

소규모 이동 애드혹 네트워크에서의 자동 노드 설정 프로토콜

(Automatic Node Configuration Protocol for Small Sized Mobile Ad-Hoc Networks)

이 혜 원 [†] 문 영 성 ^{††}
(Hyewon K. Lee) (Youngsong Mun)

요 약 이동 애드혹 네트워크(Mobile Ad-Hoc Network)는 기지국에 의존하지 않으면서 다중 홉 기반의 무선 통신을 제공한다. 또, 현재 동적인 주소할당을 위해 주로 사용되는 DHCP나 메시지를 전달을 위해 라우터를 사용하지 않고 통신에 참가하는 노드가 라우터 역할을 제공함으로써 네트워크를 구성한다. 이동 애드혹 네트워크를 위해서 현재 여러 라우팅 프로토콜이 제안되었지만, 이들은 모두 근원지에서 목적지까지의 최적화 또는 최단 경로를 찾기 위한 라우팅 프로토콜만을 기술하고 있으며 네트워크 형성 이전에 노드의 설정이 되어 있다고 가정하고 있다. 이를 보완하기 위해서 MANETConf[1] 및 예언 주소 할당 알고리즘[2] 등이 제안되고 있다. 특히 [1]에서는 단일 계층 구조에서의 노드 주소 할당 및 중복성 회피를 위한 방안을 제시하고 있다. 본 논문에서는 [1]에서 제안하는 방안을 수정 및 보완함으로써 이동 애드혹 네트워크에 참가하는 노드가 보다 효율적으로 주소를 설정할 수 있는 노드 설정 프로토콜을 제안한다. 특히, 새로이 제안되는 방안은 새로운 노드가 네트워크에 진입했을 때 교환하는 메시지 수를 줄임으로써 네트워크 내의 오버헤드를 현저하게 줄였다. 또, 2 계층 네트워크 구조를 기반으로 하기 때문에 주소 중복성 문제를 용이하게 해결한다.

키워드 : 자동 노드 설정 프로토콜, 이동 애드혹 네트워크(MANET), MUNIT(Mobile UNIT), 에이전트, 컨시그너, 주소 중복성 검사

Abstract A Mobile Ad-Hoc Network (MANET) supports a multi-hop wireless network without any prepared base station (BS). The MANET is capable of building a mobile network automatically without any help from DHCP servers for address allocation or routers to forward or route messages. Many routing protocols have been proposed for the MANET, but these specify the most optimized or shortest path from a source to a destination, and they assume that nodes are pre-configured before communication. To make up for this, address allocation algorithms, such as MANETConf [1] and prophet address allocation algorithm [2], have been proposed. Especially, MANETConf proposes address allocation algorithm with duplication address check. In this paper, we present a dynamic node configuration protocol based on 2-tiered hierarchical network architecture for mobile ad-hoc network, modified from [1]. Especially, it reduces the number of broadcast message exchange between nodes when a new node comes into a network, which lessens network overhead, remarkably. This protocol is based on two-tiered structure, and it ensures address allocation with simple duplication address detection mechanism.

Key words : Automatic node configuration protocol, Mobile Ad-Hoc Network (MANET), MUNIT (Mobile UNIT), Agent, Consignor, address duplication detection

· 본 논문은 정보통신부 정보통신연구원에서 지원하고 있는 정보통신기초 연구지원사업의 연구결과입니다.

† 학생회원 : 송실대학교 컴퓨터학부
kerenlee@sunny.ssu.ac.kr

†† 종신회원 : 송실대학교 컴퓨터학부 교수
mun@computing.ssu.ac.kr

논문접수 : 2004년 6월 1일
심사완료 : 2004년 11월 4일

1. 서론

이동 애드혹 네트워크는 기지국에 의존하지 않는 다중 홉 무선 네트워크이다. 이 기술은 현재 동적인 주소 할당을 위해 주로 사용되는 DHCP 프로토콜이나 경로 설정을 위해서 라우터를 사용하지 않고, 통신에 참가하

는 노드가 라우터 역할을 함으로써 네트워크를 구성한다. 이동성을 지원하는 무선 네트워크 또는 유선 네트워크와 이동 애드혹 네트워크의 가장 큰 차이는 하부 인프라에 의존하지 않으면서 네트워크 토폴로지가 빠른 속도로 연속적으로 바뀐다는 점이다.

현재 DSR[3], AODV[4], TBRPF[5] 등의 라우팅 프로토콜이 제안되었지만 이들은 모두 근원지에서 목적지까지의 최적화 또는 최단 경로를 찾기 위한 라우팅 프로토콜만을 기술하고 있어서 네트워크 형성 이전에 노드의 설정이 되어 있다고 가정하고 있다. 이를 보완하기 위해서 MANETConf[1] 및 예언 주소 할당 알고리즘[2] 등이 제안되고 있다. 특히 [1]에서는 단일 계층 구조에서의 노드 주소 할당 및 중복성 회피를 위한 방안을 제시하고 있다.

본 논문에서는 [1]에서 제안하는 방안을 수정 및 보완함으로써 이동 애드혹 네트워크에 참가하는 노드가 보다 효율적으로 주소를 설정할 수 있는 노드 설정 프로토콜을 제안한다. 주소 중복성 문제를 해결하기 위해서 본 논문에서 제안하는 프로토콜은 균일 계층의 네트워크 구조가 아닌 에이전트(Agent)와 컨시그너(Consignor)로 구성된 2 계층 네트워크 구조를 기반으로 하고 있다[6]. 주소 유일성 및 안정적이고 빠른 노드 설정을 보장하기 위해 이동 애드혹 네트워크는 MUNIT(Mobile UNIT)라는 작은 네트워크 단위로 분할된다. 이 구조는 또한 빠른 주소할당 및 용이한 중복성 검사를 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 이동 애드혹 네트워크에서 현재까지 제안된 노드 설정 프로토콜 및 알고리즘을 살펴보고, 3장에서는 자동 노드 설정 프로토콜을 위해서 본 논문에서 제안하는 모델 및 세부 프로토콜을 기술한다. 4장에서는 성능평가를 위한 환경을 설명하고 제안된 프로토콜의 성능을 평가한 후, 5장에서 향후 과제를 남기면서 결론을 짓는다.

2. 관련연구

2.1 애드혹 라우팅 프로토콜

현재 네트워크 상의 노드에게 이동성을 제공하기 위해서 Mobile IPv4 및 Mobile IPv6 등의 인터넷 프로토콜이 제안되고 있다[7]. 이들 프로토콜은 모두 현재의 유선 인터넷 환경을 기반으로 단말기에 이동성을 제공하는 것으로 이동 애드혹 네트워크 상의 독립적인 단말기로의 이동성 제공은 적합하지 않다.

이동 애드혹 네트워크를 위해서 현재 제안된 라우팅 프로토콜은 이동 노드가 통신 전에 라우팅 정보를 알고 있는지 여부에 따라 크게 proactive, reactive, hybrid 방식으로 분류 할 수 있다. 이들 방식을 각기 이용한 프로토콜을 그림 1에서 보이고 있다. 먼저 proactive 방식

은 모든 노드가 네트워크에 진입한 후에 네트워크의 전체 라우팅 정보를 가지고 있다. 현재, DSDV, CGSR 프로토콜에 대한 연구가 활발히 진행되고 있다. 이에 반해 reactive 방식은 노드가 필요한 경우에만 해당 노드의 라우팅 정보를 질의하는데 대표적으로 AODV, TORA, DSR 등이 있다. Hybrid 방식은 앞서 설명한 두 기법의 장점을 혼합한 것으로 ZRP[9]가 있다.

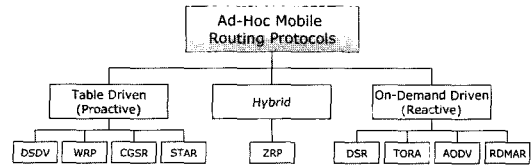


그림 1 애드혹 라우팅 프로토콜의 분류

2.2 Zeroconfiguration

IETF 내의 Zeroconf 워킹그룹[7]에서는 사용자나 관리자의 관여가 없는 네트워크 설정 기술을 연구하고 있다. 이 기술은 작은 규모의 네트워크나 교통 및 운송 수단에서의 네트워크 및 홈 네트워크 등에서 IP 네트워킹을 쉽게 운용되도록 고안되었다. 특히, 이 기법은 네트워크 내에서 주소 설정 정보가 전혀 없을 때 네트워크 설정을 위해 유용하게 사용할 수 있는 방안이지만 서비스를 제공하고 있는 토폴로지가 논리적 혹은 물리적으로 단일 네트워크 세그먼트를 가정하고 있기 때문에 유선 기반의 네트워크에 적합하다.

2.3 주소할당 알고리즘

이동 애드혹 네트워크에서의 주소할당 문제를 해결하기 위해서 현재까지 충돌감지(conflict-detection) 주소할당 알고리즘, 충돌회피(conflict-free) 주소할당 알고리즘, 최선(best-effort) 주소할당 알고리즘, 예언(prophet) 주소할당 알고리즘 등이 제안되었다.

먼저 충돌감지 주소할당 알고리즘[11]은 새로운 노드가 네트워크에 진입했을 때 충돌에 대한 고려 없이 임의로 IP 주소를 선택하여 네트워크 내의 다른 모든 노드에게서 사용 승인을 받는 일련의 '시행착오' 방식이다. 만약 다른 노드가 이 IP 주소를 사용하고 있다면, 새로운 노드는 다른 주소를 선택하여 중복이 없을 때까지 승인 요청을 수행하고 마지막에 선택한 주소를 자신의 영구 주소로 사용한다. 충돌감지 알고리즘은 가장 간단한 알고리즘으로 상태유지를 할 필요가 없다. 그러나 주소 충돌 감지를 위하여 브로드캐스트 방식을 도입하기 때문에 높은 통신 오버헤드가 발생하게 되며 적은 확장성을 갖게 된다.

충돌감지 알고리즘과는 달리 충돌회피 주소할당 알고리즘[12]은 네트워크 내의 주소 풀을 관리함으로써 주소

할당시 충돌을 회피하는 방식이다. 새로운 노드가 네트워크에 진입할 때마다 주소 풀에서 미사용중인 주소를 선택하여 할당한다. 만약 노드가 네트워크 밖으로 이동하거나 통신을 그만 두는 경우, 이웃 노드에게 주소 사용 여부를 알림으로써 이 노드에게 할당되었던 주소는 다시 주소 풀에 등록되어 새로운 노드에게 할당할 수 있다. 이 알고리즘은 주소 할당보다는 주소 풀 관리로 인한 오버헤드가 높다는 단점을 가지고 있다. 즉, 충돌 회피 알고리즘은 주소할당 관점에서는 간단하지만 주소 풀 관리측면에서는 복잡하다. 대부분의 통신이 두 이웃 노드 사이에서 발생하게 되므로 통신 오버헤드는 낮은 장점이 있다.

최선 주소할당 메커니즘은 네트워크에 새로운 노드가 진입하면 이웃의 노드들 중에 하나가 IP 주소를 선택해 주는 방식이다. 동시에 여러 노드가 네트워크에 진입하는 경우, 동일한 IP 주소를 할당 할 수 있기 때문에 주소 중복 확인 절차가 필요하다. 이 메커니즘을 사용하는 대표적인 예로 DDHCP(Distributed Dynamic Host Configuration Protocol)가 있다. MANETconf[1] 역시 최선 주소할당 메커니즘을 적용하고 있다. DDHCP는 모든 네트워크 내의 노드에 할당된 주소 상태를 유지함으로써 자원의 가용성 여부를 결정한다. 최선 알고리즘은 주소충돌 가능성, 높은 통신오버헤드, 낮은 확장성 등 충돌감지 알고리즘과 거의 유사한 특징을 갖는다. 그러나 글로벌한 상태유지를 필요로 하기 때문에 상태관리나 동기화 관점에서 좀 더 복잡하다.

이동 애드혹 네트워크에서 단일 계층구조를 기반으로 2 단계 주소할당 메커니즘을 적용한 프로토콜을 제안하고 있는 MANETconf[1]에서는 새로운 네트워크에 진입한 노드에 주소를 할당하기 위해서 주소의 유일성 여부를 검사할 때 모든 노드가 브로드캐스트 메시지를 주고받기 때문에 오버헤드가 크고, 특히 규모가 큰 네트워크에서 확장성을 제공하지 않는다는 문제점이 있다. 이외에도 네트워크의 모든 노드가 다른 모든 노드의 정보를 알고 있어야 하는 문제점이 있다. 따라서, 주소할당을 위해 주소 관리자 역할을 하고 있던 어떤 노드가 갑자기 사라진 경우 안정성 및 복원력에 있어 많은 문제점을 안고 있다.

예언 주소할당 알고리즘[2]은 앞에서 설명한 세가지 알고리즘을 혼합한 것으로 노드 초기화 및 주소 할당시 사용된다. 특히, 선임노드가 후임노드의 주소를 미리 결정할 수 있을 뿐만 아니라 잠재적인 주소의 충돌을 예측할 수 있기 때문에 선임노드를 예언노드라고 부르며 여기에 사용되는 알고리즘을 예언 주소할당이라 한다. 이 방법의 취약점은 네트워크 종료 전에 주소 관리자 역할을 하고 있던 노드가 떠날 수 없다는 것이다.

3. 자동 노드 설정 프로토콜

자동 노드 설정 프로토콜을 위해서 본 논문에서는 그림 2에서 볼 수 있는 것처럼 여러 독립적인 MUNIT로 구성된 네트워크 모델을 제안한다. 각 MUNIT는 하나의 에이전트와 하나 혹은 그 이상의 컨시그너로 구성된다. 단일 에이전트가 다른 노드 혹은 새로이 진입하는 노드에 주소할당 및 주소 유일성 보장 등의 역할을 한다. 특히, 에이전트가 각 MUNIT를 대표하여 독립적으로 관리하는 반면, 네트워크 내의 모든 에이전트는 주소 자원 정보 및 경로 정보를 공유한다는 점에서 [1]에서 제안하는 구조와 큰 차이를 보이고 있다. 컨시그너는 가장 가까운 에이전트를 선택하여 주소를 할당 받을 수 있다.

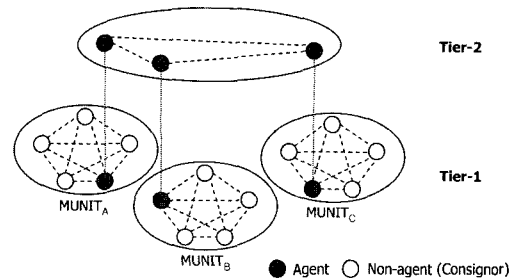


그림 2 MANET을 위한 자동 노드 설정 프로토콜의 계층적 구조

3.1 MUNIT의 구성

MUNIT에서 관리자 역할을 하고 있는 에이전트가 가지고 있는 특성은 다음과 같다. 에이전트는 다른 나머지 컨시그너 노드를 위해 중복이 없는 주소를 할당하고 MUNIT에 속해있는 노드의 네트워크 전·출입 및 주소를 관리한다. 이를 위해 자신이 할당한 MUNIT 내의 멤버뿐만 아니라 다른 에이전트가 할당한 주소 정보를 가지고 있다. 논리적으로 2 계층에 위치하고 있는 에이전트는 에이전트 간 교환되는 갱신 메시지를 통해서 네트워크 내의 다른 모든 에이전트 주소 정보를 배울 수 있을 뿐만 아니라, 각 에이전트가 관리하는 MUNIT 내의 멤버 주소 정보 또한 갱신 메시지를 통해 교환하기 때문에 각 MUNIT의 구성원 정보를 쉽게 얻을 수 있다.

에이전트는 네트워크 내에서 할당되어 사용중인 주소 (allocated address, 할당주소)와 현재 중복성 검사 중에 있는 주소(on-pending address, 예비주소) 및 휴지상태에 있는 주소(free address, 휴지주소) 정보를 가지고 있다. 특히, 에이전트의 원활하고 효율적인 MUNIT 관리를 위해서 아래의 부등식 (1)을 만족해야 한다. 이 부등식에서 h_{count} 는 에이전트와 가장 멀리 위치한 컨시그

너간의 최대 거리를, h_{count} 는 MUNIT에 등록할 수 있는 최대 컨시그너 수를 의미한다. 에이전트와 컨시그너간의 거리는 이들 사이의 홉 수를 사용한다. 부등식 (1)에서 m 과 n 은 상수로 네트워크 상의 이동 노드의 성능이나 배터리 상태로 결정된다. 본 논문에서는 m 과 n 의 최적의 값에 대해서 다루지 않는다.

$$h_{count} \leq n, c_{count} \leq m. \quad (1)$$

가장 가까운 에이전트를 선택하여 주소를 할당 받는 컨시그너는 이웃하고 있는 에이전트 주소 및 MUNIT 내의 모든 노드의 주소 정보를 가지고 있다. 컨시그너는 동일한 MUNIT에 있는 구성원은 서로의 위치 정보(홉 수 정보) 및 주소 정보를 알고 있지만 다른 MUNIT의 구성원에 대한 정보는 가지고 있지 않다는 점에서 에이전트가 가지고 있는 정보와 차이가 있다. 예를 들어, 그림 2에서 MUNIT_A에 있는 컨시그너는 이웃하고 있는 모든 노드를 알고 있지만 MUNIT_B나 MUNIT_C에 있는 컨시그너 노드에 대한 정보를 가지고 있지 않다.

홉 정보는 노드간 메시지 교환시 갱신된다. 즉, 새로운 노드가 MUNIT에 진입하여 주소를 할당받은 후 이웃 노드에 대해서 에이전트에게 이웃 노드에 대한 주소 정보를 받게 되는데, 거리 정보(홉 수)는 MUNIT 내의 노드간 메시지 교환 및 메시지 전달시에 갱신된다. 또, 에이전트간 홉 수 역시 에이전트간 메시지 교환을 통해 갱신된다. 이 외에 에이전트의 갑작스런 부재로 인한 네트워크 불안정을 막기 위해 컨시그너는 이웃한 에이전트의 위치 정보를 알고 있다.

3.2 주소 자원의 분류

본 논문에서는 이동 애드혹 네트워크에서의 주소자원을 예비주소, 할당주소, 휴지주소 등의 세 가지로 분류한다. 예비주소는 주소 중복성 검사 중에 있는 주소로, 주소 중복성 검사가 끝나면 주소의 상태는 '예비'에서 '할당'으로 바뀐다. '예비' 및 '할당'으로 등록된 주소는 다른 에이전트가 사용할 수 없다. 여러 에이전트가 한 주소를 동시에 예비 상태로 등록하여 중복성 검사를 시작하는 경우 충돌이 일어나는데, 이때 먼저 선점한 에이전트가 예비 상태의 주소를 사용하게 된다. 만약 동일한 시간에 주소가 예비 상태로 등록된 경우 이 주소를 버리고 새로운 주소를 선택하여 주소를 할당하게 된다. 또, 어떤 노드가 네트워크를 떠날 때 이 주소를 관리하던 에이전트는 주소의 상태를 '할당'에서 '휴지'로 바꾼다. 이 상태변화는 에이전트 및 해당 MUNIT 내의 구성원만이 알고 있다. 새로운 노드가 네트워크에 진입하여 주소를 요청할 때 에이전트는 주소 중복성 검사 없이 휴지주소를 바로 할당 할 수 있다.

그림 3은 주소의 상태 천이를 보여주고 있다. 예비상태에 들어간 주소는 주소 중복성 검사 후 유효한 경우

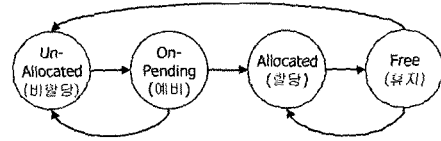


그림 3 주소 상태 천이도

에 할당 상태로 넘어간다. 할당 상태의 주소는 해당 노드가 네트워크를 떠날 때 휴지상태로 되었다가 다른 새로운 노드가 주소를 요청하면 중복성 검사 없이 바로 휴지주소를 할당할 수 있다. 따라서, 정해진 시간 이내에 주소를 요청하는 노드가 없는 경우 주소의 상태는 휴지에서 '비할당'으로 변경되며 에이전트간의 주기적인 갱신메시지를 통해 주소 상태변화에 대한 정보가 교환된다.

비할당에서 예비로의 상태 전이를 제외한 주소 상태 전이는 네트워크 전역 혹은 MUNIT 내의 멤버에게 상태 변화와 동시에 알려지지 않고 약간의 지연을 둔 채로 된다. 즉, 주소 상태 변화는 이웃 혹은 다른 에이전트에 바로 알려지지 않고 주기적인 갱신 메시지를 사용하여 나중에 알려지는데, 이는 약간의 정보 갱신 지연을 감수함으로써 통신 오버헤드를 줄일 수 있다. 예비상태의 주소는 다른 노드가 사용할 수 없을 뿐만 아니라 휴지상태의 주소 역시 해당 에이전트에게는 할당상태로 알려져 있기 때문에 지연으로 인한 주소 중복성 문제는 발생하지 않는다.

에이전트는 네트워크 내의 주소 자원을 관리하기 위해서 네트워크 내의 모든 할당된 주소와 예비주소 및 자신이 관리하고 있는 휴지주소 리스트를 가지고 있다. 네트워크 내의 할당된 주소는 에이전트와 컨시그너 주소 정보로 분류되어 관리될 수 있다. 이에 반해 컨시그너는 동일한 MUNIT 내의 이웃에 할당된 주소와 이웃하고 있는 에이전트의 주소만을 알고 있다.

표 1 주소 형식

주소 타입	구조
예비주소 (On-pending address)	{State ID=1, Agent's address, on pending address, on-pending time, the number of hop count }
할당주소 (Allocated address)	{State ID=2, Agent's address, allocated address, allocated time, the number of hop count}
휴지주소 (Free address)	{State ID=3, Agent's address, freed address, freed time, NULL}

각 주소 타입과 필드가 표 1에 정의되어 있다. 'State ID'는 주소의 상태를 인식하기 위한 것으로 1은 예비주

소 상태를, 2는 할당주소 상태를, 그리고 3은 휴지주소 상태의 주소를 의미한다. '에이전트의 주소'에는 주소를 할당한 에이전트의 주소가 저장될 것이다. '휴 수' 정보는 각 MUNIT의 경계에 위치하고 있는 노드가 어떤 것인지, 또 노드간 거리 정보를 계산하기 위해 필요하다. 이 외에 각 레코드에 기록되는 '할당된 시간'은 각 노드의 네트워크 상주시간을 계산하기 위해 필요하다. 예를 들어, 네트워크에 Agent_A와 Agent_B가 있고 Agent_A가 Consignor_C에게 예비주소를 할당하고 Consignor_D의 주소를 휴지상태로 전환한다고 가정하면 Agent_A와 Agent_B가 각각 가지고 있는 주소 정보는 아래 그림 4와 같다. 그림 4에서는 IPv4 사설 주소를 가정하고 있으며 129.168을 프리픽스로 하는 주소를 사용한다.

Agent _A	192.168.1.5	Agent _B	192.168.2.25
Consignor _C	192.168.1.55	Consignor _D	192.168.1.58

(a) 주소 정보

...				
{2	192.168.2.25,	192.168.2.25,	15:11:03:03,	3}
{1,	192.168.1.5,	192.168.1.55,	15:11:03:02,	1}
{3,	192.168.1.5,	192.168.1.58,	15:25:34:09,	null}

(b) AgentA의 주소 자원 정보 테이블

...				
{2	192.168.1.5,	192.168.1.5,	14:55:03:07,	3}
{1,	192.168.1.5,	192.168.1.55,	15:11:03:02,	4}
{2,	192.168.1.5,	192.168.1.58,	15:02:15:33,	4}

(c) AgentB의 주소 자원 정보 테이블

그림 4 에이전트의 주소 자원 정보 테이블의 예

3.3 메시지 형식

노드 설정 프로토콜에서 사용되는 메시지 종류를 표 2에서 보여주고 있다. NS 및 NR 메시지는 노드가 네트워크에서 이웃을 검색하고자 할 때 주로 사용된다. 특히, NR 메시지에는 *M* 및 *R* 플래그가 정의되어 있다. 먼저, *M* 플래그는 노드가 에이전트임을 표시하고 *R* 플래그는 노드가 에이전트인 경우 주소 할당 가능 여부를 알려준다. 주소 할당 가능의 여부는 이 에이전트가 몇 개의 노드를 관리하고 있고 새로운 노드와의 거리가 얼마인가에 따라 결정되며 이는 부등식 (1)을 만족해야 한다. *M*과 *R* 플래그는 에이전트가 응답 메시지를 생성할 때에 사용할 수 있는 플래그로, *M* 플래그가 설정되지 않으면서 *R* 플래그가 설정된 경우 수신 측에서는 무시해야 한다. 본 논문에서는 컨시그너가 이 두 플래그를 설정하지 않는 것으로 가정한다.

새로운 노드가 에이전트에게 주소 할당을 요청할 때 AREQ 메시지와 AREP 메시지가 교환된다. 응답메시지인 AREP 메시지를 전송하기 전에 에이전트는 주소 중복성 여부를 확인해야 하는데, 이를 위해 에이전트 사이에서 ADC 메시지와 ADCR 메시지 교환을 한다. 특히, ADC 메시지는 주소를 할당하려는 에이전트가 네트워크 내의 다른 에이전트에게 중복 여부를 묻는 것으로 멀티캐스트 형태로 전송되며, ADC 메시지를 유발한 에이전트를 제외한 모든 에이전트가 ADCR 메시지는 생성하여 유니캐스트 형태로 응답한다.

컨시그너 노드가 원래의 에이전트로부터 받던 비콘 (beacon) 메시지 혹은 갱신 메시지를 받을 수 없을 때 이 노드는 해당 MUNIT를 관리하던 에이전트가 변경되거나 자신이 다른 MUNIT로 이동한 것으로 간주한다. 이때 컨시그너 노드는 AS 메시지를 전송하고 원래의 에이전트를 제외한 다른 에이전트로부터 응답 메시지를 받게 된다면 새로운 에이전트를 선택할 것이다. 새로운 에이전트에 등록을 하기 위해서 노드는 TJREQ 메시지와 TJREP 메시지를 교환하고, 마지막으로 새로운 에이전트는 이전 에이전트와 CMN 메시지와 CMR 메시지를 교환함으로써 노드의 이동을 알린다. CMN 메시지를 받은 이전 에이전트는 주기적인 갱신 메시지 (NU)를 통해 관리하는 노드의 변화에 대해 네트워크 내의 다른 에이전트에게 알린다. 이 외에도 자신이 관리하는 MUNIT의 노드에게 알린다. 이때 이전 에이전트가 3.4.6에서 설명하는 것과 같이 자신의 기능을 갑작스럽게 그만 두거나 네트워크에서 사라질 수 있는데, 새로운 에이전트는 목적지가 네트워크상에서 사라졌으므로 CMN 메시지를 이전 에이전트에게 전달할 수 없을 뿐만 아니라 응답 메시지(CMR)를 받을 수 없을 것이다. 이와 같은 상황에서 새로운 에이전트는 임계 값 동안 재전송을 한다. 이 후에도 응답이 없으면, 목적지 MUNIT에서 에이전트 릴리즈가 발생한 것으로 간주하여 새 에이전트로부터 메시지를 받을 때까지 휴지상태로 들어간다. 이 휴지시간은 그다지 길지 않을 것으로 예상된다.

에이전트가 통신을 그만두거나, 에이전트 역할을 양도하고자 할 때 혹은 네트워크를 떠나고자 할 때 새로운 에이전트를 선출해야 한다. 이를 위해서 에이전트와 예비 에이전트¹⁾ 사이에서 BAN 메시지와 BAREP 메시지가 교환된다.

메시지가 브로드캐스트 형태로 전달될 때 반드시 제한을 가지고 전송되어야 한다. 본 논문에서 제안하는

1) 컨시그너 중에서 새로운 에이전트를 뽑을 때 에이전트로 선출될 노드를 예비 에이전트라고 부른다. 본 논문에서는 예비 에이전트 선출 방법에 대해서 다루지 않는다.

표 2 메시지 형식

메시지 타입	설명
neighbor_solicit (NS)	노드가 네트워크에 진입하여 이웃을 찾거나 이동 후 이웃한 에이전트를 찾기 위해 전송하는 메시지
neighbor_reply (NR)	NS 메시지에 대한 응답 메시지로 NS 메시지를 받은 노드가 전송. M 및 R 플래그가 정의되어 있다.
address_request (AREQ)	노드가 IP 주소를 요청할 때 에이전트에게 전송하는 메시지
address_reply (AREP)	에이전트가 AREQ 메시지를 보낸 노드에게 전송하는 응답 메시지로 주소 중복성 검사 후 전송한다.
address_duplication_check (ADC)	에이전트가 예비주소에 대한 중복성 검사를 할 때 네트워크 안의 다른 에이전트에게 멀티캐스트 형태로 전송하는 메시지
address_duplication_check_reply (ADCR)	ADC 메시지에 담긴 주소가 이미 할당되었는지 아니면 이미 다른 에이전트가 예비상태로 등록을 했는지 등을 확인하여 ADC 메시지를 발생한 에이전트에 유니캐스트 형태로 응답하는 메시지이다.
agent_solicit (AS)	컨시그너가 자신이 속해있던 에이전트가 바뀌었거나 위상에 변화가 왔음을 감지했을 때 에이전트를 찾기 위해 전송하는 메시지로 멀티캐스트 형태로 전송
agent_reply (AR)	AS 메시지에 대한 응답 메시지로 유니캐스트 형태로 전송된다.
to_jon_request (TJREQ)	노드가 이동한 것을 감지한 후 다른 MUNIT에 등록하기 위해 새로운 에이전트에게 전송하는 메시지
to_jon_reply (TJREP)	에이전트가 TJREP 메시지를 보낸 노드에게 전송하는 응답 메시지
consignor_move_notify (CMN)	컨시그너가 다른 MUNIT로 이동한 것을 감지했을 때 새로운 에이전트가 이전 에이전트에게 전송하는 메시지
consignor_move_reply (CMR)	CMN 메시지에 대한 응답 메시지로 이전 에이전트가 새로운 에이전트에게 전송하는 메시지
be_agent_notify (BAN)	에이전트가 네트워크를 떠나거나 이동할 때 새로운 에이전트를 선출하기 위해서 해당 MUNIT의 노드에게 전송하는 메시지
be_agent_reply (BAREP)	BAN 혹은 BAREQ 메시지에 대한 응답 메시지로 선택된 노드가 에이전트가 될 것을 수락한 경우 전송
be_agent_request (BAREQ)	노드가 이동한 컨시그너 노드에게 에이전트가 될 것을 요청 할 때 전송하는 메시지
agent_information_solicit (AIS)	노드가 네트워크 내의 에이전트의 주소 정보를 필요로 할 때 이웃하고 있는 에이전트에게 전송하는 메시지
agent_information_reply (AIR)	AS 메시지에 대한 응답 메시지로 유니캐스트 형태로 전송된다.
neighbor_update (NU)	네트워크에 변화가 생겼을 때 에이전트간에 혹은 에이전트와 컨시그너 간에 교환하는 메시지
address_release_request (ARREQ)	컨시그너가 네트워크를 떠날 때 에이전트에게 전송하는 메시지로 에이전트는 컨시그너의 주소 상태를 할당에서 휴지로 바꾼다.
address_release_reply (ARREP)	ARREQ 메시지에 대한 응답 메시지

프로토콜은 NS 메시지와 NR 메시지만이 브로드캐스트 형태로 전달되기 때문에 노드간 통신 오버헤드가 적다.

3.4 자동 노드 설정 프로토콜의 설계

3.4.1 네트워크 초기화

새로운 노드가 네트워크에 진입하면 그림 5에서 볼 수 있는 것처럼 이웃 노드를 찾기 위해 NS 메시지를 브로드캐스트하고 타이머를 동작한다. 이 노드는 타이머가 종료될 때까지 응답 메시지를 기다리는데, 응답이 없는 경우 임계 값 동안 재전송을 한다. 이 후에도 응답이 없으면 노드는 스스로 에이전트가 된다.

1. Repeating Broadcasting NS message for a threshold number of times

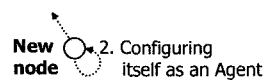


그림 5 초기화 절차

3.4.2 노드의 진입 및 주소 할당

새로운 노드가 네트워크에 진입하면 초기화 절차와 마찬가지로 이웃 노드를 찾기 위해 NS 메시지를 브로드캐스트하고 타이머를 동작한다. NR 메시지를 수신하

면 응답 버퍼링을 시작한다. 버퍼링의 순서는 노드간 응답 지연을 의미하게 된다. 부등식 (1)을 만족하는 에이전트를 찾으면 주소를 요청하고 중복성 검사 후 주소를 할당 받을 수 있다. 이상의 절차는 그림 6에서 볼 수 있다.

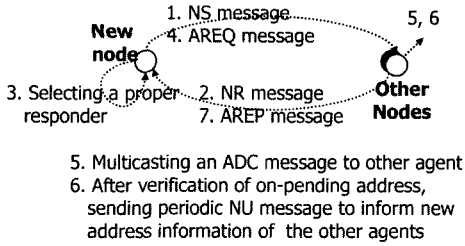


그림 6 노드의 네트워크 진입 후 주소 할당 절차

새로운 노드가 부등식 1을 만족하는 에이전트를 찾지 못한 경우 노드는 NS 메시지에 대한 응답 메시지를 보낸 컨시그너 중에서 하나를 선택하여 에이전트가 될 것을 요청한다. 노드가 BAREQ 메시지를 수신하면, 자신이 속해있던 MUNIT의 에이전트에게 AIS 메시지를 사용하여 에이전트 정보를 요청하고 응답 메시지를 받으면 새로운 데이터베이스를 생성하고 새로운 에이전트가 네트워크에 생성되었음을 네트워크에 알린다. 이 새로운 에이전트는 이제 부등식 (1)을 만족하는 한도 내에서 다른 노드에게 주소를 할당 할 수 있다.

3.4.3 주소 중복성 검사

본 논문에서는 중복 회피 정책을 사용 한다. 즉, AREQ 메시지를 받으면 에이전트는 자신의 데이터베이스를 검색하여 예비주소를 생성하여 할당하고 네트워크 주소 정보를 갱신 한 후 네트워크 내의 모든 에이전트에게 ADC 메시지를 멀티캐스트 한다. ADC 메시지를 수신한 에이전트는 자신의 데이터베이스를 검색 한 후, 자신의 데이터베이스에 예비주소가 등록되어 있지 않는 경우 응답 메시지를 생성하고 예비주소를 주소 정보에 등록한다.

예비주소 사용을 네트워크에 알린 에이전트는 (2)시간 이내에 모든 에이전트로부터 응답을 받으면 주소를 할당할 수 있다. h_{count} 는 에이전트와 가장 멀리 있는 에이전트간 거리를 t_{propa} 는 표준화된 홉간 전파시간을 의미한다.

$$w_{time} = 2 \times h_{a\ count} \times t_{propa} \quad (2)$$

이 외의 경우, 응답이 없는 에이전트에게 ADC 메시지를 임계수치까지 유니캐스트하고 응답이 없다면, 새로운 에이전트가 생성 중인 것으로 간주하여 새로운 에이전트로부터 메시지를 받을 때까지 휴지상태로 들어간다.

이 휴지시간은 그다지 길지 않을 것으로 예상된다. 주소가 중복된 경우, 즉 다른 에이전트로부터 사용 중이거나 이미 예비상태에 들어갔다는 메시지를 받으면 주소를 버리고 다른 주소를 선택하여 이 절차를 다시 시작할 것이다.

에이전트는 새로 할당된 주소를 MUNIT의 구성원에게는 즉시, 다른 에이전트에는 주기적인 갱신 메시지를 통해 알린다. 예비주소를 등록한 에이전트가 이 주소에 대한 상태변화를 알리기 전에는 다른 에이전트가 이를 사용할 수 없기 때문에 갱신메시지의 지연으로 인한 주소할당의 중복성은 일어나지 않는다. 주소 충돌이 발생하는 경우, 경쟁기반의 우선순위를 적용할 수 있다.

3.4.4 노드의 이동

컨시그너가 한 MUNIT에서 다른 MUNIT로 이동할 때 이 노드는 단순히 자신의 MUNIT만 찾으면 된다. 예를 들어, 그림 7에서 한 컨시그너가 MUNIT_A에서 MUNIT_B로 이동하는 경우, 이 노드는 새로운 에이전트를 찾기 위해 AS 메시지를 전송할 것이다. 이 노드가 Agent_A를 제외한 다른 에이전트로부터 응답을 받은 경우, 가장 적절한 에이전트(Agent_B)를 선택하여 TJREQ 메시지를 보낸다. Agent_B는 자신의 데이터베이스를 갱신하고 Agent_A에 노드의 이동을 알린다.

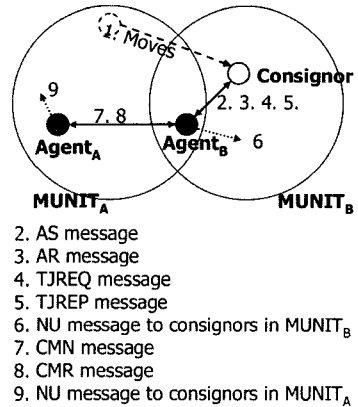


그림 7 노드의 이동시 등록 절차

3.4.5 에이전트 릴리즈

에이전트가 자신의 역할을 그만두려고 할 때 다음 에이전트를 선택한 후에 에이전트의 역할을 릴리즈 할 수 있다. 새로운 에이전트를 선택하는 알고리즘에 대해서 이 논문에서는 다루지 않는다. 네트워크 내 노드의 상주 시간이나 기기의 전파력 등을 기반으로 하여 다음 에이전트를 선출할 수 있을 것이다. 에이전트가 선출되면 이

2) 에이전트의 이동은 에이전트 릴리즈로 분류한다.

전 에이전트와 새로이 뽑힌 에이전트 사이에는 BAN 메시지와 BAREP 메시지를 교환한다. 이제 선출된 에이전트로 데이터베이스를 구성하고 다른 에이전트에게 새로운 에이전트에 대해서 알릴 것이다. 에이전트 릴리즈 후의 노드가 통신을 그만 두고 네트워크를 떠나려는 경우에는 3.4.7에서 설명하는 주소 릴리즈 절차 또한 필요하다.

3.4.6 에이전트 고장

에이전트는 배터리나 전기적인 문제로 인해 예기치 못한 상황에서 네트워크에서 갑자기 사라질 수 있다. 이 경우, 컨시그너는 주기적인 갱신메시지를 에이전트로부터 받을 수 없기 때문에 에이전트의 부재를 인지할 수 있다. 이를 위해서 즉시 새로운 에이전트가 그 역할을 수행해야 한다. 컨시그너 중에서 경쟁방식을 통해 하나를 선출할 수 있다. 새로이 뽑힌 에이전트는 가장 가까이 있는 에이전트에게 AIR 메시지를 전송하여 에이전트 데이터베이스 정보를 요청함으로써 네트워크 내의 에이전트 및 할당된 주소 정보를 배우게 된다. 자신의 MUNIT 내의 정보에 대해서는 모든 노드가 공유하고 있기 때문에 에이전트 고장으로 인해 정보 손실이나 네트워크 불안정은 발생하지 않는다. 다른 나머지 고려사항은 에이전트 릴리즈와 동일하다. 고장난 에이전트의 주소 역시 릴리즈 되어야 한다.

3.4.7 주소 릴리즈

노드가 네트워크를 떠나려고 하는 경우는 사용하던 주소를 반납해야 한다. 따라서, ARREQ 메시지는 노드가 네트워크를 떠나기 전에 에이전트에 전달되어야 한다. 노드가 갑자기 네트워크를 떠난 경우를 위해, 에이전트는 자신이 관리하는 MUNIT 내의 컨시그너가 살아 있는지 주기적으로 검사 패킷을 전송해서 확인해야 한다. 여러 번의 검사 패킷 후에도 응답이 없는 경우 에이전트는 이 주소의 상태를 휴지상태로 바꿀 것이다. 이때 주소 중복성 문제가 발생할 수 있는데, 중복이 발생한 경우 새로운 주소를 할당해야 한다.

4. 성능평가

네트워크상에 교환되는 메시지 타입은 브로드캐스트, 멀티캐스트, 유니캐스트 등이 있다. 브로드캐스트 메시지는 네트워크 내의 모든 노드에게 전파되기 때문에 이 메시지는 많은 네트워크 자원을 사용하고 높은 통신 오버헤드를 초래한다. 논문에서 제안하는 프로토콜은 네트워크 자원의 유용성을 최대화 하기 위해 NS 및 NR 메시지만이 브로드캐스트된다. 멀티캐스트 메시지는 해당 그룹의 멤버에게만 전달된다. 이 메시지는 브로드캐스트 메시지보다 적은 통신 오버헤드를 갖지만 여전히 유니캐스트 메시지보다는 높다. 유니캐스트 메시지는 해당

목적지에만 전송되기 때문에 가장 낮은 오버헤드를 갖는다.

본 논문에서 제안하는 프로토콜은 2계층의 모델을 사용함으로써 브로드캐스트 메시지 수를 현저히 줄일 뿐만 아니라 멀티캐스트 및 유니캐스트 메시지 수를 감소시킨다. 새로운 에이전트를 생성하는 상황을 제외하고는 에이전트만이 주소할당 절차에 참가하기 때문에, [1]에서 제안하는 방법과 비교했을 때 새로운 노드가 네트워크에 진입했을 때 최소의 패킷이 사용된다.

예를 들어, n 개의 노드가 네트워크에 있고, 결합율이 r 인 경우, 한 새로운 노드가 다른 이웃 노드를 찾기 위해 전송하는 브로드캐스트 메시지의 처리 부하(W)를 생각한다면, (3)와 같이 나타낼 수 있다.

$$W = \frac{n}{(1-r)} \tag{3}$$

이 논문에서 제안하는 방법은 브로드캐스트 메시지 전송에 제한을 두고 있기 때문에 부하(W' , $W' \ll W$)는 훨씬 줄어든다. 또, $m(1 < m \ll n)$ 개의 에이전트가 네트워크에 있고 주소 중복율을 l 이라고 가정하면 제안하고 있는 모델에서 주소 중복 검사를 위한 메시지의 처리 부하는 $(m+1)/(1-r)(1-l)$ 이 된다. 반면에 [1]에서의 메시지의 처리 부하는 $2n/(1-r)(1-l)$ 이다. 두 번째 비교에서는 다중 홉 라우팅을 고려하지 않은 처리 부하를 보인 것으로 메시지가 근원지 노드에서 목적지 노드에 직접 전달된다. 다중 홉 라우팅을 고려한다면 이 두 프로토콜 간의 차이는 훨씬 더 크다.

본 논문에서는 새로운 노드가 네트워크에 진입했을 때의 노드 설정을 위해 사용되는 패킷 수 측정을 위한 실험을 하였다. MAC 계층의 프로토콜로는 IEEE 802.11을 사용하고 네트워크 계층에서의 주소 할당을 위해서는 IPv6에서 제안되는 링크로컬 주소를 가정한다. 또, 전송 속도는 2Mbps이고, 패킷간 충돌은 없으며 네트워크에서 에이전트 비율은 정규화되어 정의된다고 가정한다. 또, 이 실험에서는 네트워크 초기 형성을 고려하지 않기 때문에 초기 네트워크에 노드가 이미 있다고 가정한다. 노드가 네트워크에 진입해서 주소를 할당받을 때까지의 워크로드를 산출하기 때문에 네트워크에서의 노드 생존시간은 무한하다고 가정한다. 각 노드는 20초당 진입하며 40초마다 에이전트 사이에서 갱신 메시지가 교환된다고 가정한다. 이와 같은 가정 하에 새로운 노드의 진입시 노드간 교환되는 패킷 수를 수식 (4)과 같이 나타낼 수 있다.

$$p = \alpha p_1(1-p_1)^{n-1} m_i + (1-\alpha)(2 + m_i),$$

$$m_i = a_i + b_i + c_i. \tag{4}$$

식 (4)에서 α 는 NS 메시지를 사용하여 임계 값 시도 내에 에이전트를 찾을 수 있는 확률이다. p_1 은 버퍼에

에이전트가 있을 확률이고, m_i 는 노드간 교환되는 메시지 수로 NS 메시지에 응답하는 노드의 수(a_i)와 일정한 속도로 에이전트 사이에서 교환하는 갱신 메시지의 수(b_i) 및 네트워크상의 에이전트 수($c_i = \lceil n \times \text{agent_ratio} \rceil$)의 합으로 구할 수 있다.

그림 8에서 실선의 점검색 막대 그래프와 점선의 회색 막대 그래프는 각각 에이전트 비율이 10%일 때와 25%일 때의 유니캐스트 메시지 교환을 의미한다. 또, 그리고 실선의 하얀색 그래프와 점선의 하얀색 막대 그래프는 각각 에이전트 비율이 10%일 때와 25%일 때의 멀티캐스트 메시지 교환을 의미한다. 에이전트 비율이 10%일 때와 25%일 때의 유니캐스트 메시지 교환을 비교해볼 때, 에이전트 비율이 높을수록 주소 중복 검사를 위한 에이전트 간의 유니캐스트 응답 메시지 수가 증가하기 때문에, 에이전트 비율이 높을 수록 네트워크 유용성에 나쁜 영향을 준다는 것을 볼 수 있다. 갱신 메시지의 주기가 길어진다면 증가율은 감소할 것이다. 또, 에이전트 비율이 10%일 때, 노드 수는 메시지 수에 그다지 큰 영향을 미치지 않음을 볼 수 있다. 이 외에도 새로운 노드가 진입할 때 교환되는 브로드캐스트 메시지 수는 이웃 노드 발견실패율 평균값의 무한급수로 볼 수 있다. 브로드캐스트 메시지는 멀티캐스트 메시지나 유니캐스트 메시지에 비해 굉장히 낮은 빈도로 사용되기 때문에 그림에서 나타내지는 않았지만 이의 평균값은 6에 가까워서 [1]과 비교했을 때 1/5배 만큼 전송된다. 에이

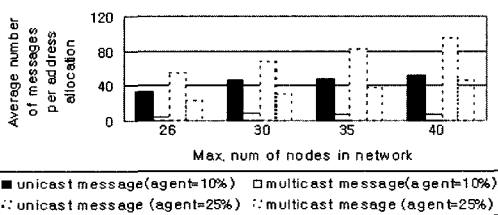


그림 8 초기에 25개의 노드가 존재할 때 에이전트 비율이 각각 10%, 25%일 경우의 통신 오버헤드

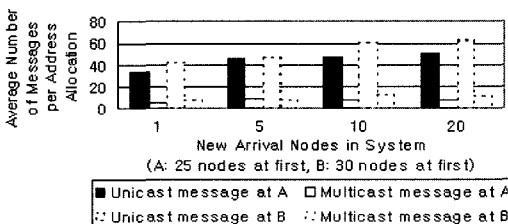


그림 9 에이전트 비율이 10%일 때, 초기에 각 25(A) 및 30개(B)의 노드가 존재하는 경우의 통신 오버헤드

전트 비율이 적절히 낮을 때, 노드 수에 관계없이 유용성은 상당히 좋을 것으로 예상할 수 있다.

그림 9는 에이전트 비율이 10%일 때 초기에 각각 25개의 노드(A)와 30개의 노드(B)가 존재하는 경우의 오버헤드를 보여주고 있다. 실선의 점검색 막대 그래프와 점선의 회색 막대 그래프는 네트워크 상에 있는 초기 노드 수가 각각 25 및 30일 때의 유니캐스트 메시지 교환을 의미한다. 이와 비슷하게, 실선의 하얀색 그래프와 점선의 하얀색 막대 그래프는 네트워크 상에 있는 초기 노드 수가 각각 25 및 30일 때의 멀티캐스트 메시지 교환을 의미한다. 양쪽 모두 노드의 증가에 따라 메시지 전송이 크게 증가하지 않음을 알 수 있다. 특히 이 각 비교에서 사용되는 메시지의 수 차이가 거의 없음을 볼 수 있다. 따라서 제안된 프로토콜을 사용하는 경우 네트워크 초기 크기는 네트워크의 유용성에 그다지 영향을 주지 않음을 알 수 있다.

5. 결론

본 논문에서는 MANETConf[1]에서 제안하는 방안을 수정 및 보완함으로써 이동 애드혹 네트워크에 참가하는 노드가 보다 효율적으로 주소를 설정할 수 있는 노드 설정 프로토콜을 제안한다. 제안된 프로토콜은 MANET에서 2 계층으로 관리되는 네트워크 구조상에서 자동으로 네트워크를 형성하고 주소를 설정한다. 또, 할당, 예비, 휴지 등의 여러 주소 타입과 노드간 교환되는 메시지 타입 또한 제안하였다. 특히, 이 프로토콜은 주소 할당시 유일성을 보장함으로써 [1]에서 제안하는 방법과 비교했을 때 동적인 MUNIT의 구성 및 감소된 오버헤드를 갖는다. 이 외에도 에이전트가 고장이나 다른 이유로 네트워크에서 사라진 경우 빠른 복원력과 안정성을 제공한다. 이 프로토콜은 MANET에서 제안되고 있는 다른 여러 라우팅 프로토콜과 양립되어 사용될 수 있다[9].

참고 문헌

- [1] S. Nesargi, R. Prakash, "MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network," IEEE INFOCOM 2002.
- [2] H. Zhou, L. Ni, M. Mutka, "Prophet Address Allocation for Large Scale MANET," IEEE INFOCOM 2003.
- [3] D. Johnson, D. Maltz, Y. Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc networks (DSR)," work in progress, Apr. 2003.
- [4] C. Perkins, E. Belding-Royer, S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," work in progress, Jul. 2003.
- [5] R. Ogier, M. Lewis, F. Templin, "Topology Dis-

- semination Based on Reverse-Path Forwarding (TBRPF)," work in progress, Dec. 2003.
- [6] Z. Hass, "A New Routing Protocol for the Reconfigurable Wireless Networks."
 - [7] <http://www.ietf.org>
 - [8] Z. Haas and S. Tabrizi, "On Some Challenges and Design Choices in Ad-Hoc Communications," Apr. 1998.
 - [9] M. Pearlman, Z. Haas, "Determining the Optimal Configuration for the Zone Routing Protocol," IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, pp1395-1414, Aug. 1999.
 - [10] S. Corson, J. Macker, "Mobile Ad hoc Networking Routing Protocol Performance Issues and Evaluation Considerations," IETF RFC 2501, Jan. 1999.
 - [11] N. Vaidya, "Duplicate Address Detection in Mobile Ad Hoc Networks," MobiHoc'02, June 2002.
 - [12] A. Misra, S. Das, A. McAuley, S. Das, "Autoconfiguration, Registration, and Mobility Management for Pervasive Computing," IEEE Personal Communication, August, 2001.



이 혜 원

1999년 숭실대학교 컴퓨터학과 학사. 2003년 숭실대학교 컴퓨터학과 석사. 2003년~현재 숭실대학교 컴퓨터학과 박사과정. 관심분야는 Sensor Network, Ad-Hoc Network, IPv6, Mobile IPv6, ENUM, QoS, MPLS



문 영 성

연세대학교 전자공학 학사. 캐나다 Univ. of Alberta 전자공학 석사. Univ. of Texas, Arlington 컴퓨터공학 박사. 1992년 미국 Supercomputing 학술대회 최우수 학생논문상 수상. 숭실대학교 컴퓨터학부 부교수. Journal of Supercomputing 편집위원. 관심분야는 Mobile IP, IPv6, Grid Networking