

A Digital Twin-based Approach for VANET Simulation in Real Urban Environments

Jonghyeon Choe*, Youngboo Kim**, Sangdae Kim***

*M. S. Student, Dept. of Computer Engineering, Kongju National University, Cheonan, Korea

**Post-doctoral Researcher, Dept. of Computer Engineering, Kongju National University, Cheonan, Korea

***Professor, Dept. of Medical Information Technology Engineering, Soonchunhyang University, Asan, Korea

[Abstract]

In this paper, we conducted a thorough investigation of existing simulators for running simulations of Vehicular Adhoc Networks (VANET) in realistic road environments, such as digital twins. After careful consideration, we chose a simulator that combines OSM (OpenStreetMap), SUMO (Simulation of Urban MObility), and OMNeT++ due to its open-source nature and efficient performance. Using this integrated simulator, we carried out VANET simulations in both simple virtual road environments and realistic road environments. Our findings revealed significant differences in VANET performance between the two types of environments, emphasizing the need to consider realistic road and traffic environments for reliable VANET operation. Furthermore, our simulations demonstrated significant performance variability, with performance degradation observed as vehicle density decreased and dynamic changes in network topology increased. These results underscore the importance of digital twin-based approaches in VANET research, highlighting the need to simulate real-world road and traffic conditions rather than relying on simple virtual road environments.

▶ **Key words:** Digital twin, VANET, Simulation, OMNeT++, OSM, SUMO

[요 약]

본 논문에서는 디지털 트윈과 같은 실제 도로 환경에서 차량용 애드혹 네트워크(VANET)의 시뮬레이션을 실행하기 위한 기존 시뮬레이터를 면밀한 조사가 수행되었다. 신중한 검토 후에, 오픈소스로 공개되어 있을 뿐만 아니라 효율적인 성능을 가지는 OSM(OpenStreetMap), SUMO(Simulation of Urban MObility), OMNeT++를 결합한 시뮬레이터를 선택되었다. 또한, 이 통합 시뮬레이터를 사용하여 단순한 가상 도로 환경과 실제 도로 환경 모두에서 VANET 시뮬레이션을 수행하였고, 그 결과 두 유형의 환경 간에 VANET 성능에 상당한 차이가 있음을 확인했으며, 안정적인 VANET 운영을 위해 현실적인 도로 및 교통 환경을 고려해야 할 필요가 있음을 확인하였다. 추가적으로, 또 다른 시뮬레이션 결과에서는 차량 밀도가 감소하고 네트워크 토폴로지의 동적 변화가 증가함에 따라 성능 저하가 관찰되는 등 상당한 성능 변동성이 있는 것으로 나타나기도 하였다. 이러한 결과들을 통해, VANET을 실제 환경에서 운용하기 위해서는, 단순한 가상 도로 환경에 의존하지 않아야 하며, VANET 연구에서 디지털 트윈 기반 접근 방식을 통해 실제 도로 및 교통 상황을 시뮬레이션해야 할 필요성을 확인할 수 있다.

▶ **주제어:** 디지털 트윈, 차량 네트워크, 시뮬레이션, OMNeT++, OSM, SUMO

- First Author: Jonghyeon Choe, Corresponding Author: Sangdae Kim
- *Jonghyeon Choe (whdgus1323@smail.kongju.ac.kr), Dept. of Computer Engineering, Kongju National University
- **Youngboo Kim (Obookim@kongju.ac.kr), Dept. of Computer Engineering, Kongju National University
- ***Sangdae Kim (sdkim.mie@sch.ac.kr), Dept. of Medical Information Technology Engineering, Soonchunhyang University
- Received: 2024. 07. 16, Revised: 2024. 07. 31, Accepted: 2024. 08. 07.

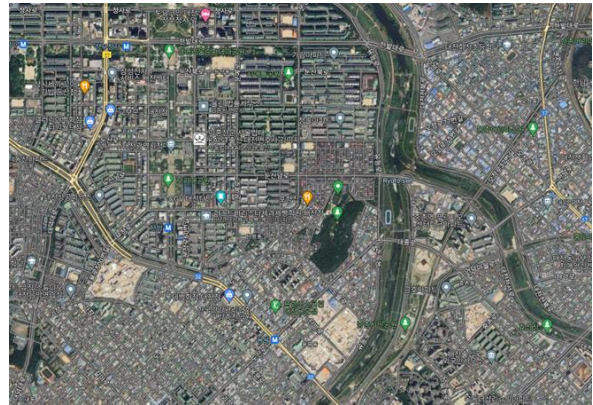
I. Introduction

디지털 트윈 기반 접근 방식은 현실 세계의 물리적 환경을 가상 공간에 정밀하게 재현하여 다양한 시나리오를 테스트하고 분석할 수 있게 해준다[1]. VANET (Vehicular Adhoc Networks)의 성능 평가를 위해 디지털 트윈을 활용하면, 현실적인 도시 환경과 교통 상황을 보다 정확하게 모델링할 수 있다. 이는 VANET의 실제 운영 조건을 반영하여 보다 신뢰성 있는 결과를 도출할 수 있게 한다. 예를 들어, 교통 혼잡, 도로 상황, 차량의 동적 이동 패턴 등 다양한 변수들을 포함하여 시뮬레이션을 수행함으로써, VANET의 성능을 종합적으로 평가할 수 있다. 이러한 접근 방식은 VANET의 설계 및 최적화에 중요한 데이터를 제공하며, 새로운 기술과 프로토콜의 실험 및 검증에도 유용할 수 있다[2].

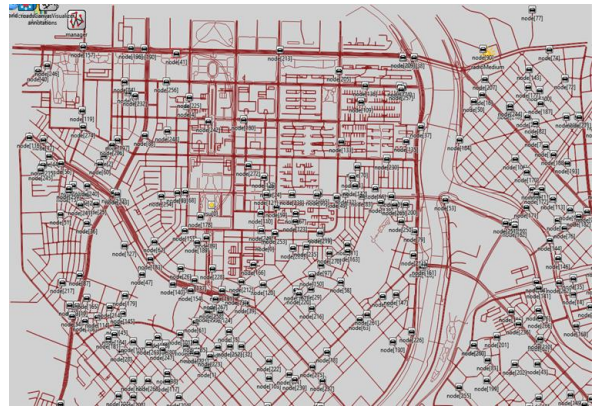
이를 위해, 본 논문에서는 통신 시스템을 위한 시뮬레이터로 OMNeT++ [3]를 사용하였으며, 현실 세계의 도로에서 차량의 이동 상황을 시뮬레이션 하기 위해서는 OSM (OpenStreetMap) [4]과 SUMO (Simulation of Urban MObility) [5]와 을 사용하였다. OSM을 사용하면, Fig. 1(a)의 대전시 둔산동 지역 같은 특정 지역에 대한 도로 환경을 VANET 시뮬레이션에 반영할 수 있고, SUMO를 이용하면, 교통 흐름, 차량 혼잡도, 대기 시간, 이동 시간 등과 같은 다양한 실제 교통 상황을 모델링할 수 있기 때문이다. 따라서, 이들을 OMNeT++와 통합하여 사용하면, Fig. 1(b)와 같이 보다 현실적인 상황에서 VANET의 성능을 분석하는 것이 가능하다.

또한, 본 연구에서는 OSM, SUMO 및 OMNeT++이 통합된 시뮬레이션 환경에서 VANET 성능을 평가하였다. 시뮬레이션 결과, Packet Delivery Ratio 측면에서 VANET 성능의 변동성이 클 뿐만 아니라, 차량의 밀집도가 낮고, 네트워크 토폴로지의 동적 변화가 더 심화될수록 성능이 악화되는 것을 확인하였다. 이러한 결과는 VANET의 실제 운영 환경을 반영한 시뮬레이션의 필요성을 강조하며, 디지털 트윈 기반 접근 방식의 유효성을 뒷받침할 수 있다.

본 논문의 이후의 구성은 다음과 같다. II절에서는 VANET (Vehicular Ad-Hoc Network)을 시뮬레이션하기 위한 다양한 소프트웨어 애플리케이션을 비교 분석한다. III절에서는 본 논문에서 사용된 OSM, SUMO 및 OMNeT++이 통합된 시뮬레이션 환경에 대해 설명한다. IV절에서는 시뮬레이션 환경 설정 및 실험 결과에 대해 분석한다. 마지막으로 V절에서 결론을 맺는다.



(a) Satellite map of Dusan-dong and neighbouring areas in Daejeon



(b) VANET simulation environment based on OSM and SUMO

Fig. 1. Satellite Map and VANET Simulation Environment for Dusan-dong and Nearby Areas in Daejeon City

II. Preliminary

차량 간 통신을 가능하게 하는 네트워크 유형인 VANET을 시뮬레이션하기 위한 다양한 소프트웨어 애플리케이션을 사용할 수 있다. 이러한 소프트웨어 도구는 다양한 시나리오에서 VANET의 성능과 동작을 평가하는 데 중요한 역할을 한다. 이 절에서는 VANET 시뮬레이션 연구에 주로 사용되는 소프트웨어들에 대해 살펴보았으며, 그 결과를 Table. 2에 요약하였다.

OMNeT++는 네트워크 시뮬레이션 수행을 위해 특별히 설계된 모듈식 C++ 시뮬레이션 라이브러리를 활용하여, 다양한 네트워크 프로토콜과 시나리오를 쉽게 모델링할 수 있는 유연하고 강력한 환경을 제공한다. 반면, SUMO는 대규모 도로망을 효과적으로 시뮬레이션할 수 있는 기능을 보여주는 오픈 소스 교통 시뮬레이터이다. 이를 통해, 연구자들은 교통 흐름, 혼잡 패턴 및 도시 이동성의 기타 중요한 측면을 분석할 수 있다. 또한, 오픈 소스 프레임

Table 1. Comparison of various simulator for VANET Simulation in Real Urban Environments

Simulator	Advantages	Disadvantages
OMNeT++ with SUMO	<ul style="list-style-type: none"> Provides a flexible and powerful environment for easy modeling of various network protocols and scenarios Veins framework integrates with SUMO for realistic simulation Scalability with modular C++ simulation library 	<ul style="list-style-type: none"> Technical complexity in the integration process
NS-3 with SUMO	<ul style="list-style-type: none"> Supports extensive network simulation Capable of performing various research with a discrete event simulator Realistic vehicle movement modeling through integration with SUMO 	<ul style="list-style-type: none"> Technical complexity and performance issues during integration
NetSim with SUMO	<ul style="list-style-type: none"> User-friendly interface and advanced features Suitable for simulating complex network scenarios Provides integration environment with SUMO 	<ul style="list-style-type: none"> Cost implication as a commercial tool Limited availability of related resources due to recent integration with SUMO
OPNET	<ul style="list-style-type: none"> Accurately simulating complex network environments Supports modeling of various network scenarios Offers robust customization and scripting support 	<ul style="list-style-type: none"> Cost implication as a commercial tool Lack of direct integration with SUMO
JiST/SWANS	<ul style="list-style-type: none"> Compatibility with various operating systems and programming languages due to Java-based architecture Suitability for large-scale simulations 	<ul style="list-style-type: none"> Lack of direct integration with SUMO

워크인 Veins [6]를 이용하면, OMNeT++ 과 SUMO를 통합된 환경에서 사용할 수 있고, OSM으로부터 지도 데이터를 입력 받아 현실의 도시 환경에서 VANET 시나리오를 사실적으로 시뮬레이션할 수 있다.

NS-3 [7]은 광범위한 네트워크 시뮬레이션에 대한 지원과 확장이 용이한 이산 이벤트 시뮬레이터로, 다양한 목적성과 광범위한 확장성으로 인해 VANET과 관련된 다양한 연구를 수행할 수 있다. 또한, NS-3에서도 SUMO와의 통합을 통해 VANET과 관련된 연구에서, 현실적인 차량 이동 모델링과 정확한 네트워크 시뮬레이션을 수행하여, 종합적인 VANET 분석이 가능하다. 그러나 통합 과정에서의 기술적 복잡성, 성능 문제, 데이터 일관성 유지 등과 같은 도전 과제의 극복이 요구된다 [8].

NetSim [9]은 VANET을 비롯한 다양한 네트워크에 대한 시뮬레이션을 수행하는 데 널리 사용되는 상용 이산 이벤트 시뮬레이터다. 사용자 친화적인 인터페이스와 고급 기능 덕분에 복잡한 네트워킹 시나리오를 시뮬레이션하는 데 가장 적합하고, 최근에는 SUMO와 통합된 환경을 구축하는 방법을 제공하기 시작하였다 [10]. 마찬가지로 OPNET [11]은 복잡한 네트워크 환경을 정확하게 시뮬레이션하는 기능으로 잘 알려진 또 다른 상용 도구이나, 기본적으로 SUMO와의 직접적인 통합을 제공하지 않고, 커스터마이제이션과 스크립팅을 통해 가능할 수는 있다.

JiST/SWANS [12]는 Java 기반 시뮬레이터로 VANET을 포함하는 대규모 시뮬레이션을 수행하는 데 적합하다.

다양한 운영 체제 및 프로그래밍 언어와의 호환성으로 연구자들이 다용도로 사용할 수 있습니다. 하지만, OPNET가 마찬가지로, JiST/SWANS와 SUMO와의 통합을 위해서는 추가적인 커스터마이제이션이 요구된다.

결론적으로, VANET 시뮬레이션을 위한 다른 소프트웨어와 비교하여, OSM, SUMO 그리고 OMNeT++의 통합된 환경을 구축하는 것은 상업적인 제약이 없으면서도, 현실적인 도시 환경 및 교통 환경에서 정밀한 네트워크 시뮬레이션을 수행할 수 있다는 장점이 있다.

III. System Structure

이 절에서는 Fig. 2와 같이 실제 도시 환경에서 VANET을 시뮬레이션하기 위한 시뮬레이터의 구조에 대해 설명한다. 시뮬레이터는 크게 교통 시뮬레이션 모듈(Traffic Simulation Module)과 차량 네트워크를 시뮬레이션하기 위한 네트워크 시뮬레이션 모듈(Network Simulation Module)로 구성되어 있다.

1. OSM

OSM (Open Street Map) 은 도로, 건물, 자연 지형 등의 정보를 제공하는 오픈 소스 지도 프로젝트로, 지도 및 도로 네트워크에 관한 정보를 여러 형식으로 변환할 수 있어 다양한 유형의 GIS (Geographic Information System) 도구에 활용될 수 있다. 우리의 VANET 시뮬레

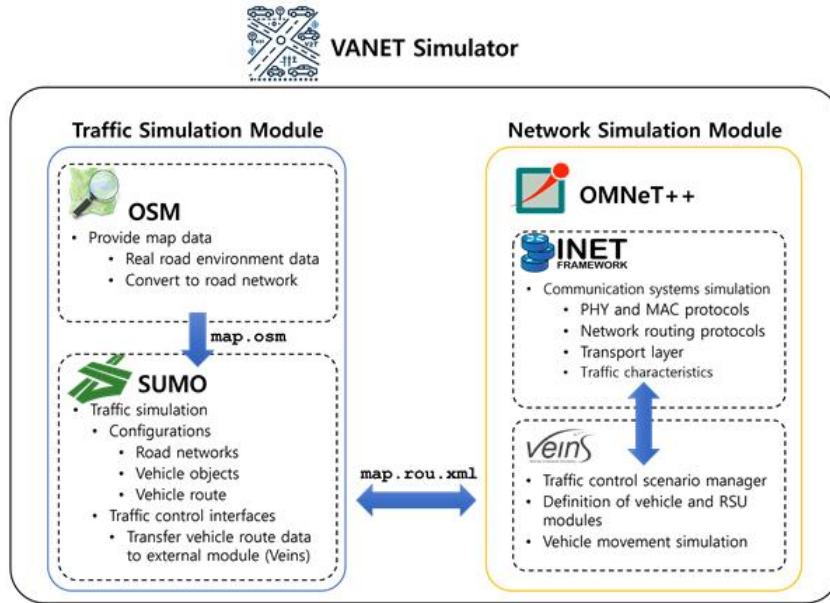


Fig. 2. System structure of VANET simulation in real urban environment

이터에서도 SUMO와의 연동을 통해 현실의 도시 환경에서 차량이 이동하는 상황을 반영하기 위해 OSM이 사용되었다.

또한, OSM에서는 사용자가 원하는 지도 상의 영역을 설정하면, 해당 영역에 관한 지리 정보가 XML 기반의 .osm 파일 형식으로 저장되며, 이 파일이 SUMO에 입력되어 현실 도로에서의 차량 이동 경로를 생성해 준다. SUMO에 입력되는 OSM 데이터의 주요 구성 요소들은 다음과 같다.

- **bounds:** 이 요소는 OSM 데이터 파일의 지리적 경계를 최소 및 최대 경도를 통해 정의한다.
- **node:** 장소, 지형지물, 교통 시설과 같은 지리적 객체의 위치를 나타내며, 고유 식별자와 위도 및 경도 정보를 속성으로 갖는다.
- **way:** 일련의 노드로 구성된 선형 요소로, 도로, 경로, 강 등의 선형적으로 표현할 수 있는 지리적 객체이다.
- **relation:** 버스 노선이나 행정 구역 같은 특정한 지리적 구조나 논리적 관계 나타내기 위해 다수의 node, way 및 다른 relation들을 그룹화 하는 요소이다.
- **tag:** node, way 및 다른 relation 등의 이름이나 유형 같은 추가 정보를 정의할 수 있는 요소이다.

2. SUMO

SUMO (Simulation of Urban MObility) 는 도로 네트워크와 차량 경로 생성을 위해 사용되는 교통 시뮬레이션 도구로, OSM과 연동하면, 실제 도시 내 교통 상황을 모델링하고 시뮬레이션하는 데 매우 유용한 도구가 될 수 있다.

SUMO를 통해 사용자는 차량 속도, 사용 중인 차량 유형, 보행자 흐름, 신호등 기능, 도로 교통량, 차선 변경 기능 등 다양한 측면을 조작할 수 있을 뿐만 아니라 필요한 경우 차량이 이용할 대체 경로를 지정할 수 있다. 차량이 도로를 따라 이동하는 속도를 조절하고 버스, 트럭, 자전거 및 모터사이클과 같은 특정 유형의 차량을 지정할 수도 있다. 또한 사용자는 보행자가 따라야 할 경로와 안전하게 도로를 건널 수 있는 위치를 정의하고, 교차로에 위치한 신호등의 신호 주기 타이밍을 조정할 수도 있으며, 도로의 특정 구간에서의 차량 밀도 관리 및 차량의 차선 변경 빈도 및 규칙에 관한 설정도 가능하다. 필요한 경우 특정 상황에서 우회 경로를 생성할 수도 있다. SUMO의 이러한 기능들을 통해 현실적인 차량 이동을 시뮬레이션할 수 있어 실제 교통 상황을 보다 정확하게 반영할 수 있다.

시뮬레이션에서는 먼저, OSM에서 생성한 파일인 map.osm 을 SUMO로부터 입력 받아 netconvert 명령어를 사용하여 도로 네트워크에 관한 정보가 저장되어 있는 map.net.xml 파일을 생성한다. 이후, SUMO에서 제공하는 모듈 중에 randomTrips.py를 사용하여, 차량들의 랜덤한 경로를 map.trips.xml에 저장한다. 마지막으로, 차량들의 랜덤한 이동 경로를 OMNeT++ 시뮬레이터의 Veins 확장 모듈이 이용할 수 있도록 포맷을 변환하기 위해, duarouter 명령어를 사용하여, map1.rou.xml을 생성한다.

SUMO에서 *.rou.xml 데이터 파일은 차량 경로와 관련된 정보를 정의하는 파일로, 시뮬레이션에서 각 차량의 경로와 관련된 다양한 속성과 요소들을 포함하고 있다. 주요 구성 요소와 속성들은 다음과 같다.

- vehicle: 개별 차량을 정의하는 요소로, 차량의 고유 식별자 및 유형(자동차, 버스 트럭 등)을 지정하는 속성을 갖는다. 뿐만 아니라, 차량이 시뮬레이션에서 출발하는 시간 및 출발 속도와 위치 등을 지정하는 속성들도 갖는다.
- route: 차량의 이동 경로를 정의하는 요소이며, 이 요소의 주요 속성인 edges에는 차량이 이동할 도로 구간의 목록이 심플로 구분되어 나열된다.
- flow: 이 요소는 일정 시간 동안의 다수의 차량들의 대한 교통 흐름을 정의하는 요소로, 생성될 차량의 총대수 및 유형, 속도, 출발 위치와 경로(route)와 같은 속성을 통해 정의된다.

3. OMNeT++

OMNeT++는 네트워크 시뮬레이션 프레임워크로, 다양한 통신 시스템의 성능을 측정할 수 있는 도구이다. 또한, 모듈화된 구조를 갖추고 있어 모듈의 확장 및 모듈 간의 상호 작용이 용이하다. 본 연구에서는 SUMO로부터 얻은 차량들의 이동 경로 정보를 OMNeT++에서 이용하기 위해 Veins 라는 프레임워크를 추가하였으며, OMNeT++의 확장 모듈인 INET 프레임워크를 이용하여 차량 통신 시스템을 시뮬레이션 하였다.

3.1 VEINS

VEINS (Vehicular Environment In Network Simulation)는 차량 네트워크(VANET)의 통신을 시뮬레이션하기 위한 프레임워크이다. VEINS는 OMNeT++와 SUMO를 통합하여, 차량의 이동과 네트워크 통신을 동시에 시뮬레이션할 수 있도록 한다.

SUMO의 *.rou.xml 파일을 이용하여 OMNeT++의 VEINS가 차량의 이동과 네트워크 통신을 동시에 시뮬레이션하는 절차는 다음과 같다. 이 절차는 SUMO에서 생성된 차량 이동 데이터를 OMNeT++로 전달하여, 두 시뮬레이터의 협력을 통해 차량 네트워크(VANET)를 시뮬레이션하는 과정을 포함한다.

먼저, VEINS는 TraCI (Traffic Control Interface)를 사용하여 SUMO와 OMNeT++ 간의 실시간 통신을 VEINS 프로젝트 내의 트래픽 컨트롤 시나리오 매니저의 역할을 하는 모듈인 TraCIScenarioManager 을 사용하여 설정한다. 이후, SUMO 와 OMNeT++ 시뮬레이션을 실행하면, TraCIScenarioManager 가 SUMO 와 OMNeT++ 간의 연결을 설정하고 실시간으로 데이터를 교환할 수 있다. 즉, SUMO는 차량의 위치, 속도, 이동 경로 등의 데이터를

VEINS를 통해 실시간으로 OMNeT++에 제공 한다.

3.2 INET

OMNeT++에서 INET 확장 프레임워크는 유선 네트워크, 무선 네트워크, 모바일 애드혹 네트워크(MANET), 차량 네트워크(VANET) 등과 같은 IP 기반 네트워크의 다양한 요소를 모델링하는 데 필요한 포괄적인 라이브러리와 도구를 제공하기 위해 사용된다. 특히, INET은 OSI 7 계층 모델에 따라 모듈이 구성하여, 다양한 네트워크 프로토콜을 시뮬레이션할 수 있으면서도, 효율적인 시뮬레이션 환경을 제공하여, 복잡한 네트워크 시나리오에서도 높은 처리 성능을 보여, 더 빠르게 시뮬레이션을 실행하고 결과를 분석할 수 있게 한다는 장점이 있다.

본 연구에서도 INET은 차량 간 통신을 위한 무선 네트워크 통신 환경을 구성하기 위해 사용되었다. 즉, SUMO를 통해 차량의 이동 정보가 OMNeT++에 제공되면 이를 네트워크 통신에 반영하여, INET을 통해 차량 간의 네트워크 패킷 전송, 신호 강도, 패킷 지연 등의 네트워크 통신 데이터를 시뮬레이션하는 것이다.

IV. Simulation Study

1. Simulation configurations

본 연구에서는 INET 프레임워크에서 라우팅 프로토콜을 적용하여, 기존의 단순 페이로드만 전송하는 기본 애플리케이션인 VeinsInetSampleApplication을 INET에서 제공하는 UDP와 TCP 기반의 완성된 애플리케이션으로 대체하였다. 기본 애플리케이션에서는 단순한 데이터 페이로드를 전송하는 것만 가능했기 때문에, 현실적인 VANET 실험 환경을 구성하기 어렵기 때문이다.

또한 새롭게 구현한 RSU 모듈에 라우터 기능을 추가하여 UDP와 TCP를 기반으로 하는 애플리케이션을 이용해 파일 전송, 스트리밍 서비스, 웹 브라우징 등과 같은 다양한 유형의 트래픽을 시뮬레이션하도록 설정하였다.

이를 통해, INET의 라우팅 프로토콜과 완성된 UDP 및 TCP 애플리케이션을 사용함으로써, 단순한 페이로드 전송이 아닌 실제 VANET 환경에서 사용하는 데이터를 이용하여 현실적인 VANET 시뮬레이션을 할 수 있다.

한편, SUMO에서 생성된 *.rou.xml 파일을 VEINS의 모빌리티 모델을 통해 OMNeT++에서 사용려는 경우, *.rou.xml 내에 저장된 차량 속도 정보에만 의존한다는 문제가 있어, 신호등, 교통체증 상황, 인접한 보행자 및 차량의 영향과 같은 도로 상황을 실시간으로 반영하지 못해

현실성이 떨어지는 문제가 있다. 이를 해결하기 위해 VEINS의 모빌리티 모델을 수정하여 실시간으로 도로 상황을 반영하여 차량의 속도가 제어되도록 하였으며, 이를 통해 시뮬레이션에서 차량이 움직일 때 다양한 속성들과의 상호작용을 고려하여 차량의 속도를 조절할 수 있도록 하였다.

모의 실험은 먼저, 디지털 트윈 기반의 접근 방식의 유효성과 필요성을 입증하기 위해, 가상의 단순한 형태의 도로와 실제 도로 환경인 대전시 둔산동 지역에서 시뮬레이션을 수행한 결과를 비교한다. 이 후, Fig.1과 같이 대전시의 둔산동 인근 지역으로 지도의 규모를 확장하여, 차량의 최대 속도와 차량의 밀집도가 차량 네트워크의 성능에 미치는 영향을 확인한다. 모든 모의실험에서 RSU는 10 byte의 패킷을 1,000 ms 간격으로 전송하며, 특정 차량을 목적으로 설정하여, 라우팅 프로토콜인 AODV에 의해 패킷이 다양한 차량을 거쳐 목적지 차량까지 이동하게 된다. 그 밖의 시뮬레이션의 주요한 설정들은 Table 2에서 확인할 수 있다. 추가적으로 시뮬레이터에서 실험이 진행되는 상황을 <https://www.youtube.com/watch?v=7RpCHoeVa4E>에서 영상을 통해 확인할 수 있다.

2. Deployment strategy of RSUs

VANET에서 RSU를 차량이 더 빈번하게 이동하는 영역에 배치하면, 더 많은 차량에게 데이터를 전달할 수 있는 기회가 증가하므로, 목적지까지 패킷이 원활하게 전달될 수 있다. 모의 실험에서는 RSU를 차량의 이동이 빈번한 영역에 위치하도록 설정하기 위해 다음과 같은 방법을 사용하였다. 먼저 Fig.1과 같은 대전시 둔산동 인근 지역에 대하여 차량들의 이동을 시뮬레이션한다. 이때, 차량 노드들에 대한 경로를 단위 시간별로 기록하여 2차원 좌표 값의 형식으로 저장된 경로 데이터를 생성한다. 다음으로, (1)과 같이 각 좌표를 보정 하면 특정 범주의 영역에 대하여, 차량 노드들이 통과하는 빈도수를 계산할 수 있다.

$$p_i = (\hat{x}, \hat{y}) = \left(\left\lfloor \frac{x}{\delta} \right\rfloor \times \eta, \left\lfloor \frac{y}{\delta} \right\rfloor \times \eta \right) \quad (1)$$

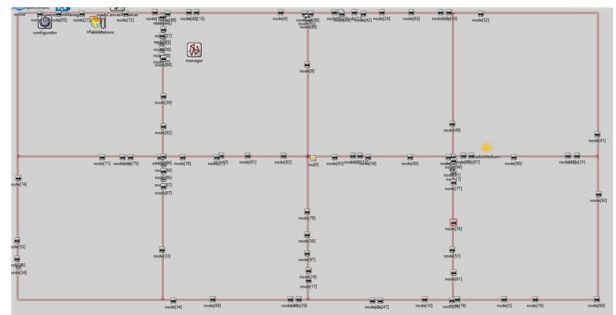
(1)은 좌표 값 (x, y) 를 이산화하고 특정 구간 내의 좌표를 대표하는 값 p_i 로 변환하기 위한 규칙으로, 주요 요소는 다음과 같다. (\hat{x}, \hat{y}) 는 이산화된 좌표이며, δ 와 η 는 좌표의 이산화에 사용할 간격 또는 구간의 크기를 설정하는 값이다. 예로 들어, $(x, y) = (117.57, 274.25)$ 인 경우, δ 와 η 를 모두 10으로 설정하면, 1의 자리에서 버림 연산을 수행하여, $(\hat{x}, \hat{y}) = (110, 270)$ 이 된다. δ 와 η 가 모

Table 2. Simulation configurations

Parameter	Value
Simulation time	100 sec
Wifi Phy mode	IEEE 802.11p
Packet Size	10 byte
Routing Protocol	AODV
Traffic Type	UDP
Channel Model	SimpleObstacleShadowing
Data rate	6 Mbps
Noise Floor	-98dBm
Maximum Communication Range	250m
Node Transmit power	2mW
RSU Transmit Power	20mW
RSU Bandwidth	10MHz
RSU Radio Band	5.9GHz
RSU Application Send interval	1,000ms

두 1인 경우는 $(\hat{x}, \hat{y}) = (117, 274)$ 이다. 추가적으로, η 의 값을 δ 와 동일한 값이 아닌 다른 값으로 조정하면, 구간의 범주를 조절할 수도 있다.

이러한 변환 과정을 모든 차량의 경로 데이터에 반영하면, 특정 구간 p_i 와 이 구간을 통과하는 차량의 빈도수를 얻을 수 있다. 실험에서 η 과 δ 값은 모두 10으로 설정하였으며, 차량이 가장 많이 통과하는 영역 한 곳을 RSU의 위치로 설정하였다. 또한, 매 실험 마다 랜덤한 교통 트래픽을 사용하므로 RSU의 위치도 새롭게 계산되지만, 주로 교차로의 특성을 가지는 영역에서 RSU의 위치가 설정된다.



(a) Grid type road environment



(b) Road environment in Dusan-dong, Daejeon

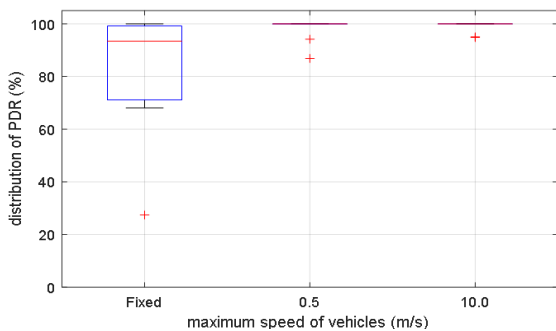
Fig. 3. Grid type road environment in simulation and road environment in Dunsan-dong, Daejeon

3. Comparative Analysis: VANET Performance in Realistic and Simple Road Environments

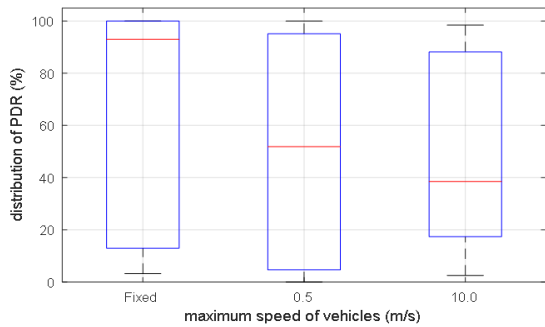
이 절에서는 모의실험을 통해, 현실 도시 환경과 가상의 도로 환경에서 VANET에서의 packet delivery ratio (PDR) 성능을 관찰하고, 가상의 도로 환경에서는 발견되지 않는 성능 상의 문제가 현실 도로 환경에서 발생할 수 있음을 확인한다. 이를 통해, 디지털 트윈 기반의 접근 방식이 VANET에서의 새로운 문제를 발견할 수 있음을 알 수 있다.

Fig. 3은 시뮬레이션에서 사용할 두 도로 환경을 나타낸 것으로, Fig. 3(a)는 SUMO에서 도로 네트워크를 직접 구성하는 기능을 이용하여 그리드 유형으로 구성된 단순한 형태의 도로 환경이며, Fig. 3(b)는 Fig. 3(a)의 그리드 유형 도로 환경과 비슷한 규모를 갖는 대전시 둔산동 지역의 도로 환경을 OSM과 SUMO를 이용해 구성한 것이다.

실험은 100대의 차량에 대해, 속도가 고정되어 있는 경우(Fixed)와 최대 속도가 각각 0.5m/s 및 10.0m/s인 경우에 대하여 10회씩 수행되었으며, 그 결과인 PDR의 분포를 boxplot을 통해 Fig. 4에 나타냈다.



(a) Distribution of PDR performance in grid type road environments



(b) Distribution of PDR performance in Dunsan-dong, Daejeon

Fig. 4. Distribution of packet delivery ratio performance in grid type road environment and the road environment in Dunsan-dong, Daejeon

먼저 Fig. 4(a)에서 단순한 그리드 유형의 도로 환경에서 PDR의 분포를 보면, 차량의 최대 속도가 0m/s (Fixed)인 경우, 중앙값은 93.43%으로 비교적 높은 값을 보이고, 25 퍼센타일 값이 71.07%로 PDR 값 간의 약간의 편차가 있음을 확인할 수 있다. 다음으로 차량의 최대 속도가 0.5m/s 및 10.0m/s 인 경우 PDR의 중앙값은 모두 99.97%로 매우 높은 값을 갖는다. 이 때, 25 퍼센타일 값은 각각 99.91% 및 99.95%여서 두 경우 모두 그리드 유형과 같은 단순한 도로 환경이라면, 거의 모든 상황에서 높은 PDR을 유지할 수 있음을 확인할 수 있었다.

반면, 가상의 도로 환경보다 더 복잡한 형태를 갖는 현실의 도로 환경에서 PDR 성능의 편차는 매우 큰 것으로 확인 된다. Fig. 4(b)는 대전시 둔산동 지역의 PDR 성능의 분포를 나타낸 것으로, 차량의 최대속도가 증가하는 순으로 상위 25%는 각각 99.99%, 94.12% 및 86.30%이며, 하위 25%는 각각 30.05%, 10.54% 그리고 17.56%여서 단순한 도로 환경에 비교해 봤을 때, 네트워크 성능의 변동성이 크며, 특정 상황에서 성능이 크게 저하될 가능성이 매우 높다는 사실을 확인할 수 있다.

이는 네트워크 토폴로지의 복잡성과 도로 조건의 다양성이 성능에 큰 영향을 미칠 수 있음을 보여준다. 반면, 가상 도로 환경에서는 네트워크 토폴로지가 단순하고 예측 가능하여, PDR 값의 변동성이 적고 안정적인 성능을 보일 가능성이 높다. 따라서, 이 결과는 VANET 시뮬레이션에서 디지털 트윈 기반의 접근 방식을 통해, 현실적인 도로 환경이 반영된 신뢰성 있는 평가의 중요성을 보여주며, 단순한 그리드 네트워크만으로는 실제 운영 환경에서 발생할 수 있는 문제를 완전히 파악하기 어려울 수 있음을 의미한다.

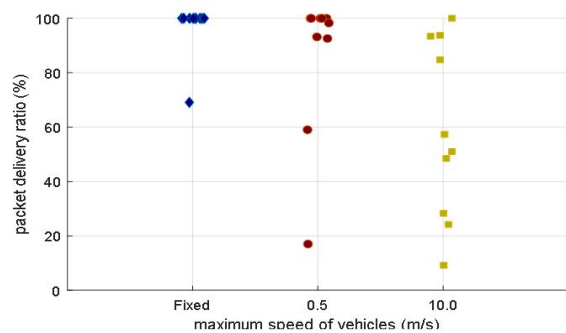


Fig. 5. The impact of the vehicle's top speed on VANET performance

4. Effect of vehicle speed

이 절에서는 모의실험을 통해 대전시 둔산동 인근 지역에 구축된 차량 네트워크 환경에서 차량들의 최고 속도가 미치는 영향을 PDR의 측면에서 확인한다. 시뮬레이션에서 차량들은 도로 상황에 따라 차량이 멈추거나 감속이 발생하지만, 가속을 하여도 최대 속도까지만 증가하게된다.

모의실험에서는 차량의 수가 900대로 고정되어 있는 상황에서, 최고 속도가 각각 0 (Fixed)이여서 차량들이 이동하지 않는 경우 그리고 0.5 m/s 및 10.0 m/s인 경우에 대해서 수행되었으며, 이를 통해 네트워크의 토폴로지가 차량 이동에 의해 변하는 수준에 따라 PDR 성능을 확인할 수 있다. 각 최고 속도에 실험은 서로 다른 교통 트래픽을 갖도록 10회씩 수행되었으며, 그 결과를 Fig. 5에서 확인할 수 있다.

먼저, 차량이 고정된 경우인 Fixed에서는 총 10번의 실험 중에서 하나의 경우에서만 약 70% PDR을 나타내었고, 나머지 9번의 경우에는 모두 PDR이 100%여서 네트워크 토폴로지의 변화가 없는 경우 대부분의 트래픽은 목적지 차량까지 높은 신뢰성을 가지며 전달될 수 있는 것을 확인할 수 있었다. 하지만, 차량의 최고 속도가 0.5m/s 여서 네트워크 토폴로지가 동적으로 변화하는 경우에는 총 5번의 실험에서 100%의 PDR을 나타내고 3회의 실험에서는 90% 이상의 PDR을 보여, 패킷 전달이 원활하게 이루어졌음을 확인할 수 있지만, 일부 실험에서의 PDR 값이 각각 59%와 17%로 크게 떨어지는 것을 확인할 수 있었다. 즉, 이동하는 차량으로 구성된 네트워크가 시간에 따라 조금이라도 변화하게 되면, PDR 값의 변동성이 커지게 된다.

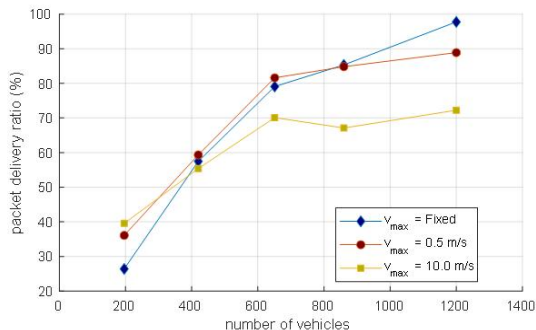


Fig. 6. The impact of vehicle density on VANET performance

차량의 최고 속도가 0.5m/s 보다 큰 10.0m/s 인 경우에는 PDR 값이 더 크게 변동하는 것을 확인할 수 있다.

일부 실험에서는 PDR 값이 100에 도달한 반면, 다른 실험에서는 매우 낮은 PDR 값을 보인다. 특히, 9.2, 28.3, 24.1과 같은 값은 매우 낮아 현실 도시 환경 및 차량 이동 환경에서 차량의 이동 속도가 빠를수록 네트워크 성능이 저하될 가능성이 높아지고, 일관적이면서도 신뢰성 있는 통신을 유지하기 어려울 수 있음을 확인할 수 있다.

5. Effect of vehicle density

이 절에서는 앞 절과 동일하게 대전시 둔산동 인근 지역에 구축된 차량 네트워크 환경에서 차량들의 밀집도가 VANET의 성능에 미치는 영향을 확인한다. 이를 위해, 차량의 수를 196, 420, 651, 860, 그리고 1,200대까지 증가시키면서 PDR을 측정하였으며, 추가적으로 차량의 최고 속도의 영향도 확인하여 그 결과를 Fig. 4에 나타내었다.

한편, 이전 절에서 차량의 최대 속도의 영향을 확인하였던 실험과 마찬가지로 각 결과에서의 PDR 값의 편차가 커 이를 가능할 수 있도록 Table. 3와 같이 PDR 값의 최소·최댓값을 명시하였다.

Fig. 6에서 네트워크 내 차량의 수가 증가하여 밀집도가 증가하면, PDR의 평균값도 증가 추세를 보이는 것을 확인할 수 있다. 차량 들이 고정되어 있는 경우의 PDR 값은 차량의 수가 증가하면, 26%, 58%, 79%, 그리고 마지막으로 98%로 향상되며, 차량 들의 최고 속도가 0.5m/s인 경우에서 PDR은 36%에서 59%, 82%, 85%로 증가하고 89.9%에 도달한다. 이는 차량이 밀집해 있으면, 적어도 하나 이상의 이웃한 차량이 존재할 가능성이 증가하여, 목적지 까지 패킷이 전달될 경로도 존재할 가능성이 높아지기 때문이다.

차량의 최고 속도가 10.0m/s로 올라가면 차량의 밀집도가 증가함에 따라 패킷 전송 비율도 여전히 개선되지만, 차량의 이동성이 낮은 다른 두 경우와 비교하여, PDR의 현저한 감소가 발생한다. PDR 값은 40%에서 55%, 70%로 변동하는 패턴을 따르다가 67%으로 하락 후 72%까지 소폭 증가하여, 차량이 고정되어 있는 경우와 비교하였을 때, 최대 25% 까지 낮은 성능을 보인다.

이는 차량들이 고속으로 이동하면, 네트워크 토폴로지가 더 급격하게 변화함에 따라, 라우팅 프로토콜에 의해 유지되고 있는 라우팅 테이블에 저장된 이웃 노드에 관한 정보가 더 이상 유효하지 않아, 목적지 까지 패킷을 효과적으로 전달할 수 없기 때문이다.

Table 3. PDR performance of VANET for a various number of vehicles and maximum vehicle speed

number of vehicles maximum speed	196		420		651		860		1200	
	min.	max.	min.	max.	min.	max.	min.	max.	min.	max.
0 m/s (fixed)	0.0	100.0	0.0	100.0	11.5	100.0	1.3	100.0	85.0	100.0
0.5 m/s	0.0	100.0	0.0	100.0	0.0	100.0	29.3	100.0	64.4	100.0
10.0 m/s	0.0	92.9	0.0	95.6	19.6	98.4	10.0	94.0	30.9	96.6

6. Comparison of simulator performance

이 절에서는 2절에서 비교 분석한 시뮬레이터들 중 OSM 및 SUMO와 통합된 환경을 구축할 수 있으면서도 상용 도구가 아닌 NS-3와 OMNeT++에 대해서, 시뮬레이션 실행 시간을 측정하여, VANET 연구를 위한 시뮬레이션 도구로서 OMNeT++가 더 효율적이라는 것을 확인한다.

Table 4는 NS-3와 OMNeT++ 시뮬레이터를 SUMO와 통합된 환경을 구축하여, 수행하였을 때의 실행 시간을 비교한 결과이다. N_v 는 시뮬레이션에 생성된 차량의 수를 나타내며, Run Time은 시뮬레이션을 실행하는 데 소요된 시간을 초 단위로 나타낸 것이다. 시뮬레이터의 실행 시간이 측정된 시스템의 환경은 CPU 12th Gen Inter(R) Core(TM) i7-12700 2.10GHz, GPU NVIDIA GeForce RTX 3070 (8GB), 메모리 32.0GB, 디스크 SAMSUNG MZVL2512HCJQ-00B00 (477GB, SSD)이며, NS-3가 구동되는 운영체제는 Ubuntu 22.04.4 LTS, OMNeT++은 Windows 11 Pro에서 수행되었다. 실행 시간 측정 결과, 차량의 수가 비교적 적은 200대 이하의 경우에 대해서 NS-3와 OMNeT++의 실행 시간은 각각 196초와 12초로 NS-3는 OMNeT++보다 약 16배 더 긴 실행 시간을 보인다. 차량의 수가 중간 정도인 약 600여대의 경우에는 NS-3의 실행 시간은 11,300초로 급격히 증가한 반면, OMNeT++의 실행 시간은 180초로 크게 증가하지 않아, 그 결과, NS-3는 OMNeT++보다 약 62배 더 긴 실행 시간을 보인다. 차량 수가 가장 많은 경우인 약 1,000대 이상의 경우에 대해서 NS-3는 OMNeT++보다 약 159배 더 긴 실행 시간을 보여 격차가 더 증가함을 확인할 수 있다.

즉, OMNeT++는 모든 차량 수에 대해 NS-3보다 훨씬 짧은 실행 시간을 보였으며, 특히 차량 수가 증가할수록 두 시뮬레이터 간의 성능 차이가 더욱 두드러졌다. 이러한 결과는 OMNeT++가 대규모 VANET 연구에서 더 나은 효율성을 제공함을 의미한다.

V. Conclusions

본 논문에서는 디지털 트윈 기반의 접근 방식을 적용하여 현실적인 도시 환경에서 차량 네트워크의 성능을 평가하기 위한 시뮬레이터의 통합된 환경으로, OSM, SUMO 그리고 OMNeT++을 이용하였다. 통합된 환경에서의 시뮬레이션에서는 VANET의 성능이 낮은 차량 밀집도와 네트워크 토폴로지의 변화로 인해 급격히 저하되는 문제가 발생할 수 있음을 뿐만 아니라, 성능이 일관되지 않다는 문제도 확인하였다.

Table 4. Comparison of simulation run times

N_v	NS-3		OMNeT++	
	N_v	Run Time (sec)	N_v	Run Time (sec)
196	196	196	193	12
651	11,300	11,300	617	180
1,200	42,444	42,444	1,093	267

향후 연구에서는 이와 같은 문제를 더 심층적으로 분석하고, VANET 성능의 일관성을 높이기 위한 방안을 고안할 필요가 있다. 예를 들어, 현실 도시 환경에서 RSU의 최적 배치 전략, 고성능 라우팅 프로토콜의 개발, 차량 간 협력 통신 방법 등의 접근법을 통해 VANET의 성능을 향상시킬 수 있다. 또한, 다양한 도시 환경과 교통 조건에서의 시뮬레이션을 통해 일반화된 결과를 도출하고, 실제 적용 가능성을 높이는 것도 중요하다.

REFERENCES

- [1] Tao, F., Zhang, H., Liu, A., & Nee, A. Y. C. (2018). Digital Twin in Industry: State-of-the-Art. *IEEE Transactions on Industrial Informatics*, 15(4), 2405-2415.
- [2] Boschert, S., & Rosen, R. (2016). Digital Twin—The Simulation Aspect. In *Mechatronic Futures* (pp. 59-74). Springer, Cham.
- [3] Varga, A., & Hornig, R. (2008). An overview of the OMNeT++ simulation environment. In *Proceedings of the 1st international conference on Simulation tools and techniques for*

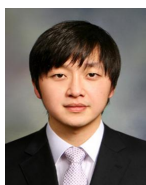
- communications, networks and systems & workshops (p. 60).
- [4] Haklay, M., & Weber, P. (2008). OpenStreetMap: User-generated street maps. *IEEE Pervasive Computing*, 7(4), 12-18.
- [5] Krajzewicz, D., Erdmann, J., Behrisch, M., & Bieker, L. (2012). Recent Development and Applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4), 128-138.
- [6] Sommer, C., & German, R. (2011). Veins: The open source vehicular network simulation framework. In *Proceedings of the 2011 IEEE Vehicular Networking Conference* (pp. 213-214).
- [7] Henderson, T. R., Roy, S., Floyd, S., & Riley, G. F. (2006). ns-3 project goals. In *Proceedings of the 2006 workshop on ns-2: the IP network simulator* (pp. 13-es).
- [8] Riley, G. F., & Henderson, T. R. (2010). The ns-3 network simulator. In *Modeling and Tools for Network Simulation* (pp. 15-34). Springer, Berlin, Heidelberg.
- [9] NetSim. (2021). NetSim Product Brochure. Retrieved from <https://www.tetcos.com/pdf/NetSimBrochure.pdf>
- [10] Baldo, N., Miozzo, M., Guerra, A., & Dini, P. (2010). Integrating real world applications into NS-3. In *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques* (pp. 415-422).
- [11] McCabe, J. D., & Lawrence, S. J. (1998). *Practical computer network analysis and design*. Morgan Kaufmann.
- [12] Barr, R., Haas, Z. J., & van Renesse, R. (2004). JiST: an efficient approach to simulation using virtual machines. *Software: Practice and Experience*, 34(6), 541-567.

Authors



Jonghyeon Choe received the B.S. degree in Computer Engineering from Kongju National University, Korea, in 2023. Jonghyeon Choe is currently pursuing a Master's degree in the Department of Computer Engineering at

Kongju National University since 2023. His areas of interest include vehicular communication networks and wireless networks.



Youngboon Kim received the B.S., M.S. and Ph.D. degrees in Information and Communication Engineering from Dongguk University, Korea, in 2011, 2013 and 2020, respectively.

Dr. Kim has been serving as a Post-doctoral researcher and lecturer at both Kongju National University and Dongguk University since 2020. His research interests encompass wireless networks, vehicular communication, computer security, and convergence security.



Sangdae Kim received his B.S. and Ph.D. degrees in computer engineering from the Chungnam National University, Daejeon, Republic of Korea in 2013 and 2019, respectively.

He was a Postdoctoral Researcher with the Software Research Center, Chungnam National University, and the Division of Computer Science and Engineering, Kongju National University. He is currently an Assistant Professor in Dept. of Medical Information Technology Engineering Soonchunhyang University. His research interests include routing protocols for real-time and reliable communication, mobility support strategy for seamless communication in wireless sensor networks (WSNs), and industrial wireless sensor networks (IWSNs).