

# 이동 애드혹 네트워크에서 DSR 프로토콜을 위한 경로 축소 방법

## (A Route Shortening Mechanism for DSR protocol in Mobile Ad-Hoc Networks)

하 은 용<sup>†</sup>  
(Eun Yong Ha)

**요 약** 이동 무선 애드-혹 네트워크에서 이동 노드들은 일반적인 호스트 기능뿐만 아니라 이동 노드의 위치 변화에 따른 네트워크 토폴로지가 변화하는 상태에서 데이터 패킷 전달의 역할도 수행하는 라우팅 기능을 수행한다. 이런 애드-혹 네트워크를 위한 라우팅 프로토콜이 여러 가지 있다. 본 논문에서는 여러 제안된 프로토콜 중에서 네트워크에서 라우팅으로 인한 프로토콜 오버헤드가 작은 요구/응답 방식으로 동작하는 DSR 프로토콜을 대상으로 성능을 향상시키는 새로운 경로 축소 방법을 제안했다. 기존 경로 축소 방법은 소스 경로 상에 있는 노드만이 경로 축소에 참여하는데 반해, 본 논문에서 소스 경로 밖에 있는 인접 노드들도 청취한 패킷으로부터 수집한 정보를 경로 캐쉬로 관리해서, 저장된 캐쉬 경로 정보를 토대로 진행 중인 연결에 대해 더 짧은 경로를 발견한 경우에 기존 경로 대신 새 경로를 사용하도록 경로 축소 작업을 수행하는 방법을 제시했다. ns2를 이용한 시뮬레이션 결과, 제안한 방법이 이동 노드의 위치 변화가 활발한 경우에 기존 방법보다 경로 축소 횟수가 많이 발생했고, 또한 토폴로지의 변화에 대해 신속한 경로 축소를 수행해서 패킷 전송 성공률도 높아졌음을 확인했다. 즉, 약간의 프로토콜 오버헤드를 치루고 노드의 이동성에 따라 두 배에서 다섯 배 정도의 경로 축소로 신속한 패킷 전송과 전체적인 노드의 에너지 소모를 줄일 수 있음을 확인하였다.

**키워드** : 애드혹 네트워크, DSR 프로토콜, 경로 축소

**Abstract** Mobile nodes in ad-hoc wireless networks play roles of router as well as host. Movement of nodes causes network topology changes, which make existing routing information be modified. Therefore many routing protocols for ad-hoc networks were suggested in the literature. In this paper, we suggest an enhanced automatic route shortening method for dynamic source routing(DSR) protocol. DSR is a request/response based protocol which has low routing overhead owing to node movement. The current automatic route shortening is performed on the only nodes which belong to the source route of packets. On the contrary, our suggested method allows all neighbor nodes hearing the packet to participate in automatic route shortening. It makes all possible route shortenings be performed. So we maintain maximal shortened routes of ongoing data connections. Simulation results with ns2 show that our method pays small extra protocol overhead for ARS, but increases the ratio of successful packet transmissions and the number of ARSs performed in our mechanism is from 2 to 5 times higher than in original ARS mechanism and therefore it will improve the network-wide energy consumption in wireless ad-hoc networks.

**Key words** : Ad-Hoc network, DSR(dynamic source routing), ARS(automatic route shortening)

· 본 연구는 한국과학재단 목적기초연구(R05-2003-000-12092-0)지원으로 수행되었음

· 본 연구는 안양대학교 안식년 기간중 연구되었음

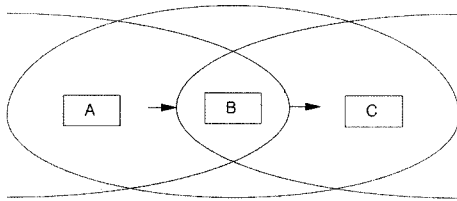
† 종신회원 : 안양대학교 컴퓨터공학과 교수  
eyha@anyang.ac.kr  
논문접수 : 2007년 8월 2일  
심사완료 : 2007년 9월 18일

: 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.  
정보과학회논문지: 정보통신 제34권 제6호(2007.12)  
Copyright©2007 한국정보과학회

1. 서론

에드혹 네트워크란 기존 네트워크 인프라 구조(infra-structure)나 중앙 집중적인 관리를 사용하지 않고, 무선 이동 노드들이 동적으로 임시적인 네트워크를 구성하는 것을 의미한다. 에드혹 네트워크는 유선 인프라 구조를 구성할 필요가 없기 때문에 저비용으로 망을 구성할 수 있어서 응급 상황, 전시 상황 등과 같은 유선망 구성이 힘든 상황에서 적용될 수 있는 기술이다.

하지만 각 이동 노드들이 호스트로서의 역할뿐 아니라 무선망 인터페이스의 전파 전송의 제약으로 인해 직접 연결이 어려운 노드들 간의 통신을 지원하기 위해서 다른 이동 노드들에게 패킷을 전달하는 라우터로서의 기능도 수행해야 한다. 에드혹 네트워크에 참여하는 노드들은 네트워크 상에 임의의 노드들 간에 경로를 결정하는 라우팅 프로토콜을 지원해야한다. 즉, 소스 노드가 목적 노드에게 직접 패킷을 전송할 수 없는 경우에는 그림 1과 같이 소스 노드는 인접 노드에게 패킷을 전송하고, 다시 중간 노드들은 목적 노드까지 패킷을 전달하는 방식으로 동작한다.



source A -> intermediate node B -> destination C

그림 1 에드혹 네트워크에서 패킷 전달

지금까지 문헌상에 제안된 프로토콜들은 네트워크 노드들 간에 주기적으로 토폴로지에 대한 정보를 주고받아서 네트워크 토폴로지에 대한 정보를 토대로 라우팅을 수행하는 proactive 프로토콜과 필요할 때에만 request/reply 방식으로 라우팅을 수행하는 reactive 프로토콜로 분류할 수 있다.

proactive 방식은 실제 데이터 전송과 관계되지 않은 경우에도 별도로 주기적으로 네트워크 토폴로지 정보를 상호 교환하는데 따른 오버헤드가 큰 반면에 경로 설정할 때 신속하게 할 수 있다는 장점이 있다. reactive 방식은 다소 경로 설정을 하는데 시간이 걸리지만 꼭 필요한 정보 교환을 하기 때문에 네트워크 자원을 덜 소비한다는 장점이 있다. proactive 방식의 프로토콜에는 OLSR(Optimized Link State Routing Protocol)[1]과 WRP(Wireless Routing Protocol)[2]이 있고, reactive 방식의 프로토콜에는 DSR(Dynamic Source Routing)[3],

AODV(Ad hoc On Demand Distance Vector)[4], TORA(Temporally Ordered Routing Algorithm)[5] 등이 있고, 두 방식을 혼합한 프로토콜에는 ZRP(Zone Routing Protocol)[6]이 있다.

몇 가지 대표적인 프로토콜에 대해서 간단히 살펴보면 다음과 같다.

• DSR(Dynamic Source Routing)[3]

이 방법은 각 패킷의 헤더에 패킷이 통과할 모든 노드들의 리스트가 포함된 형태의 소스 라우팅 방법으로 중간 노드들이 패킷 전달을 위해 최신 경로 정보를 유지할 필요가 없는 장점을 갖는다. 또한 주기적인 경로 광고 패킷이나 인접 노드 검사 패킷을 보낼 필요가 없다. 목적지까지의 경로 결정은 경로 발견 단계(route discovery phase)와 경로 유지 관리 단계(route maintenance phase)로 진행되는데, 먼저 소스 노드가 경로 요구(Route Request) 패킷을 발송(flooding)하면 목적지까지의 경로를 알고 있는 노드(목적지 노드 포함)가 경로 응답(Route Reply) 패킷으로 응답을 해서 경로를 찾아내고, 만약 네트워크 토폴로지가 변해서 목적지까지 경로를 찾을 수 없는 경우, 즉 경로 탐색 타이머가 만료되면 소스 노드가 자신의 캐쉬에 있는 다른 경로로 시도하든지 다시 경로 발견 단계를 수행해서 경로를 찾아낸다.

• AODV(Ad-hoc On Demand Distance Vector)[4]

각 노드가 목적지에 대한 다음 홉(next-hop) 라우팅 테이블을 유지하고, 이 라우팅 엔트리는 일정 시간 동안 사용되지 않으면 삭제된다. 경로 설정은 소스 노드에 라우팅 테이블에 목적지에 대한 정보가 있을 경우에는 바로 데이터 전송이 시작되고, 없는 경우에는 ERS (expanding ring search) 방법을 사용해서 목적지 노드에 대한 정보를 얻을 때까지 경로 요구(RREQ) 패킷을 전달한다. 목적지 노드에 대한 정보를 갖고 있는 노드가 경로 응답(RREP) 패킷을 소스 노드에게 전달해서 경로를 설정한다. 활동 중인 경로 상에 중간 노드에서 링크 에러를 발견하면 경로 에러(RERR) 패킷을 상위 경로로 보내서 상위 노드에서의 라우팅 정보 테이블을 갱신할 수 있도록 하고, 소스 노드에서는 경로 설정이 필요한 경우에 다시 할 수 있도록 한다.

이동 무선 에드혹 네트워크를 위한 라우팅 프로토콜은 망 토폴로지가 동적으로 빈번히 변하고, 무선 링크의 낮은 대역폭과 높은 전송오류, 휴대용 배터리 전력의 제한 등을 고려해서 라우팅 루프 발생을 없애고, 라우팅 정보 교환 오버헤드를 최소화할 수 있도록 설계해야 한다.

특히 이동 에드-혹 네트워크에서는 노드의 이동으로 인해 소스와 목적지 간의 경로가 변하기 때문에 경로를 유지 관리하는 것이 중요한 프로토콜 기능 중에 하나가

되었다. 이와 관련하여 경로 관리를 효과적으로 하기 위해 노드에 경로 캐쉬를 유지하고, 소스와 목적지 간의 경로의 길이를 가능한 짧게 유지함으로써 전체 네트워크의 패킷 전송량을 줄이려는 연구를 하고 있다.

본 논문에서는 DSR 프로토콜에서 경로 축소 방법을 개선하는데 목표를 두고 있다. 기존 DSR 프로토콜에서 정의된 경로 축소는 연결 설정 초기에 결정된 소스 경로(source route)의 길이를 줄이기 위해서 소스 경로 상에 노드들만이 참여한다. 즉, 경로 상의 노드는 수신한 패킷에 포함된 소스 경로를 확인해서 자신이 패킷을 처리할 노드가 아니라 그 이후의 노드라고 판단하면 경로 축소가 가능함을 판단해서 현재 패킷을 전송한 노드에서 이후의 패킷들은 바로 자신에게 전송하도록 경로를 조정한다. 하지만 경로 축소는 소스 경로 밖에 존재하는 노드에서도 가능한 경우가 존재한다.

따라서, 본 논문에서 제안하는 방법은 소스 경로 밖에 있는 이동 노드들도 경로 축소 작업에 참여하도록 한다. 외부에 있는 노드들이 패킷을 수신한 경우 패킷에 포함된 소스 경로를 분석하고, 자신의 경로 캐쉬에 존재하는 정보를 사용해서 자신을 통해서 목적지까지 가는 경로가 현재 소스 경로 보다 짧은 경우에 경로 축소 작업을 수행한다. 즉, 제안하는 방법은 애드혹 네트워크를 구성하는 모든 이동 노드들이 경로 축소의 가능성이 있을 때 모두 참여해서 경로 길이를 최대한 줄이는 작업을 수행한다. 제안한 방법을 ns2를 이용해서 시뮬레이션한 결과 경로 축소 횟수가 이동 노드의 이동성에 따라 두 배 혹은 다섯 배 정도 향상되었음을 확인했고, 노드의 이동을 고려해서 신속하게 경로 축소를 함으로써 패킷에 대한 성공적인 전송률도 향상되었음을 확인했다. 따라서 제안한 경로 축소는 결국 약간의 프로토콜 오버헤드를 감수하고 데이터 연결에 대해 최대한 짧은 경로를 유지함으로써 네트워크 전체적인 데이터 전송량을 줄이고 또한 이로 인한 전체 노드의 전원을 절감할 수 있는 효과를 볼 수 있음을 확인하였다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 DSR 프로토콜에 대해서 설명하고, 3장에서는 기존 경로 축소 방법에 대해 분석하고, 4장에서는 새로운 자동 경로 축소 방법의 개념에 대해 설명하고, 5장에서는 새로운 경로 축소 방법의 설계에 대해서 설명하고, 6장에서는 시뮬레이션을 통해 제안한 경로 축소 방법의 성능을 평가하고, 마지막으로 결론을 맺는다.

## 2. 기존 DSR 프로토콜 분석

DSR[3] 프로토콜은 애드-혹 네트워크를 위한 라우팅 프로토콜로서 크게 두 부분으로 구성된다. 첫째, 소스 노드가 목적지 노드까지의 경로를 찾기 위한 경로 발견

단계(route discovery phase)가 있고, 둘째, 이동 노드들의 빈번한 이동으로 인한 환경 변화에 따른 경로 유지 단계(route maintenance phase)가 있다. 특히 경로 유지 단계는 링크 에러 또는 기존 경로의 길이 보다 더 좋은 경로가 존재할 경우에 새로운 경로로 대체하는 즉 경로를 최적화 하는 기능을 수행한다.

### 2.1 경로 발견(route discovery)

DSR 프로토콜은 노드 간에 전달할 메시지가 있는 경우에만 경로를 설정한다. 경로 발견 과정은 그림 2에서 처럼 소스 노드(A)는 목적지 노드(E)에 대한 경로를 발견하기 위한 요구 메시지인 ROUTE REQUEST(RREQ) 메시지를 인접 노드에게 방송한다.

RREQ를 수신한 인접 노드는 두 가지 분류된다. 자신이 목적지 노드가 아닌 경우에는, 여기서는 B, C, D 노드, RREQ 메시지에 포함된 지금까지 발견된 source route (SR)에 자기 노드 식별자를 추가해서 인접 노드에게 RREQ 메시지를 전달한다. 예를 들면, 노드 C의 경우 현재까지 발견된 경로 [A,B]에 자신의 식별자를 추가한 [A, B, C]를 포함한 RREQ를 인접 노드에 전달한다. 이 과정은 목적지 노드에 RREQ가 도착할 때까지 반복된다.

목적지 노드 경우에는, 여기서 노드 E, 발견된 경로 [A, B, C, D, E]를 포함하는 ROUTE REPLY(RREP) 메시지를 생성해서 소스 노드에게 응답한다. RREP 메시지를 소스 노드에게 전달하는 과정은 네트워크의 링크가 양방향 링크냐 단방향 링크냐에 따라 다르게 동작한다.

양방향 링크의 경우는 발견 경로의 역경로를 추적해서 소스 노드에게 전달되고, 단방향 링크의 경우에는 발견된 경로를 포함한 RREP 메시지를 RREQ 메시지에 piggybacking 해서 다시 목적지에서 소스까지의 경로를 발견하는 과정을 수행한다.

특히, 경로 발견 과정을 보다 효율적으로 하기 위해서 이동 노드들은 경로 발견 과정에서 수신한 RREQ와 RREP 메시지에 포함된 source route 정보를 자신의 경로 캐쉬(route cache)에 저장해둔다. 이 경로 캐쉬 정보는 자신이 RREQ에 대한 목적지 노드가 아니라도 자신

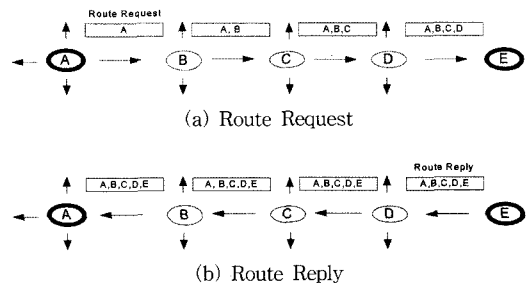


그림 2 경로 발견 과정

의 캐쉬에 목적지 노드까지의 경로가 존재할 경우 이 경로를 참조해서 바로 소스 노드에게 RREP 메시지를 전달해서 경로 발견 과정을 신속히 처리할 수 있도록 한다. 캐쉬 정보의 유효성 관리는 별도의 연구 대상이 될 수 있다.

**2.2 경로 유지(route maintenance)**

경로 발견 과정에서 결정된 source route(SR)를 이용해서 패킷을 전송 또는 전달하는 과정에서 각 이동 노드는 SR 상의 다음 노드로의 패킷 전달에 대한 책임을 갖는다. 따라서 링크의 상태를 점검해서 링크의 에러나 이동 노드의 위치 변화로 인해 더 이상 기존 SR을 유지할 수 없는 경우에 ROUTE ERROR 메시지를 패킷을 전송한 노드로 전달해서 안전한 경로를 유지할 수 있도록 하는 작업이 경로 유지 단계이다.

그림 3에서처럼 노드 C가 데이터 패킷을 받고 노드 D로 전달하는 과정에서 <C-D> 간 링크의 에러로 더 이상 전달이 불가능할 때 노드 C는 ROUTE ERROR 메시지를 만들어서 SR의 소스 노드인 노드 A에게 전달해서 새로운 경로를 설정하도록 한다.

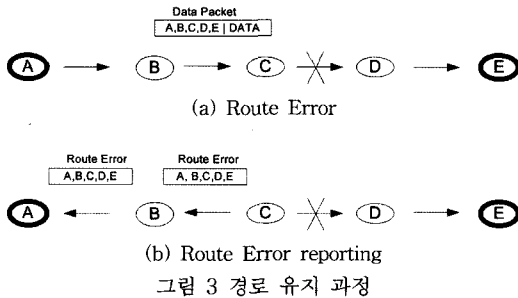


그림 3 경로 유지 과정

**3. 기존의 DSR 프로토콜의 경로 축소 방법**

DSR 프로토콜은 이동 무선 애드-후 네트워크에서 동작하는 라우팅 프로토콜으로써 특별히 각 이동 노드들이 promiscuous mode 로 동작해서 인접 노드들이 방송된 모든 패킷들을 청취할 수 있는 기능을 제공한다. 이것은 이동 노드들이 빈번히 위치 변화를 하는 환경에서 경로 발견 시 각 노드에 저장한 경로 캐쉬 정보를 이용해서 경로 발견을 신속히 처리할 수 있는 기회를 제공할 뿐만 아니라 경로의 길이가 긴 기존 경로를 보다 짧은 경로로 최적화할 수 있는 단서를 제공한다는 점에서 중요한 의미를 갖는다.

DSR에서는 모든 이동 노드들이 지나가는 모든 패킷들을 청취해서 경로에 대한 정보를 습득한다. 이 캐쉬된 경로 정보를 이용해서 기존 경로를 자동으로 축소시키는 기능을 자동 경로 축소(automatic route shorting: ARS)라 한다.

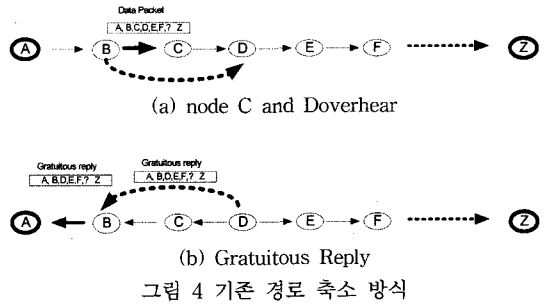


그림 4 기존 경로 축소 방식

그림 4에서 노드 B가 다음 노드 C에게 전달한 패킷 속에는 소스 경로(source route:SR) 정보가 포함되어 있다. 여기서 SR 정보는 [A, B, C, D, E, F, ..., Z]이다. 그런데 노드들의 이동으로 그림 4처럼 SR 상에 있는 노드 C와 노드 D가 동시에 노드 B가 전달한 패킷을 수신한 경우 노드 D의 경우는 SR 상에서는 자신이 노드 B의 인접이 아니지만 결국 노드 C를 통해 조금 후에 전달된 동일한 패킷을 받게 될 것이라고 알 수 있다.

따라서 노드 D는 기존 SR에서 B->C->D로 전달되는 경로를 B->D로 대체해서 경로를 짧게 할 수 있다고 판단을 할 수 있다. 이때 노드 D는 소스 노드 A에게 gratuitous reply 메시지를 보내서 이후의 패킷부터는 짧아진 경로로 보내도록 조치를 취한다. 즉, 새로운 SR 정보는 [A, B, D, E, F, ..., Z]가 된다. 이 과정이 기존 경로 축소 방식이다.

**4. 새로운 경로 축소 방법**

기존 경로 축소 방식은 패킷에 포함된 SR 경로 상에 있는 노드들에 국한해서 경로 축소 작업을 수행한다. 하지만 경로 밖에 있는 노드 역시 전달되는 패킷들을 청취할 수 있고 청취한 패킷에서 경로 정보를 획득해서 이를 토대로 자신의 경로 캐쉬(route cache)를 유지하는데, 이 캐쉬 정보 속에는 기존 경로 상의 노드가 수행한 경로 축소 결과 만들어진 경로보다도 훨씬 짧은 경로를 가지고 있을 수 있다.

본 논문에서는 경로 외부에 있는 노드들도 경로 축소에 참여할 수 있는 새로운 경로 축소 방법을 제안한다.

그림 5에서 노드 B가 전달한 패킷 속에 포함된 SR 정보는 [A, B, C, D, E, F, ..., Z]이다. 이 패킷은 소스 경로 상에 있는 노드 C와 D 뿐만 아니라 소스 경로 밖에 있는 노드 G도 수신한다. 그런데 노드 G의 경우 자신의 경로 캐쉬에 저장된 경로 정보를 검색해 보니 그림 5에서처럼 노드 F로 점프하는 경로가 있음을 확인했다. 이렇게 발견된 경로는 기존 경로보다 짧은 경로이다. 따라서 노드 G는 새로운 경로축소 처리 결과인 짧은 경로 [A, B, G, F, ..., Z]를 gratuitous reply에 포함

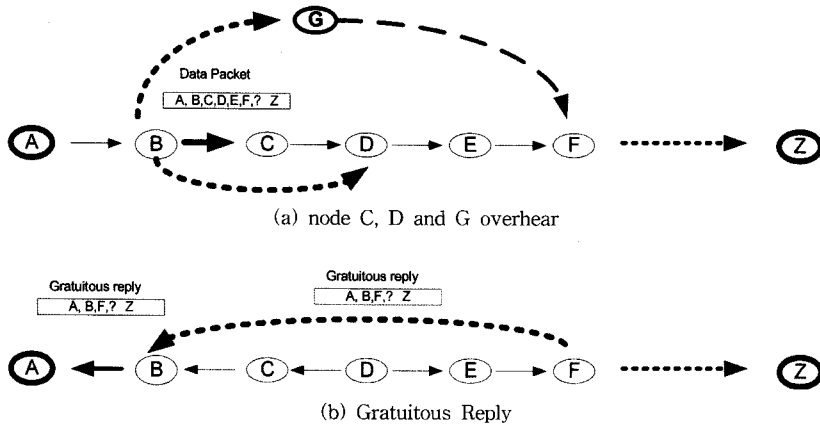


그림 5 제한한 경로 축소 방식

해서 소스 노드 A에게 전달한다. 이제 소스 노드 A는 더 짧아진 경로를 통해 이후의 패킷들을 목적지 노드 Z에게 전송한다.

### 5. 경로 축소 방법 설계

새로운 경로 축소는 노드의 특성에 따라 다음과 같이 세 가지로 분류한다. 그림 5에서 노드 B가 패킷을 전달할 때, 패킷을 수신한 노드 C, 노드 D와 노드 G에 따라 제한한 경로 축소를 실행하는 형태가 세 가지로 분류된다.

- 경로 축소 작업이 필요 없는 노드

DSR 프로토콜에서 노드를 식별하기 위해 net\_id 또는 MAC\_id 를 사용한다. 식별 정보는 소스 경로 패킷(SRPacket)의 소스 경로(source route) 필드에 포함되어서 전달된다. 따라서 각 노드는 수신한 SR패킷내 SR 필드 값의 현재 식별자와 자신의 식별자를 비교해서 자신이 SR패킷을 처리할 노드인지 아닌지를 판단한다.

그림 5에서 노드 C는 수신한 SR패킷의 현재 처리할 노드의 식별자와 자신 식별자가 같은 경우로 경로 축소 작업이 필요 없고, 수신한 SR패킷을 다음 노드로 전달하거나, 프로토콜 절차에 따라서 처리한다. 알고리즘 1은 처리과정을 설명하고 있다.

```

receive SRpacket;
if (current Node ID of SR field in SRpacket == My Node ID) {
    no ARS operation;
    if (intermediate node ) forward SRpacket;
    else (destination node ) process SRpacket according to rules;
}
    
```

알고리즘 1. 경로 축소를 하지 않는 노드에서 SR 패킷 처리

- 기존 경로 축소 작업을 하는 노드

그림 5에서 노드 D의 경우에는 수신한 SR패킷을 처리할 노드가 아님을 확인한다. 그리고 SR패킷 내에 소스 경로 필드에 자신의 식별자가 존재하는 지 확인하기 위해 소스 경로에서 현재 이후의 노드 리스트를 자신의 식별자와 비교한다. 자신이 경로 상의 노드임이 확인되면, 기존의 경로 축소 작업을 실행하고, 소스 노드에게 Gratuitous Reply 메시지를 보내서 추후의 SR패킷은 새롭게 축소된 경로로 전송하도록 조치를 취한다.

```

receive SRpacket;
if (current Node ID of SR field in SRpacket != My Node ID) {
    for (each Node ID of SR field after current Node ID)
        if (current Node ID == My Node ID) { // on the source route
            save the route shortening point;
            do the route shortening operation;
            send Gratuitous Reply to the source node;
            forward SRpacket;
        }
    next Node ID;
}
    
```

알고리즘 2. 기존의 경로 축소를 하는 노드에서 SR 패킷 처리

- 새로운 경로 축소 작업을 수행하는 노드

그림 5에서 노드 G와 같은 경우, 자신이 현재 promiscuous mode 동작 상태에서 수신한 SR패킷의 소스 경로를 검사해서 자신이 경로 밖에 노드인지를 확인한 후, 새로운 경로 축소 작업을 적용한다.

SR패킷을 수신한 노드는 저장된 경로 캐쉬(route

cache)를 검색해서 SR패킷의 소스 경로 상에 현재 이후의 노드 중에서 경로 캐쉬에 저장된 경로를 통해서 갈 수 있는 경로가 있는지를 확인한다. 해당 경로 캐쉬를 찾은 경우에는 기존 소스 경로의 길이와 해당 경로를 경유해서 목적지 노드로 가는 경로의 길이를 비교한다. 비교 결과 새로 발견한 경로가 더 짧은 경우, 기존 소스 경로를 새로운 경로로 대체해서 경로 축소 작업을 수행한다. 그리고 소스 노드에게 Gratuitous Reply 메시지를 전달해서 이후의 패킷에 대해서 새로운 경로로 전달되도록 처리한다.

```

receive SRpacket;

if (My Node ID does not belong to the SR field in SRpacket)
// out of the source route
for (each Node ID from the end node of SR field to the current node) {
    find the route to the Node ID in my route cache;
    if (found) {
        make the new route to the destination node;
        compare the current route length to the new route length;
        if (new route is short) {
            save the route shortening point;
            do the route shortening operation;
            send Gratuitous Reply to the source node;
            forward SRpacket;
        }
        break;
    }
}
    
```

알고리즘 3. 새로운 경로 축소를 하는 노드에서 SR 패킷 처리

6. 시뮬레이션을 통한 성능 평가

새로 제안한 경로 축소 방법의 성능 평가를 위해 ns-2 시뮬레이터를 사용하였다. 시뮬레이션을 위한 기본 환경 변수 값은 표 1과 같다. 노드는 670m \* 670m 영역에서 이동하고, 30 개의 이동 노드가 존재한다. 이동 노드들은 평균 20m/sec의 속도로 이동하고, 평균 정지 시간은 100초에서 600초까지의 범위 값을 갖는다. 또한 데이터 트래픽은 CBR(constant bit rate) UDP 트래픽으로 최대 연결 수는 10개까지 가능하고, 패킷은 초당 4 개를 전송한다. MAC 계층은 IEEE 802.11를 사용하였고, 전체 시뮬레이션 타임은 1000 simulation units을 사용하였다.

경로 축소 작업을 실행한 횟수에 대한 시뮬레이션 결

표 1 시뮬레이션 환경 변수

시뮬레이션 환경 변수	변수값
노드 수	30
노드 최대 이동 속도	20 m/sec
노드 평균 정지 시간	100,200,300,400,500,600 sec
이동 영역	670 m × 670 m
최대 데이터 트래픽 CBR 연결 수	10
패킷 전송률	4 pakets/sec
MAC 계층	IEEE 802.11
CBR 데이터 패킷 크기	512 bytes
시뮬레이션 시간	1000 simulation units

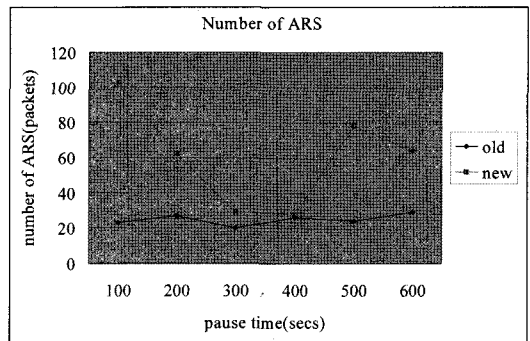


그림 6 경로 축소 실행 횟수

과는 그림 6에 나타나 있다. 그림 6에서 보는 것처럼 제안한 경로 축소 방법이 기존 방법에 비해서 이동 노드의 정지 시간에 대해 대부분의 경우에 더 많은 경로 축소를 실행했음을 알 수 있다.

노드의 위치 이동이 비교적 활발한 경우, 즉 정지시간이 100초 또는 200초 인 경우, 자동 경로 축소가 3배에서 5배 정도로 많이 실행되었음을 확인할 수 있다. 이 결과는 빠른 노드의 이동이 현재 데이터 연결의 목적지 노드까지의 소스 경로가 중간 링크의 직접 연결성 단절로 기존 소스 경로를 더 이상 사용할 수 없게 되었고, 이것은 결국 새롭게 경로를 설정해서 데이터 연결을 복원해야 하는 상황을 반영한 것으로 설명할 수 있다. 결과적으로 노드의 이동성이 높으면 전체 네트워크의 토폴로지의 변화가 심해서 경로를 축소할 연결이 많이 생긴다는 사실을 보여주고 있다.

경로의 축소는 결국 빠른 데이터 전송을 보장하는 것이고 이는 노드의 전원을 덜 사용하게 되어서 무선 애드혹 네트워크의 수명을 기존 방법보다도 더 연장할 수 있음을 보여 주는 결과로 해석할 수 있다.

정지시간이 600초인 경우에도 약 2배의 경로 축소가 수행되었다. 이것은 노드의 이동이 덜한 상황에서, 즉 네트워크 토폴로지의 변화가 적은 경우에도 경로 축소의 필요성이 있음을 보여주고, 안정된 토폴로지에서의

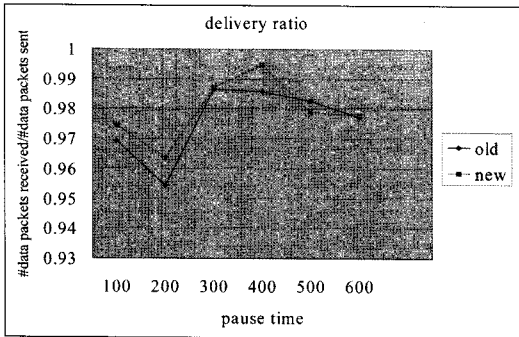


그림 7 패킷 전송 성공 비율

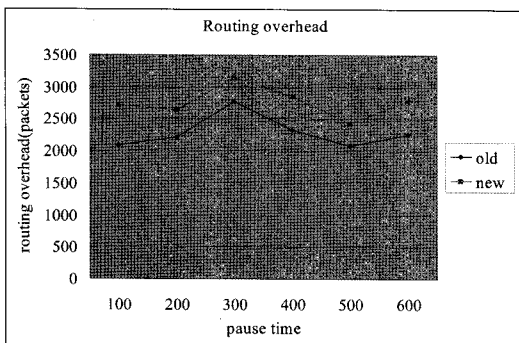


그림 8 프로토콜 오버헤드

경로 축소도 역시 네트워크의 수명을 연장하는 역할을 할 수 있음을 보여준다.

그림 7은 소스노드에서 전송한 패킷이 목적지 노드에 정확히 도착한 경우에 대한 패킷 전송 성공 비율로써, 패킷 수신비율 = 수신한 패킷수/전송한 패킷수로 계산된다. 결과에서 보면 제안한 새로운 방법이 기존 방법에 비해서 더 좋은 성능을 보이고 있음을 알 수 있다.

이것은 기존 경로 축소 방법을 사용해서 데이터 전송을 할 경우에 네트워크 토폴로지의 변화를 기존 방법이 즉각적으로 반영하지 못해서 현재 사용 중인 연결의 소스 경로가 유효하지 못해 결과적으로 패킷전송 성공 비율이 떨어짐을 보여주는 결과로 해석된다. 즉, 제안한 경로 축소 방법을 사용하는 경우에 네트워크 토폴로지의 변화에 따른 소스 경로에 대한 정확한 경로 단축이 신속한 패킷 전송을 돕기 때문에 패킷 전송 성공 비율도 높게 나타난 것으로 해석된다.

그림 8은 경로 설정 및 유지를 하는데 전송한 제어 메시지의 수를 표시하고 있다. 결과에서 보듯이 새로운 경로 축소 방법은 그림 6에서처럼 2배에서 5배 정도의 경로 축소를 실행하기 때문에 이로 인한 gratuitous-reply 메시지를 보내야 하는 오버헤드가 존재한다. 그런데 결과에서 보듯이 제어 메시지의 수는 0.5 배 정도 많

이 나왔을 뿐 많은 오버헤드를 발생시키지 않은 것을 볼 수 있다.

따라서 새로운 경로 축소 방법을 사용할 경우에 조금의 프로토콜 오버헤드 상승으로 패킷 성공 전송 비율 향상, 경로 축소 횟수 증가로 인한 경로 단축에 따른 네트워크의 수명 시간 연장 등의 여러 좋은 성능 향상을 얻을 수 있음을 보여주고 있다.

## 7. 결론

이동 무선 애드혹 네트워크에서 이동 노드들은 일반적인 호스트 기능뿐만 아니라 이동 노드의 위치 변화에 따른 네트워크 토폴로지가 변화는 상태에서 데이터 패킷 전달의 역할도 수행하는 라우팅 기능도 갖고 있어야 한다. 따라서 새로운 라우팅 프로토콜이 필요하다.

본 연구에서는 여러 제안된 프로토콜 중에서 네트워크에서 라우팅으로 인한 프로토콜 오버헤드가 작은 요구/응답 방식으로 동작하는 DSR 프로토콜을 대상으로 성능을 향상시키는 새로운 경로 축소 방법을 제안했다.

기존 경로 축소 방법은 소스 경로 상에 있는 노드만이 경로 축소에 참여하는데 반해, 제안한 새로운 경로 축소 방법은 소스 경로 밖에 있는 이동 노드들도 경로 축소 작업에 참여해서 가능한 경우에 모든 경로 축소를 수행해서 애드혹 네트워크 내에 데이터 연결에 대해 가능한 짧은 거리의 경로를 유지해서 데이터 전송을 신속하게 처리하도록 하였다.

시뮬레이션 결과를 볼 때, 새로운 경로 축소 방법은 약간의 부가적인 프로토콜 오버헤드를 치루고, 대신 패킷 성공 전송 비율을 기존 보다 많이 향상시켰고, 또한 경로 축소 횟수도 기존의 2-5배 정도 향상시켜 경로 단축에 따른 신속한 패킷 전송을 할 수 있는 등 여러 좋은 성능 향상을 얻을 수 있었다.

이동 무선 애드혹 네트워크는 가급적 오랫동안 지속되어야 한다. 따라서 휴대용 전원의 한계로 최대한 네트워크를 오래 지속하기 위해서는 효율적인 방법이 라우팅 프로토콜 레벨에서 개발되어야 한다. 추후 멀티캐스트 서비스를 적용한 환경에서 경로 축소 방법의 활용 면에 대해서 연구할 것이다.

## 참고 문헌

- [1] T. Clausen, P. Jacquet, and Project Hipercom, "Optimized Link State Routing Protocol," IETF MANET, RFC 3626, Oct. 2003.
- [2] S. Murthy and J.J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks," ACM Mobile Networks and Applications Journal, Special Issue on Routing in Mobile Communications Networks, Oct. 1996.

- [3] David B. Jhonson, David A. Maltz and Yih-Chun Hu. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks for IPv4, RFC 4728, Feb. 2007.
- [4] Charles E. Perkins, E. Belding-Royer and S. Das. Ad Hoc On Demand Distance Vector(AODV) routing. IETF MANET, RFC 3561, Jul, 2003.
- [5] V. Park and M.S. Corson, IETF MANET Internet Draft, "Draft-ietf-MANET-tora-spec-03.txt," Nov. 2000.
- [6] M.R. Pearlman and Z.J. Haas, "Determining the Optimal Configuration for the Zone Routing Protocol," IEEE Journal on Selected Areas in Communications, Special Issue on Wireless Ad Hoc Networks, Vol.17, No.8, pp. 1395-1414, Aug. 1999.
- [7] Network Simulator ns-2, <http://www.isi.edu/nsnam/ns/>



하 은 용

1986년 서울대학교 컴퓨터공학과(공학사)  
 1988년 서울대학교 대학원 컴퓨터공학과  
 (공학석사). 1997년 서울대학교 대학원 컴  
 퓨터공학과(공학박사). 1997년~현재 안양  
 대학교 컴퓨터공학과 교수. 관심분야는  
 USN/RFID, MANET, 임베디드시스템