

## 대규모 이동 애드 혹 망을 위한 스케일러블 동적 소스 라우팅

최명수<sup>0</sup> 정재일  
 한양대학교 전자통신전파공학과  
 mschoi@mnlab.hanyang.ac.kr<sup>0</sup> jijung@hanyang.ac.kr

### A Scalable Dynamic Source Routing for Large Mobile Ad Hoc Networks

Myoungsoo Choi<sup>0</sup> Jaeil Jung  
 Dept. of Division of Electrical and Computer Engineering, Hanyang University

#### 요 약

소스 호스트와 목적지 호스트간 경유 경로를 패킷 헤더에 모두 포함하는 DSR(Dynamic Source Routing)은 경유 호스트 수 증가에 비례 해서 패킷 헤더와 개별 호스트들이 유지 하는 라우트 캐쉬의 크기가 커져야 한다. 따라서 경로 크기와 선형적 관계로 유발되는 확장성 문제로 DSR을 수천 개 이상의 호스트로 구성 되는 대규모 애드-혹 망에 적용하기 어렵다. IETF(Internet Engineering Task Force)는 이러한 단점의 회피 방법으로 Flow State 매카니즘을 제안 하고 있다. 그러나 이 방법은 Route Discovery를 위한 Route Request 와 Route Response 패킷에 전체 경로를 포함 해야 하는 것은 물론 Route Cache에도 전체 경로를 저장 해야 하기 때문에 DSR 확장성 문제를 해결 할 수 없다. 본 논문에서는 경로 크기에 비례해 증가하는 DSR의 오버헤드를 패킷 헤더는 6-hop, 라우트 캐쉬는 12-hop 크기로 한정 하는 경로 분할 라우팅 방법을 제안한다.

#### 1. 서 론

무선 전송기술의 발전과 휴대용 컴퓨팅 기기의 대중화에 따라 mobile ad hoc network의 필요성이 증가 하고 있다. Mobile ad hoc network(MANET)[1]은 이동이 자유로운 무선 호스트로 구성 되는 임시 네트워크이다. 네트워크에 존재 하는 호스트들이 모두 동일한 무선 송신 범위 안에 함께 존재 하지 않기 때문에 MANET에서 패킷은 다중 홉 방식으로 전송 된다. 한편 MANET은 고정된 라우터와 같은 기반 구조(infrastructure)를 포함 하지 않기 때문에 개별 호스트가 라우터의 역할을 병행 해야 한다[2]. MANET의 라우팅은 호스트간 위상 변화에 대응 하는 방법에 따라 table-driven 방식과 on-demand 방식으로 구분 된다[3].

Table-driven 라우팅 방식으로 동작 하는 호스트는 이웃 호스트간 링크 상태(Link State)에 변화가 발생 할 때마다 자신의 모든 이웃 호스트들에게 그 변화를 알린다. 이 변화를 통보 받은 호스트들 역시 수신한 링크 상태 변화를 자신의 이웃 호스트들에게 전송 한다. 따라서 라우팅을 위한 정보 전송 양은 네트워크 크기에 비례 하게 되어 table-driven 라우팅 방식은 확장성 문제를 유발 한다.

On-demand 라우팅 방식으로 동작 하는 호스트는 필요 할 때만 목적지 호스트에 도달 하기 위한 경로를 찾는다. 그리고 찾은 경로에 변화 발생시 변화 공유 및 대체 경로 탐색과 같은 유지보수는 변화가 나타난 경로에 관계

하는 호스트간에만 필요 하다. 따라서 table-driven 방식 보다 확장성에서 유리 하다.

On-demand 방식은 source routing과 next-hop routing으로 구분 된다. DSR(Dynamic Source Routing)은 source routing 방식에 해당 한다[4]. DSR은 소스 호스트가 목적지 호스트에 대한 경로를 알지 못 할 때 경로 탐색을 위해 ROUTE\_REQ 패킷을 브로드캐스팅 한다. ROUTE\_REQ을 수신한 이웃 호스트도 목적지에 대한 경로를 알지 못하면 자신의 주소를 수신한 ROUTE\_REQ 헤더에 추가해 다시 브로드캐스팅 한다. 목적지 호스트가 ROUTE\_REQ를 수신 하면 ROUTE\_REQ 헤더에 저장 되어 있는 경로의 역순으로 ROUTE\_REPLY를 유니캐스팅 한다. 이런 과정을 통해 목적지에 도달 할 수 있는 경로를 발견한 소스 호스트는 데이터를 전송 하기 위해 패킷 헤더에 경로 탐색에서 얻은 목적지까지의 전체 경로를 추가 해 전송 한다. 그 결과 DSR의 패킷 헤더는 목적지까지의 거리에 비례해 커지게 되어 on-demand 방식에도 불구하고 확장성의 문제를 유발 한다. 즉 경로 크기에 비례해 증가 하는 헤더를 전송 하기 위해 낭비 되는 무선 대역폭 과 경로 크기에 비례해 커져야 하는 라우트 캐쉬는 DSR을 대규모 네트워크에 적용 할 수 없게 한다.

본 논문에서는 상기한 DSR의 확장성 문제를 해결 하기 위해 경로 분할 라우팅 방법을 제안 한다. 2장에서 경로 분할 라우팅의 개념과 구현 방법을 설명 하고 3장에서

IETF(Internet Engineering Task Force)에서 제안하고 있는 Flow State 방법이 DSR 확장성 문제의 완전한 해결책이 될 수 없는 이유를 설명한다. 그리고 4장에서는 결론을 소개한다.

## 2. 경로 분할 라우팅

1967년 Stanley Milgram은 "small world experiment"를 통해 무작위의 두 미국인이 평균 6명의 아는 사람을 통해 연결되어 있다는 "six degrees of separation" 현상을 소개하였다. 경로 분할 라우팅은 대규모 MANET에 존재하는 source-destination간 경로를 여러 개의 작은 경로로 분할한다. 이때 분할된 작은 경로의 크기는 "six degrees of separation" 개념에 기초해 6-hop으로 하였다. 경로 분할 라우팅에서 모든 호스트는 라우트 캐쉬에 source-destination pair당 최대 12-hop 길이의 경로를 갖는다. 12-hop은 upstream 6-hop과 downstream 6-hop으로 구성된다.

### 2.1 경로 분할 라우팅에서 Route Discovery

소스 호스트가 목적지 호스트에 대한 경로 탐색 시 DSR은 서론에서 언급한 것처럼 ROUTE\_REQ를 브로드캐스팅한다. 이를 수신한 중간 호스트는 라우트 캐쉬에서 목적지 호스트에 대한 경로를 검색한다. 검색 결과 라우트 캐쉬에서 경로를 찾지 못하면 DSR은 ROUTE\_REQ 헤더에 단순히 중간 호스트의 주소를 추가 하지만 본 논문에서 제안하는 경로 분할 라우팅은 그림 1의 방법으로 추가해 헤더의 route record 크기가 최대 6-hop을 넘지 않게 한다.

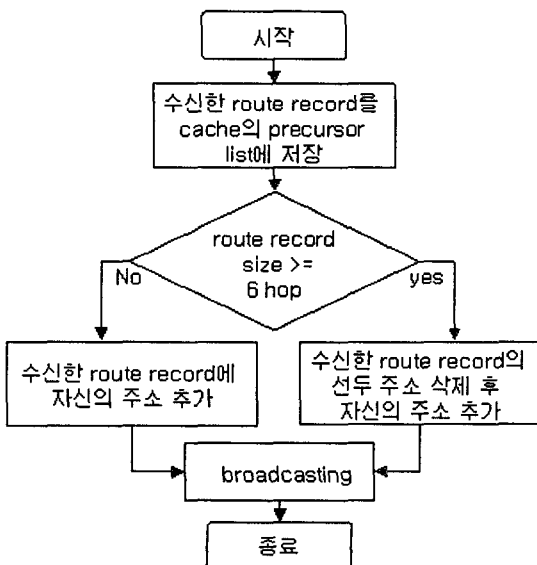


그림 1 중간 호스트의 ROUTE\_REQ 처리 방법

최종적으로 소스 호스트가 전송한 ROUTE\_REQ가 목적지 호스트에 도달 하면 목적지 호스트는 수신한 ROUTE\_REQ 헤더에 포함 되어 있는 최대 6-hop (그림 1 참조) route record를 자신의 라우트 캐쉬의 precursor list에 저장한 후 그 역순으로 ROUTE\_REPLY를 유니캐스팅으로 전송한다. ROUTE\_REPLY를 수신하는 중간 호스트는 그림 2의 방법으로 동작한다.

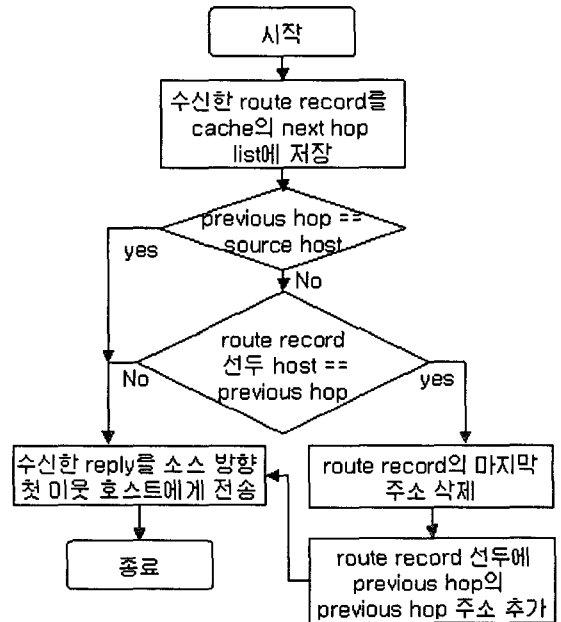


그림 2 중간 호스트의 ROUTE\_REPLY 처리 방법

그림 1 과 그림 2의 방법으로 모든 호스트들은 경로의 크기에 관계 없이 source-destination pair당 라우트 캐쉬 엔트리의 크기가 최대 12-hop을 넘지 않는다. 한편 그림 1의 방법으로 ROUTE\_REQ 헤더의 route record 크기가 6-hop을 넘지 않으며 그림 2의 방법으로 ROUTE\_REPLY의 route record 크기 역시 6-hop을 넘지 않는다.

### 2.2 경로 분할 라우팅에서 Data Forwarding

DSR에서 소스 호스트가 데이터 패킷 전송 시 헤더에 전체 경로를 추가 하지만 본 논문에서는 라우트 캐쉬에 저장 되어 있는 최대 6-hop의 next hop list만 추가 한다. 따라서 데이터 패킷을 중계하는 중간 호스트들은 경우에 따라 데이터 패킷 헤더의 route record를 갱신 해 주어야 한다. 먼저 data forwarding을 수행 하는 중간 호스트의 next hop이 목적지인 경우와 목적지가 아니고 자신이 route record의 마지막 호스트도 아닌 경우 route record 갱신 없이 경로상의 next hop에게 전달 한다. 반면 next

hop이 목적지가 아니고 자신이 route record의 마지막 호스트인 경우는 route record를 자신의 라우트 캐쉬에 저장 되어 있는 next hop list로 갱신 해서 수신한 데이터 패킷을 중계 한다. 그림 3에 중간 호스트의 data forwarding 방법을 정리 하였다.

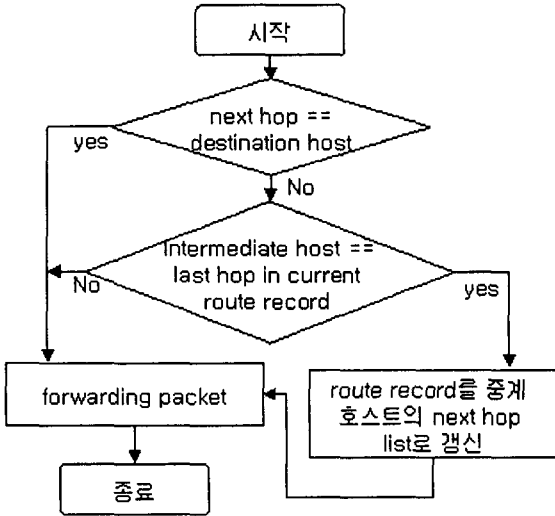


그림 3 중간 호스트의 Data Forwarding 방법

Data forwarding을 위한 중간 호스트의 그림 3과 같은 처리 절차는 본 논문에서 제안 하는 경로 분할 라우팅이 데이터 패킷에 전체 경로를 대신 분할 된 경로만 포함해 전송 하기 때문이다. 중간 호스트들은 데이터 패킷을 헤더에 포함 되어 있는 분할 경로에서 그 다음 분할 경로로 중계 해야 하는지를 판단 해서 필요 시 헤더의 route record를 갱신 한다.

### 3. DSR의 Flow State

“ Flow state” 는 DSR의 라우팅 동작에 관련 한 오버헤드 감소를 위해 패킷 헤더에 route record를 포함 하지 않고 data forwarding을 가능케 하는 방법 이다[5]. Flow state 방법은 “ route discovery” 후 라우트 캐쉬에 저장 되어 있는 경로에 flow id를 할당 하기 위해 데이터 패킷에 route record를 표현 하는 source route header와 flow state header를 함께 포함해 전송 한다. 이를 수신 하는 중간 호스트들은 flow state table을 생성한다. source-destination간 모든 중간 호스트들이 flow state table을 생성 하게 되면 flow state가 “ established end-to-end” 상태에 있다고 말한다. 이 때부터 중간 호스트들은 소스 호스트가 전송 하는 flow id를 라우트 캐쉬 엔트리의 식별자로 사용 해서 라우팅 정보를 얻을 수 있다.

Flow state를 사용 하기 위해서는 위에 설명 한 것처럼 전체 경로 상의 모든 호스트들이 flow state table을 생성 해야 하는 오버헤드가 있다. 그리고 여전히 route discovery에 source routing을 그대로 사용 하고 라우트 캐쉬에 전체 경로를 저장 하기 때문에 flow state는 단지 data forwarding에 관련된 오버헤드만 감소 시킨다.

### 4. 결론

본 논문에서 제안한 경로 분할 라우팅은 source-destination간 경로 크기 증가와 선형적 관계를 갖고 있는 DSR의 패킷 헤더 크기를 최대 6-hop으로, 라우트 캐쉬를 최대 12-hop 크기의 부하로 한정 한다. 따라서 MANET의 크기 증가에 따른 경로 길이 증가에 따라 라우팅 부하가 같이 증가 하지 않게 되어 대규모 MANET에 대한 DSR의 확장성이 확보 된다.

본 논문은 스케일러블 DSR 구현을 위해 경로 크기에 비례한 route discovery, data forwarding 및 라우트 캐쉬 부하 개선을 위해 노력 하였다. 향후 경로 분할 라우팅과 DSR의 route maintenance와의 상호 연동 관계에 대한 연구가 추가 진행 되어야 할 과제 이다.

### 5. 참고 문헌

- [1] IETF MANET Working Group, <http://www.ietf.org/html.charters/manet-charter.html>
- [2] C.-K. Toh, AD Hoc Mobile WIRELESS NETWORKS : PROTOCOLS AND SYSTEMS, Prentice Hall PTR, 27-28 ,2002
- [3] E. M. Royer and C.-K. Toh, A Review of current routing protocols for ad hoc mobile wireless networks, IEEE Personal Communications, Apr., 46-55, 1999
- [4] David B. Johnson and David A. Maltz, Mobile Computing, Kluwer Academic Publisher, 1996
- [5] IETF MANET Working Group, <http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt>