

인프라구조 기반 이동 애드혹 네트워크의 이동성 관리에서 깔때기 효과의 해결 방안

정회원 이 성 욱*, 준회원 오 지 충**, 정회원 김 제 욱**, 종신회원 오 훈***

Resolving the Funneling Effect in the Node Mobility Management of Infrastructure-based Mobile Ad Hoc Networks

Sung Uk Lee* *Regular Member*, Chi-Trung Ngo** *Associate Member*,
Je-Wook Kim** *Regular Member*, Hoon Oh*** *Lifelong Member*

요 약

본 논문에서는 인프라구조 기반의 애드혹 네트워크에서 노드의 효율적인 이동성 관리를 위한 방식을 제시한다. 이러한 종류의 네트워크는 자연적으로 IG를 기준으로 하는 트리 형태의 네트워크 모델을 갖게 된다. 따라서, IG에 가까운 노드가 많은 메시지를 처리하게 됨으로써 병목현상을 발생시키고 네트워크의 성능을 저하시키는데 이러한 문제는 깔때기 효과 (Funneling Effect)로 알려져 있다. 이 문제를 해결하기 위하여 주기적으로 트리를 구성하고 트리의 깊이에 따라서 기울진 시간 동기화를 수행하여 결합전송의 정도를 최적화하는 방식을 제안한다. 시뮬레이션을 통하여 깔때기 효과가 크게 개선되고 성능이 개선됨을 입증하였다.

Key Words : Funneling Effect, mobility management, ad hoc networks, skewed synchronization, message aggregation

ABSTRACT

In this paper, we present a new method for efficient mobility management in infrastructure-based ad hoc networks. This type of network inherently takes the tree topology model in which an Internet gateway (IG) becomes a root. Accordingly, the nodes near the IG process get much more number of messages, thereby causing a bottleneck phenomenon that degrades network performance. This problem is known as a funneling effect. We present a method to maximize message aggregation in which the nodes in the networks form tree topology and perform skewed time synchronization according to the depths of the nodes during the tree construction process. We proved by resorting to simulation that the funneling effect is alleviated and network performance is improved greatly.

I. 서 론

지난 20년 동안에 걸쳐 많은 연구가 진행된 이동 애드혹 네트워크 (MANET: Mobile Ad Hoc Networks)는 무선 이동 디바이스 (MN: Mobile

Node)들이 인프라구조 없이 멀티 홉 통신을 할 수 있는 통신 기술로서 많은 주목을 받았다. 네트워크 구축 비용의 저렴함과 네트워크 전개의 용이성 측면에서 MANET의 장점에도 불구하고, 인터넷 연결성의 부족으로 인하여 MANET의 활용성이 제한되고 있다고

* 본 연구는 울산시 및 교육과학기술부의 지역과학연구단지 개발 프로그램의 지원으로 수행되었음.

* 포항산업기술과학연구원, ** 울산대학교 전기공학부 조선훈융합연구실(hoonoh@ulsan.ac.kr), (° : 교신저자)
논문번호: KICS2011-08-366, 접수일자: 2011년 8월 23일, 최종논문접수일자: 2011년 12월 1일

할 수 있다. 이러한 인터넷 연결성을 해결하는 방안으로 MANET와 인터넷을 인터넷 게이트웨이(IG: Internet Gateway)를 사용하여 통합하고자 하는 많은 시도가 있었다. 하지만 이러한 이중 네트워크를 통합하는데 있어서 가장 큰 문제는 IG가 무선 멀티홉을 거쳐서 자신이 지원할 수 있는 MN들을 효율적으로 관리하는 것이다. 노드의 이동, 생성 및 소멸로 인하여 토폴로지가 빈번하게 변하는 MANET에 속하는 MN은 주기적으로 IG에 자신을 등록하여 자신의 위치를 갱신하여야 하기 때문에 많은 오버헤드를 발생시킨다. 이러한 문제는 노드 이동성 관리라고 알려져 있으며, 이에 대한 많은 연구가 진행되었다.

이용 가능한 IG를 찾고 MN이 IG에 등록하기 위한 경로를 얻는데 사용되는 세 가지 접근 방식 즉, 사전적(Proactive) 방식^[1], 반응적(Reactive) 방식^[2], 하이브리드 접근방식^[3]이 있다. 사전적 방식에서는 IG가 자신의 존재를 알리는 광고(Advertisement) 메시지를 전체 네트워크에 플러딩한다. 이 메시지를 수신할 때 각 MN은 역경로를 설정하고 자신의 등록 메시지가 경로를 따라서 IG에 주기적으로 보낸다. 대조적으로, 반응적 방식에서는 MN이 인터넷 액세스 혹은 외부 네트워크의 다른 노드들과 통신을 필요로 할 때에만 IG의 위치를 찾는 IG탐색(Solicitation) 메시지를 네트워크의 제한된 영역에 플러딩한다. 해당 MN이 IG로부터 혹은 IG까지 유효한 경로를 가지고 있는 어떤 MN으로부터 응답 메시지를 받으면 응답경로의 역경로를 따라서 IG에 등록하거나 응답 메시지를 보낸 MN을 거쳐서 IG에 등록한다. 사전적 방식은 전체 네트워크에 주기적으로 플러딩을 수행함으로써 높은 오버헤드를 발생하고 반응적 방식은 플러딩 범위를 제한할 수 있으나 등록 지연 문제가 발생한다. 이 두 방식의 문제점을 보완하기 위하여 사전적 방식에서 사용하는 광고 메시지의 플러딩 범위를 제한하고, 플러딩 범위 밖에 있는 노드들은 반응적 방식을 사용하고 IG까지 경로를 가지고 있는 기 등록된 노드들을 거쳐서 등록하도록 하는 하이브리드 접근방식^[3]이 제안되었다. 하지만 이들 방식들은 방식에 따라서 제한적이긴 하지만 모두 광고메시지 혹은 요청 메시지를 플러딩 함으로써 높은 제어 오버헤드를 발생 시킨다고 할 수 있다.

최근에 플러딩 방식의 단점을 극복하기 위하여 트리 기반의 방식^[4,6]이 제안되었다. 트리 기반의 방식에서는 IG로부터 트리가 형성되고 MN은 트리 경로를 따라서 등록 메시지를 MN에 보낸다. 이 방식에서는 트리 토폴로지의 효율적인 관리에 대하여 집중적으로

논의하였다. 초기에 트리 토폴로지가 형성되고 나서 노드의 이동에 따라서 동적으로 유지보수가 이루어진다. 각 MN은 가능한 IG까지 최단 거리를 유지하려고 한다. 트리 기반의 방식은 플러딩에 의존하지는 않지만 다음과 같은 결점을 가지고 있다. 먼저, 노드의 움직임에 따라서 트리 유지보수를 수행해야 한다. 즉, 부모들 잃은 노드는 다른 부모를 찾아야 하고 경우에 따라서는 게이트웨이까지 더 짧은 경로를 제공하는 노드로 부모노드를 변경하여야 한다. 둘째, 어떤 노드가 게이트웨이까지 거리를 변경하는 경우에 모든 후손 노드에게 변경된 정보를 알려 주어야 한다. 마지막으로 어떤 노드가 토폴로지 유지보수 동안에 자신의 후손노드에게 가입하는 경우에 루프가 형성될 수 있다. 루프문제를 해결하기 위한 효율적인 알고리즘이 제시되었다^[5]. 부가적으로 트리 토폴로지의 보수를 불규칙적으로 수행해야 하기 때문에 등록 지연이 증가하고 전송지터가 증가하는 경향이 있다.

하지만, 이러한 방식들은 모든 노드들이 특정 노드(IG)로 빈번하게 메시지를 보내는 이러한 유형의 네트워크에서 상존하는 문제 즉, 낮은 깊이에 위치한 노드가 높은 깊이에 있는 노드보다 더 많은 등록 메시지를 처리해야 하는 문제를 해결하지는 못한다. 결과적으로 IG 주위에 있는 노드는 높은 트래픽 혼잡현상으로 인하여 메시지 충돌 및 손실이 증가하게 된다. 이러한 현상은 갈매기 효과(Funneling Effect)^[7]로 알려져 있다. 이 문제를 해결하기 위해서는 IG에 가까운 노드들에게 더 많은 대역폭을 할당 해야 하지만, 이 방법은 현실적으로 어렵다고 할 수 있다.

본 논문에서는 이러한 문제를 해결하기 위하여, 뒤 틀린 시간 동기화 기법을 사용하는 최적화된 메시지 결합(Message Aggregation) 방식을 제안한다. IG가 자신의 존재를 알리기 위하여 주기적으로 보내는 광고 메시지를 이용하여 트리를 구성하고 각 노드가 후손 노드로부터 전달되는 모든 등록 메시지를 결합하여 보낼 수 있도록 시간을 기울지게 동기화(Skewed Synchronization)한다. 트리 구성 과정에서 각 노드는 자신의 깊이에 반비례하는 식으로 등록 메시지 전송 시간을 설정한다. 이렇게 함으로써 부모노드는 자식 노드보다 약간 늦게 등록 메시지를 전송하게 된다. 이러한 기울진 동기화를 통한 등록 메시지 결합을 최대화 함으로써 메시지 전송 횟수를 줄일 수 있을 뿐만 아니라 네트워크에서 전송되는 총 바이트의 수를 크게 줄일 수 있다. 메시지 전송 횟수를 줄이는 경우에 MAC 계층의 제어 메시지(RTS, CTS, ACK)의 수도 비례해서 크게 줄어든다. 깊이가 낮은 노드 즉, IG에

가까운 노드에서 훨씬 더 오버헤드 감소가 커진다.

본 논문의 구성은 다음과 같다. 2장에서 네트워크 모델을 정의하고 제안하는 방식을 간략히 설명한다. 3장에서는 뒤틀림 동기화를 사용하는 이동성 관리 기법을 자세하게 설명한다. 4장에서는 기존의 방식과의 성능을 비교 평가하고, 5장에서 결론을 맺는다.

II. 연구배경

2.1 네트워크 모델

본 논문에서는 고려하는 네트워크는 복수의 IG (Internet Gateway)와 많은 MN(Mobile Node)들로 구성된다. 모든 MN은 동일한 기능을 가지며, 어떤 MN에서 수신한 메시지를 목적지를 향해서 다른 MN으로 전달할 수 있는 라우터 기능을 갖는다. IG는 두 개의 네트워크 인터페이스 카드를 장착하고 있으며, 하나를 사용하여 MN과 통신할 수 있고 다른 하나를 사용하여 인터넷 상의 유선 호스트와 통신할 수 있다. IG의 무선 전송 거리는 MN의 그것과 같고 통신 거리가 길어지면 더 많은 수의 MN들의 통신과 통신 간섭이 발생하기 때문에 통신 거리가 적절하게 제한된다. 따라서, 무선 멀티홉 통신을 요구한다.

MN은 인터넷 상의 유선 호스트와의 통신을 요청할 수 있을 뿐 아니라 유선 호스트나 혹은 다른 MANET에 있는 MN으로부터 통신 요청을 받을 수 있다. MN이 이러한 인터넷 연결 서비스를 받기 위해서는 자신이 특정 IG에 서비스 요청을 위한 등록을 수행하여야 하고 주기적으로 갱신을 하여야 한다. 그리고 해당 IG는 등록된 MN이 자신의 서비스를 받고 있다는 사실을 해당 MN의 홈 네트워크에 알려주어야 한다. 이러한 등록 과정이 정상적으로 이루어진다고 가정하면, 어떤 MN의 홈 네트워크로 전송된 데이터 패킷은 해당 MN이 임시로 등록된 IG로 전달되고, IG는 무선 멀티홉 전송을 통하여 MN으로 데이터를 전달하게 된다. 이러한 MN의 등록 갱신이 제대로 이루어지지 않는 경우에는 무선 멀티홉을 통한 잘못된 데이터 전송으로 인하여 데이터를 잃게 되고 귀중한 통신 자원을 낭비하게 된다.

위에서 기술한 유형의 네트워크 모델을 인프라구조 기반의 이동 애드혹 네트워크 (IFMANET)라고 정의한다. IFMANET는 무선 1-홉만을 허용하는 셀룰라 IP 네트워크의 확장이라고 할 수 있다. 하지만, IFMANET에서는 모든 MN이 주기적으로 무선 복수 홉을 거쳐서 IG에 등록을 수행하여야 한다. MN은 모든 등록 행위 마다 IG까지 경로를 설정해야 할 수도

있다. 더구나 IG 주위에 위치한 MN들은 더 많은 MN들이 전송하는 메시지를 상위로 전달해야 하기 때문에 트래픽 혼잡현상으로 인하여 메시지 충돌 및 손실이 증가하게 된다. 이러한 현상은 갈매기 효과 [7]로 알려져 있다. 네트워크 성능을 개선하기 위하여 효율적인 등록 방식을 제공해야 하고 이와 더불어 IG 주위의 MN에 부과되는 높은 처리 부하를 완화하여 처리 부하의 불균형 문제가 해결하여야 한다.

이러한 네트워크의 특징은 태생적으로 트리 토폴로지의 구성형태를 갖는다. 각 MN은 IG까지 최단 경로를 제공하는 다음 노드를 자신의 부모로 지정한다. 복수의 최단 경로가 있고 다음 노드가 서로 다른 경우에는 다음 노드들 중에서 임의로 부모 노드를 지정할 수 있다. 설명의 편의를 위하여, 노드와 노드 사이의 직접 연결을 링크라고 정의하고, 특히 부모와 자식간의 링크를 트리링크라고 정의한다. 또한, 트리에 가입한 노드를 트리노드 가입하지 않은 노드를 고아노드로 명명한다.

2.2 문제해결 접근방식

본 논문에서는 IFMANET에서 발생하는 갈매기 효과를 해결하기 위하여 메시지 결합전송 (Message Aggregation and Transmission) 즉, 복수의 메시지를 통합하여 하나의 메시지로 전송하는 메시지 전송의 최적화 방법을 논의한다. 결합전송은 각 MN이 메시지의 전송 횟수를 줄여서 메시지 헤드와 트레일러의 불필요한 중복 전송을 줄일 수 있을 뿐만 아니라, 그 결과 IEEE 802.11 MAC 프로토콜 [8]에서 RTS, CTS, ACK와 같은 제어신호 메시지의 감소를 가져다 준다. 따라서, MN 들 사이의 전송 충돌을 감소시킬 수 있다. 특히, 메시지 결합을 수행함으로써 낮은 깊이에 위치한 노드들 즉, 후손 노드들을 많이 가진 노드들이 높은 트리깊이에 위치한 노드들 보다 훨씬 더 크게 처리 부하를 줄일 수 있다.

트리의 중간에 위치한 노드가 자신의 후손 노드들로부터 모든 등록 메시지를 수신할 때까지 기다려서 수신한 메시지를 결합하여 전송한다면 메시지 결합의 정도를 극대화 될 수 있다. 이를 위하여 그림 1과 같이 각 노드의 메시지 전송시간을 기울지게 동기화하면 된다. 즉, 노드의 트리 깊이가 $j > i$ 인 경우에, 깊이 i 에 있는 노드는 깊이 j 에 있는 노드보다 항상 늦게 전송을 하도록 시간을 동기화 한다.

이렇게 전송시간이 동기화되면 MN은 자신의 후손들로부터 전달된 모든 등록 메시지를 자신의 등록 메시지와 함께 결합하여 한 번에 자신의 부모 노드에게

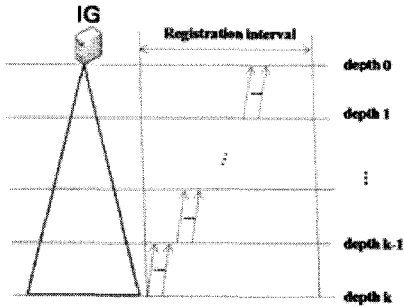


그림 1. 뒤틀림 시간 동기화를 사용하는 등록 메시지의 전송 전승할 수 있다. 이상적으로 트리 전체에서 기울지게 동기화가 된다면 모든 노드는 등록 주기 (Registration Interval)마다 등록 메시지를 한 번만 전송하게 된다. 이러한 트리 깊이에 따른 처리 부하의 불균형을 해소함으로써 깔때기 효과를 완화하는데 크게 기여할 수 있다.

2.3 제어 메시지 정의

제안하는 방식을 설명하는데 사용되는 몇 가지 제어 메시지를 정의한다.

- IG_Advertisement = (MessageId): IG 는 주기적으로 자신의 존재를 알리기 위하여 이 메시지를 브로드캐스트 한다.
- J-REQ = (MessageId, HopToIG): MN은 부모노드로 IG 혹은 다른 MN에 결합하기 위하여 이 메시지를 보낸다. MessageId 는 이 메시지를 다른 메시지와 구별하기 위한 유일한 ID 번호이다. HopToIG 는 IG까지의 홉 수를 나타낸다.
- REG = (N₁, N₂, ..., N_k): MN이 등록 (Registration) 메시지를 IG에 등록하기 위하여 이 메시지를 사용한다. N_i, i=1..k, 는 등록될 노드의 ID 번호이다.
- x.REG: Node x의 등록 메시지를 나타낸다.

III. 시간 동기화 및 결합 전송

3.1 트리 구성 및 유지 보수

IG는 IG-Advertisement 메시지를 주기적으로 전송하고 이를 이용하여 트리 구성을 시작한다. 이 메시지를 수신하는 MN은 IG의 자식 노드가 되기 위하여 J-REQ 메시지를 보낸다. IG는 J-REQ를 받으면 해당 MN을 자식 노드로 받아들인다. MN은 송신한 J-REQ에 대한 ACK (IEEE 802.11 MAC 계층 제어 신호)를 IG로부터 수신하면 IG의 자식 노드가 된다.

어떤 고아노드가 J-REQ를 보내고 성공적으로 트리

노드가 되면, 이 J-REQ를 엿듣는 다른 고아노드는 같은 방식으로 J-REQ를 보내서 부모를 갖고 트리노드가 된다. 이러한 과정은 J-REQ를 엿듣는 고아노드가 더 이상 없을 때까지 계속된다. 모든 고아노드는 순차적으로 하나의 제어 메시지를 사용하여 트리노드가 되고, 이러한 반복을 통하여 최종 트리가 구성된다.

그림 2는 트리의 구성 예를 보여준다. 복수의 고아노드가 IG-Advertisement 혹은 J-REQ 를 수신 혹은 엿듣는 경우에 ID가 낮은 고아노드가 먼저 J-REQ를 보낸다고 가정하자. 먼저 IG가 IG_Advertisement 메시지를 브로드캐스트 하면, 노드 1, 2, 3이 수신하게 된다. 노드 1이 IG에 J-REQ를 보내면 노드 2, 4, 5가 이 J-REQ 메시지를 엿들을 수 있다. 노드 2는 HopToIG = 1을 가지고 IG에 가입할 수 있기 때문에 노드 1을 부모로 선택하지 않는다. 노드 4, 5는 J-REQ를 보냄으로써 노드 1의 자식 노드가 된다. 노드 2가 J-REQ 를 IG에 보내면 노드 1, 3, 4, 5, 6이 엿들게 되고 노드 1, 3은 위와 같은 논리로 배제된다. 노드 4, 5는 노드 1에 이미 가입을 하였으며 노드 6이 노드 2에 가입을 하게 된다. 이런 식으로 모든 고아노드는 자신의 부모노드를 갖게 되고 트리가 형성된다.

트리가 형성되고 나서 노드 등록 프로세스가 시작된다. 노드 4는 노드 7, 8의 등록 메시지 즉, 7.REG=(7)과 8.REG=(8)를 먼저 수신하고 자신의 등록 메시지를 결합하여 4.REG = (4, 7, 8)을 노드 1에 전송한다. 노드 1은 4.REG = (4, 7, 8)와 5.REG = (5, 9)를 각각 노드 4와 노드 5로부터 수신하고 이들과 자신의 REG를 결합하여 1.REG = (4, 5, 7, 8, 9)를 IG에 전송한다.

트리 형성과 등록 프로세스는 등록 주기마다 반복해서 수행된다. 트리 형성 과정은 사전적 방식에서 IG가 Advertisement 메시지를 전체 네트워크 플러딩하여 각 MN이 IG에 대한 최단 경로를 형성하도록 하는 과정과 유사하다. 이 방식에서는 모든 노드가 하나의 메시지를 브로드캐스트 한다. 이렇게 형성된 경로를 따

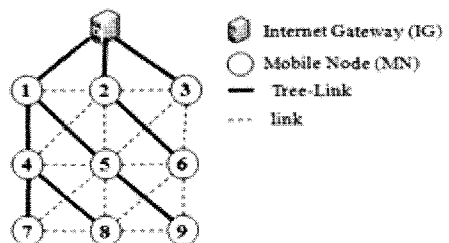


그림 2. 메시지 결합전송의 예

라서 각 노드는 주기적으로 등록을 수행한다. 본 논문
에 제안하는 트리 형성 방식에서는 IG만 하나의 메시
지를 브로드캐스트하고 모든 MN 노드는 부모를 설정
하기 위하여 하나의 메시지를 유니캐스트 한다.

경로 관리 측면에서 보면, 트리 방식에서는 각 MN
은 트리 경로를 사용하여 IG에 등록을 수행한다. 경로는
트리링크의 연속으로 이루어지며, 어떤 MN이 자
신의 부모에 대한 트리링크를 잃은 경우에 해당 트리
링크만 복구하면 자신의 후손 노드는 모두 경로를 유
지하게 되는 집합적 경로 관리를 수행한다는 점에 있
어서 기존의 사전적 방식과는 크게 대비된다. 표 1에
이 두 가지 방식의 차이를 비교하였다.

트리의 유지보수는 다음과 같이 이루어진다. 트리
형성과정에서 각 노드는 자신의 이웃 노드들과 각 이
웃 노드의 HopToIG에 대한 정보를 가진다. 어떤 MN
이 자신의 부모를 잃게 되면 자신의 이전 HopToIG보
다 짧거나 혹은 동일한 경로 길이를 제공하는 다른
MN을 선택하여 부모를 변경한다. 더 긴 경로를 제공
하는 노드에 가입을 허용하면 자신의 후손 노드를 부
모노드로 선택할 가능성이 있으며, 이 경우에 루프를
형성할 수 있다. IG가 IG-Advertisement를 전송할 때
마다 트리를 재생성 하기 때문에 적절한 부모노드를
찾지 못하면 등록 메시지를 버리고 다음 등록주기가
될 때까지 기다린다.

표 1. 사전적 방식과 트리 구성방식의 비교

항목 \ 방식	사전적 방식	트리구성 방식
IG	브로드캐스트	브로드캐스트
MN	브로드캐스트	유니캐스트
전송메시지 수 (노드 수 = n)	n	n
경로형성	플러딩에 의한 개별경로	유니캐스트에 의 한 트리경로
경로유지보수	개별 경로 관리	집합적 경로 관리

3.2 기울진 시간 동기화

본 절에서는 MN이 트리 깊이 i에 위치한 노드가
트리 깊이 i+1에 위치한 노드보다 항상 등록 메시지를
늦게 보낼 수 있도록 등록 메시지의 전송시작 시간을
생성하는 방법을 기술한다. 모든 MN이 결합 프로세
스를 사용하여 부모를 설정할 때, MN은 자신의 트리
깊이 (즉, IG까지의 홉 거리) 에 상대적으로 등록 메
시지의 전송시작 시간을 설정한다. 각 MN은 자신의
트리 깊이가 더 높을수록 등록 메시지의 전송을 시작

하기 전에 덜 기다리도록 자신의 등록 메시지 전송 시
작 시간을 설정한다. 따라서, 깊이 d에 있는 MN의 대
기 시간, W(d)를 결정하기 위하여 다음과 같은 공식을
사용한다.

$$W(d) = W_1 * (a)^{d-1}$$

여기서, $d \geq 1$, W_1 트리 깊이 1에 위치한 노드의
등록 메시지 전송 전 대기 시간을 나타내고, a는 지수
함수의 기저로서 $0 < a < 1$ 을 만족한다. a는 1보다 적
은 분수 값이기 때문에 W(d)는 최대 값 W_1 을 갖는 감
소함수이다.

설명의 편의를 위하여, 깊이 i에 있는 노드와 위치
i+1에 위치한 노드의 대기 시간 차이를 wdiff(i) (= $|W(i) - W(i+1)|$)라고 정의하도록 한다 ($1 \leq i < \text{트리의 최대 깊이}$).

$$\text{wdiff}(i) = W_1 \cdot (1-a)a^{i-1}$$

여기서 wdiff(i) 는 W_1 에 비례한다. 따라서, W_1 의
크기는 모든 트리 깊이에서 충돌의 가능성에 영향을
미친다. 한편, wdiff(i) 는 a의 지수 함수이다. a가 더
작아짐에 따라서, wdiff(i) 는 깊이 (즉, i)가 증가할 때
더 급격하게 지수적으로 감소한다. a가 0에 가까우면
깊이가 증가함에 따라서 더 빨리 0에 접근한다. 두 파
라미터에 대하여 적절한 값들을 선택해야 한다.

그림 3은 깊이 d와 a의 변화에 따라서 서로 다른
깊이 d에 있는 노드들의 대기 시간 W(d) 값을 보여준
다. 편의상 W_1 은 임의 값인 시간 단위 3으로 설정하
였다. 만일 a = 1이면, 모든 노드는 깊이와 관계없이
3 시간단위를 기다린 후에 동시에 등록 메시지의 전
송을 시작하게 된다. a값이 매우 작으면 (즉, a = 0.1),
깊이 3, 4, 5에 위치한 노드는 거의 동시에 등록을 시

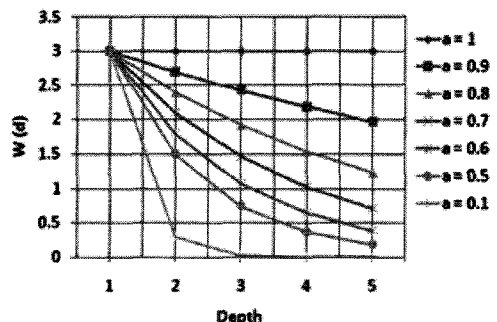


그림 3. 깊이 d와 파라미터 a에 따른 W(d)의 분포도

작하게 된다. 따라서, 이들 값들은 높은 깊이에 있는 노드들의 충돌률을 크게 높일 수 있기 때문에 적절하지 않다. $a = 0.9$ 이면, 트리 깊이에 따른 시간 분배 즉 $wdiff(i)$ 가 균일하게 분포되지만 트리에 속한 모든 노드가 1의 짧은 간격 즉, 시간 단위 2와 3 사이에서 자신의 등록 메시지를 전송하게 된다. 따라서, 이 값은 전체 등록 가능 시간 $W_1=3$ 에 대하여 너무 짧은 간격 내에서 모든 노드가 등록을 시작하게 되어 충돌률을 높일 수 있다. $a = 0.7$ (0.8)이면, 서로 다른 트리 깊이에 위치하는 모든 노드들은 2.3 (1.8) 시간단위 내에서 등록을 시작하게 된다. 이 경우에 $wdiff(i)$ 는 깊이가 낮아질 때 약간 증가 하지만 깊이 i 에 그렇게 민감하지 않으며, 이 현상은 바람직하다고 할 수 있다. 이유는 더 많은 IG에 가까운 MN들이 더 많은 등록 메시지를 처리하기 때문이다. 이러한 분석은 시뮬레이션을 사용하여 W_1 과 a 에 대한 적절한 값을 결정할 때 크게 도움이 된다.

3.3 시간 동기를 이용한 결합 전송

모든 MN이 트리 구성 동안에 설정된 등록 메시지의 전송시작 시간에 따라서 등록 메시지를 보낸다. 깊이 i 에 위치한 MN은 모든 후손 노드들로부터 등록 메시지를 수신 완료하거나 혹은 자신의 정의된 전송시간 $W(i)$ 경과할 때까지 기다린다. MN은 수신한 등록 메시지들과 자신의 등록 메시지를 결합하고 자신의 부모 노드에게 보낸다. 때로는, 시간 동기화 에러로 인하여 자신의 특정 후손들로부터 등록 메시지를 늦게 수신할 수도 있다. 이 경우에는 해당 등록 메시지를 수신하는 즉시 부모 노드에게 별도로 전송한다.

MN이 자신의 부모 노드를 잃게 되면, 트리 유지보수 방식에 따라서 다른 부모를 얻은 후에 전송을 완료한다. 만일 적절한 부모 노드를 발견하지 못하면 해당 등록 메시지를 폐기한다. 트리가 주기적으로

재구성되기 때문에 한 번의 등록 실패는 큰 문제를 발생시키지는 않는다. 등록 메시지를 전송하는 도중에 부모 노드가 파손된 것을 알게 되면 트리 유지 보수 방법에 따라서 자신의 부모노드를 즉시 변경하여 전송하면 된다.

IV. 성능평가

4.1 시뮬레이션 환경

제한하는 방식이 갈때기 효과를 다루는 성능을 평가하기 위하여 그림 4에 제시한 세 가지 시뮬레이션 시나리오 S1, S2, S3를 사용하였다. S1은 IG가 영역

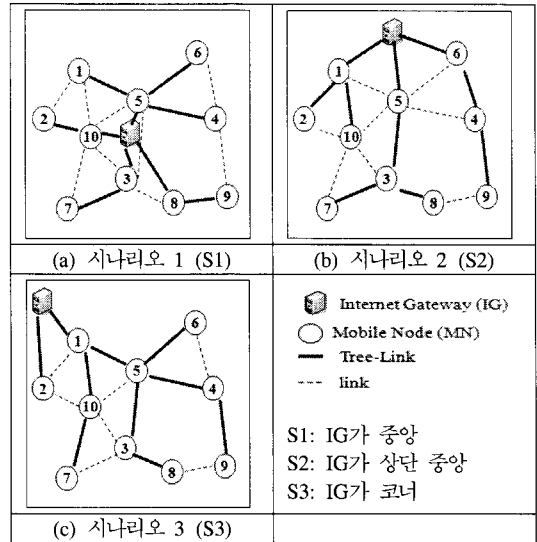


그림 4. 시뮬레이션 시나리오 패턴

의 중앙, S2는 상단 중앙, S3는 코너에 위치하는 차이를 가지고 있다. S1은 IG가 가장 안정된 중앙에 위치하고 있으며 형성된 트리의 깊이가 가장 낮을 것이다. 반면에 S3의 경우는 IG에 코너에 위치하여 IG에 가까운 노드의 수가 적어서 병목현상이 가장 많이 발생하는 트리를 구성하게 된다. 따라서, 이 세가지 시나리오의 다양한 특성을 반영한다는 점에 있어서 적합한 선택이라고 할 수 있다.

본 시뮬레이션에서는 세가지 시나리오의 각각에 대하여 노드의 수를 변경하고 노드 이동성을 변화시키면서 시뮬레이션을 수행하여 성능을 평가한다. 시뮬레이션은 QualNet 5.0.2을 사용하였으며, 주요 파라미터 및 해당 값은 표 2에 제시하였다. 그림 5에서 분석한 결과에 따라서, $a = 0.8$ 을 선정하였다. W_1 값은 전체 노드의 등록 메시지 전송 시작 시간의 범위를 결정하기 때문에 노드 수에 따라 다소 차이가 있다. 여러 번

표 2. 시뮬레이션 파라미터 및 값

파라미터	값
이동성 패턴	Random Waypoint
Pause Time	30 seconds
노드의 수	이동노드 수는 변함 (하나의 IG)
영역의 크기	1000 x 1000 m2
전송거리	250 m
무선 대역폭	2 Mbps
시뮬레이션 시간	600 seconds
IG_Advertisement 전송 간격	3 seconds

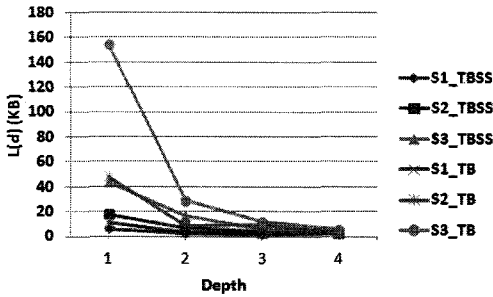


그림 5. 서로 다른 깊이 에 위치하는 노드가 처리하는 바이트의 수 (nNodes = 75, mSpeed = 20 m/s)

의 시뮬레이션을 통하여 50개에 100개의 노드를 사용하는 경우에 각 노드의 등록 메시지 전송 지연 시간이 가장 낮을 때의 값을 선정하였으며, $W_1=280ms$.

첫 번째 파트에서는 깔때기 효과를 다루는 능력을 평가한다. 서로 다른 트리 깊이에 위치하는 노드들의 등록메시지 처리 (송신 및 수신) 부하를 평가하기 위하여 트리 깊이 d의 함수로서 $L(d)$ 를 사용하였다.

$$L(d) = 1/n(d) \sum_{i=1..n}(R(i, d) + T(i, d))$$

여기서, $n(d)$ 는 깊이 d에 있는 평균 노드의 수로서 다음과 같이 주어진다.

$$n(d) = \text{SUM}_{i=1..n} t(i, d) / \text{SimTime},$$

여기서, SimTime은 전체 시뮬레이션 시간, n은 전체 노드의 수, $t(i, d)$ 는 노드 i가 깊이 d에 머물렀던 시간을 나타낸다. 또한, $R(i, d)$ 는 노드 i가 깊이 d에 있는 동안 수신한 바이트 수이며, $T(i, d)$ 는 노드 i가 깊이 d에 있는 동안 전송한 바이트 수를 나타낸다.

MN은 시뮬레이션 시간이 경과함에 따라서 이동할 수 있고 자신의 트리 깊이가 변경될 수 있다. 따라서, $L(d)$ 는 트리 깊이 d에 머물렀던 노드들이 깊이 d에 머무르는 동안에 평균적으로 처리 (수신 및 송신)했던 등록 메시지들의 바이트 수를 나타낸다. TB-SS 방식이 효과적으로 메시지 결합 전송을 수행한다면 트리 깊이가 낮을수록 기존의 TB 방식에 비해서 처리 부하의 개선 정도가 커질 것으로 기대할 수 있다.

두 번째 파트에서는 일반적으로 사용되는 성능 지표를 사용하여 개선 정도를 평가한다. 여기서 사용할 두 가지 성능 지표는 제어 메시지의 부하 (Control Overhead)와 등록 메시지의 등록 성공률 (Registration Ratio)이다. TB-SS에서 사용한 제어 메

시지는 IG-Advertisement, J-REQ, REG 를 사용하고, TB는 Hello, J-REQ, J-RES, REG를 사용하였다. 노드의 수(nNodes)는 50, 75, 100로 변경하고, 최대 노드 이동 속도 (mSpeed)는 10, 20, 30, 40 (m/s)로 변경시키면서 시뮬레이션을 수행하였다.

4.2 TB-SS와 TB의 깔때기 효과 비교

그림 5는 TBSS와 TB를 사용할 때 서로 다른 트리 깊이에 있는 MN들의 처리 부하를 바이트 수로 나타내었다. TB의 경우에 낮은 깊이에 위치하고 있는 노드들의 처리 부하가 높은 깊이에 위치하고 있는 노드들에 비해서 매우 크다는 것을 알 수 있다. 더구나, 트리 깊이가 가장 클 것으로 판단되는 시나리오 S3의 경우에는 트리 깊이 1에 위치하는 적은 수의 노드가 더 많은 후손 노드들의 등록 메시지를 처리해야 하기 때문에 IG에 가까운 노드일수록 처리 부하가 급격히 증가함을 알 수 있다.

트리 깊이가 낮아질수록 TB-SS와 TB 그래프 사이에 차이가 커진다는 것을 알 수 있다. 이러한 큰 차이는 시간 동기화를 사용하는 결합전송이 트리 깊이가 낮은 노드들의 부하를 줄이는데 크게 기여한다는 것을 알 수 있다. 결과적으로, TB-SS가 TB에 비해서 깔때기 효과를 크게 완화한다는 것을 알 수 있다.

4.3 TB-SS와 TB의 성능비교

본 절에서는 깔때기 효과를 크게 완화하는 것으로 입증된 TB-SS가 TB에 비해서 얼마나 성능을 개선하는지를 보이기 위하여 두 개의 성능지표를 사용한다.

$$\text{Control Overhead (A)} = \frac{1}{nNodes} \sum_{m \in M(A)} \text{sizeof}(m)$$

여기서,

$M(TB)$: {Hello, J-REQ, J-RES, REG }

$M(TB-SS)$: {IG_Advertisement, J-REQ, REG }

$\text{sizeof}(m)$: 메시지 m의 바이트 수

$$\text{Registration Ratio} = \frac{1}{nNodes} \sum_{i=1..nNodes} \frac{\text{Received}(IG, i)}{\text{Transmitted}(i)}$$

여기서,

$\text{Transmitted}(i)$: 노드 i가 전송한 REG 메시지의

$\text{Received}(IG, i)$: 노드 i가 전송한 메시지를 IG가 성공적으로 수신한 REG의 수

위의 두 성능지표는 깔때기 효과와 관련성이 높다고 할 수 있다. 등록 메시지의 경우에, 모든 노드가 주기적으로 등록 메시지를 전송한다. 따라서, 등록 메시

지의 지연시간은 등록 주기에 비해서 매우 미미하기 때문에 성능 지표로서는 적합하지 않다. 이리 이유로 등록 메시지의 지연 및 지터에 성능 평가는 배제한다.

다음은 시나리오, 노드 이동성, 그리고 노드 밀도를 변경 시키면서 시뮬레이션을 수행되었다. 이 세 가지 파라미터는 제어 오버헤드에 직접적으로 영향을 주고 그 결과 등록 메시지의 전송 효율에 영향을 준다.

4.3.1 시나리오에 따른 성능 평가

그림 6, 7은 시나리오의 변화에 따른 성능 평가 결과를 보여 준다. 1000 x 1000 (m²)영역에서 75개의 노드, 최대 노드 이동 속도는 20m/s로 설정 하였다.

제어 오버헤드는 트리의 크기 및 깊이에 비례한다. 본 시뮬레이션에서는 하나의 IG를 사용하기 때문에 트리 규모는 동일하다고 할 수 있다. S1은 IG가 디멘전의 중앙에 위치하기 때문에 트리의 깊이가 비교적 낮으며, S3는 IG가 코너에 위치하기 때문에 트리의 깊이가 평균적으로 크다고 유추할 수 있다. 따라서, 그림 6의 제어 오버헤드는 S1, S2, S3의 순서에 따라

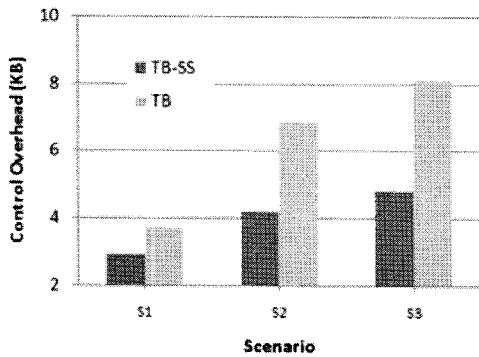


그림 6. 시나리오의 변경에 따른 제어 오버헤드 (nNodes = 75, mSpeed = 20m/s)

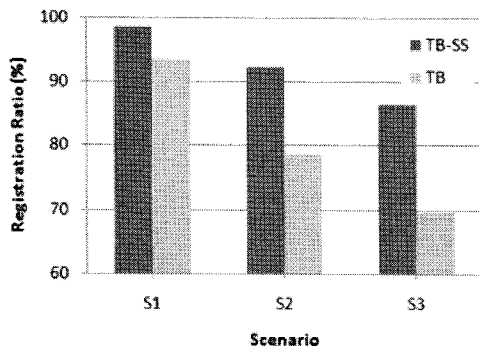


그림 7. 시나리오의 변경에 따른 등록률 (nNodes = 75, mSpeed = 20m/s)

서 증가한다는 것을 보여주고 있다.

TB-SS와 TB를 비교하면, 결합전송을 사용하는 TB-SS가 오버헤드를 40%이상 개선한다는 것을 알 수 있다. 그림 7을 보면, 시나리오 S1의 경우보다 S3의 경우에 즉, 트리 깊이가 높을수록 훨씬 더 전송률을 개선하고 있다는 것을 보여준다. S2의 경우에는 20% (78에서 93), S3의 경우에는 26% (69에서 87)의 전송률 개선을 보여준다.

4.3.2 노드 이동성에 따른 성능 평가

그림 8과 9는 시나리오 S3에서 50개의 노드를 사용하여 mSpeed를 10에서 40m/s까지 변경하면서 TB-SS와 TB의 제어오버헤드와 등록률을 측정하였다.

그림 8을 보면, TB-SS의 경우에 제어오버헤드가 이동성에 거의 관계가 없다는 것을 보여준다. 노드 이동 속도가 증가하더라도 주기적인 트리 구성에서 사용하는 IG_Advertisement 및 J-REQ와 관련된 오버헤드는 거의 일정하지만 REG 메시지의 손실이 조금씩 증가하기 때문에 mSpeed = 40 m/s에서 전체적인 오버헤드가 약간 감소함을 보여주고 있다. TB의 경우에는 속도가 증가하면 REG 메시지의 전송 손실이 커

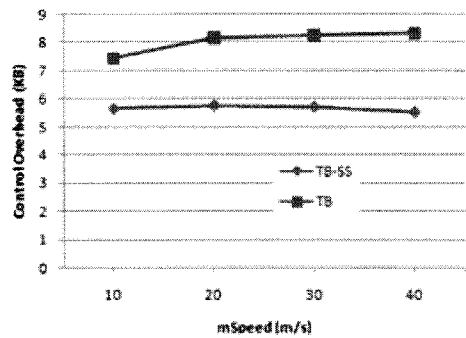


그림 8. 노드 이동성에 따른 오버헤드 (S3, nNodes = 50)

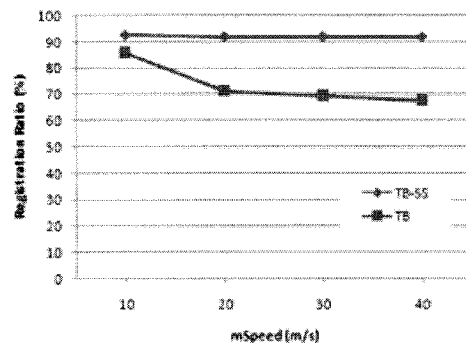


그림 9. 노드 이동성에 따른 등록률 (S3, nNodes = 50)

지지만 Hello메시지 재전송이 많아진다. 또한, 노드 이동성이 증가하여 트리 링크가 파손 빈도가 증가하고, J-REQ와 J-RES의 전송 빈도가 증가하기 때문에 제어오버헤드가 증가함을 알 수 있다.

그림 9는 노드의 이동성에 따라서 등록률을 보여준다. TB-SS의 경우에는 주기적으로 트리를 구성하고 트리 구성 직후에 경로가 잘 유지되는 상태에서 등록을 수행하기 때문에 노드의 이동성에 관계없이 매우 안정된 전송률을 보여 주고 있다. 하지만, TB의 경우에 속도가 증가하면 그림 8에서와 같이 제어오버헤드가 증가하고, 트리 파손의 경우에 링크 파손이 감지될 때까지 갱신이 이루어질 수 없기 때문에 전송률이 떨어지는 것을 알 수 있다.

4.3.3 노드 밀도에 따른 성능 평가

그림 10과 11은 시나리오 S3에서 40m/s의 노드 이동성을 사용하여 노드 수를 50에서 100개로 증가시키면서 TB-SS와 TB의 성능을 평가하였다. 그림 10을 보면, TB-SS의 제어오버헤드는 완만하게 증가하는 자

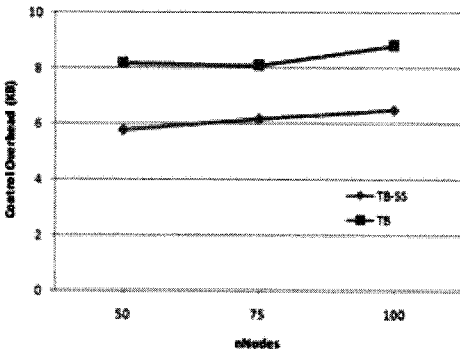


그림 10. 노드 밀도에 따른 제어 오버헤드 (S3, mSpeed = 40m/s)

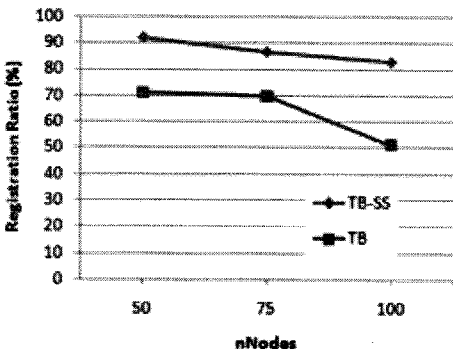


그림 11. 노드 밀도에 따른 등록률 (S3, mSpeed = 40m/s)

연스러운 패턴을 보이고 있으며, TB의 경우에도 기본적으로 증가하는 패턴을 보이나, 노드 수가 70개 인 경우에 비해서 노드 수가 50개 일 때 약간 높은 오버헤드를 보이고 있다. 그 이유는 50개의 노드를 사용하는 경우에 네트워크 파티션이 좀 더 자주 발생하기 때문이다. 반면에 TB-SS의 경우에는 IG가 IG-Advertisement를 전송할 때 트리 구성이 시작되고 네트워크 파티션이 발생하면 트리 구성이 멈춰서 제어 오버헤드가 증가하지 않기 때문이다.

그림 11은 노드 수의 증가에 따른 등록률을 보여준다. 노드 수가 많아지면 노드 사이의 전송 간섭이 증가하고 깔때기 효과로 인하여 전송 성공률이 떨어지는 것은 자연스러운 현상이다. TB의 경우에 노드의 수가 50개인 경우에 위에 설명한 바와 같이 네트워크 파티션 빈도가 증가하고 오버헤드가 상대적으로 증가하기 때문이다. 노드의 수가 100인 경우에, 깔때기 효과를 완화하는 TB-SS에 비해서 TB의 경우에 깔때기 효과로 인하여 등록 성공률이 급격히 떨어지는 현상을 볼 수 있다.

V. 결 론

본 논문에서는 주기적으로 트리를 구성하고 트리의 깊이에 따라서 기울진 시간 동기화를 수행하여 결합 전송의 정도를 최적화 하는 방식을 제시하였다. 일반적인 트리 유형의 네트워크에서 트리 깊이가 낮은 곳에 위치하는 노드가 메시지를 파다하게 처리 (수신 혹은 전송)함으로써 깔때기 효과가 발생한다. 깔때기 효과는 메시지 전송의 충돌을 증가시키고 네트워크의 처리 성능을 저하시킨다. 이를 해결하기 위하여 기울진 시간 동기화를 이용하여 결합전송을 최적화 함으로써 트리 깊이가 낮은 노드의 메시지 전송 횟수를 크게 줄이는 방법을 제시하였으며, 시뮬레이션을 통하여 깔때기 효과가 크게 개선되고 성능이 개선됨을 입증하였다.

참 고 문 헌

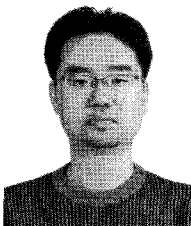
- [1] H. El-Moshri, M. A. Mangoud, and M. Rizk, "Gateway discovery in ad hoc on demand distance vector (AODV) routing for Internet connectivity," in *Radio Science Conference, NRSC 2007*, pp. 1-8, Mar. 2007.
- [2] J. Broch, D. A. Maltz and D. B. Johnson, "Supporting Hierarchy and Heterogeneous

Interfaces in Multi-Hop Wireless Ad Hoc Networks,” in *Proc. Int’l. Symp. Parallel Architecture, Algorithms, and Networks*, Perth, Australia, pp. 370-375, Jun 1999.

- [3] P. Ratanchandani, R. Kravets, "A hybrid approach to Internet connectivity for mobile ad hoc networks," *Proc. IEEE WCNC 2003*, pp. 1522 - 1527, Mar. 2003.
- [4] T. D. Han, H. Oh, "A topology management routing protocol for mobile IP support of mobile ad hoc networks," *In the 8th international conference on ad hoc networks and wireless, AdHocNow2009*, Murcia, Spain, LNCS 5793, pp. 341 - 346, Sep. 2009.
- [5] T. D. Han, H. Oh, "Detecting and Resolving a Loop in the Tree-based Mobility Management Protocol," *Proc. 5th International Conference on Advances in Grid and Pervasive Computing, GPC 2010*, pp. 583-592, May 2010
- [6] T. D. Han, H. Oh, "Quasi-tree mobility management for internet connectivity of mobile ad hoc networks," *Wireless Networks*, vol. 17, no. 2, pp. 493-506, Feb. 2011
- [7] G. S. Ahn, E. Emiliano. M, A. T. Campbell S. G. Hong, F. Cuomo, "Funneling-MAC: a localized, sink-oriented MAC for boosting fidelity in sensor networks," *Proc. 4th international conference on Embedded networked sensor systems*, pp. 293-306, Nov. 2006
- [8] IEEE standard 802.11. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 2007

이 성 욱 (Sung Uk Lee)

정회원

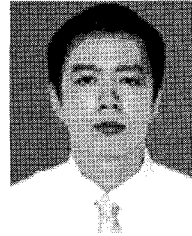


1998년 경북대학교 전자공학
학사
1998년~2001년 대우조선해양
2004년 University of Florida
전기컴퓨터 석사
2010년 University of Florida
전기컴퓨터 박사

2010~현재 포항산업과학연구원 선임연구원
<관심분야> 이동통신, 무선 애드혹 및 센서 네트워크

오 지 중 (Chi-Trung Ngo)

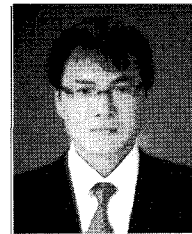
준회원



2010년 베트남 호치민 공과대
학 공학사
2010년~현재 울산대학교 전기
공학부 석사과정
<관심분야> 무선통신, 상황인
지컴퓨팅

김 제 욱 (Je-Wook Kim)

정회원



2004년 울산대학교 공학사
2007년 울산대학교 정보통신공
학석사
2010년~현재 울산대학교 자동
차선박대학원 박사과정
<관심분야> 상황인지컴퓨팅, 무
선통신

오 훈 (Hoon Oh)

중신회원



1981년 성균관대학교 전자공학
학사
1993년 텍사스A&M대학교 전
산학 석사
1995년 텍사스A&M대학교 전
산학 박사
1996년 삼성전자 중앙연구소
수석연구원

2005년~현재 울산대학교 컴퓨터정보통신공학부 부
교수
<관심분야> 실시간 시스템, 임베디드 시스템, 상황
인지컴퓨팅, 이동 애드 혹 및 센서 네트워크 프
로토콜