

VANET내 RSU에서 (m,k)-firm 기반 요청 데이터 처리를 위한 스케줄링 알고리즘

남재현*

A Scheduling Algorithm for Servicing (m,k)-firm-based Request Data from RSU Using in VANETs

Jae-Hyun Nam*

Department of Computer Education, Silla University, Busan 469-58, Korea

요 약

VANET은 차량들을 노드로 간주하는 에드혹 네트워크이고, 이들 노드들은 도로상에서 RSU들과 통신을 수행한다. VANET환경에서 요청한 데이터를 요청한 차량에게 정확하고 적절하게 전달하기 위해서는 스케줄링이 매우 중요하다. 또한 차량이 해당 RSU 영역내에 존재할 때 차량이 요청한 데이터를 수신할 수 있게 하는 것이 중요하다. RSU는 스케줄링 알고리즘에 따라 처리할 우선순위가 가장 높은 데이터를 선택한다. 본 논문에서는 RSU가 과부하 때문에 처리하지 못한 데이터를 다른 RSU로 전달되는 환경에서 우선순위가 제일 높은 요청 데이터를 선택할 수 있는 스케줄링 알고리즘을 제안한다. 요청 데이터의 우선순위를 계산하기 위해 수정된 DBP 기법을 이용하였다. 시뮬레이션 결과 다양한 조건하에서 제안된 기법이 다른 스케줄링 기법보다 성능이 우수함을 볼 수 있다.

ABSTRACT

VANET is ad-hoc network in which vehicles are treated as node and these nodes are communicating with each other as well as RSUs on the road. In VANET, scheduling is a very important issue as the request data are to be delivered to the recipient vehicle properly and accurately. It is also critical that any request data which is sent to the vehicle is received by it in the time period for which the vehicle will be in range of RSU. RSU picks the one with the highest priority to serve based on the scheduling scheme. In this paper, we propose a scheduling algorithm to select the one with the highest priority in RSU, in which an RSU can transfer the overload requests to other RSUs. We use the modified DBP scheme to calculate the priority of request data. Simulation results show that our scheme outperforms other scheduling schemes under various conditions.

키워드 : VANET, 데드라인, RSU, 우선순위 스케줄링, DBP

Key word : VANET, Deadline, RSU, Priority Scheduling, DBP

Received 02 October 2015, Revised 30 October 2015, Accepted 09 November 2015

* Corresponding Author Jae-Hyun Nam(E-mail:jhnam@silla.ac.kr, Tel:+82-51-999-5627)

Department of Computer Education, Silla University, Busan 469-58, Korea

Open Access <http://dx.doi.org/10.6109/jkiice.2015.19.11.2665>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

차량용 애드혹 네트워크, 또는 VANET(Vehicular Ad-Hoc Networks)는 모바일 네트워크를 구성하기 위해 네트워크 내의 노드로써 이동용 차량을 사용하는 기술이다. VANET에서는 실시간으로 차량들에게 정보를 전달하는 기능이 핵심이다. 차량들은 IEEE 108.11 AP(Access Point)와 같은 RSU(Road Side Unit)를 통해 RSU에 저장된 데이터를 접근하거나 인터넷 연결을 할 수 있다.

VANET에서 차량으로부터의 데이터 접근 요청은 요청을 발생한 차량이 RSU 영역을 벗어나기 전에 완전하게 서비스를 받을 수 있어야 하기 때문에 엄격한 시간제약 특성을 지닌다[1]. 따라서 데이터 접근 요청을 하는 차량 수가 많을수록 데드라인 내에 데이터 접근을 성공시키기 위한 스케줄링 알고리즘 설계는 매우 중요해진다. 이러한 문제를 해결하기 위해 [1]에서는 D*S/N 알고리즘을 제안했다. 하지만 실제 도로상에는 하나의 RSU만 존재할 수 없기 때문에 여러 개의 RSU들과 같이 작업할 수 있는 스케줄링 알고리즘도 제안되었다[2].

또한 다양한 제약사항이 있는 무선 센서네트워크에서 실시간 데이터 전송 서비스에 적합한 (m,k)-firm 데드라인 모델은 스트림은 연속적으로 전송된 k개의 패킷 중에서 m개의 패킷이 데드라인 안에 전송될 경우 실시간 서비스를 만족하는 모델이다. (m,k)-firm 데드라인 모델에서 δ_{ij} 를 i번째 태스크의 j번째 작업이라고 하고, δ_{ij} 의 (m,k)-firm 데드라인 변수를 각각 m_i, k_i 라고 할 때, δ_{ij} 을 포함하여 이전에 도착한 가장 가까운 k_i 개의 작업을 나타낸 열을 $(\delta_{ij-k+1}, \dots, \delta_{ij-1}, \delta_{ij})$ 로 나타낼 수 있다. 이때 작업열의 각 작업이 데드라인을 만족시킴을 M, 만족시키지 못할 경우를 m이라고 표시할 경우, M의 개수가 m_i 보다 작다면 동적 실패(Dynamic Failure)가 발생했음을 나타낸다. 또 동적 실패가 발생하기까지 연속적으로 놓쳐야 하는 데드라인의 개수를 동적 실패까지의 거리로 표현한다. Hamdaoui 등은 이와 같은 거리를 이용하여 동적 실패까지 짧은 거리에 있는 태스크를 먼저 스케줄링 하는 DBP(Distance Based Priority)을 제안하였다[3].

본 논문에서는 데드라인을 만족하지 못하여 자신의 RSU내에서 처리되지 못하고 인접 RSU로 전달되는 요

청 데이터와 정상적으로 해당 RSU로 전달되는 데이터 간에 우선적으로 처리하는 데이터를 선택하는 우선순위 스케줄링 기법을 제안한다. 제안된 기법은 하나의 RSU내에서 차량으로부터 오는 요청을 처리하기 위해 2개의 큐를 두어 각 큐 헤더에 있는 데이터의 요청 우선순위를 계산한 후 처리할 요청데이터를 선택하는 기법이다. 우선순위를 계산할 때 요청데이터의 크기, 데드라인, 그리고 해당 영역에서 처리되지 못하고 인접 RSU로 전달된 횟수 등을 고려하였다. 성능평가 결과 본 논문에서 제안된 기법이 서비스율과 데드라인 실패율이 기존의 기법보다 뛰어나다는 것을 볼 수 있었다.

본 논문의 구성은 다음과 같다. II장에서는 관련연구에 대해서 기술하였고 III장에서는 본 논문에서 제안한 기법에 대해 기술하였다. IV장은 성능평가에 대해서 기술하였고, 끝으로 V장에서는 결론을 나타내었다.

II. 관련연구

Fig. 1은 교차로 상에서 OMNI 안테나를 사용하는 하나의 RSU가 존재하는 시스템 모델이다[1].

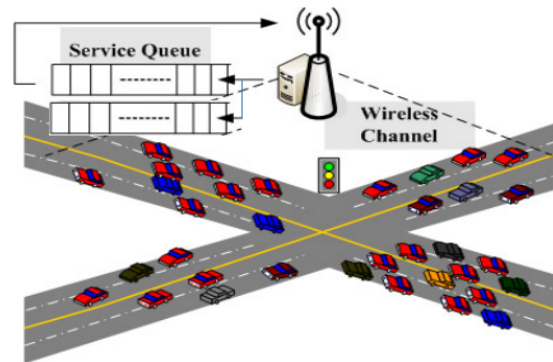


Fig. 1 The Architecture of Vehicle-Roadside Service Scheduling

Fig. 1에서 보는바와 같이 다수의 차량들이 RSU 영역내에 들어오면 RSU와 데이터 교환을 시작할 수 있다. RSU는 non-preemptive 서비스를 제공하는 서버로서 동작한다. 모든 차량들은 데이터 접근을 원할 경우 RSU로 요청메시지를 전송한다. [1]에서 제안한 요청메시지 형식은 4-tuple $\langle v-id, d-id, op, deadline \rangle$ 을 사용하였다.

v-id는 차량식별자, d-id는 요청 데이터 아이템 식별자, op는 업로드와 다운로드를 구별하는 동작, deadline은 요청한 차량이 RSU 영역을 빠져 나오기 전에 처리해야 될 시간을 말한다. 하지만 4-tuple 형식의 요청메시지에는 데이터 타입이 빠져있어 인접 RSU로부터 전달된 데이터와 일반데이터를 구분할 수 없다. 본 논문에서는 4-tuple 형식의 요청메시지에 데이터 타입을 삽입하여 5-tuple <v-id, d-id, d-type, op, deadline> 형식의 메시지를 사용한다. 여기서 d-type은 전달된 데이터와 일반데이터를 구분하는 필드이다. 모든 요청은 RSU 서버에 도달하면 큐에 저장된다. [1]에서는 데이터타입을 고려하지 않았기 때문에 큐를 하나만 사용했지만 본 논문에서는 데이터 타입에 따라 별도의 큐에 저장되도록 했다. RSU로 들어오는 요청 데이터 아이템들을 d-type 필드를 통해 일반 데이터와 전달된 데이터로 구분하고, 구분된 데이터 아이템들은 각각 별도의 큐에 FCFS 스케줄링 기법으로 저장된다.

RSU는 아래와 같은 스케줄링 기법을 통해 큐에 저장된 요청 데이터 아이템을 선택한다.

- First Come First Serve(FCFS) : 먼저 도착한 요청을 먼저 서비스한다. 이 기법은 데이터 크기와 요청 데드라인은 고려하지 않는다.
- First Deadline First(FDF) : 가장 긴급한 요청을 먼저 서비스한다. 이 기법은 데이터 크기를 고려하지 않기 때문에 서비스 시간이 무시된다.
- Smallest Data Size First(SDF) : 가장 적은 크기를 지닌 데이터를 먼저 처리한다. 이 기법은 긴급하게 처리해야 될 요청을 처리하지 못한다.

[1, 5]에서 제안한 기법들은 기존의 스케줄링 기법들의 문제점을 해결하기 위해 데이터 크기와 데드라인을 고려한 스케줄링 알고리즘을 제안했다. [1]에서 제안한 알고리즘은 FDF와 SDF를 결합시켰다. 이 알고리즘은 모든 요청들에 대해 DS_Value를 계산하여 가장 적은 DS_Value를 지닌 요청을 우선적으로 처리한다.

$$DS_Value = (DeadLine - CurrentClock) \times DataSize \quad (1)$$

[1]은 스케줄링 우선순위를 결정하기 위해 데드라인과 데이터 크기만 고려하였고, [2]에서는 데드라인, 데

이터크기, 그리고 데이터 타입까지 고려한 2단계 스케줄링 기법을 제안했다. 하지만 이러한 기법들은 하나의 RSU내에서 서비스를 처리하는 것을 가정하고 있다.

차량들이 하나의 RSU내에 머무는 시간이 매우 짧기 때문에 다수의 RSU를 네트워크로 연결하고, RSU간에서 서로 통신하는 환경에서 하나의 RSU내에서 데드라인 내에 처리하지 못한 요청데이터를 처리할 수 있는 방안이 필요하다[5]. [5]에서는 통신범위를 벗어나기 전에 차량의 요청을 처리하기 위해 [1]에서 제안한 DS_Value값을 이용하여 우선순위를 결정한다.

하지만 우선순위를 결정할 때 요청 데이터의 데드라인, 크기뿐만 아니라 해당 RSU내에서 처리되지 못하고 다른 RSU로 전달된 횟수로 고려해야 한다. 이러한 점을 반영하기 위해 본 논문에서는 [3]에서 제안한 거리값과 [1]에서 제안한 값을 요청데이터의 우선순위를 결정하는 기법을 제안한다.

III. 제안 기법

본 장에서는 차량으로부터 전송된 요청데이터를 처리하기 위해 필요한 파라미터와 (m,k)-firm 스케줄링 기법을 적용한 기법에 대해 설명한다.

RSU들은 백본에 연결되어 있고 각 RSU는 다른 RSU들의 부하량을 알 수 있는 방법이 있다고 가정한다 [6]. 각 RSU는 주기적으로 자신의 부하량 정보를 이웃 RSU들에게 브로드캐스트한다.

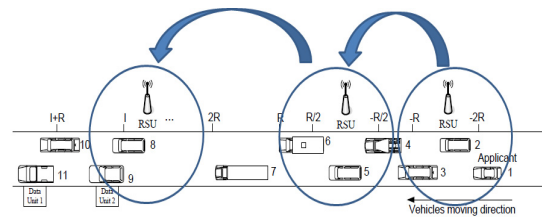


Fig. 2 Request data transfer in multiple RSUs

Fig. 2는 현재 RSU에서 부하가 많이 발생하여 차량의 요청을 데드라인 내에 처리할 수 없을 경우 차량이 이동하는 인접 RSU로 요청 데이터를 전송하는 과정을 나타낸 것이다. 따라서 차량으로부터 전송된 데이터들은 인접한 RSU로부터 전송된 데이터 아이템과 정상 데이

터 아이템들의 혼합으로 구성되어 있기 때문에 다수 차량의 요청을 처리하기 위해서는 우선순위 부여가 필요하다.

RSU에서 차량의 요청 데이터를 처리하기 위해 몇 가지 정의가 필요하다. Unserved_set(US)은 차량으로부터 요청은 받았지만 아직 서비스가 되지 않은 모든 데이터 아이템들을 담고 있는 셋(set)이다. US내의 모든 데이터 아이템들은 요청 데드라인 전에 처리되어야 하고 만약 처리되지 못할 경우 fail로 간주된다.

Request Valid Time(RVT)은 요청 생성과 요청 데드라인 사이의 기간으로 정의된다. 즉, Request deadline = Request generation time + RVT 이다. 만약 데드라인을 만족하지 못할 경우 해당 요청은 invalid 상태가 된다. 데드라인을 만족할 수 없는 데이터는 차량이 이동하는 방향의 인접 RSU로 전달된다. Request Priority (RP)는 하나의 요청에 대한 우선순위를 나타낸 것이다. 이것은 다음과 같이 나타낼 수 있다.

$$RP_N = \text{Unserved_size} * (\text{Deadline} - \text{Current Time}) \quad (2)$$

여기서 RP_N 은 일반 데이터 큐에서의 우선순위를 의미하고, Deadline은 하나의 RSU내에서 차량의 요청을 처리해야 될 시간이다. Unserved_size는 US내에 속해 있는 데이터 아이템들의 크기들을 합한 값이다.

각 스케줄링 라운드동안 RSU는 대기 요청의 전체 set을 스캔한 후 가장 우선순위가 높은 요청 데이터를 선택한다. 뿐만 아니라 데드라인 전에 처리할 수 없는 요청 데이터들은 삭제하거나 인접한 RSU로 전달시킨다. 이 때 고려사항은 인접 RSU의 workload 상태, 요청을 발생시킨 차량 속도, 요청데이터의 데드라인 등이다. 각 RSU는 차량의 움직임을 예측하여 요청을 전달할 인접 RSU를 선택한다. 전달된 요청 데이터는 인접 RSU내의 별도의 큐에 저장되어 해당 요청을 발생시킨 차량이 해당 RSU내로 진입할 때까지 대기한다. 만약 특정 차량의 요청에 대한 우선순위가 낮을 경우 요청한 데이터를 계속적으로 수신하지 못하는 기아 현상이 발생할 수 있다[7].

본 논문에서는 이러한 문제를 해결하기 위해 RSU내에서 요청데이터를 처리할 때 인접한 RSU로부터 전달된 요청 데이터의 우선순위를 변경시키기 위해 [3]에서 제안한 DBP 기법을 수정하여 이용하였다.

각 RSU는 인접한 RSU로부터 전달된 요청데이터를 처리하기 위해 하나의 테이블을 유지한다. 테이블 엔트리는 (Neighbor RSU ID, Position, Request Deadline, (m,k))이다. 여기서 Neighbor RSU ID는 요청데이터를 전달한 RSU의 ID이고, Position은 방향값, Request Deadline은 요청데이터를 해당 영역내에서 처리해야 될 시간을 의미한다. 그리고 (m,k)는 데드라인 내에 처리되지 못하고 인접 RSU로 전달된 형태를 나타낸 일련의 스트링이다. 하나의 RSU내에 처리해야 될 데이터를 데드라인 내에 처리하지 못하고 인접한 RSU로 전달할 경우 (m,k)는 0으로 설정한다. 이 데이터가 다시 데드라인 내에 처리되지 못하고 다른 RSU로 전달되면 (m,k)는 00으로 설정된다. 이러한 과정은 전달된 데이터가 RSU 내에서 처리될 때까지 계속된다. 데드라인 내에 처리되지 못해 인접한 RSU로 전달되어야 하는 데이터는 무한정 전달될 수 없기 때문에 본 논문에서는 k변만 전달된다고 가정하였다. 전달받은 데이터 처리를 위한 우선순위 계산을 위해 수정된 DBP는 다음과 같다.

$$DBP = \text{len}((m,k)) + 1 \quad (3)$$

$$RP_T = \text{Unserved_size} * (\text{Deadline} - \text{Current Time}) / DBP \quad (4)$$

여기서 RP_T 는 전달 데이터 큐에서의 우선순위를 의미한다.

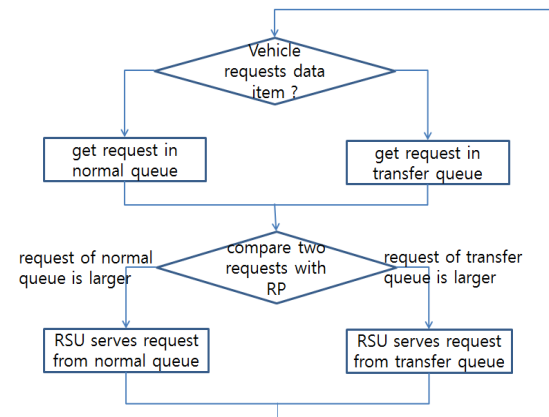


Fig. 3 Flow chart of the proposed scheduling

RSU내에서 전달받은 데이터와 반경내에서 요청된 정상 데이터를 처리하기 위한 우선순위는 Fig. 3과 같이 결정된다. Fig. 3은 [4]에서 제안한 2단계 스케줄링 기법과 유사한 흐름도를 보여주고 있다. 하지만 [4]에서는 RSU가 차량으로부터 데이터 요청을 받을 경우 우선순위를 결정하기 위해 $D*S/W$ 값을 이용하였지만 본 논문에서는 식 (2)와 식 (4)를 통해 얻은 RP_N 과 RP_T 값을 비교하였다. RSU가 차량으로부터 요청을 받을 경우 일반 데이터 큐와 전달 큐의 헤더에서 각 요청 데이터를 가져온 후 RP 값을 계산한다. 이 때 일반 데이터 큐의 요청 데이터의 RP_N 값은 식 (2)를 통해 계산되고 전달 큐의 요청 데이터의 RP_T 값은 식 (4)를 이용하여 계산된다. 계산된 RP 값들을 비교한 후 적은 값을 지닌 요청 데이터를 먼저 처리한다.

IV. 성능평가

본 논문에서는 제안된 알고리즘의 성능을 평가하기 위해 MATLAB을 이용하여 시뮬레이션을 수행했다. 차량 움직임 패턴은 Manhattan Mobility Model 특성을 따른다[8]. 차량들은 시뮬레이션 기간 동안 다운로드 또는 업로드 요청을 생성한다.

각 RSU들은 자신의 대기 큐를 가지면서 차량들의 요청을 처리하기 위해 스케줄링 알고리즘을 수행하고, 해당 RSU에서 처리하지 못한 요청 데이터들은 본 논문에서 제시한 알고리즘을 이용하여 인접한 RSU에 전달함으로써 부하량을 분산시킨다. 시뮬레이션을 단순화하기 위해 차량들이 접근 가능한 데이터 셋은 고정되어 있다고 가정한다. 즉, 차량들은 데이터 셋에 새로운 엔트리를 추가하지 못한다. 시뮬레이션에 사용된 파라메타와 기본 값은 Table. 1에 나타나 있다.

Table. 1 Simulation Parameters

Parameter	Value
Number of Vehicle	100 ~ 400
Transmission Rate	64kbps
Vehicle Velocity	0~40km/h
Wireless Coverage Range	200m
Packet Size	10 ~ 512KByte
Traffic Type	CBR

Fig. 4는 차량으로부터 전달되는 요청데이터 간격에 따른 서비스율을 나타낸 것이다. 요청 간격이 줄어들면 많은 요청이 생성되는 것이기 때문에 RSU에 과중한 워크로드를 발생시킨다. 그림에서 보면 RSU 워크로드가 증가함에 따라 본 논문에서 제시한 알고리즘의 성능이 다른 기법보다 감소하는 비율이 느리게 나타난다.

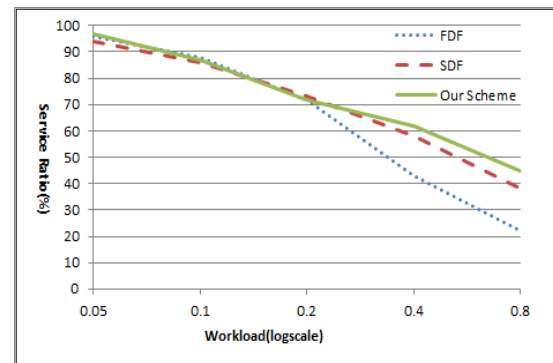


Fig. 4 Service ratio for FDF, SDF and Our scheme

Fig. 5는 차량 수에 따른 데드라인 실패율을 나타낸 것이다. 차량 수가 증가하면 일정기간동안 많은 요청이 발생되기 때문에 데드라인 실패가 발생할 확률이 증가한다. 그림에서 보면 제안할 알고리즘이 기존 기법보다 전체적으로 성능이 우수하다는 것을 알 수 있다.

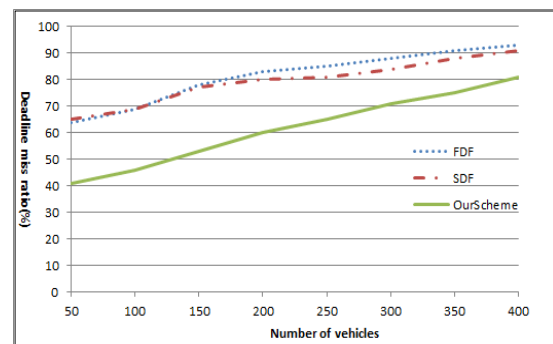


Fig. 5 Deadline miss ratio to the number of vehicles

V. 결 론

본 논문에서는 RSU내 과중한 워크로드로 인해 차량으로부터 전달받은 요청데이터 서비스 실패 확률을 감소하기 위한 스케줄링 알고리즘을 제안했다. 각 RSU는 데드라인 전에 처리할 수 없는 요청 데이터들은 삭제하거나 인접한 RSU로 전달시킨다. 만약 특정 차량의 요청에 대한 우선순위가 낮을 경우 요청한 데이터를 계속적으로 수신하지 못하는 기아 현상이 발생할 수 있다.

본 논문에서는 RSU내에서 요청데이터를 처리할 때 인접한 RSU로부터 전달된 요청 데이터의 우선순위를 변경시키기 위해 DBP 기법을 적용하여 기아 현상을 완화하였다. 시뮬레이션 결과 본 논문에서 제안한 기법의 성능이 FDF나 SDF보다 우수함을 보였다.

REFERENCES

[1] Y.Zhang, J. Zhao, and G. Cao, "On Scheduling Vehicle-Roadside Data Access," in *Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks*, New York, pp. 9-18, 2007.

[2] C. Lochert, B. Scheuermann, M. Caliskan, and M. Mauve,

"The feasibility of information dissemination in vehicular ad-hoc networks," in *Proc. of Wireless on Demand Network Systems and Services (WONS'07)*, pp. 92-99, 2007.

[3] M. Hamdaoui and P. Ramanathan, "A dynamic priority assignment technique for streams with (m,k)-firm deadlines," *IEEE T. on Computers*, pp.1443-1451, 1995.

[4] S.Sahebgharani and M.Shahverdy, "A Scheduling Algorithm for Downloading Data from RSU Using Multicast Technique," *International Journal of Advanced Computer Science*, Vol. 4, No. 3, pp. 115-120, 2014.

[5] G.G.M.N. Ali, and E.Chan "Co-operative data access in multiple Road Side Units (RSUs)-based Vehicular Ad Hoc Networks (VANETs)," *Australasian Telecommunication Networks and Applications Conference (ATNAC)*, pp. 1-6, 2011.

[6] Luan, T.H., Ling, X., and Shen, X., "MAC in motion: Impact of mobility on the mac of drive-thru internet", *IEEE Transactions on Mobile Computing 11(2)*, pp. 305-319, 2012.

[7] K. Liu and V. C. S. Lee, "Analysis of data scheduling for multi-item requests in multi-channel on-demand broadcast environments," *Technical Report, Computer Science Department, City University of Hong Kong*, 2008.

[8] F. Bai, N. Sadagopan and A. Helmy, "IMPORTANT: a framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks", in *Proc. of IEEE INFOCOM, vol.2*, pp. 825-835, 2003.



남재현(Jae-Hyun Nam)

1989 부산대 컴퓨터공학과(학사)
 1992 부산대 컴퓨터공학과(공학석사)
 2002 부산대 컴퓨터공학과(공학박사)
 1993 ~ 2002 동주대학 조교수
 2002 ~ 현재 신라대학교 컴퓨터교육과 교수
 ※ 관심분야 : 무선센서네트워크, VANET