신율이 저하될 수 있다. 이러한 문제점을 해결하기 위해 본 논문에서는 다수 RSU들을 우선망으로 연결하고 RSU 협업을 통해 안전/비안전 데이터를 모두 전송할 수 새로운 스케줄링 기법을 제안한다. 비안전 데이터는 사용자의 요청에 따라 처리되는 요구 기반 데이터이기 때문에 RSU에서 부하가 발생할 경우 데드라인 실패율이 방지하기 위한 기법을 제공한다. 안전 데이터에서는 수신율을 고려하여 데이터를 받지 못한 차량들에게도 고려하여 더 많은 차량들이 서비스를 받을 수 있게 한다.

본 논문의 나머지 구성은 다음과 같다. II장에서는 기존에 제안된 RSU 협업 전략에 대해 기술하고 III장에서는 제안하는 RSU 협업 전략에 대해 기술한다. IV장에서는 제안하는 기법의 우수성 입증하기 위해 성능 평가를 수행한 결과를 기술하고 V장에서는 본 논문의 결론을 기술한다.

## II. 관련연구

VANET 환경에서 V2I 기반으로 다수 사용자의 요청을 처리하기 위해서는 사용자를 수신하고 데드라인 실패율을 최소화하기 위한 스케줄링 기법에 필요하다.

DSIN은 서비스율을 향상시키기 위해 데드라인, 요청 수, 데이터의 크기를 고려하여 스케줄링한다[7]. 서비스율이란 전체 요청 중에서 데드라인 내에 처리된 요청 수로써 서비스율이 높을수록 좋은 스케줄링이다. DSIN은 스케줄링은 데드라인과 데이터의 크기가 작고 요청 수가 높은 데이터의 우선 순위가 높다. Zhang 등은 데드라인, 데이터 크기, 요청 수를 상호 비교하여 배포할 데이터를 선별하는  $D*S$ ,  $D*S/N$ ,  $D*S/R$  스케줄링을 제안하였다[8].  $D*S$ 는 데드라인과 데이터 크기를 고려한다. 동일한 데드라인을 가지는 데이터가 존재한다고 데이터의 크기가 작은 것을 먼저 처리하고 데이터의 크기가 동일하면 데드라인이 작은 것을 처리한다.  $D*S$ 는 데드라인과 데이터의 크기가 작을수록 우선 순위가 높다. 다운로드를 최적화하기 위해  $N$ (요청 수)을 추가하고 데드라인, 데이터 크기, 요청 수를 상호 비교하여 다운로드할 때  $D*S/N$  스케줄링을 사용하고 업로딩할 때에는  $D*S/R$  스케줄링을 사용한다. 이를 통해 서비스 비율 및 데이터 품질 사이의 상충 관계를 연구하였다.

단일 RSU 스케줄링 기법은 RSU의 통신 반경이 작고 RSU와 차량 간의 연결 시간이 짧아 차량들에게 안정적인 데이터를 제공하는데 한계가 있다. 이러한 한계점을 극복하기 위해 다수 RSU를 백본망을 통해 연결하고 RSU 간에 서로 통신하는 다중 RSU 협업 전략의 기법이 제안되었다. 일반적으로 다중 RSU 환경에서는 RSU의 부하 상태를 파악하여 요청 메시지를 다른 RSU에게 전송한다[9]. multiple RSU 기법은 통신 범위를 벗어나기 전에 사용자의 요청을 처리하기 위해 기본적인 RSU 스케줄링 전략으로 DSIN 기법을 사용하여 우선 순위를 할당한다. RSU의 부하가 증가하여 다수 차량의 요청을 처리할 수 없을 경우 RSU 상호 간의 위치를 고려하여 낮은 부하를 갖는 RSU에게 요청의 일부를 전송한다. 만약 RSU1에서 부하가 발생하고 RSU2에서는 부하가 발생하지 않았다고 가정할 때 RSU1에서 수신한 요청을 RSU2에게 전송하려면 차량은 RSU1에서 RSU2 방향으로 이동해야 하며 차량의 요청에 대한 허용지연시간이 차량의 이동 시간과 데이터를 제공받는 시간보다 커야 한다.

MPO에서는 차량들의 움직임을 예측하고 RSU 간의 부하 균형 맞춤과 동시에 다수의 차량에게 많은 데이터를 제공한다[10]. 차량의 이동을 예측하기 위해 Manhattan mobility model의 이동성 모델을 사용한다. 차량들의 이동을 예측하고 특정 RSU에서 실패할 요청은 다른 RSU에게 양도한다. Manhattan mobility model의 이동성 모델은 기본적으로 격자 도로 토폴로지를 사용하고 차량은 지도의 수평 및 수직 거리의 격자를 따라 이동한다. 노드 운동의 선택에 확률 접근 방식을 사용한다. 차량의 이동성에 같은 방향은 0.5, 왼쪽 또는 오른쪽은 0.25의 확률을 갖는다. 4가지 시나리오가 있으며 RSU에서 부하 때문에 처리 못하는 요청이 발생하면 Manhattan mobility model로 차량의 이동을 예측한다. 차량의 이동 경로를 예측하여 다음 RSU의 `RSU_available_time`을 검사한다. 이때, `RSU_available_time`과 `duration`의 합이 데드라인보다 커야 한다. `RSU_available_time`이란 차량이 다음 RSU로 이동할 때 데이터를 제공받을 수 있는 시간이며 `duration`은 차량이 데이터를 다운로드 받는 시간이다.

다중 RSU 협업 전략 기법은 단일 RSU 모델을 문제점을 해결하여 다수의 요청을 처리할 수 있다. 또한, 요청이 많이 생성되는 핫스팟에 대해 효과적으로 처리가 가능하다. 하지만 기존의 다중 RSU 협업 전략은 안전 데이터와 비안전 데이터를 구분하지 않고 스케줄링을 수행한다. 이로 인해 특정 RSU에 사용자의 요청이 증가할 경우 데드라인 실패가 증가한다. 제안하는 기법에서는 안전 데이터와 비안전 데이터의 특성을 고려하여 차별화된 스케줄링 기법에 제공한다. 비안전 데이터는 사용자마다 요청하는 데이터의 유형을 상이하고 데이터 크기가 다양하기 때문에 요청수, 데드라인, 데이터 크기를 고려하여 스케줄링한다. 또한, 교통 상황에 따라 차량의 속도와 RSU의 배치가 다르기 때문에 차량의 정확한 데이터 수신 시간을 계산하여 처리하도록 한다. 안전 데이터는 사용자의 요청이 없이도 전송해야 할 데이터이기 때문에 데드라인과 데이터 수신율을 고려하여 스케줄링한다. 안전 데이터는 데드라인 실패가 발생할 경우 인적, 물적 피해가 발생할 수 있기 때문에 안전 데이터가 발생한 RSU 방향으로 이동한 차량에게 안전

데이터를 미리 제공할 수 있도록 한다.

### III. 제안하는 RUS 협력 전략

#### 1. 제안하는 시스템 구조

VANET 환경에서 RSU 스케줄링은 RSU 통신 범위 내에서 다수의 차량에게 안전 데이터와 비안전 데이터를 전송하는데 목적이 있다. 이를 위해 기존 RSU 스케줄링에서는 요청 수, 데이터 크기, 데드라인 등을 고려하였다. 제안하는 기법은 데드라인 실패율과 응답 시간을 감소시키기 위해 안전 데이터와 비안전 데이터에 특성에 따라 각기 다른 스케줄링 정책을 제공한다. [그림 1]은 제안하는 스케줄링 기법의 구조를 나타낸 것이다. 제안하는 기법은 사용자의 요청을 수신하기 위한 요청 큐(request queue), 차량에게 전송할 데이터를 선별하는 스케줄러(scheduler), 네트워크를 통해 전송할 데이터를 저장하는 아이템 큐(item queue)로 구성되어 있다. RSU는 서버로부터 전달된 위치 주변 데이터와 차량들로부터 수신한 데이터들을 저장 관리하고 차량들의 요청을 수집해야 한다. 이를 위해 요청 큐는 두 개의 업로드 큐(upload queue)와 하나의 다운로드 큐(download queue)로 구성되어 있다. 다운로드 큐는 사용자의 요청을 수신하여 저장한 큐로 요구 기반 서비스를 제공하기 위해 사용한다. 업로드 큐는 차량, 이웃한

RSU, 서버 등으로부터 전송된 데이터를 저장하는 큐로 안전/비안전 데이터를 각각 저장한다.

안전 데이터는 생명 또는 안전과 직접적으로 관련되어 있기 때문에 비안전 데이터를 우선적으로 RSU에 저장하고 차량에게 전송해야 한다. 이에 반해 비안전 데이터는 파일 공유, 위치 주변 정보와 같이 인포테인먼트 특성을 가지고 있기 때문에 데드라인 실패가 최대한 발생하지 않도록 최선의 데이터를 제공해야 한다. 따라서 제안하는 기법에서는 안전/비안전 데이터를 별도로 저장 관리하기 위해 두 개의 입력 큐로 구성되어 있다. 요청 큐에 저장된 정보는 스케줄러에 의해 우선 순위가 부여되고 우선 순위에 따라 아이템 큐에 배치된다. 아이템 큐에 저장된 데이터는 무선 통신을 통해 차량들에게 브로드캐스팅된다.

#### 2. RSU 협업 스케줄링 고려사항

단일 RSU 스케줄링의 한계점을 해결하기 위해 다수 RSU들을 유선망으로 연결하고 다수 차량에게 데이터를 제공하는 다중 RSU 기반의 협업 스케줄링 기법을 제안한다. 안전 데이터를 스케줄링 하기 위해 데드라인, 수신율을 고려한다. 데드라인은 차량이 안전 데이터와 관련된 위치에 도달하는 시간 즉, 사고 또는 위험 지역에 도달하기 전까지의 시간이다. 수신율은 하나의 RSU 내에서 안전 데이터를 수신한 차량의 비율을 나타낸 것이다. 수신율  $rr_{i,j}$ 은 동일한 데이터에 대해 많은 차량이

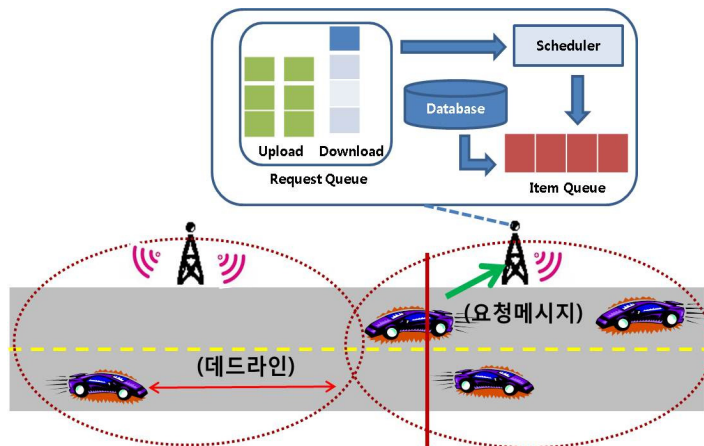


그림 1. 시스템 모델

수신하지 못한 데이터를 우선적으로 전송하기 위해 사용되며 (식 1)과 같다. 이때,  $c_i$ 는 RSU  $i$ 의 통신 반경 내에 존재하는 차량의 수이고  $r_j$ 는 안전 데이터  $j$ 를 수신한 차량의 수를 나타낸다.

$$rr_{ij} = \frac{r_j}{c_i} \quad (1)$$

안전 데이터가 사고나 인명 피해와 직접적으로 관련된 데이터이기 때문에 이를 수신하지 못할 경우 인적 물적 피해를 받을 수 있다. 즉, 특정 RSU에 부하가 발생하여 특정 안전 데이터를 전송하지 못할 경우 많은 피해가 발생할 수 있다. 이러한 문제를 해결하기 위해 제안하는 기법에서는 특정 RSU 내에서 안전 데이터가 발생할 경우 안전 데이터가 발생한 RSU 방향으로 이동하는 차량이 존재하는 RSU에 안전 데이터를 전송한다. 즉, 안전 데이터가 발생한 위치 주변에 존재하는 RSU에 안전 데이터를 전송하여 차량들이 안전 데이터를 미리 수신할 수 있도록 한다. 예를 들어, [그림 2]와 같이 RSU(D)에서 안전 데이터가 발생할 경우 RSU(C), RSU(A)에 RSU(D)에 발생한 안전 데이터를 전송한다. 이를 통해 차량의 이동을 분산시키거나 실제 사고가 발생한 RSU에서 부하가 발생하여도 안전 데이터를 미리 수신하였기 때문에 사고에 대처할 수 있다.

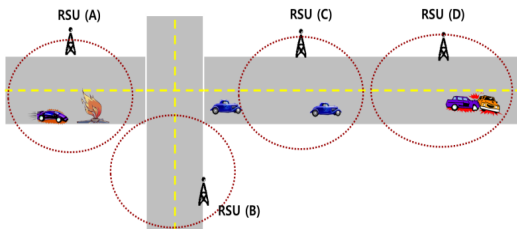


그림 2. 다중 RSU에서 안전 데이터 전송

일반적으로 비안전 데이터는 사용자의 요청에 의해 전달되는 요구기반 데이터이다. 따라서 특정 RSU 내에 부하가 증가할 경우 다수 사용자의 요청을 처리하지 못해 데드라인 실패율이 증가된다. 이를 처리하기 위해 제안하는 기법에서는 부하가 발생한 RSU에 차량들이

요구한 요청의 일부를 이웃한 RSU에 전달하여 부하 분산을 수행한다. 현재 RSU에서 부하가 발생하여 차량의 요청을 처리할 수 없을 경우 차량이 이동하는 다음 RSU에게 요청 메시지를 전송한다. RSU는 다수 차량의 요청을 처리하기 위해 우선 순위를 부여한다. 만약 특정 차량의 요청에 대한 우선 순위가 낮을 경우 요청한 데이터를 계속적으로 수신하지 못하는 기아 현상이 발생할 수 있다. 이러한 문제점을 해결하기 위해 제안하는 기법에서는  $dc$ 를 사용한다. 요청 메시지를 다음 RSU에게 전송하면 요청 메시지의  $dc$  값을 증가시킨다. 이를 통해 다음 RSU 내에서 우선 순위를 증가시킬 수 있다. 이때, RSU 간의 메시지 전송은  $n$ 번까지만 허용한다.

[그림 3]은 RSU에서 비안전 데이터를 전송하는 과정을 나타낸 것이다. RSU(A)에서 비안전 데이터에 대한 요청이 증가하여 RSU(A)에 부하가 발생하였다고 가정하자. RSU(A)는 RSU의 부하로 인해 사용자의 요청을 처리하지 못해 데드라인 실패가 발생할 것으로 예상되면 사용자의 요청을 차량이 이동하는 방향에 존재하는 RSU(B)로 전달한다. 이때, RSU(A)로부터 수신한 차량의 요청에 대해  $dc$  값을 증가시킨다. RSU(B)는 RSU(A)로부터 전달된 요청을 수신하고 데드라인 실패가 발생하지 않으면서 요청된 데이터를 전송할 수 있는지 판별한다. 만약 RSU(B)가 RSU(A)로부터 전달된 요청을 처리하지 못할 경우 RSU(B)는 RSU(A)의 요청을 RSU(C)로 재전송한다. 부하가 발생한 RSU의 요청을 다른 RSU에게 전송하는 수를 2번까지만 허용한다고 가정하면 RSU(A)에서 전달된 요청은 RSU(C)까지만 전달된다. RSU(C)에서도 차량의 요청을 처리하지 못할 경우 해당 메시지를 제거한다.

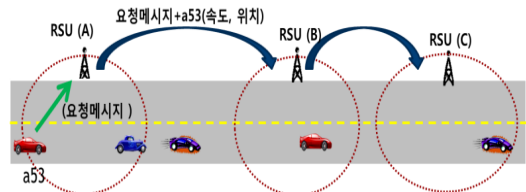


그림 3. 다중 RSU에서 비안전 데이터 전송

기존 다중 RSU 기법에서는 특정 RSU에서 차량의 요청을 처리하지 못해 다음 RSU로 전달될 경우 차량이 다음 RSU에 도착할 시간을 고려하지 않고 있다. 즉, 다음 RSU에 차량의 요청이 전달될 경우 RSU 통신 반경에 도착할 시간을 고려하지 않고 요청 메시지를 전송하기 위한 스케줄링을 수행한다. 그러나 RSU 배치가 불특정하고 교통 상황에 따라 차량의 이동 속도나 방향이 다양할 수 있다. 따라서 제안하는 기법에서는 부하 발생으로 특정 RSU가 다음 RSU로 차량의 요청을 전송할 때 다음 RSU의 통신 범위에 들어오는 시점을 파악하기 위해 요청 메시지에 차량의 속도와 위치를 함께 전송한다. 이를 통해 요청 메시지를 수신한 RSU는 (식 2)을 이용하여 차량이 RSU의 통신 범위에 들어오는 시간을 파악하고 스케줄링 시점을 결정한다. (식 2)에서  $v_{ij}$ 는 RSU  $i$ 에 존재하는 차량  $j$ 의 속도,  $d_{kj}$ 는 차량  $j$ 의 위치에서 RSU  $k$  까지의 거리,  $at_{kj}$ 는 RSU  $k$ 의 통신 범위 내에 차량  $j$ 가 도착할 예상시간을 나타낸다.

$$at_{kj} = \frac{d_{kj}}{v_{ij}} \quad (2)$$

### 3. 다중 RSU 협업 스케줄링 절차

다중 RSU 협업을 통한 데이터 전송을 위해 안전 데이터는 데드라인 실패가 없이 반드시 전송되어야 한다. 또한, 안전 데이터는 차량이 안전 데이터가 발생한 RSU로 이동하는 차량이 존재하는 RSU에 데이터를 미리 전송하기 때문에 데드라인 뿐만 아니라 수신율을 고려하여 전송한다. 일반적으로 비안전 데이터는 차량의 요구에 따라 전송해야 할 데이터 크기가 매우 다양할 수 있다. 이에 반해 안전 데이터는 상태 정보를 전송하는 경우가 대부분이기 때문에 전송할 데이터 크기에 차이가 발생하지 않는다. 따라서 제안하는 기법에서 안전 데이터에 우선 순위를 부여하기 위해 안전 데이터의 크기는 고려하지 않는다.

[그림 4]는 다중 RSU 협업 전략을 통해 안전 데이터를 차량에게 전송하기 위한 우선 순위를 부여하는 과정을 나타낸 것이다. [그림 4]의 (a)와 같이 RSU(A)에서 안전 데이터 {s2, s1, s3}가 발생하고 RSU(D)에서는 안

전 데이터 {s6, s9}가 발생하여 이를 RSU(A)에 전달하였다고 가정하자. 먼저 RSU(A)에서는 차량으로 전송할 안전 데이터의 우선 순위를 부여하기 위해 [그림 4]의 (b)와 같이 데드라인, 수신율을 기준으로 정렬을 수행한다. 우선 데드라인 실패를 최소화하기 위해 데드라인을 기준으로 정렬을 수행하고 각 데이터에 대한 수신율을 기준으로 정렬을 수행한다. {s6, s9}는 수신율이 {s2, s1, s3}보다 작지만 데드라인이 크기 때문에 *item\_queue*에 여유 공간이 있을 때 추가한다. 차량에게 전송할 데이터를 저장하기 위한 큐 *item\_queue*의 크기를 50라고 할 때 RSU(A)에서 발생한 데이터 {s2, s1, s3}를 *item\_queue*에 저장한 후에도 여유 공간이 존재한다. 따라서 RSU(D)에서 전달된 s6도 추가적으로 *item\_queue*에 저장되어 차량에게 전송한다. 최종적으로 전송할 데이터는 {s2, s1, s3, s6}으로써 *item\_queue*에 선별한 데이터를 저장하여 전송한다.

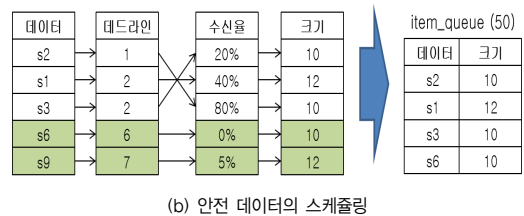
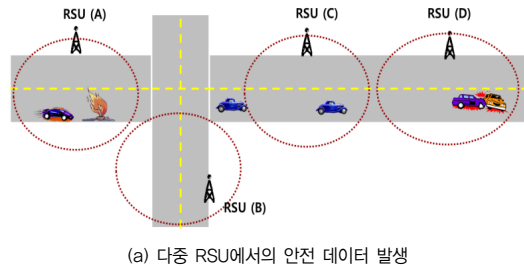


그림 4. 다중 RSU에서의 안전 데이터 스케줄링

비안전 데이터는 최대한 많은 차량에게 데드라인 실패가 없이 데이터를 전송하기 위한 우선 순위를 부여한다. 또한, 다중 RSU 협업 전략을 통해 부하가 발생한 RSU의 요청을 다른 RSU에서 처리할 수 있도록 한다. RSU는 다수의 차량으로부터 다양한 요청이 수신하기 때문에 비안전 데이터에 대한 요청 수가 다양하다. 요

청이 많은 데이터를 차량에게 전송할 경우 다수의 차량이 요청한 정보를 수신할 수 있기 때문에 전체적인 서비스 만족도를 향상시킬 수 있다. 따라서 제안하는 기법에서는 데드라인과 함께 요청 수를 고려하여 우선 순위를 부여한다. 비안전 데이터는 인포테인먼트적인 특성을 가지고 있기 때문에 전송할 데이터 크기가 다양할 수 있다. 만약 동일한 데드라인과 요청 수를 갖는 데이터가 존재한다고 할 때 고정된 대역폭 내에서 데이터 크기가 작은 데이터를 전송할 경우 많은 데이터를 차량에게 전송할 수 있다. 따라서 데이터 크기는 전송할 데이터에 대한 우선 순위를 부여하는데 중요한 기준이 된다. 제안하는 기법에서는 비안전 데이터를 전송할 우선 순위를 부여하기 위해 데드라인, 요청 수, 데이터 크기를 고려한다.

비안전 데이터는 요청을 전달한 시점에서 시간이 지날수록 의미 없는 데이터가 될 수 있다. 예를 들어, 특정 차량이 위치 주변 정보를 요청하였다고 할 때 RSU의 부하 및 데이터 우선 순위로 인해 현재 요청을 먼 거리에 있는 RSU 내에서 처리된다면 위치 주변 정보에 대한 의미가 상실될 수 있다. 또한, 요청 수에 따라 우선 순위가 부여된다고 할 때 요청이 거의 없는 데이터는 우선 순위가 낮아 데이터를 전송하지 못한다. 이러한 요청은 다른 RSU에 요청을 전달하여도 계속적으로 서비스를 제공받지 못하는 기아 현상이 발생할 수 있다. 이러한 문제점을 해결하기 위해 제안하는 기법에서는  $dc$ 를 사용한다.  $dc$ 는 우선 순위 또는 RSU 부하로 인해 특정 RSU 내에서 요청한 서비스가 처리되지 못하고 다른 RSU 내에서 처리될 때 임의의 수만큼 요청 수를 증가시킨다. 이를 통해 처음 차량의 요청을 수신 RSU 내에서 비안전 데이터를 전송하지 못할 경우 우선 순위를 증가시킬 수 있다.

특정 RSU의 부하로 인해 비안전 데이터를 차량에게 제공하지 못한다면 차량이 이동한 방향에 존재하는 이웃한 RSU에 비안전 데이터를 요청을 전달하여 다른 RSU에서 비안전 데이터를 수신할 수 있도록 한다. 예를 들어, [그림 4]의 (a)와 같이 특정 차량이 RSU(A)에서 비안전 데이터를 요청했지만 RSU(A)의 부하로 인해 RSU(B)에 차량이 요청한 비안전 데이터를 전달하

였다고 가정하자. 만약 RSU(B)에 부하가 발생하지 않았다면 RSU(A)에서 전달된 요청은 RSU(B)에서 차량에게 전송된다. 그러나 RSU(B)에도 부하가 발생한다면 다음 RSU인 RSU(C)에서 차량의 요청을 전달한다. RSU 내에서 비안전 데이터를 전송하기 위한 우선 순위를 결정하기 위해 데드라인, 데이터 크기,  $dc$ 를 추가한 요청수를 고려한다. 비안전 데이터의 다중 RSU 협업 기법에서는 데드라인 실패가 발생하지 않은 데이터를 선별하여 *item\_queue*에 저장한다. 이번 전송 주기에 꼭 전송해야 할 데이터들의 크기가 *item\_queue*의 크기를 초과할 경우에는 요청 수와 데이터 크기를 고려하여 전송할 데이터를 재선정한다.

[그림 5]는 다중 RSU 협업을 통해 비안전 데이터를 배포하기 위한 스케줄링 과정을 나타낸 것이다. *item\_queue*의 크기를 100이라 가정하고 RSU(B)에서 차량들이 요청한 비안전 데이터를 {c, x, a, b, g, t, w}라고 하자. 데이터를 전송하는 주기는 6이며 {t, w}는 이전 RSU 내에서 처리되지 못하고 RSU(B)로 전달된 요청이라고 하자. 특정 RSU 내에서 요청한 결과를 수신하지 못하고 다음 RSU로 전달될 때  $dc$  값을 1씩 증가시킨다고 가정하면 t는 두 번의 RSU 내에서 처리되지 못하고 RSU(B)로 전달되어  $dc$  값을 2로 설정하고 이와 유사하게 w의  $dc$  값 1로 설정한다.  $dc$  값을 이용하여 우선 순위를 증가시키기 위해 요청 수에  $dc$  값을 합한다. 가장 먼저 데드라인 실패를 최소화하는 데드라인을 기준으로 하여 정렬을 수행하고 정렬된 값은 다시 요청 수에 따라 정렬한다. 비안전 데이터 {t, w}는 이번 전송 주기 내에 데이터를 전송하지 않아도 데드라인 실패가 발생하지 않는다. 따라서 이번 전송 주기에 전송하지 못하면 데드라인 실패가 발생하는 {a, x, g, c, b}를 우선 *item\_queue*에 저장된다. 하지만 *item\_queue*의 크기가 100이기 때문에 남은 여유 공간에는 다음 주기에 전송될 데이터를 추가적으로 저장한다. t와 w는 데드라인과 데이터 크기가 동일하지만  $dc$  값을 고려할 경우 요청 수가 큰 t를 *item\_queue*에 추가 저장한다. 최종적으로 전송할 데이터는 {a, x, g, c, b, t} 이다.



그림 5. 다중 RSU에서의 비안전 데이터 스케줄링

#### IV. 성능평가

제안하는 기법의 우수성을 보이기 위해서 Dual Core CPU 2.99GHz, 4G RAM, Windows 7 Ultimate K OS를 사용하는 컴퓨터에서 자바로 구현하여 실험 평가하였다. [표 1]은 성능 평가에 사용된 요소들이다. 차량 수와 차량 속도를 변경하면서 Multiple RSU[9], MPO[10]과 데드라인 실패율과 평균 응답 시간을 비교 평가하였다.

표 1. 성능평가 요소

| 요소     | 범위               |
|--------|------------------|
| 차량의 수  | 300 ~ 1000대      |
| 차량의 속도 | 30km/h ~ 100km/h |
| 데이터 종류 | 500개             |
| 데이터 크기 | 10 ~ 512 Kbytes  |
| RSU 수  | 16개              |

[그림 6]은 차량의 수에 따른 데드라인 실패율을 나타낸 것이다. 성능 평가 결과 제안하는 기법이 기존 기법 보다 전체적으로 우수한 성능을 보였다. 차량의 수가 300대 일 때는 7% 성능이 우수하였으며 가장 많은 1000대에서는 11% 우수한 성능을 나타내었다.

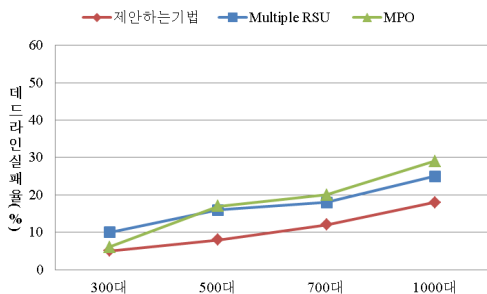


그림 6. 차량 수에 따른 데드라인 실패율

[그림 7]은 차량의 속도에 따른 데드라인 실패율을 나타낸 것이다. 차량의 속도가 느릴 때는 RSU 반경 내에 많은 차량이 있어 데드라인 실패율이 증가하고 차량의 속도가 빠를 때는 한 개의 RSU 반경 내에 겹치는 차량이 없어서 데드라인 실패율이 감소된다. 제안하는 기법은 기본 기법에 비해 평균 11% 성능 향상을 나타내었다.

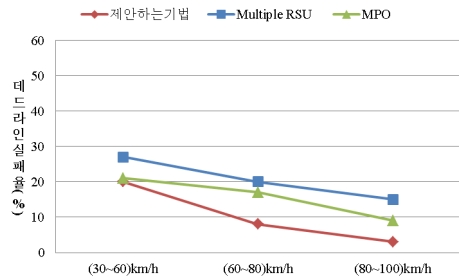


그림 7. 차량 속도에 따른 데드라인 실패율

다중 RSU 환경에서 평균 응답 시간에 대한 성능 평가를 수행하였다. [그림 8]은 다중 RSU 환경에서 차량의 수에 따른 평균 응답 시간을 나타낸 것이다. 기존 기법들은 차량의 수에 따라 평균 응답 시간이 증가 하지만 제안하는 기법에서는 비슷한 값을 유지한다.

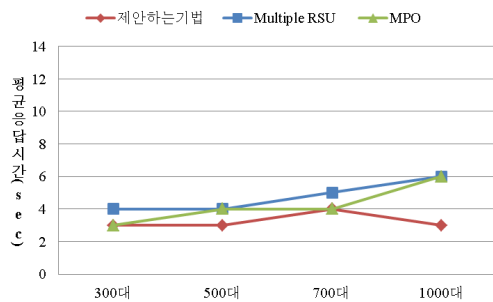


그림 8. 차량 수에 따른 평균 응답 시간

[그림 9]은 차량의 속도 변화에 따른 평균 응답 시간을 나타낸 것이다. 속도가 증가함에 따라 기존 기법과 제안하는 기법 모두 평균 응답 시간이 감소한다. 속도가 차량의 속도가 빠를수록 제안하는 기법이 기존 기법에 비해 평균적으로 12% 우수한 성능을 나타내었다.



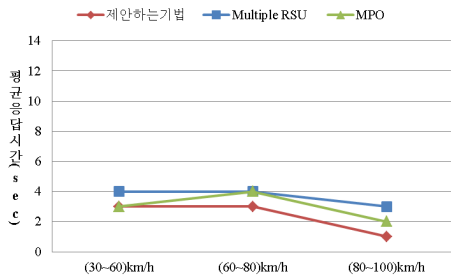


그림 9. 차량 속도에 따른 평균 응답 시간

## V. 결론

본 논문에서는 RSU들 간의 통신을 통해 안전 데이터와 비안전 데이터를 전송하기 위해 협업 스케줄링 기법을 제안하였다. 안전 데이터에 대해서는 데드라인 실패율을 감소시키기 위해 안전 데이터가 발생한 RSU 방향으로 이동하는 차량에게 데이터를 미리 전송한다. 또한, 비안전 데이터에 대해서는 RSU들 사이에 부하 분산을 수행하여 안정적인 데이터 전송이 가능하도록 하였다. 차량 수와 속도를 변화시키면서 성능 평가를 수행한 결과 데드라인 실패율과 평균 응답 시간 측면에서 제안하는 기법이 기존 기법보다 모두 우수함을 입증하였다. 차량의 이동 변화에 따라 특정 도로 상태가 매우 다양한 변화된다. 이에 따라 차량의 이동 변화를 예측하고 이를 기반으로 차량의 요청을 전송하기 위한 기법들이 필요하다. 향후 연구로 차량의 이동 변화를 예측하고 도로 상황에 따라 전송 스케줄링 동적으로 적용하기 위한 연구를 진행할 예정이다.

## 참고 문헌

[1] H. T. Cheng, H. Shan, and W. Zhuang, "Infotainment and Road Safety Service Support in Vehicular Networking: From a Communication Perspective," *Mechanical Systems and Signal Processing*, Vol.25, No.6, pp.2020-2038, 2011.

[2] P. Tomar, B. K. Caurasia, and G. S. Tomar, "State of the Art of Data Dissemination in

VANETs," *International Journal of Computer Theory and Engineering*, Vol.2, No.6, pp.957-962, 2010.

- [3] 홍승완, 임종태, 복경수, 유재수, "VANET에서 데드라인 실패를 감소시키기 위한 RSU 스케줄링 기법", *데이터베이스연구*, 제29권, 제1호, pp.37-50, 2013.04.
- [4] Y. Zhang, J. Zhao, and G. Cao, "On Scheduling Vehicle-Roadside Data Access," *Proc. International Workshop on Vehicular Ad Hoc Networks*, pp.9-18, 2007.
- [5] K. Liu and V. C. S. Lee, "RSU-based Real-time Data Access in Dynamic Vehicular Networks," *Proc. International Conference on Intelligent Transportation Systems*, pp.1051-1056, 2010.
- [6] D. Aksoy and M. J. Franklin, "R×W : A Scheduling Approach for Large-Scale On-demand Data Broadcast," *IEEE/ACM Transactions on Networking*, Vol.7, No.6, pp.846-860, 1999.
- [7] G. G. M. Nawaz Ali, E. Chan, and W. Li, "Two-Step Joint Scheduling Scheme for Road Side Units (RSUs)-Based Vehicular Ad Hoc Networks (VANETs)," *Proc. DASFAA Workshops*, pp.453-464, 2011.
- [8] Y. Zhang, J. Zhao, and G. Cao, "Service Scheduling of Vehicle-Roadside Data Access," *Mobile Networks and Applications*, Vol.15, No.1, pp.83-96, 2010.
- [9] G. G. Md Nawaz Ali and E. Chan, "Co-operative Data Access in Multiple Road Side Units (RSUs)-based Vehicular Ad Hoc Networks(VANETs)," *Proc. Australasian Telecommunication Networks and Applications Conference*, pp.1-6, 2011.
- [10] Y. Gui and E. Chan, "A Motion Prediction Based Cooperative Scheduling Scheme for Vehicle-Roadside Data Access," *Proc. International Conference on Networking and*

Information Technology, pp.195-204, 2011.

[11] 조준모, “운전자의 안전을 위한 도심지역 자동차 애드혹 통신망의 뇌파전송 성능평가”, 한국콘텐츠학회논문지, 제11권, 제6호, pp.26-32, 2011.

[12] 이은주, 이권익, 좌정우, 양두영, “차량간 통신을 위한 위치 정보 기반의 AODV 라우팅 프로토콜”, 한국콘텐츠학회논문지, 제8권, 제3호, pp.47-54, 2008.

저 자 소 개

북 경 수(Kyung Soo Bok)

정회원



- 1998년 2월 : 충북대학교 수학과 (이학사)
- 2000년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2005년 2월 : 충북대학교 정보통신공학과(공학박사)

- 2005년 3월 ~ 2008년 2월 : 한국과학기술원 정보전자연구소 Postdoc
- 2008년 3월 ~ 2011년 2월 : 가인정보기술 연구소 차장
- 2011년 3월 ~ 현재 : 충북대학교 전자정보대학 정보통신공학부 초빙부교수

<관심분야> : 데이터베이스 시스템, 자료저장시스템, 이동객체 데이터베이스, RFID 및 센서네트워크, 모바일 P2P 네트워크, 빅데이터 등

홍 승 완(Seung Wan Hong)

준회원



- 2011년 8월 : 충북대학교 컴퓨터공학과(공학사)
- 2011년 9월 ~ 현재 : 충북대학교 정보통신공학과 석사과정

<관심분야> : 데이터베이스시스템, 센서네트워크, VANET, 클라우드컴퓨팅 등

차 채 홍(Jae Hog Cha)

준회원



- 2012년 2월 : 충북대학교 정보통신공학과(공학사)
- 2012년 3월 ~ 현재 : 충북대학교 정보통신공학과 석사과정

<관심분야> : 데이터베이스시스템, 소셜 네트워크, 빅데이터, 모바일 P2P 네트워크 등

임 종 태(Jong Tae Lim)

정회원



- 2009년 2월 : 충북대학교 정보통신공학과(공학사)
- 2011년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2011년 3월 ~ 현재 : 충북대학교 정보통신공학과 박사과정

<관심분야> : 데이터베이스 시스템, 시공간 데이터베이스, 위치기반 서비스, 모바일 P2P 네트워크, 빅데이터 등

유 재 수(Jaesoo Yoo)

종신회원



- 1989년 2월 : 전북대학교컴퓨터공학과(공학사)
- 1991년 2월 : 한국과학기술원 전산학과(공학석사)
- 1995년 2월 : 한국과학기술원 전산학과(공학박사)

- 1995년 3월 ~ 1996년 8월 : 목포대학교 전산통계학과 전임강사
- 1996년 8월 ~ 현재 : 충북대학교 정보통신공학부 및 컴퓨터정보통신연구소 교수
- 2009년 3월 ~ 2010년 2월 : 캘리포니아주립대학교 방문교수

<관심분야> : 데이터베이스시스템, 빅데이터, 센서네트워크 및 RFID, 소셜네트워크서비스, 분산객체컴퓨팅, 바이오인포매틱스 등