KIPS Tr. Comp. and Comm. Sys.
Vol.3, No.12 pp.449~454 pISSN: 2287-5891

http://dx.doi.org/10.3745/KTCCS.2014.3.12.449

Supplements an Initial Creation and User Addition in VANET Cloud Architecture

Taehyeong Kim† · JooSeok Song††

ABSTRACT

While the era of driverless car has come, Vehicular Ad hoc NETwork(VANET) is getting important. Original VANET has a limit that cannot use computation power, storage space of On Board Unit(OBU) installed in a vehicle efficiently. VANET cloud computing(VCC) solves the limit to focus on using abilities of each vehicle. This article proposes VCC architecture for supplementing user addition and initial cloud creation that have been researched insufficiently.

Keywords : Vehicular ad Hoc Network, Vehicular ad Hoc Network Cloud Computing, Road Side Unit, Driverless Car

초기 생성과 사용자 추가를 고려한 VANET 클라우드 아키텍처

김 태 형† · 송 주 석††

요 약

무인 자동차의 시대가 도래하면서, 차량 간 통신 네트워크인 Vehicular Ad hoc NETwork(VANET)의 중요성이 점점 더해지고 있다. 지금까지 VANET의 연구가 많이 진행되었으나, 기존의 연구는 각 자동차의 통신적인 측면만 연구하고, 차량에 설치된 On Board Unit(OBU)의 계산 능력, 저장 공간 등을 효율적으로 사용하지 못하는 한계점이 있었다. 그러나 최근 나온 VANET cloud computing(VCC) 개념은 높아진 각 자동차의 능력을 효율적으로 이용하여, 유용하게 사용하는 것에 초점을 맞추어 이러한 한계점을 해결하였다. 그러나 지금까지 진행된 연구에서는 VCC의 초기 클라우드 생성 부분과 사용자 추가 부분의 연구가 미흡한 실정이다. 본 논문은 이러한 부분을 보완한, VCC 아키텍처를 제안한다.

키워드 : 차량 애드혹 네트워크, 차량 애드혹 클라우드 컴퓨팅, 노변 장치, 무인 자동차

1. 서 론

2000년대 초반 Vehicular Ad hoc NETwork(VANET) 연구가 시작되면서[2] 차량과 차량, 차량과 Road Side Unit (RSU)가 통신하는 기술이 개발되고 발전해왔다. 이러한 기술은 교통사고나 정체정보 같은 도로정보를 운전자에게 제공해서 사고 예방 및 도로 정체 예방 등을 가능하게 하였다.

VANET은 크게 RSU와 OBU, 차량으로 구성되어있다. RSU는 도로와 같은 외부에 설치되어 주변의 정보를 각 차량에 전달하고, OBU는 각 차량에 설치되어 RSU를 통해 전달된 정보를 운전자에게 전달한다. 그리고 차량과 RSU 간의 통신을 Vehicle-to-roadside(V2R) 또는 Vehicle-to-Infrastructure (V2I), 차량과 차량 간의 통신을 Vehicle-to-Vehicle(V2V)이라고 한다. 통신프로토콜은 Dedicated Short Range Communication (DSRC), Long Term Evolution(LTE), Zigbee, 3G 등 다양한 프로토콜을 사용할 수 있다.

지속적인 연구가 진행되면서 conditional privacy, 키 관리, 라우팅 프로토콜 등등의 많은 연구가 진행되었고 많은 발전이 있었다. 또한 컴퓨터 기술이 발전하면서 On board unit(OBU)의 성능 또한 과거에 비해 월등히 발전하였다. 그로 인하여 각각의 자동차의 계산 능력, 저장소 등의 활용에 대한 필요성이 대두되었다. 이런 문제를 해결하기 위해 2011년 Olariu et al.이 클라우드 컴퓨팅을 기존의 VANET에 적용하는 아이디어를 제시하면서 해결 방안을 제시하였다[1].기존의 VANET는 각각의 자동차의 통신 성능을 높이는 것을 목표로 하였지만, 이 논문은 각각의 자동차의 높아진 계산 능력을 모으거나 저장소를 공유하여 효율적인 자

※ 이 논문은 2014년도 정부(교육부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(2014R1A1B3004161).
※ 이 논문은 2014년도 한국정보처리학회 춘계학술발표대회에서 'VANET 클라우드 컴퓨팅 아키텍처'의 제목으로 발표된 논문을 확장한 것임.
† 준 회 원 : 연세대학교 컴퓨터과학과 석사과정
†† 종신회원 : 연세대학교 컴퓨터과학과 정교수
Manuscript Received : July 17, 2014
First Revision : September 15, 2014
Accepted : September 15, 2014
* Corresponding Author : JooSeok Song(jssong@emerald.yonsei.ac.kr)

원 관리를 할 수 있게 하는 것을 목표로 하였다. 또한 기존의 클라우드 컴퓨팅의 장점을 그대로 가져와 이용자가 언제든 모든 정보를 인터넷상의 서버에서 가져와 이용할 수 있게 하였다. 이러한 장점들은 기존의 VANET 연구들보다 훨씬 복잡한 계산을 쉽게 할 수 있어 Intelligent Transport System (ITS) 구축의 문제점 해결에 하나의 해결책이 될 수 있게 하였다.

기본적으로 VCC의 구조는 다른 차량들이 구성한 VANET 클라우드를 사용하는 것과, RSU를 통해 방대한 데이터베이스 클라우드 자료를 얻는 것으로 나눌 수 있다. 이러한 구조를 효율적으로 구성하기 위해 많은 연구들이 진행되었고, 많은 제안들이 나왔다. 그러나 아직까지 초기 네트워크 설정과 사용자 추가 부분에 관한 연구는 미흡한 실정이다. 그래서 본 논문에서는 이러한 미흡한 부분을 보완한 VCC의 아키텍처를 제안한다. 2절에서는 지금까지 제안된 VCC 관련 연구를 살펴보고, 3절에서는 제안하는 아키텍처 구조를 소개한다. 4절에서는 제안된 구조의 성능 평가결과를 도출하고, 끝으로 5절에서는 논문의 결론에 대해 기술한다.

2. 관련 연구

클라우드 컴퓨팅 서비스는 크게 Software as a Service (SaaS), Platform as a service(PaaS), Infrastructure as a Service(IaaS)로 나눌 수 있다[3]. SaaS는 사용자가 클라우드상에서 소프트웨어를 사용할 수 있게 해주는 서비스로, 예로는 Google docs, gmail 등이 있다. PaaS는 사용자가 클라우드상에서 플랫폼을 이용할 수 있게 해주는 서비스로, 예로는 Window Azure, Google App Engine 등이 있다. IaaS는 사용자가 클라우드상에서 저장 공간, 네트워크와 같은 인프라스트럭처를 이용할 수 있게 해주는 서비스이다. 그러나 이 서비스들을 VCC에 적용시키기에는 문제가 있다. 예를 들면, PaaS는 VCC에 적합하지 않은 서비스이기 때문이다. 그래서 Olariu et al.은 이 세 가지 서비스를 Network as a Service(Naas), Storage as a Service(STaaS), Cooperation as a Service(CaaS)의 세 가지 범주로 VCC에 맞게 분류하였다[4]. NaaS는 사용자에게 인터넷과 같은 네트워크를 제공해주는 서비스이고, STaas는 사용자에게 추가적인 저장소를 대여해주는 서비스이다. 마지막으로 CaaS는 사용자에게 RSU와 같은 곳을 통해 날씨 정보, 도로 상황과 같은 운전자 안전을 위한 교통 정보 등을 제공해주는 서비스이다.

Hussain et al.은 VCC를 Vehicular Cloud(VC), Vehicles using Clouds(VuC), Hybrid Clouds(HC)로 나누어 정의했다[5]. VC는 각각의 자동차가 VCC 제공자들로 이루어진 클라우드 서비스를 이용하는 것을 의미하고 VuC는 각각의 자동차가 RSU를 통해 데이터베이스 클라우드 서비스를 개별적으로 이용하는 것을 의미한다. 마지막으로 HC는 앞의 두 가지 유형을 합친 것이다. VC는 주로 사용자가 많은 계산이나 저장소가 필요할 때 사용되고, VuC는 사용자가 도로 상황과 같은 정보를 얻기 위하여 사용된다. HC와 같은 경우는 많은 정보를 토대로 많은 계산을 동시에 하기 위해 사용된다. 예를 들면, 미국 서부에서 동부까지 가는 데 최단의 길을 구하는 것과 같은 많은 정보와 계산이 동시에 필요할 때 사용된다.

Olariu et al.은 VCC의 활용 방안에 관해 제시하였다[1]. 유동성 있는 교통 신호, 대규모 피난, 주차장에 주차되어있는 차들의 자원 활용 등이 그것이다. 유동성 있는 교통 신호의 경우는 교통사고 직후나 행사 때 사용된다. 이때는 평소와 다르게 특정 구간이 막히기 때문에 교통 신호를 빨리 조절해줘야 한다. 특정 구간이 막히면 VCC가 Central Authority(CA)에게 의견을 전달하여 교통 신호를 유동적으로 바꾸게 한다. 대규모 피난의 경우에는 클라우드를 통해 원하는 피난처로 빠르게 갈 수 있게 인도해준다. 주차장에 주차된 차량 활용 방안으로는, 주차되어있는 차량의 자원을 이용하여 클라우드를 생성하고, 이러한 자원을 소규모 기업에서 활용하는 방안이 있다.

3. 제안하는 아키텍처 구조

2010년, VCC의 개념이 소개된 이후로 많은 연구가 이루어졌고, VCC의 활용 방안, 정의 등에 초점을 맞추어 많은 연구가 진행되어왔다. 그러나 초기 연결 과정과 전체적인 보안에 관련된 연구는 미흡한 실정이다. 예를 들면, Olariu et al..[1]의 논문에서는 VCC의 정의와 활용 방안에 초점을 맞추고 있고, Hussain et al.[5]의 논문에서는 VCC의 정의에 관해 초점을 맞추고 있다.

제안하는 VCC 아키텍처 구조는 VANET 클라우드 및 하이퍼바이저 구조, 클라우드 초기 생성 및 사용자 추가, 내부 감시 구조로 이루어져 있다. 지금까지 연구가 이루어지지 않은 초기 네트워크 부분과 전반적인 VCC 보안 부분을 다루고 있어서, 이후 VCC의 연구에 많은 도움을 줄 것이라고 확신한다.

3.1 VANET 클라우드 및 하이퍼바이저 구조

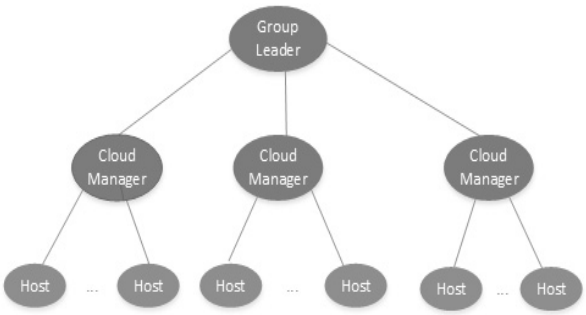

Fig. 1. VCC structure

VANET 클라우드의 구조는 Fig. 1과 같다.

그룹리더는 클라우드 그룹의 대표이고, 클라우드 매니저는 클라우드 서비스를 지원하는 각 차량을 의미한다. 호스

트는 클라우드 서비스 사용자이며 각각의 호스트는 Fig. 2의 구조를 가지고 있다. 그룹리더는 클라우드 매니저를 관리 및 인증하는 역할을 한다. 또한 사용자가 사용 요청할 경우 사용자를 각 클라우드 매니저에 할당하는 역할을 한다. 그리고 일정 간격으로 비콘 메시지를 보내 클라우드 사용을 원하는 차량을 모집한다. 클라우드 매니저는 호스트들을 관리하고, 자원을 호스트와 공유한다. 또한 가용한 자원에 관해 그룹리더에게 항시 보고하여 그룹리더가 사용자를 모집할 때 참고하도록 한다. 기본적으로 Fig. 1과 같은 구조지만 클라우드를 이루는 모든 참여자는 클라우드 매니저이자 호스트가 될 수 있다. 자세히 말하면, 모든 참여자는 자원을 이용하는 동시에 다른 사용자와 공유도 할 수 있다. 그러므로 호스트는 호스트이자 클라우드 매니저가 될 수 있고, 클라우드 매니저는 클라우드 매니저이자 호스트가 될 수 있다. 그룹리더도 마찬가지이다. VCC는 참여하는 모든 유저가 유기적으로 자원을 공유할 수 있다.

Fig. 1과 같이 계층 구조를 가지는 이유는 대역폭 낭비를 막기 위해서이다. 만약 계층 구조를 가지고 있지 않으면 클라우드 서비스 제공자가 계속 비콘 메시지를 보내야 한다. 또한 서비스 사용을 원하는 차량이 브로드캐스트 메시지를 보냈을 시 서비스 제공이 가능한 모든 차량이 개인에게 답을 해줘야 한다. 그러나 Fig. 1처럼 계층 구조를 가지면 그룹리더가 서비스 제공자인 클라우드 매니저들이 할 일을 대행하여 낭비되는 데이터를 줄일 수 있고 전체의 대역폭을 절약할 수 있다. 그리고 클라우드 매니저는 계속 비콘 메시지를 보내는 등의 일을 하지 않아도 되므로 전체적인 클라우드의 효율 또한 증가시킬 수 있다.

가상화 시작 시 쓰이는 하이퍼바이저는 다수의 운영체제를 단일 호스트에서 동시에 실행할 수 있게 하는 프로그램을 말한다. 하이퍼바이저의 종류는 크게 Bare metal과 Hosted 종류로 나뉜다[6].

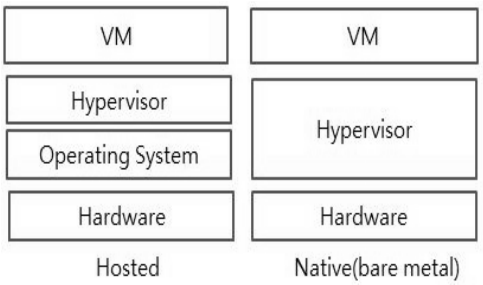

Fig. 2. Hypervisior type[6]

Fig. 2는 하이퍼바이저의 종류를 보여주는 그림이다. Hosted 방식은 하이퍼바이저가 운영체제 호스트의 운영체제 위에서 가상화하는 방식으로 윈도우에서 사용하는 VMWARE가 대표적인 예다. Bare metal은 하드웨어 자체를 가상화하는 방식으로 대표적인 예로는 VMWARE의 ESX가 있다. Hosted 방식은 하이퍼바이저가 호스트의 운영체제 위에서 가상화되기 때문에 공격자가 호스트의 운영체제를 공격하여 권한을 얻거나 방해하여 전체의 효율을 떨어뜨리는 공격이 가능하다. 그러나 Bare metal은 하드웨어 자체를 가상화하기 때문에 이러한 공격이 불가능하다. 그러므로 가상화 시 Bare metal 하이퍼바이저 형식을 사용하는 것을 제안한다.

3.2 클라우드 초기 생성 및 지원자/사용자 추가

클라우드를 사용하기 위해서는 각 지원자의 인증이 필요하다. 여기서 지원자란 클라우드 서비스를 제공하려는 사람/차량을 의미한다. 그러나 인증 전, 지원자들의 클라우드 형성 가능성에 관한 검증이 필요하다. 만약 인증을 먼저 한 후 다수의 클라우드 형성에 부적합한 지원자들이 나오면 인증에 관한 많은 대역폭이 낭비되기 때문이다[5]. 그래서 인증 전 통신 파워, 플랫폼 버전과 같은 사전 검사를 거친다. 그 후 인증을 위해서 지원자들이 생기면 먼저 지원자 사이에서 대표를 뽑는데[5], 안정성을 위해 가장 가까운 RSU와 지원자 간의 예상 연결 지속시간이 가장 긴 지원자를 대표로 한다. 수식은 Su et al.의 논문[7]을 인용하였다. 위 논문에서는 두 개의 움직이는 노드의 예상 연결시간을 구했지만, 본 논문에서는 RSU가 고정되어있으므로, 목적지가 고정되어있게 식을 응용하여 사용하였다. 대표를 뽑을 시에는 각 후보들이 위의 식을 사용하여 Dt값을 구한 뒤, 그 값을 브로드캐스트한다. 그중 가장 높은 Dt값을 가진 후보를 그룹리더로 뽑는 방식을 사용한다.

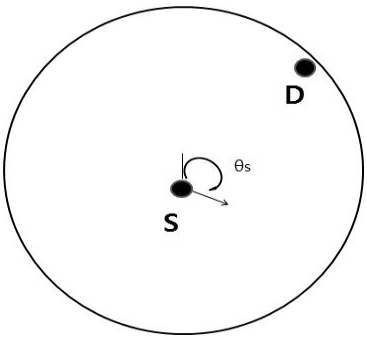

Fig. 3. link between vehicle and RSU[7]

$$D_t = \frac{-(ab+cd)+\sqrt{(a^2+c^2)r^2-(ad-bc)^2}}{a^2+c^2}$$

$$a = v_i\cos\theta_i,$$
$$b = x_i - x_j,$$
$$c = v_i\sin\theta_i,$$
$$d = y_i - y_j$$

수식에서 Dt는 예상 지속시간이고, r은 전송 반경이다. V는 속도를 의미하며 $S(x_i, y_i)$, $D(x_j, y_j)$는 좌표를 가진다. θ는 자동자의 방향에 따른 각도를 의미하는 것으로, Fig. 3에서 볼 때 자동차는 오른쪽 하단 대각선 방향으로 이동하는 중이다.

a는 S의 x좌표의 속도와 진행방향 각도의 코사인값을 곱

한 것이며, b는 S와 D의 x좌표의 차이값이다. c는 S의 x좌표의 속도와 진행방향 각도의 사인값을 곱한 것이며 d는 S와 D의 y좌표값의 차이이다.

Fig. 3은 자동차와 RSU 간의 연결을 보여주는 그림이다. 그림에서 s는 지원자, d는 RSU이고, 원은 전송 반경을 의미한다. 그림처럼 RSU는 지원자의 전송 반경 안에 항상 들어와 있어야 한다.

대표가 뽑힌 후 대표는 공개키 알고리즘[9]을 통해 RSU와 통신하여 먼저 인증과정을 거친다. 이때 각 자동차는 만들어질 당시에 자동차 안에 CA의 공개키를 가지고 있고 RSU는 CA의 공개키와 비밀키를 알고 있다고 가정한다.

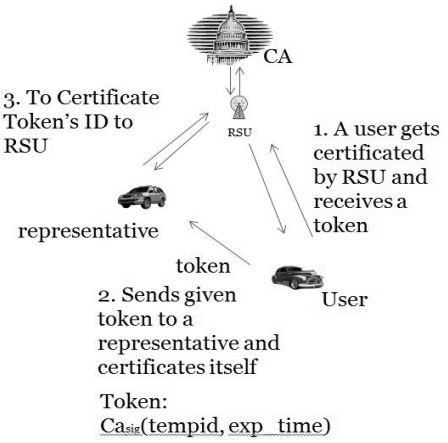

Fig. 4. Key certification process

이때 대표의 지원자나 사용자 추가에 관한 오버헤드를 줄이고, Denial of Service(DoS) 공격을 예방하기 위해 지원자들은 RSU를 통해 받은 토큰을 통해 본인을 인증한다. 토큰은 각 지원자들이 대표에게 인증받기 바로 직전에 RSU와의 통신으로 본인을 인증받고 획득한다. 토큰은 CAsig(일회용 id, 만기 시간)로 이루어져 있다. CAsig는 CA의 서명이며 만기 시간은 재사용 공격 방지를 위해 사용한다. 지원자는 받은 일회용 id와 CA의 서명으로 암호화되어있는 토큰을 대표에게 주고 대표는 받은 일회용 id와 토큰의 id를 대조하여 지원자를 인증한다. Fig. 4는 이러한 인증과정을 나타낸 그림이다. Fig. 4에서 1번은 사용자가 RSU를 통해 CA에게 본인을 인증받고 토큰을 받는 것을 나타내고 있다. 2번은 사용자가 RSU를 통해 받은 토큰을 대표에게 보내고 대표에게 자신을 인증받는 과정을 나타내고 있다. 3번은 대표가 CA에게 사용자에게서 받은 토큰에 들어있는 일회용 id와 토큰 id가 일치하는지 확인받는 과정을 나타내고 있다.

3.3 통신의 프라이버시 보호

VANET의 특성상 각 자동차의 위치 및 자동차 정보 등이 통신 중 노출될 위험이 크다. 이러한 통신의 프라이버시 보호를 위해 Group Signature[13]를 사용한다. Group Signature란 그룹의 구성원이 통신 중 자신을 숨기기 위해 그룹의 서명으로 메시지를 서명하는 기술이다. 그룹 리더(대표)가 public key, private key 한 쌍으로 이루어진 그룹 키와 public key, private key 한 쌍으로 이루어진 그룹 관리키를 생성한다. Group manage key는 그룹 리더가 오염된 구성원을 찾는데 사용된다. 각 사용자는 자신만의 독특한 그룹 관리키를 그룹 리더와의 통신을 통해 할당받는다[14]. 구성원들은 보낼 메시지를 그룹 키와 그룹 관리키를 사용하여 차례로 암호화하여 전송한다. 상대방은 모든 사용자의 그룹 키가 동일하므로 그룹 구성원 중 어떤 사용자가 메시지를 보냈는지 알 수 없다. 또한 만약 그룹 구성원이 오염되었을 때는 그룹 리더가 그룹 관리키를 이용하여 찾아낼 수 있다.

기존의 VANET에서 group signature 방식은 대표를 뽑는 것이 힘들고, 대표의 키 생성 과정에서 오버헤드가 많이 생긴다는 단점이 있었다. 그러나 VCC에서는 대표가 이미 뽑혀있고, 대표가 모든 클라우드 구성원의 남는 자원을 공유하여 쓸 수 있기 때문에 기존의 한계를 극복할 수 있다.

3.4 사용

지금까지 나온 VCC의 경우는 교통 혼잡 방지를 위한 신호등 시간 조절, 주차장에 주차되어있는 차의 자원을 이용하여 소규모 사업 이용, 대규모 대피를 위한 인프라 구축 등에 사용할 수 있다고 한다[1]. 본 저자는 그 이외에 응급 자동차를 위한 빠른 차선 확보, 무인 자동차를 위한 인프라 구축에도 사용할 수 있다고 생각한다. 기존의 VANET는 전송할 수 있는 거리가 짧아서, 모든 차가 서행하는 경우 멀리서 빠르게 달려오는 응급차를 위한 차선 확보에 맞지 않는 점이 있었다. 차선이 막히는 경우에는 차선 확보를 위해 여유 시간이 걸리기 때문이다. 그러나 VCC의 경우에는 먼 거리에 있는 차도 빠르게 상황을 확인할 수 있기 때문에, 차선 확보에 조금 더 유리할 수 있다. 현재 무인 자동차는 센서 또는 위성에 의지하고 있다. 그러나 센서의 경우는, 비가 많이 와서 차선이 잘 보이지 않을 때나, 차선이 유실된 경우에 센서의 차선이나 도로 감지에 문제가 생겨서 위험할 수 있다. 또한 위성의 경우에도, 터널이나 다리 밑과 같은 위성신호가 끊기는 곳에서의 위험, 날씨에 따른 신호 감지 저하 등의 위험이 있다. 그러나 VCC를 이용하여 차선, 도로의 크기 등 목적지까지 가는 도로의 자세한 정보를 일정 부분 다운로드받을 수 있다면 센서나 위성만 이용한 것보다 훨씬 안전하게 주행할 수 있다.

4. 성능 평가

성능 평가는 클라우드 초기 생성 및 지원자 및 사용자를 추가할 때 사용되는 지원자와 RSU 간의 예상 연결 지속시간을 검증하기 위하여 진행하였다. 시뮬레이터는 NS2로 하였으며, 50개의 노드를 사용하였다. 차의 속도는 12~24m/s로 하였으며 통신 프로토콜은 AODV를 사용하였다. 자세한 초기설정에 관한 설명은 Table 1과 같다.

Table 1. Simulation Setup

Parameter	Value
Simulator	NS2
Channel type	Wireless
Routing protocol	AODV
Network Interface Type	Physical Wireless
MAC protocol	IEEE 802.11
Traffic type	UDP
Queue Length	50 packets
Radio Propagation Model	Two Ray Ground
Antenna	Omni Antenna
Topology dimension	1500 * 1500m
Speed	12~24m/s
Node number	50

시뮬레이션 시나리오 맵은 Fig. 5와 같다. 가로, 세로 각 2줄의 도로를 만들고, 도로의 중간에 RSU를 설치하였다. RSU는 도로에 지나다니는 모든 자동차에 매 초마다 브로드캐스트 메시지를 보내고, 응답을 받으며 모든 자동차와 링크를 유지한다. 성능평가는 NS2를 이용하여 측정한 연결 지속시간값과 3.2에 있는 예상 연결시간 공식을 이용한 값을 비교하는 것으로 진행하였다.

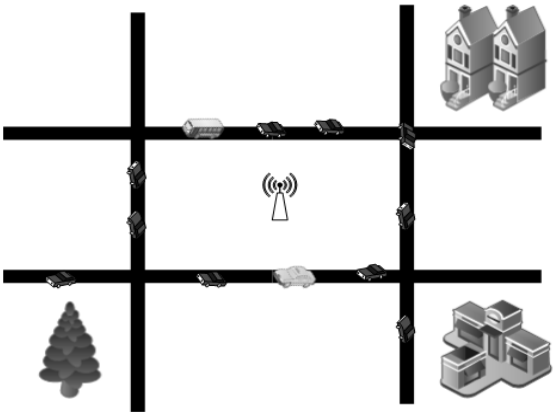

Fig. 5. Simulation scenario map

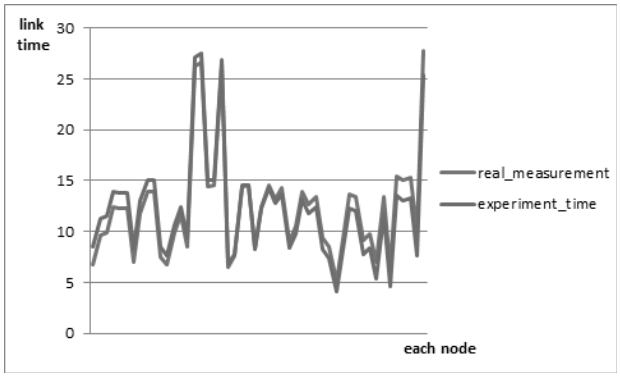

Fig. 6. Simulation result

Fig. 6은 성능 평가 결과이며 가로축은 각 노드를 의미하고, 세로축은 각 노드와 RSU 간의 연결 지속시간(초)이다. real_measurement는 NS2를 이용하여 측정한 실제 연결지속시간값이며, experiment_time은 3.2에 있는 공식을 이용하여 나온 값이다. 두 개의 값을 비교한 결과 대부분 일치하는 것을 확인할 수 있으며, 약간의 차이가 나는 것은 노드 간의 네트워크 충돌로 인한 오차로 추정된다.

5. 결 론

VCC 기술이 현실화되면 기존에 연구된 VANET 기술보다 훨씬 더 유용하게 사용될 것으로 예상된다. 현재의 VANET 기술은 먼 곳의 상황까지 예측하는 것에는 한계가 있지만 VCC의 경우에는 클라우드를 통해 먼 곳의 상황을 바로 알 수 있기 때문에 훨씬 더 빨리 대응이 가능하기 때문이다. 그러나 연구된 기간이 짧아서 아직 많은 연구가 더 필요한 상황이다. 본 논문은 지금까지 연구되지 않은 보안적인 측면의 VCC 아키텍처를 제안하였다. 이러한 아키텍처를 사용하면 보다 효율적이고 안전한 VCC 구축이 가능할 것이라고 예상된다.

앞으로 내부 감시 구조와 잦은 핸드오버를 효율적으로 관리하는 방법, 효율적인 키 관리 방법, 핸드오버 시 기존에 유지하던 클라우드 유저들의 자료 처리 문제 등에 관해 더 연구할 예정이다.

References

[1] Olariu, Stephan, Ismail Khalil, and Mahmoud Abuelela, "Taking VANET to the clouds," *International Journal of Pervasive Computing and Communications,* Vol.7, No.1, pp.7-21, 2011.

[2] El Zarki, Magda, et al., "Security issues in a future vehicular network," *European Wireless,* Vol.2, 2002.

[3] Armbrust, Michael, et al., "A view of cloud computing," *Communications of the ACM,* Vol.53, No.4, pp.50-58, 2010.

[4] Olariu, Stephan, Tihomir Hristov, and Gongjun Yan, "The next paradigm shift: from vehicular networks to vehicular clouds," Basagni, S., Conti, M., and Giordano, S. Stojmenovic, I) (Eds), "Mobile Ad hoc networking: the cutting edge directions," Wiley and Sons, New York, 2012.

[5] Hussain, Rasheed, et al., "Rethinking vehicular communications: merging VANET with cloud computing," Cloud Computing Technology and Science (CloudCom), *2012 IEEE 4th International Conference on.* IEEE, 2012.

[6] Popek, Gerald J., Robert P. Goldberg, "Formal requirements for virtualizable third generation architectures," *Communications of the ACM,* Vol.17, No.7, pp.412-421, 1974.

[7] Su, William, Sung Ju Lee, and Mario Gerla, "Mobility prediction and routing in ad hoc wireless networks," *International Journal of Network Management,* Vol.11, No.1, pp.3-30, 2001.

[8] Noureddine, Hadi, et al., "A New Link Lifetime Prediction Method for Greedy and Contention-based Routing in Mobile Ad Hoc Networks," Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on. IEEE, 2010.

[9] Rivest, Ronald L., Adi Shamir, and Len Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM,* Vol.21, No.2, pp.120-126, 1978.

[10] Rowland, Craig H., "Intrusion detection system," U.S. Patent, No.6, 405,318., 11 Jun., 2002.

[11] Modi, Chirag, et al., "A survey of intrusion detection techniques in cloud," *Journal of Network and Computer Applications,* Vol.36, No.1, pp.42-57, 2013.

[12] Qiao, Yan, et al., "Anomaly intrusion detection method based on HMM," *Electronics Letters,* Vol.38, No.13, pp.663-664, 2002.

[13] Chaum, David, and Eugène Van Heyst, "Group signatures," Advances in Cryptology—EUROCRYPT'91. Springer Berlin Heidelberg, 1991.

[14] Guo, Jinhua, John P. Baugh, and Shengquan Wang, "A group signature based secure and privacy-preserving vehicular communication framework," Mobile Networking for Vehicular Environments 2007, pp.103-108, 2007.

김 태 형

e-mail : thkim85@emerald.yonsei.ac.kr
2013년 동국대학교 컴퓨터공학과(학사)
2013년~현 재 연세대학교 컴퓨터과학과 석사과정
관심분야 : 차량 간 통신 네트워크, 차량 간 클라우드 컴퓨팅

송 주 석

e-mail : jssong@emerald.yonsei.ac.kr
1976년 서울대학교 전기공학과(학사)
1979년 한국과학기술원 전기전자공학 (공학석사)
1988년 Univ. of California at Berkeley, 컴퓨터과학(박사)
1988년~1989년 Assistant Professor in Naval Postgraduate School
1989년~현 재 연세대학교 컴퓨터과학과 정교수
2006년 한국정보보호학회 회장 역임
관심분야 : 유/무선 통신, 정보보호 등