

MANET 라우팅 프로토콜을 적용한 중첩 이동 네트워크의 경로 최적화 방안 연구

*장성진, 박용진

한양대학교 대학원 전자통신전파공학과

e-mail : sjjang@hyuee.hanyang.ac.kr, park@hyuee.hanyang.ac.kr

A Study on the Route Optimization in MARNET Routing Protocol application Nested NEMO

*Sung-Jin Jang, Yong-Jin Park

Dept. of Electronic and Communication Engineering
Hanyang University

Abstract

NEMO(Network MOBility) is a complex model of mobile network which refers to a forming of Nested Mobile Network by a moving transportation like vessel, bus, or a train which then moves a Mobile Node, or by a movement of a PAN(Personal Area Network). In a Nested NEMO, pinball routing is the primary obstacle as exemplified by the IPv6 environment. This paper will focus on improving such models as RRH, RBU+ and MANET Approach, that attempted to solve pinball routing by route optimization.

I. 서론

무선 네트워크 환경에 Mobile IPv6를 사용하게 될 경우에 MONET 워크 그룹을 통해 다양한 가능성에 대해 많은 연구가 진행되고 있다. 본 논문은 Mobile IPv6를 적용한 무선 네트워크 환경에서 이동하는 Mobile Node가 모여 형성된 Nested Mobile Network와 같은 복잡한 네트워크 모델에서 발생하는 Pinball Routing 문제를 해결하기 위한 경로 최적화 방법 중에 관련연구로 MANET 라우팅 프로토콜을 Nested Mobile Network에 적용한 논문이 발표되었으나 Mobile Node의 대응 노드인 CN이 외부에 존재하는

경우에 Pinball Routing 문제가 발생 되었고 이 문제점을 해결하기 위해 Mobile Node가 이동 중 Binding Update 과정에서 MN의 Home Agent에 TLMR(Top Level Mobile Router)의 CoA를 바인딩 정보로 전달하도록 함으로써 경로 최적화와 Pinball Routing 문제를 해결 할 수 있는 모델을 제안하고자 하며 Nested Mobile Network의 Route Optimization을 위한 개선방법을 제안 하려고 한다.

II. 본론

NEMO의 기본 동작을 그림 1과 같이 나타낸다. 이동 라우터는 홈 망에서 IPv6 주소, 즉 홈 주소를 할당 받으며, 다른 망으로 이동한 경우 그 망으로부터 또 하나의 주소 CoA(Care of Address)를 할당받아 이동 라우터가 현재의 위치를 알릴 수 있는 방법으로 사용한다. 그러나 이동 라우터가 어느 위치에 존재하든지 홈 주소에 의해 식별을 하며 이동망 내의 단말들 또한 항상 홈 주소를 이용하여 통신하게 된다. 이와 같이 이동 라우터는 홈 주소 및 CoA를 자동 할당 받을 때, DHCP 서버로부터 할당 받거나, 인접 라우터의 RA(Router Advertisement) 메시지의 프리픽스 정보를 이용하여 자동으로 구성할 수 있다. 이동 라우터가 CoA를 할당받으면, 홈주소와 CoA로 구성된 바인딩 정보를 홈 에이전트에게 등록한다. 홈 에이전트는 이

바인딩 정보를 이용하여 이동 라우터와 양방향 터널을 설정하여 이동 네트워크 내의 단말과 인터넷과의 연결 서비스를 제공하게 된다.

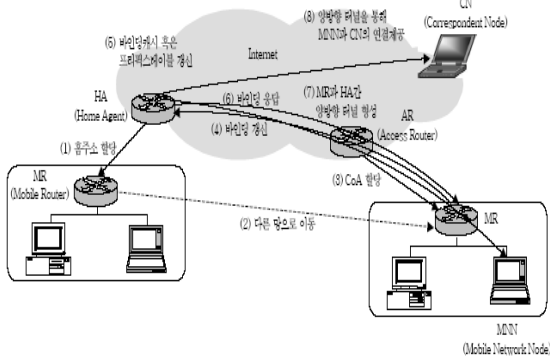


그림1. NEMO의 기본동작

그러나 MN의 대응노드인 CN이 Nested NEMO 내부에 존재하는 경우에는 OLSR 라우팅 프로토콜에 의해 준비된 경로로 외부 HA를 거치지 않고 데이터를 전송할 수 있지만, CN이 Nested NEMO외부에 존재하는 경우에는 그림 2와 같이 HA를 통한 터널이 형성됨을 알 수 있다.

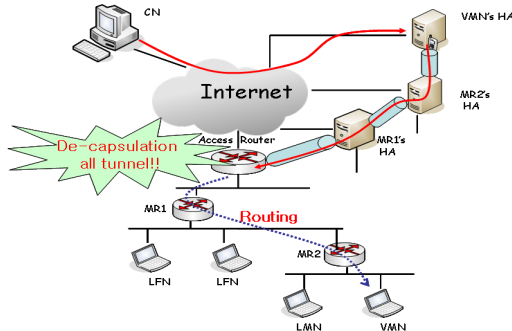


그림2. 인터넷상의 CN과 Nested NEMO 내부의 VMN간의 패킷전송

그림 2와 같이 CN이 Nested NEMO 내부가 아닌 인터넷상의 임의의 위치에 존재하는 경우에는 패킷의 전송 경로가 단축되지 않는다는 것을 알 수 있으며, 이 경우에는 NEMO Basic support의 패킷전송 방법을 사용하는 TLMR 상위의 AR 또는 TLMR이 터널링 되어 온 모든 헤더를 decapsulation 함으로써 TLMR 또는 AR의 부하가 증가 될 수 있다는 것을 보여준다.

III. 제안모델

RRH와 RBU+는 Nested Mobile Network 내부에 AD-Hoc 라우팅 알고리즘을 적용한 것으로 Route Optimization을 구현하기 위한 시도이지만 내부 MN의

HoA와 CoA등의 정보를 TLMR이 수집하기 위해 내부 네트워크에 병목현상과 전송 지연이 발생할 수 있는 단점을 가지고 있다. 제안모델은 이러한 단점을 해결하기 위해 Nested Mobile Network 상의 라우팅 테이블 형성과 내부 토폴로지 유지를 위한 방법으로 MN 또는 MR이 CN과 HA에 전송하는 Binding Update Message를 이용한다. 이를 위해 MIPv6에서 Option 헤더를 추가로 구성해야 하며, Option 헤더의 형식은 Mobility options으로 정의된 Address Option들과 같고 Type value를 별도로 정의 하였다. Option Header는 MR 또는 MN의 Binding Update 패킷이 전송될 때 MR이 자신의 Egress interface의 Care-of Address를 'ECAMR' field에 삽입하여 전송하게 된다. 상위 MR은 해당 필드를 참고하여 Next Hop field를 채우고 자신의 Routing Table을 갱신한 MR은 해당 패킷의 'ECAMR' field에 자신의 Egress interface의 Care-of Address로 해당 필드를 변경하여 전송한다. 이렇게 전송된 패킷이 CN에 전송되게 되면 CN은 MN에 전송할 패킷의 경로가 'ECAMR' field에 있는 Router를 통해 전송되도록 Binding Cache entry를 변경하게 된다. CN과 MN간의 패킷전송은 TLMR의 Egress interface의 CoA를 통해 전송되고 패킷이 TLMR에 도착하면 각 MR은 BU Message에 의해 형성된 Routing Table을 참조하여 패킷을 전달하게 된다.

IV. 결론 및 향후 연구 방향

위와 같은 방법을 사용하면 중첩된 깊이가 깊어질수록 발생하는 패킷 당 헤더 오버헤드와 Binding Cache의 재귀적인 검색을 통한 검색과정에서 생기는 불필요한 시간을 줄일 수 있게 되어 Nested Mobile Network의 Route Optimization을. 기존의 알고리즘을 적용한 방법보다 Binding Update 시간을 줄일 수 있게 된다.

참고문헌

[1] V. Devarapalli, R. Wakikawa, A. Petrescu, P. Thubert, "Network Mobility (NEMO) Basic Support Protocol", Request for Comments: 3963
 [2] T. Clausen, P. Jacquet, "Optimized Link State Routing Protocol (OLSR)", Request for Comments: 3626
 [3] Hosik Cho, Eun Kyoung Paik and Yanghee Choi, "RBU+: Recursive Binding Update for End-to-End Route Optimization in Nested Mobile Networks," 7th IEEE International Conference on High Speed Networks and Multimedia Communications HSNMC'04, Toulouse, France, July 2004.