

# 하이브리드 애드-혹 네트워크를 위한 크로스-레이어 서비스 검색 기법

김문정<sup>†</sup> · 엄영익<sup>††</sup>

## 요 약

무선 애드-혹 네트워크와 유선 네트워크를 연동하는 하이브리드 애드-혹 네트워크에서 효율적인 서비스 검색 기법은 중요하다. 본 논문에서는 무선 애드-혹 네트워크를 인터넷으로 확장한 하이브리드 애드-혹 네트워크를 위해 중복을 허용하는 다중 경로 소스 라우팅 프로토콜을 사용하는 효율적인 크로스-레이어 서비스 검색 기법을 제안한다. 이 기법은 다중 경로 소스 라우팅 프로토콜의 장점뿐만 아니라 크로스-레이어 서비스 검색 기법의 장점을 동시에 갖는 기법이다. 서비스와 경로 검색이 동시에 이루어지는 크로스-레이어 서비스 검색 기법이 서비스 검색 후 해당 경로를 다시 검색하여 서비스를 제공받는 기존 기법보다 적은 오버헤드를 가진다는 것은 직관적으로 이해할 수 있으므로, 본 시뮬레이션을 통해서 이 기법이 각 노드에서 다중 라우팅 경로를 유지함으로써 경로 단절시 보다 빠른 라우팅 복구가 가능하며, 다중 경로의 수를 제한하고 링크/노드 중복을 허용하는 다중 경로를 유지하도록 함으로써 라우팅 오버헤드를 줄일 수 있음으로 보인다.

키워드 : 크로스-레이어 서비스 검색, 다중-경로 라우팅, 링크/노드 중복을 허용하는 다중 경로, 하이브리드 애드-혹 네트워크

## Cross-Layer Service Discovery Scheme for Hybrid Ad-hoc Networks

Moon Jeong Kim<sup>†</sup> · Young Ik Eom<sup>††</sup>

### ABSTRACT

Efficient service discovery mechanism is a crucial feature for a hybrid ad-hoc network supporting extension of a wireless ad-hoc network to the Internet. We propose an efficient cross-layer service discovery mechanism using non-disjoint multi-path source routing protocol for hybrid ad-hoc networks. Our scheme has advantages of multi-path routing protocol and cross-layer service discovery. Intuitively, it is not difficult to imagine that the cross-layer service discovery mechanism could result in a decreased number of messages compared to the traditional approach for handling routing independently from service discovery. By simulation, we show that faster route recovery is possible by maintaining multiple routing paths in each node, and the route maintenance overhead can be reduced by limiting the number of multiple routing paths and by maintaining link/node non-disjoint multi-path.

Keywords : Cross-Layer Service Discovery, Multi-Path Routing, Link/Node Non-Disjoint Multi-Path, Hybrid Ad-Hoc Network

### 1. 서 론

현대 사회는 인터넷 기술과 휴대 전화, 무선 LAN, 무선 PAN 등 무선 통신의 발전으로 다양한 형태의 기기를 네트워크로 연결하여 언제 어디서나 원하는 정보를 주고 받을 수 있는 유비쿼터스 네트워크 환경이 되고 있다. 또한 유비쿼터스 네트워크, 센서 네트워크, 홈 네트워크 등은 미래 사회에 대한 키워드가 되고 있다. 무선 애드-혹 네트워크(wireless ad-hoc network)는 이러한 네트워크들의 원천 기술에 해당하며, 어떠한 중앙 관리 체계 및

기존의 유선 네트워크 하부구조의 도움 없이 이동 노드들만으로 구성될 수 있는 일시적인 네트워크이다[1-3]. 현재 무선 애드-혹 네트워크의 연구 및 표준화 작업이 IETF의 MANET WG을 중심으로 활발히 진행되고 있고, 특히 무선 애드-혹 네트워크와 유선 네트워크를 연동하는 하이브리드 애드-혹 네트워크(hybrid ad-hoc network) 및 서비스 검색을 응용계층과 네트워크 계층이 협력하여 지원하는 크로스-레이어 서비스 검색(cross-layer service discovery) 기법에 대한 연구가 진행되고 있다[4-7].

현재 진행되는 연구는 무선 애드-혹 네트워크에서의 다중 경로에 관한 연구와 하이브리드 애드-혹 네트워크를 위한 게이트웨이 검색기법, 그리고 크로스-레이어 기법의 필요성 등에 관한 연구가 개별적으로 진행되고 있다. 본 논문은 하이브리드 애드-혹 네트워크 환경에서 링크/노드 중복을 허용하는 다중 경로를 이용하여 크로

\* 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음(ITA-2009-(C1090-0902-0046)).

† 정 회 원 : 영동대학교 컴퓨터공학과 교수

†† 종신회원 : 성균관대학교 정보통신공학부 교수

논문접수 : 2008년 10월 13일

수정일 : 1차 2009년 1월 19일

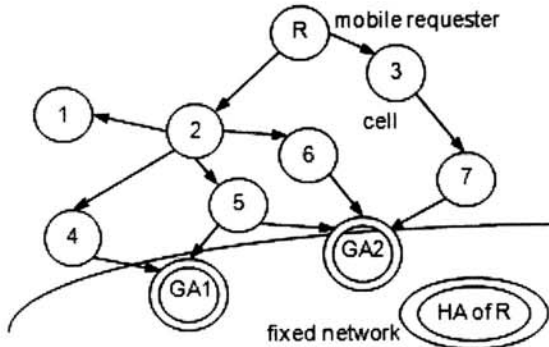
심사완료 : 2009년 1월 20일

스-레이어 서비스 검색 기법을 사용하는 기법을 설계하고자 한다. 무선 애드-혹 네트워크 내의 이동 노드들에게 인터넷 서비스를 제공하며 서비스 검색을 위한 오버헤드를 줄이는 기법이다. 본 논문의 2절에서는 하이브리드 애드-혹 네트워크와 크로스-레이어 서비스 검색 기법에 대해 간략히 소개하고 3절에서는 제안기법에 대해 설명하겠다. 4절에서는 크로스-레이어의 성능은 직관적으로 비교가 가능하므로 링크/노드 중복을 허용하는 다중 경로와 중복을 허용하지 않는 다중 경로 간의 성능 비교를 보이고 5절에서 결론은 맺는다.

## 2. 관련 연구

### 2.1 하이브리드 애드-혹 네트워크

하이브리드 애드-혹 네트워크란 무선 애드-혹 네트워크와 유선 네트워크가 연동되는 환경으로, 무선 애드-혹 네트워크 환경 내의 이동 노드들에게 인터넷 서비스를 제공하기 위해서 필요한 네트워크이다. 하이브리드 애드-혹 네트워크의 시나리오는 (그림 1)에서 보인다.

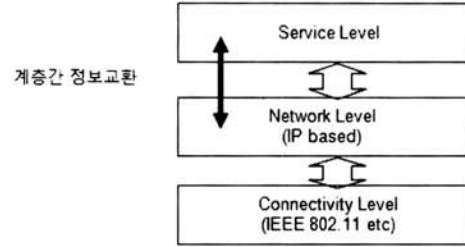


(그림 1) 하이브리드 애드-혹 네트워크

(그림 1)에서 보이는 바와 같이, 유선 네트워크 및 기지국 서비스를 제공 받을 수 없는 이동 애드-혹 네트워크 내의 이동 노드(MH1)는 무선 애드-혹 네트워크를 위한 라우팅 프로토콜을 통해 기지국(GA1와 GA2)를 검색하고, 이들 중 하나의 기지국을 통해 인터넷 서비스를 제공받을 수 있게 된다. 본 논문에서는 링크/노드 중복을 허용하는 다중 경로를 사용하므로 (그림 1)에서 R과 GA1의 경로는 R-2-4-GA1 와 R-2-5-GA1의 두 가지 경로를 유지할 수 있다. 하이브리드 애드-혹 네트워크를 위해서는 참여하는 모든 이동 노드들이 무선 인터페이스를 가지며 라우팅 기능을 제공해야 한다. 또한 유/무선 인터페이스뿐만 아니라 모바일 IP와 무선 애드-혹 네트워크를 위한 라우팅 프로토콜을 동시에 지원하는 게이트웨이 에이전트(gateway agent)가 필요하다. 하이브리드 애드-혹 네트워크를 위해 다양한 프로토콜들이 연구되고 있으며, 게이트웨이 에이전트 발견 과정을 기반으로 proactive, reactive, 그리고 hybrid 게이트웨이 에이전트 탐색 기법으로 구분될 수 있다[4].

### 2.2 크로스-레이어 서비스 검색 기법

기존 서비스 검색 기법은 서비스 레벨에서 수행되는 것이 일반적이지만, 최근에는 네트워크 계층과 협력하는 크로스-레이어 서비스 검색 기법에 대한 연구가 진행되고 있으며 이의 구조는 (그림 2)에서 보이는 바와 같다.



(그림 2) 크로스-레이어 구조

크로스 레이어 서비스 검색 기법은 임의의 서비스를 제공하는 노드와 그 노드의 위치를 동시에 검색하는 기법으로 서비스 계층과 네트워크 계층이 협력하는 기법이다. 따라서 이러한 기법은 레이어가 분리된 전통적인 기법에 비해 서비스 검색으로 인한 메시지 수가 줄어드는 결과를 초래한다[5-7].

현재 제안되는 크로스-레이어 서비스 발견 기법들은 단일 경로를 이용하므로 서비스 검색으로 인한 메시지 수는 줄어드는 반면, 경로 단절로 인한 재경로 설정에 대한 오버헤드는 고려되지 않고 있다. 본 논문에서는 서비스 검색을 위해 네트워크 계층의 지원을 받는 동시에 다중경로를 유지하도록 함으로써, 경로 단절시 보다 빠른 경로 및 제어패킷의 오버헤드를 줄이는 효과를 갖으며 이를 무선 애드-혹에서 부터 인터넷과의 통합 네트워크 기법을 제안하고자 한다. 특히, 다중 경로를 유지하는 과정에서 경로에 포함하는 링크/노드의 중복을 허용하는 경우가 중복을 허용하지 않는 경우에 비해 좋은 성능을 보인다[8,9].

## 3. 제안 기법

본 논문에서는 임의의 서비스를 제공하는 노드와 그 노드의 위치를 동시에 검색하는 크로스-레이어 서비스 검색 기법을 제안한다. 이 때 서비스를 요청하는 노드와 서비스를 제공하는 노드 간에 다중 경로를 유지하여, 하나의 경로가 단절되었을 때 재설정 과정을 거치지 않고 바로 다른 경로를 사용할 수 있도록 한다.

### 3.1 용어

본 논문의 제안 기법에서, 각 이동 호스트(MH: Mobile Host)는 서비스 테이블(ST: Service Table), 서비스 캐시(SC: Service Cache), 에이전트 캐시(AC: Agent Cache), 그리고 라우팅 캐시(RC: Routing Cache)를 유지하고 게이트웨이 에이전트(GA: Gateway Agent)는 애드-혹 네트워크 내의 이동 호스트들에게 인터넷 서비스를 제공하

기 위해 서비스지원 목록(SL: Service supporting List)을 사용한다. 또한 본 제안 기법은 서비스경로요청패킷(SRQP: Service and Route reQuest Packet), 서비스경로응답패킷(SRRP: Service and Route Reply Packet), 서비스요청패킷(SQP: Service reQuest Packet), 그리고 서비스데이터패킷(SDP: Service Data Packet)을 사용한다.

이동 애드-혹 네트워크 내의 각 이동 호스트들은 자신이 소유한 서비스의 목록을 유지하기 위한 ST와 자신이 소유하지 않은 서비스이지만 경로를 알고 있는 경우 해당 경로에 대한 정보를 유지하기 위한 SC를 관리한다. 임의의 이동 호스트(R: Requester)이 특정 서비스를 요청하는 경우, SRQP를 사용하며, 이를 수신한 이동 호스트 중 해당 서비스를 소유한 호스트 또는 GA는 SRRP로 응답할 수 있다. R은 수신한 SRRP 중에 서비스에 대한 정보가 없는 패킷은 GA로부터 송신된 것이므로 이 정보를 자신의 AC에 저장한다. R이 서비스 정보를 포함한 SRRP를 처음 수신하면 서비스를 소유한 호스트로 SQP를 이용하여 서비스를 요청하고, 두 번째부터 수신하는 SRRP의 정보는 SC에 저장한다. 일정시간동안 서비스 정보를 포함한 SRRP를 하나도 수신하지 못하고 서비스 정보를 포함하지 않은 SRRP를 수신한 경우에는 해당 GA로 SQP를 전송할 수 있다. SRRP를 수신한 GA는 자신이 관리하는 SL 내에 해당 이동 호스트에 대한 정보를 저장한 후 대신하여 서비스를 검색하고, 해당 서비스가 검색되면 자신의 서비스 지원 목록을 참고하여 SDP를 생성하고 이를 해당 이동 호스트로 전달한다. 본 논문에서는 두 이동 호스트 간 다중 경로를 유지하기 위해 RC를 사용한다.

### 3.2 기본 시나리오

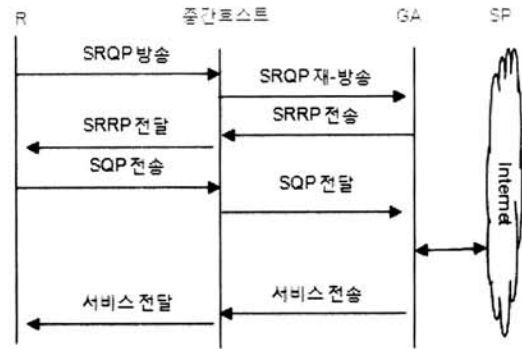
본 논문에서 제안하는 서비스 검색 기법의 동작 시나리오는 (그림 3)에서 보인다.



(그림 3) 기본 동작 시나리오

임의의 서비스를 검색하고자하는 노드, R은 먼저 자신의 ST와 SC를 검색한다. 이 때, 해당 서비스를 발견하지 못하면 (그림 3)에서와 같이 SRQP를 방송하게 된다. 이를 수신한 다른 이동 호스트들은 자신의 ST와 SC를 검색한 후 역시 해당 서비스를 검색하지 못하면 자신의 ID를 경로정보에 추가하여 다시 방송한다. 서비스를 가진 노드(SP)가 해당 SRQP를 수신하면 자신이 가진 서비스

의 자세한 정보를 포함하여 SRRP를 R에게 전송한다. GA가 SRQP를 수신하면 서비스에 대한 정보를 포함하지 않은 SRRP를 생성하여 R에게 전송한다. R은 서비스를 포함한 SRRP를 보낸 SP로 SQP를 송신하여 서비스를 요청할 수 있다. 만일 서비스를 포함한 SRRP를 수신하지 못하고 서비스를 포함하지 않은 SRRP만 수신한 경우 (그림 4)와 같은 요청도 가능하다.



(그림 4) 인터넷 확장 시나리오

SRQP를 방송한 후 일정시간 내에 R이 서비스를 포함한 SRRP를 하나도 수신하지 못하고 서비스를 포함하지 않은 SRRP만 수신한 경우는 (그림 4)에서와 같이 SRRP를 전송한 GA에게 SQP를 송신한다. 이를 수신한 GA는 R을 대신하여 인터넷 상에서 서비스 검색 및 요청을 수행하여 적당한 서비스를 R에게 전달하는 역할을 수행한다.

이 때, 하나의 R은 동일한 SRRP를 다중 경로로 전달받을 수 있으며 제한된 수의 다중 응답으로 노드 R과 SP 또는 GA 간에 다중 경로를 유지하여 하나의 경로가 단절되었을 때 재설정 과정을 거치지 않고 바로 다른 경로를 사용할 수 있는 장점을 가진다.

### 3.3 알고리즘

본 절에서는 본 논문에서 제안하는 기법의 보다 자세한 처리절차를 설명한다.

#### 3.3.1 요청자의 작업

서비스 검색을 원하는 이동 호스트 R(requester)은 먼저 자신의 ST를 확인하여 원하는 서비스를 발견하면 사용한다. 만일, 발견하지 못하면 자신의 SC에서 원하는 서비스에 대한 정보를 검색하여, 발견하면 해당 경로 정보를 통해 SP에게 서비스를 요청하게 된다. 자신의 ST 또는 SC에서 원하는 서비스에 대한 정보를 발견하지 못한 경우 먼저 TTL(Time To Live)을 1로 하여 (그림 5)에서 보이는 작업을 수행한다.

이동 호스트(R)가 서비스 검색을 원할 때, 먼저 TTL을 1로 하여 이웃 호스트들에게만 SRQP 패킷을 방송한다. SRQP를 수신한 호스트 중에서 서비스를 제공할 수 있는 호스트(SP)는 해당 서비스에 대한 정보를 포함한 SRRP를 생성하여 R로 전송한다. 하나의 SP는 다른 경

```

Tw : timeout for waiting a SRRP
{
  set FIRST to 0;
  create and broadcast SRQP;
  while (during Tw) {
    if (received SRRP with service-info.)
      if (FIRST equals 0) { // first SRRP
        set FIRST to 1;
        create and send SQP to the SP;
      } else insert new entry in SC;
    else if (received SRRP without service-info.)
      insert new entry in AC;
  }
}
    
```

(그림 5) 요청자의 작업 알고리즘

로를 통해 전달된 SRRP에 대해 다중 응답을 할 수 있다. 플래그 FIRST가 0이라면 SRRP를 처음 수신한 경우이므로, 해당 SP로 SQP를 전송하여 서비스를 요청한다. 그리고 플래그 FIRST를 1로 설정하여, 두 번째부터 수신하게 되는 SRRP에 대해서는 서비스 요청없이 자신의 SC에 해당 서비스 정보를 저장한다. 만일 서비스에 대한 정보를 포함하지 않은 SRRP를 수신하면 이는 GA로부터 수신된 정보이므로 바로 AC에 저장한다.

Timeout이 되었을 때, FIRST 값을 확인한다. 이 값이 0이라면 서비스를 가진 이동 호스트를 발견하지 못한 경우이므로 AC에 저장되어 있는 GA 정보가 있는지 확인한다. GA에 대한 정보가 있는 경우, 적당한 GA를 선택하여 서비스 요청을 위해 SQP를 전송한다. 만일 적당한 시간(Timeout) 동안에 SP 또는 GA 모두 발견하지 못한 경우라면 TTL을 3으로 하여 다시 (그림 5)의 작업을 수행한다.

### 3.3.2 SRQP를 수신한 이동 호스트의 작업

SRQP 패킷을 수신한 이동 호스트는 (그림 6)에서 보이는 작업을 수행한다.

SRQP 패킷을 수신한 임의의 이동 호스트(M)는 자신의 ST와 SC를 먼저 확인한다. 해당 서비스를 발견하면,

```

if (found requested service in its ST) { // service provider
  if (duplicate information for (SourceID, SeqNo) in History) {
    increase Pathcreated by one;
    if (Pathcreated greater than Pathmax) discard SRQP; // end
    else create SRRP and send back it;
  } else { // new SRQP
    insert new entry in History;
    set Pathcreated to 1;
    create SRRP and send back it;
  }
} else if (not duplicate the information in History) {
  decrease TTL by 1;
  if ((TTL equal zero) or (its own ID exists within header of SRQP))
    discard SRQP; // end
  else {
    append its own ID
    to the end of SRQP and re-broadcast it;
  }
}
    
```

(그림 6) SRQP를 수신한 노드의 작업 알고리즘

M은 서비스제공자(SP)로 역할을 할 수 있다. 해당 요청 호스트(R)로부터 처음 수신한 SRQP인 경우, 해당 서비스의 정보를 포함한 SRRP를 생성하여 R로 전송한다. R로부터 중복 수신된 SRQP인 경우, 최대 다중패킷수를 넘지 않는 범위 내(Path<sub>created</sub> ≤ Path<sub>max</sub>)에서 서비스의 정보를 포함한 SRRP를 생성하여 R로 전송한다. 수신한 SRQP에서 요구하는 서비스를 자신의 ST와 SC에서 발견하지 못한 경우, 이미 수신했던 SRQP과 동일한 R로부터 송신된 SRQP이거나, TTL을 1만큼 감소시켰을 때 0이 되거나, 수신한 SRQP의 라우팅 경로에 자신의 주소가 있는 경우 해당 패킷을 버린다. 그렇지 않다면 수신한 SRQP의 경로 끝에 자신의 경로를 추가하여 재-방송한다.

### 3.3.3 게이트웨이 에이전트의 작업

SRQP를 수신한 GA는 서비스 정보를 포함하지 않은 SRRP를 생성하여 R로 전송하고, SQP를 수신한 GA는 R에 대한 정보를 SL에 저장한 후 R을 대신하여 인터넷상의 해당 서비스를 검색한 후 적절한 서비스를 R로 전달한다.

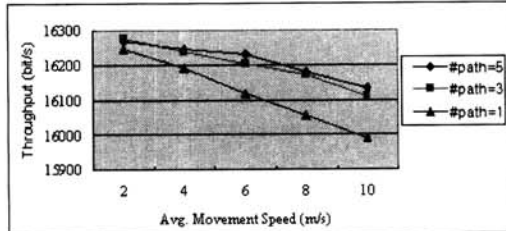
### 3.3.4 SRRP를 수신한 호스트의 작업

SRRP 패킷을 수신하자마자, R은 수신된 SRRP 내의 라우팅 정보를 이용하여 해당 SP로 서비스를 요청한다. R은 자신의 RC에 처음 수신하는 SRRP 패킷의 라우팅 정보를 능동경로(active route)로 저장하고, 다른 SRRP 패킷들 내의 경로정보는 수동경로(passive routes)로 저장한다. 임의의 중간 노드가 하나의 SRRP 패킷을 수신하면 해당 노드는 자신의 RC 내에 해당 경로 정보를 저장한 후 해당 SRRP 경로내의 다음 호스트로 해당 패킷을 전달한다.

## 4. 성능 평가

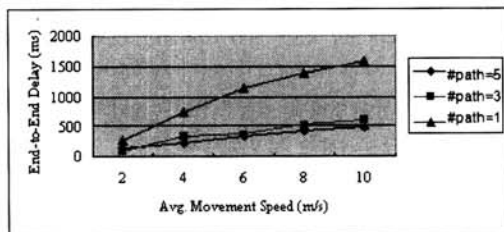
본 논문에서는 제안 기법을 시뮬레이션하기 위해 Simlib이라는 시뮬레이션 툴을 사용하였다[10]. 본 논문에서는 제안 프로토콜의 성능 평가를 위하여 총 시뮬레이션 시간은 1180초로하고 처음 1000초 후에 180초 동안 데이터 패킷을 전송하도록 수행하였다. 본 시뮬레이션을 위해 이동 호스트들의 이동 모델은 random waypoint model을 적용하였으며 pause time은 '0'으로 하였다. 데이터 패킷은 512 bytes CBR(Constant Bit Rate)로 가정하였으며, 초당 4개의 패킷을 전송하고, 홉 간의 전송 지연 시간은 30ms라 가정하였다. 700m X 500m 네트워크 내에 총 40개의 이동 호스트들이 존재하며, 전송 범위를 250m로 가정하였다. 각 시뮬레이션은 총 5회 시행하여 평균을 계산하였다. 본 논문에서는 경로의 수를 1, 3, 그리고 5로 하는 경우에 이동 호스트들의 평균 이동 속도에 따른 비교를 보였다. 서비스와 경로 검색이 동시에 이

루어지는 크로스-레이어 서비스 검색 기법이 서비스 검색 후 해당 경로를 다시 검색하여 서비스를 제공받는 기존 기법보다 오버헤드가 줄어든다[11]. 본 논문에서는 다중경로를 이용함으로써 얻어지는 이점에도 불구하고 다중경로를 유지하기 위한 오버헤드가 크지 않음을 보이고자 하며, 오버헤드는 전송되는 제어패킷들의 수로 계산된다.



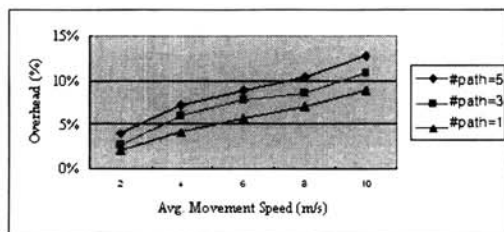
(그림 7) 처리량

(그림 7)은 이동 호스트들의 이동 속도 변화에 따른 처리량을 보인다. 이동 호스트들의 평균 이동속도가 높아짐에 따라 단일 경로를 사용하는 경우 경로 단절이 빈번히 발생하여 처리량의 급속한 감소를 보이지만, 다중 경로를 사용하는 경우 보다 빠른 재설정 가능성이 높아 처리량이 급속히 감소하지 않음을 보인다.



(그림 8) 단대단 지연

(그림 8)은 이동 호스트들의 이동 속도 변화에 따른 단대단 지연을 보인다. 역시, 이동 호스트들의 평균 이동속도가 높아짐에 따라 단일 경로를 사용하는 경우 경로 단절이 빈번히 발생하여 단대단 지연이 급속히 증가하는 반면, 다중 경로를 사용하는 경우 보다 빠른 재설정 가능성이 높아 단대단 지연의 큰 차이가 발생하지 않는다.



(그림 9) 오버헤드

(그림 9)의 결과는 다중 경로를 유지하기 위한 오버헤드와 단일 경로를 유지할 때 경로 단절로 인한 경로 재설정 오버헤드로 오버헤드측면에서는 차이가 없으므로 보인다.

서비스와 경로를 동시에 요청하는 본 논문의 제안기법을 사용하는 경우가 서비스를 요청하고 경로를 다시 요청하는 전통적인 기법에 비해 메시지 오버헤드가 줄어드는 결과는 직관적으로 비교가 가능하다. 따라서 본 논문에서는 크로스 레이어를 사용하면서 다중 경로를 사용하는 경우와 단일 경로를 사용하는 경우에 대해 시뮬레이션 결과를 보였다. 다중 경로를 유지하기 위한 오버헤드와 단일 경로의 끊김으로 인한 경로 재설정 오버헤드로 인해 오버헤드 측면에서는 단일 경로와 다중 경로의 차이가 거의 없었다.

## 5. 결 론

크로스-레이어 서비스 검색 기법은 서비스 검색 절차를 경로 검색 절차와 동시에 수행함으로써 생성되는 제어 패킷의 수를 줄이는 효과를 갖는다. 중복을 허용하는 다중 경로 소스 라우팅 프로토콜의 장점은 경로 실패시 경로 재발견 빈도를 줄인다는 것이다. 각 호스트가 다중 경로를 유지하기 때문에 보다 빠른 경로 복구가 가능하고 다중경로의 수를 제한하고 링크/노드의 중복을 허용하는 다중 경로를 유지하기 때문에 경로 유지 오버헤드를 줄일 수 있다.

본 논문에서는 무선 애드-혹 네트워크를 위해 중복을 허용하는 다중 소스 라우팅 프로토콜을 이용하는 효율적인 서비스 검색 기법을 제안한다. 본 논문에서 제안하는 기법은 크로스 레이어 서비스 검색기법을 제공함으로써 서비스 검색 시간은 단축할뿐만 아니라 다중 경로를 이용함으로써 경로 단절로 인한 경로 재설정 비용을 절감하며 보다 빠른 경로 재설정을 지원한다.

## 참 고 문 헌

- [1] S. Corson and J. Macker, "Mobile ad hoc (MANET): Routing Protocol Performance Issues and Evaluation Considerations," RFC 2501, IETF, 1999.
- [2] 권혜연, 신재욱, 이병복, 최지혁, 남상우, "이동 Ad-hoc 네트워크 서비스," 한국전자통신연구원, Vol.18, No.4, pp.23-35, 2003.
- [3] D. Johnson, Y. Hu, and D. Maltz, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4," RFC 4728, IETF, 2007.
- [4] F. M. Abduljalil and S. K. Bodhe, "A Survey of Integrating IP Mobility Protocols and Mobile Ad Hoc Networks," IEEE Communications Surveys & Tutorials, Vol.9, No.1, pp.14-30, 2007.
- [5] J. A. Garcia-Macias and D. A. Torres, "Service Discovery in Mobile Ad-Hoc Networks: Better at the Network Layer?," Proc. International Conf. on Parallel Processing, IEEE Computer Society, Norway, pp.452-457, 2005.

- [6] A. Varshavsky, B. Reid, and E. Lara, "The Need for Cross-Layer Service Discovery in MANETs," Technical Report CSRG-492, UofT Computer Science, U.S.A., 2004.
- [7] V. Atanasovski and L. Gavrilovska, "Efficient Service Discovery Schemes in Wireless Ad Hoc Networks Implementing Cross-layer System Design," Proc. 27th International Conf. on Information Technology Interfaces, Cavtat/Dubrovnik, Croatia, pp.496-501, 2005.
- [8] M. J. Kim, B. H. Kim, D. H. Lee, and Y. I. Eom, "Non-disjoint Multi-path Source Routing Protocol for Ubiquitous Network," Proc. of the International Ubiquitous Information Technology Applications Conf., pp.424-433, 2007.
- [9] M. J. Kim, D. H. Lee, and Y. I. Eom, "A Transformation Scheme for Enhanced Wireless Ad-Hoc Networks," Proc. of the International Communications in Computing Conf., pp.36-42, 2007.
- [10] A. M. Law and W. D. Kelton, 'Simulation Modeling and Analysis,' 3rd ED., McGraw-Hill, 2000.
- [11] G. P. Halkes, A. Baggio and K. G. Langendoen, "A Simulation Study of Integrated Service Discovery," Lecture Notes in Computer Science 4272, Springer Berlin, Heidelberg, pp.39-53, 2006.



### 김문정

e-mail : tops@youngdong.ac.kr

1988년 성균관대학교 정보공학과(학사)

2000년 성균관대학교 전기전자및컴퓨터공학과(공학석사)

2005년 성균관대학교 전기전자및컴퓨터공학과(공학박사)

2006년~2007년 고려대학교 BK21유비쿼터스정보보호사업단 연구교수

2008년 성균관대학교 정보통신공학부 연구교수

2009년~현 재 영동대학교 컴퓨터공학과 교수

관심분야: 무선 애드-혹 네트워크, 서비스 디스커버리, 이동 에이전트 등



### 염영익

e-mail : yieom@skku.edu

1983년 서울대학교 계산통계학과(이학사)

1985년 서울대학교 전산과학과(이학석사)

1991년 서울대학교 전산과학과(이학박사)

1993년~현 재 성균관대학교 정보통신공학부 교수

2000년~2001년 Dept. of Info. and Comm. Science at UCI 방문교수

2006년~2007년 성균관대학교 이동통신교육센터 센터장

2007년~현 재 성균관대학교 정보통신처장

2008년~현 재 한국정보과학회 학회지 편집 부위원장

관심분야: 시스템 소프트웨어, 분산 컴퓨팅, 미들웨어, 시스템 보안 등