Ad Hoc 망에서 신뢰성있는 데이터 전송을 위한 경로 재설정 기법

준회원 한 정 안*, 백 종 근**, 정회원 김 병 기***

A re-route method for reliable data transport in Ad Hoc Networks

Jung-Ahn Han*, Jong-Gun Paik** *Associate Members* Byung-Gi Kim*** *Regular Member*

요 약

Ad hoc 망은 기지국이나 엑세스 포인트(AP) 같은 인프라가 없는 무선이동통신기술이다. 즉, ad hoc 망에서는 이동노드들이 라우터와 서버의 역할을 모두 한다. 기존의 경로재설정 방법에서는 노드의 이동으로 인하여 경로가 끊어졌을 경우 송신노드는 경로를 다시 재설정해야한다. 하지만 경로가 끊어진 후 다시 재설정하는 방법은 QoS를 보장해야하는 실시간 데이터 전송에서는 적합하지 않을 수 있다. 그러므로 본 논문에서는 경로가 끊어지는 것을 미리 예방하는 경로재설정 기법을 제안한다. 중간노드가 이동하면서 어떤 임계영역으로 들어가는 경우 자신의 다음 노드에게 핸드오프 패킷을 발생시키고, 핸드오프 패킷을 수신한 노드는 경로가 끊어지기 전에 경로요청패킷을 중간노드의 바로 이전노드에게 브로트캐스트함으로써 중간 노드가 경로를 벗어나더라도 지속적으로 새로운 경로를 통하여 데이터 패킷을 전송할 수 있다.

Key Words : Wireless Ad-hoc networks, Routing

ABSTRACT

An ad hoc network is infra(Base Sstation or Access Point) free wireless mobile communication technology. Mobile nodes function as routers and servers in ad hoc networks. Many routing protocols for ad hoc network have been proposed. If any route is broken owing to moving node, source must repair broken route again. But route repair technology after route collapses is not suitable to transmit real-time data packet for QoS guarantee. So this paper presents route repair technology that prevents route from breaking. If intermediate node moves to critical section, the node issues handoff packet and sends the packet to the next node. After next node receives handoff packet, the node broadcasts route request packet to the previous node for intermediate node. Finally, even if intermediate node moves out of the routing region, the source can continuously transmit data packets to the destination through the new path.

Ⅰ. 서 론

지금까지 인터넷 사용자들은 랜(LAN)이나 케이블 모뎀 등의 유선을 통하여 제한된 시·공간에서 통신을 하였다. 그러나 인터넷이 빠르게 발전하면서 사용자들의 욕구도 높아져 언제, 어디서나 통신을

* 숭실대학교 대학원 컴퓨터학과(dawndew@archi.ssu.ac.kr) ** ㈜베타리서치&컨설팅 모바일 리서치팀 (pkj4338@veta.co.kr)
*** 숭실대학교 컴퓨터학부(bjkim@comp.ssu.ac.kr)
논문번호 : #030356-0814, 접수일자 : 2003년 8월 29일
※ 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음

할 수 있기를 원하게 되었다. 이러한 사용자들의 욕구를 충족시키기 위해 이동단말기의 하드웨어나 무선 이동 통신 기술개발 등이 빠른 속도로 이루어지고 있다.

이러한 추세를 가장 잘 반영하는 무선 이동 통신 기술로 ad hoc 망을 예로 든다. Ad hoc 망 기술은 기존의 기지국(Base Station:BS)이나 또는 엑세스포인트(Access Point:AP)가 유선 통신망에 연결된 인프라(infrastructure)기반 망과는 달리 고정된 라우터(router)가 없다. 즉, 모든 노드(node)들이 이동할 수 있다. 모든 노드들이 이동하는 환경에서, 서로 직접적인 무선 전송 범위에 위치하지 않은 노드간의 원활한 데이터 전송을 위해 다중 홉 무선 링크(multi hop wireless link)로 구성하고, 여러 개의 중간 노드들에 의해 데이터를 포워딩/라우팅(forwarding/routing) 하는 새로운 형태의 통신망이다. 즉, 신속한 통신망 구축이 가능하고, 기존의 통신 인프라에 의존하지 않으며, 단말기 이동에 빨리 적응할 수 있는 장점이 있다[1,2].

이러한 ad hoc 망은 라우팅 프로토콜(routing protocol)에 따라 table-driven 방식과 on-demand 방식으로 나눈다. table-driven 라우팅 프로토콜은 각 노드가 망 내의 다른 모든 노드와 라우팅 정보를 꾸준히 유지한다. 이러한 table-driven 방식은 라우팅 정보를 저장하기 위해 하나 또는 그 이상의 테이블(table)을 필요로 하고, 망의 정보를 브로드캐스트(broadcast)하여 모든 노드들이 항상 같은 망의 토플로지(network topology)에 대한 정보를 갖도록 테이블을 갱신한다.

On-demand 라우팅 프로토콜은 송신 노드의 요구가 있을 때에만 경로를 설정한다. 송신 노드가 목적지 노드까지의 경로가 필요하면 망 내에 있는 경로 탐색 프로세스(route discovery process)가 작동한다. 이 프로세스는 경로를 찾거나 또는 가능한 모든 경로의 검사가 끝나면 완료된다. 이 경로는 목적지 노드까지 접근이 불가능하거나 또는 경로가 더 이상 필요하지 않을 때까지 유지된다[2].

본 논문에서는 ad hoc 망에서 QoS 보장을 위한 경로 재설정 기법을 제안한다. 이 라우팅 기법은 ad hoc 무선망의 라우팅 프로토콜 방식 중 on-demand 방식을 사용한다. 이 라우팅 기법은 기존의 방법들이 경로가 끊어졌을 경우 다시 경로를 재설정 하는 것과는 다르게 핸드오프 기법을 사용하여 설정된 경로가 끊어지는 것을 미리 예방한다. 그리고 경로를 부분적으로 빠르게 재설정 하기위해 지역적으로 재설정하는 기법을 사용한다.

본 논문은 ad hoc 무선망에서 QoS 보장을 위한 경로 재설정 기법에 관한 연구로서 총 6장으로 구성한다. 2장에서는 연구배경에 대해 알아보고, 3장에서는 ad hoc 무선망이 라우팅 프로토콜에 따라 어떻게 분류하는지 살펴본다. 4장에서는 on-demand 라우팅 방식에서 지금까지 제안된 여러 경로 재설정 알고리즘에 대해 살펴보고 그것들의 특징과 문제점에 대해 알아본다. 5장에서는 본 논문에서 제안하는 경로 재설정 기법에 대하여 자세한 설명을 한다. 마지막으로 6장에서는 시뮬레이션을 통해 5장에서 제안한 경로 재설정 기법을 기존의 경로 재설정 알고리즘들과 비교하여 성능을 평가한다. 마지막으로 7장에서는 본 연구결과에 대한 전체적인 평가와 고찰로 결론을 맺는다.

Ⅱ. Ad hoc 라우팅 프로토콜 분류

현재의 ad hoc 라우팅 프로토콜은 일반적으로 table-driven방식과 on-demand방식으로 나눌 수 있다.

2.1 Table-driven 라우팅 프로토콜

Table-driven 라우팅 프로토콜은 각 노드가 망 내의 다른 모든 노드와 라우팅 정보를 꾸준히 유지한다. 이러한 table-driven 라우팅 프로토콜 방식은 라우팅 정보를 저장하기 위해 하나 또는 그 이상의 테이블(table)을 필요로 하고, 망의 정보를 브로드캐스트(broadcast)하여 모든 노드들이 항상 같은 망의 토플로지(network topology)에 대한 정보를 갖도록 테이블을 갱신한다[2].

다음은 대표적인 table-driven 방식의 라우팅 프로토콜이다.

2.1.1 DSDV(Destination-Sequence Distance-Vector) 라우팅 프로토콜

Table-driven 방식의 대표적인 라우팅 프로토콜로 DSDV(Destination-Sequence Distance-Vector) 라우팅 프로토콜이 있다[4,5]. 이 DSDV 라우팅 프로토콜은 Bellman-Ford 알고리즘[6]을 기반으로 만든 table-driven 라우팅 프로토콜이다. DSDV 라우팅 프로토콜에서 망 내의 모든 이동 노드들은 그들이 연결할 수 있는 다른 모든 노드들에 대해 라우팅 정보를 가진다. 각각의 라우팅 정보는 목적지 노드가 표시한 순차번호(sequence number)를 가진다. 순

차번호는 이동 노드들의 라우팅 루프(loop)를 방지하기 위해 사용한다. 라우팅 테이블에서 갱신한 내용은 일관성을 위해 정기적으로 망을 통해 브로드캐스트(broadcast)한다. 또한 이동 노드들은 경로에 대한 고정시간(settling time)을 기록한다. 고정시간의 길이를 이용하여 갱신된 라우팅 정보에 대한 브로드캐스트를 지연시켜, 이동 노드들은 망의 트래픽(traffic)을 감소시키고, 보다 최적의 경로를 찾은 후 브로드캐스트함으로써 경로를 최적화한다.

2.1.2 CGSR(Clusterhead Gateway Switch Routing) 프로토콜

Clusterhead gateway switch routing (CGSR) 프로토콜은 주소할당 방식과 망의 구성에서 DSDV 방식과 다르다. CGSR은 DSDV를 기본 골격으로 하며 DSDV와 같이 많은 오버헤드(overhead)가 있다. 그러나, 송신 노드와 목적지 노드 사이가 아닌, 계층적인 클러스터헤드(clusterhead)와 게이트웨이(gateway)간의 라우팅을 사용한다. 패킷은 먼저 클러스터헤드로부터 게이트웨이로 전송되고, 다시 클러스터헤드로, 그러한 방식을 반복하여 목적지로 전송한다. 각 노드는 클러스터 멤버 테이블(cluster member table)을 유지하여야 한다. 그리고 DSDV를 이용하여 클러스터 멤버 테이블을 정기적으로 이웃 노드에게 브로드캐스트한다. 각 노드들은 클러스터 멤버 테이블 외에, 목적지에 도달하기 위한 라우팅 테이블도 유지한다.

2.2 On-demand 라우팅 프로토콜

Table-driven 라우팅 프로토콜과 다른 접근 방식으로 on-demand 라우팅 프로토콜이 있다. 이러한 방식의 경우, 송신 노드의 요청에 따라 경로를 설정한다. 송신 노드가 목적지 노드로의 경로가 필요하다면, 망 내의 경로 탐색 프로세스가 작동한다. 이 프로세스는 경로가 탐색되거나, 또는 가능한 모든 경로에 대해 검사가 끝나면 완료된다. 이렇게 생긴 경로는 목적지 노드까지 접근이 불가능해지거나, 또는 경로가 더 이상 필요 없을 때까지 유지된다. 다음은 대표적인 on-demand 라우팅 프로토콜들이다.

2.2.1 DSR(Dynamic Source Routing) 프로토콜

DSR 프로토콜[7]은 송신 노드의 라우팅에 기반을 둔 on-demand 라우팅 프로토콜이다. 이동 노드들은 라우팅 테이블과 같은 역할을 하는 라우트 캐쉬(cache)를 계속 유지한다. 이 라우트 캐쉬는 송신 노드까지의 경로를 포함하고 있고, 새 경로가 입력될 때마다 지속적으로 갱신한다. 이 프로토콜은 경로를 획득하고 관리하는 경로 탐색과 경로 관리에 관한 부분으로 이루어진다.

경로 획득 절차인 경로 탐색 절차를 살펴보면, 목적지 노드에 전송할 데이터를 갖고 있는 송신 노드는 우선 자신의 라우트 캐쉬에서 목적지 노드에 대한 경로를 검색하고, 경로가 존재한다면 이 경로를 이용하여 데이터를 전송한다. 만약 캐쉬에 경로가 존재하지 않는다면 송신 노드는 자신의 주소를 경로요청 패킷에 추가하고 이것을 브로드캐스트한다. 이 패킷을 받은 이웃 노드들은 자신의 캐쉬를 확인하여 목적지 노드까지의 경로가 존재하면 라우트 캐쉬에 존재하는 목적지 노드까지의 경로를 추가하여 송신 노드에게 경로응답 패킷을 전송하고 경로가 존재하지 않는다면 경로요청 패킷에 자신의 주소를 추가하여 다음 노드로 전달한다. 이렇게 하여 최종적으로 목적지 노드가 경로요청 패킷을 수신하면 더 이상 경로요청 패킷을 전송하지 않고 지금까지 축적된 경로를 추가하여 송신 노드에게 획득한 경로를 이용하여 경로 응답 패킷을 전송한다. 경로 응답 패킷을 수신한 송신 노드는 이 경로를 이용하여 데이터를 전송한다.

라우트 캐쉬를 사용하는 방법은 경로 탐색절차를 빠르게 수행하고 경로요청 패킷수를 줄이는 장점이 있으나 노드 이동과 시간의 경과에 따라 잘못된 라우트 캐쉬 정보로 인하여 다른 노드의 라우트 캐쉬까지 영향을 미침으로써 자원의 낭비와 경로 획득 시간 지연, 그리고 TCP 성능저하를 초래한다.

2.2.2 AODV(Ad hoc On-demand Distance Vector) 라우팅 프로토콜

AODV 라우팅 프로토콜[8]은 DSDV 라우팅 프로토콜의 알고리즘을 사용한다. DSVP가 전체 경로에 대한 라우팅 정보를 유지하는데 비하여 AODV는 필요한 경로에 대한 라우팅 정보를 유지한다. 경로 설정을 위해 필요하지 않은 노드들은 라우팅 테이블에 포함되지 않고 그에 대한 정보를 전송하지 않는다. 송신 노드부터 목적지 노드까지 경로가 미리 만들어져 있지 않다면, 경로 탐색 프로세스를 사용한다. 경로요청(Route Request: RREQ) 패킷을 이웃 노드들에게 브로드캐스트하고, 이웃 노드들은 다시 그 이웃들에게 브로드캐스트하는 방법으로 경로를 찾아간다.

송신 노드가 브로드캐스트한 경로 요청 패킷이

망을 통하여 전송하는 동안, 중간 노드들은 패킷을 브로드캐스트한 이웃 노드의 주소를 라우팅 테이블에 기록하여 둠으로써 역경로를 만든다. 이후에 받게 되는 동일한 경로요청 패킷들은 버린다. 경로요청 패킷이 목적지에 도달하면, 목적지 노드와 중간 노드들은 경로요청 패킷을 전송하는 중에 만들어진 역경로로 경로 응답(Route Reply: RREP) 패킷을 보낸다. 경로응답 패킷을 전송하는 과정에서 노드들은 라우팅 테이블에 전송 경로를 기록한다.

2.2.3 ABR(Associativity-Based Routing) 프로토콜

ABR 프로토콜[9]의 핵심은 ad hoc 망에서 오래 지속되는 경로를 사용하는 것이다. 오래 지속되는 경로는 이동성이 적은 이동 노드들로 이루어진 경로이기 때문에 계속적으로 유지할 가능성이 크고 경로 재설정 가능성이 적으므로 높은 효율을 기대할 수 있다. 또한 오래 지속되는 경로 하나만을 관리하기 때문에 패킷 중복을 피할 수 있다. 그러나 ABR 프로토콜의 각 노드는 정기적으로 자신의 존재를 알리는 신호(beacon)를 만들어 브로드캐스트해야한다. 이를 이웃 노드에서 수신한 노드는 관련 테이블을 갱신한다. 그러므로 정확한 이동 노드의 시·공간 및 연결 상태를 반영하기 위해 신호간격(beacon interval)은 작아야하며 이것은 추가적으로 전력 소모를 필요로하는 원인이된다. ABR의 기본적인 목적은 ad hoc 이동 망에서 오래 지속되는 경로를 사용하는 것이고, 이를 위해 경로 탐색(route discovery), 경로 재설정(route reconstruction), 경로 삭제(route deletion)의 과정으로 이루어진다.

2.2.4 TORA(Temporally Ordered Routing Algorithm)

TORA는 동적인 이동 통신환경에 적합한 루프프리(loop-free) 분산 라우팅 프로토콜이다[10]. TORA는 1997년에 IETF에서 처음 제안하였고, 송신 노드와 목적지 노드간의 경로는 다중 경로를 설정한다. TORA의 핵심은 경로설정을 위한 라우팅 패킷의 오버헤드를 최소화하기 위해 제어메시지를 토폴로지 변화가 일어나는 작은 노드의 집합 안에서만 교환한다. 이를 위해 노드들은 이웃 노드들에 대한 라우팅 정보를 유지한다. 이 프로토콜은 경로생성(route creation), 경로 유지(route maintenance), 경로 삭제(route erasure)의 과정으로 이루어진다.

Ⅲ. on-demand 라우팅 방식

본 장에서는 ad hoc 무선망의 on-demand 방식과 관련하여 지금까지 제안된 여러 다른 종류의 라우팅 프로토콜들을 기술하고, 그것들이 가진 여러 문제점들을 기술한다.

3.1 DSR과 AODV의 경로 재설정 과정

그림 1은 DSR과 AODV의 경로 재설정 과정을 보여준다. (a)에서 노드 A는 노드 B에 이상이 생긴 것(예를 들어 노드가 이동하여 경로를 벗어나는 경우)을 알게 되고 이것에 대한 경로 에러 패킷을 송신 노드에게 보낸다. (b)에서 경로 에러 패킷을 받은 송신 노드는 경로 탐색 프로세스를 사용하여 경로 요청 패킷을 목적지 노드로 브로드캐스트한다. (c)에서 경로 요청 패킷을 받은 목적지 노드는 송신 노드에게 새로운 경로를 포함하는 경로 응답 패킷을 유니캐스트한다.

이러한 라우팅 프로토콜의 경우에서는 중간 노드에 이상이 생길 경우 송신 노드로부터 목적지 노드까지 모든 경로가 다시 재설정 된다. 즉, 경로가 끊어졌을 경우에 송신 노드는 경로 요청 메시지를 보내고 목적지 노드로부터 경로 응답 메시지를 받아야 경로가 설정된다. 그러므로 이러한 방법은 경로를 재설정하는데 걸리는 시간이 너무 길다. 게다가 송신 노드가 데이터 패킷과 경로 요청 패킷, 그리고

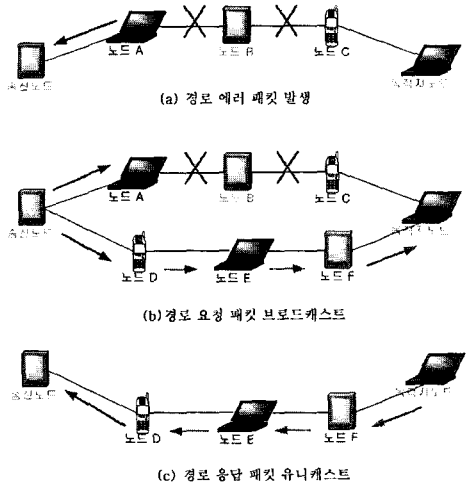


그림 1. DSR과 AODV의 경로 재설정 과정

경로 응답 패킷도 받아야 하므로 송신 노드에 너무 많은 로드(load)가 발생할 수 있다.

3.2 ABR과 TORA의 경로 재설정 과정

그림 2는 ABR과 TORA의 경로 재설정 과정을 보여준다. (a)에서 노드 A가 노드 B에 이상이 생긴 것(예를 들어 노드가 이동하여 경로를 벗어나는 경우)을 알게 되면 송신 노드에게 경로 에러 패킷을 보내는 DSR이나 AODV와는 달리 (b)에서처럼 노드 A 자신이 바로 경로 탐색 프로세스를 사용하여 경로 요청 패킷을 수신 노드로 브로드캐스트한다. 경로 요청 패킷을 받은 목적지 노드는 (c)에서처럼 노드 A 에게 새로운 경로를 포함하는 경로 응답 패킷을 유니캐스트한다.

이러한 라우팅 프로토콜의 경우에는 중간 노드에 이상이 생길 경우, 이상이 생긴 바로 전 노드로부터 목적지 노드까지 경로가 다시 재설정 된다. 하지만 이러한 방법은 경로가 끊어졌을 경우, 중간 노드 A가 경로 요청 메시지를 보내고 목적지 노드로부터 경로 응답 메시지를 받았을 때 경로가 설정되기 때문에 (c)에서처럼 설정된 경로가 가장 짧은 경로가 아닐 수도 있다는 단점이 있다. 그러므로 데이터 패킷을 전송하는 시간이 증가할 수 있다.

3.3 NRR(Novel Route Reconstruction) 프로토콜의 경로 재설정 과정

그림 3과 그림 4는 *NRR[11]* 프로토콜의 경로 재설정 과정을 보여준다. 그림 3의 (a)에서처럼 노드 B에 이상이 생겼을 경우, 노드 C가 목적지 노드에 이웃한 바로 전 노드라면 노드 C는 경로 에러 패킷을 목적지 노드로 보낸다. 그러면 목적지 노드가 (b)에서처럼 바로 경로 요청 패킷을 송신 노드로 보내고 이 패킷을 받은 송신 노드는 (c)에서처럼 경로 응답 패킷 없이 바로 경로를 설정하여 데이터 패킷을 보낼 수 있다.

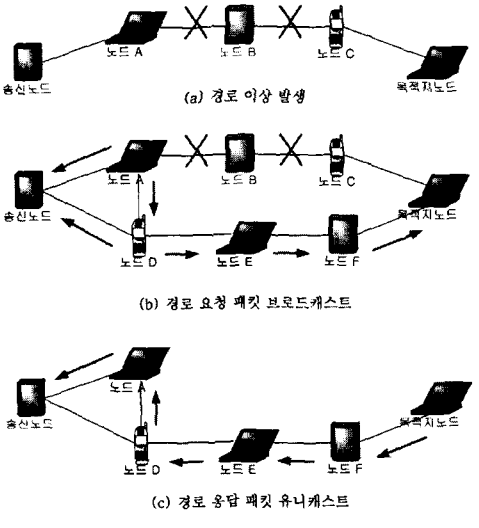


그림 2. ABR과 TORA의 경로 재설정 과정

만약 그림 4의 (a)에서처럼 노드 C가 목적지 노드에 이웃한 노드가 아니라면 노드 C는 목적지 노드에게 경로 에러 패킷을 보내지 않고 (b)에서처럼 노드 C 스스로 경로 요청 패킷을 송신 노드 쪽으로 브로드캐스트한다. 이 패킷을 받은 송신 노드는 (c)에서처럼 노드 C까지의 경로를 재설정하고 데이터 패킷을 보내게 된다. 이러한 방법은 경로 응답 패킷을 사용하지 않기 때문에 전체 경로상의 트래픽 오버헤드(overhead)를 감소시킬 수 있고, 경로 재설정 시간도 단축시킬 수 있다. 하지만 이 방법도 결국 경로가 깨졌을 경우 다시 경로를 재설정 하는 방법이므로 QoS(Quality of Service)를 보장해야하는 실시간 데이터 전송에서는 사용하기 어려울 수 있다.

3.4 RDMAR(Relative Distance Microdiscovery Ad hoc Routing) 프로토콜

그림 5는 RDMAR 프로토콜[12,13]을 사용한 경로 재설정 방법을 보여준다. (a)에서처럼 노드 B에 이상이 생기면 노드 A는 현재 경로에 이상이 생긴 것을 송신 노드에게 알리지 않는다. 즉, 경로 에러 패킷을 송신 노드로 보내지 않고 (b)에서처럼 노드 A가 경로 재설정을 위해 경로 복구 요청(route repair request;RRREQ) 패킷을 브로드캐스트 한다. 이 때, 노드 A가 브로드캐스트하는 경로 복구 요청 패킷은 RDM 알고리즘[13]을 사용하여 홉(hop)수에 제한을 두고 지역적으로 경로를 재설정 한다. 기존의 경로 상에 있던 노드 중에서 경로 복구 요청 패킷을 받은 노드 C는 (c)에서처럼 경로 응답 패킷을 새로 설정된 지역적인 경로를 통해 노드 A에게 유니캐스트한다. 노드 A가 노드 C로부터 경로 응답 패킷을 받으면 지역적으로 경로 재설정이 이루어진다. 이러한 방법은 기존의 경로 재설정 방법과는 큰 차이가 있다. 경로를 전체적으로 재설정하지 않아도 되므로 경로 재설정 시간을 많이 줄일 수 있다. 하지만 대부분의 기존 경로를 계속 사용하면서 지역적으로 경로 재설정을 하기 때문에 가장 짧은 경로가 되지 않을 가능성이 크다. 그리고 이러한 방법 역시 경로가 끊어졌을 경우에 다시 경로를 재설정

하는 방법이므로 QoS를 보장해야하는 실시간 데이터 전송에서는 문제점을 나타낼 수 있다.

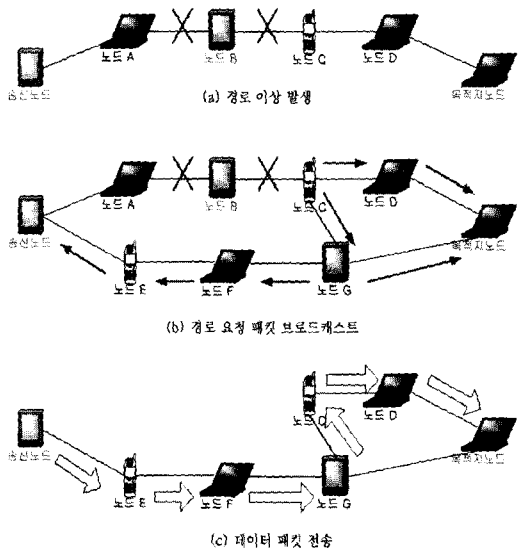


그림 4. NRR의 경로 재설정 과정 II

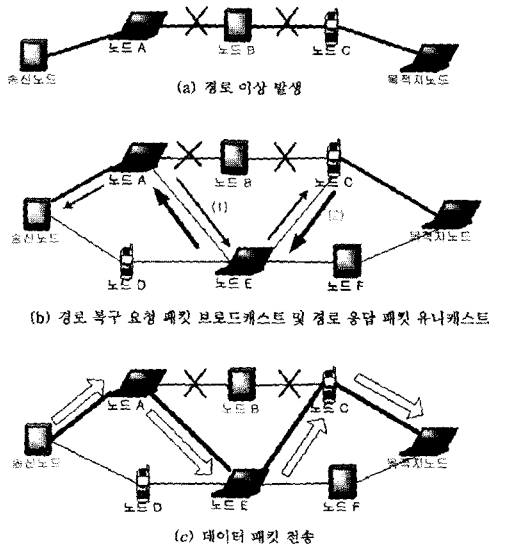


그림 5. RDMAR 프로토콜을 사용한 경로 재설정 과정

IV. 제안 경로 재설정 기법

Ad hoc 무선망의 on-demand 방식과 관련하여 지금까지 제안된 라우팅 프로토콜들은 경로가 끊어졌을 경우에 경로를 재설정하는 프로토콜이 대부분이고, 추가적으로 경로를 최대한 오래 유지하도록 하는 알고리즘이 있지만 이 알고리즘 역시 경로가 끊어지는 것을 예방할 수는 없다. 즉, 경로가 끊어졌을 경우 다시 경로를 재설정해야하는 프로토콜들은 QoS(Quality of Service)를 보장해야 하는 실시간 데이터 전송에 적합하지 않다. 그러므로 본 논문에서는 ad hoc 무선망에서 QoS 보장을 위한 경로 재설정 기법을 제안한다. 이 방법은 하나의 중간 노드가 이동하려고 할 때 경로상의 인접한 다음 노드에게 핸드오프 패킷을 보내고 이 패킷을 받은 노드는 경로가 끊어지기 전에 지역적으로 경로를 재설정하여 전체 경로가 끊어지지 않도록 하는 것이다.

본 논문에서 제안하는 경로 재설정 기법은 크게 두 가지 과정으로 나눈다. 첫째로 본 논문에서 제안하는 경로 재설정 기법은 데이터 전송 유지를 위해 핸드오프(handoff) 기법을 사용한다. 어느 한 노드가 움직이고 있고 그 노드가 현재 설정되어 있는 경로상의 중간 노드라면, 그 노드는 자신이 어떤 임계구역으로 들어 갔을 때 자신이 곧 경로를 벗어날 것임을 알리는 핸드오프 패킷을 인접한 다음 노드에게 보낸다. 핸드오프 패킷을 받은 다음 노드는 중간 노드가 경로를 벗어나기 전에 다른 경로를 재설정함으로써 데이터 전송이 끊어지지 않도록 유지한다.

둘째로 경로를 재설정하는 과정이다. 경로를 벗어나려는 노드로부터 핸드오프 패킷을 받은 노드는 중간 노드가 경로를 벗어나기 전에 다른 경로를 재설정한다. 이 때, 경로를 재설정하는데 걸리는 시간을 최소로 하기 위해 RDM 알고리즘을 사용하여 경로 요청 패킷의 홉수를 제한하고 지역적으로 경로를 재설정한다. 또한, 이동한 중간 노드의 다음 노드가 경로 요청 패킷을 브로드캐스트하기 때문에 경로 응답 패킷이 필요없으므로 좀 더 빠른 시간 안에 경로를 재설정 할 수 있다.

4.1 데이터 전송 유지를 위한 핸드오프 (handoff) 기법

4.1.1 각 노드의 전송 범위에 따른 위치

Ad hoc 무선망에서 경로를 구성하고 있는 노드들 중 송신 노드와 목적지 노드를 제외한 모든 중간 노드들은 자신의 이전 노드와 다음 노드의 전송 범위가 겹쳐지는 부분에 위치하고 있어야 한다. 즉, 이전 노드로부터 데이터를 전송 받고, 다음 노드에게 데이터를 전송할 수 있는 중계 범위 내에 위치

해야 한다. 또한, 중간 노드의 전송 범위 내에 이전 노드와 다음 노드가 모두 있어야 한다. 중간 노드가 중계 범위 내에 있다 하더라도 이전 노드나 다음 노드가 중간 노드의 전송 범위 내에 있지 않다면 중간 노드는 데이터를 중계할 수 없다.

그림 6은 중간 노드 B가 데이터를 제대로 중계할 수 있는 범위를 나타낸다. 노드 B가 전체 경로상에서 중간 노드의 역할을 제대로 하려면 이 노드는 중계 범위안에 있어야 한다. 만약 노드 B가 중계 범위를 벗어나면 이 경로는 끊어진다. 또한, 노드 A와 C는 중간 노드 B의 전송 범위 내에 있어야한다. 만약 노드 A 또는 C가 노드 B의 전송 범위를 벗어난다면 경로는 끊어진다. 이와 같이 각 노드의 전송 범위내에 인접 노드가 제대로 위치하고 있을 때에만 설정된 경로로 데이터를 올바르게 전송한다.

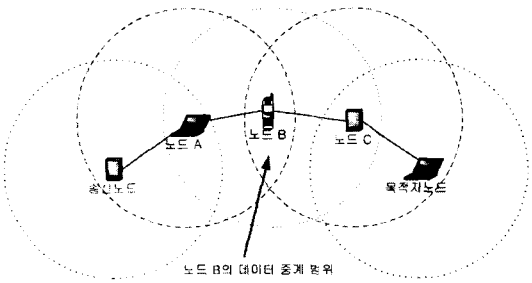


그림 6. 중간 노드의 데이터 중계범위

4.1.2 ML(Movement Limit)과 d_{cs}의 유도

Ad hoc의 각 노드들은 이동하기때문에 데이터 패킷을 전송할 수 있는 중계 범위를 벗어날 수 있다. 본 논문에서 제안하는 경로 재설정 기법은 핸드오프 과정을 거침으로써 중간 노드가 중계 범위를 벗어나기 전에 다른 경로를 미리 재설정하여 경로가 끊어지는 것을 예방할 수 있다. 이때, 모든 노드들은 GPS(Global positioning System)[14]로부터 지속적으로 자신들의 위치에 대한 정보를 얻을 수 있다고 가정한다. 또한, 노드들간의 이동 속도는 틀릴 수 있지만 각 노드의 속도는 일정하다고 가정한다.

그림 7는 본 논문에서 매우 중요한 파라미터(parameter)인 ML(Movement Limit)과 d_{cs}을 유도하기 위한 여러 가지 파라미터들을 보여준다. 노드 B는 이동하면서 GPS로부터 얻은 정보를 통하여 임계영역의 거리, d_{cs}를 계산하고 인접 노드 A, C와 정기적으로 주고 받는 신호를 통하여 그것들의 정보를 알게됨으로써 노드 B가 데이터를 제대로 중계할 수 있는 거리, ML_b를 계산한다.

그러면 ML_b와 d_{cs}를 유도하는 과정을 살펴보자. 먼저 노드 A와 노드 B 사이의 거리 d_{ab}는 수식(1)과 같이 구한다.

$$d_{ab} = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \quad (1)$$

데이터 패킷의 전송 방향에서 노드 B가 데이터 패킷을 중계할 수 있는 최대 거리, 즉 노드 B의 FL(Forward Limit)은 수식(2)와 같이 구한다.

$$FL_b = r_a - d_{ab} \quad (2)$$

같은 방법으로, 노드 B와 노드 C 사이의 거리 d_{bc}는 수식(3)과 같다.

$$d_{bc} = \sqrt{(x_b - x_c)^2 + (y_b - y_c)^2} \quad (3)$$

데이터 패킷이 전송 방향의 역방향으로 노드 B가 데이터 패킷을 중계할 수 있는 최대 거리, 즉 노드 B의 BL(Backward Limit)은 수식(4)와 같이 구한다.

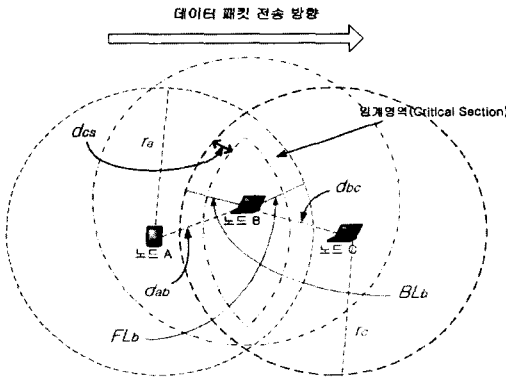


그림 7. ML_b와 d_{cs}을 유도하기 위한 여러 가지 파라미터

$$BL_b = r_c - d_{bc} \quad (4)$$

위의 수식(1),(2),(3),(4)를 이용하여 다음과 같이

노드 B의 ML(Movement Limit)를 구할 수 있다.

$$ML_b = min(FL_b, BL_b) \quad (5)$$

ML_b값이 임계영역의 거리, d_{cs}와 비교하여 같거나 작아지면 노드 B는 노드 C에게 핸드오프 패킷을 보낸다. 여기서 임계영역의 거리, d_{cs}는 다음과 같이 구할 수 있다.

$$d_{cs} = t_{rep} \cdot v_{node} \quad (6)$$

이 식에서 t_{rep}는 경로를 재설정하는데 걸리는 시간이고, v_{node}는 노드의 이동속도이다. t_{rep}는 수식(7)과 같이 구할 수 있다. 전체 경로를 설정하는데 걸리는 시간을 전체 홉 수로 나누면 하나의 링크(link)를 설정하는데 걸리는 시간이 나오게 되고, 그것을 제한 홉수 만큼 곱하면 경로를 부분적으로 재설정하는데 걸리는 총 시간이 된다.

$$t_{rep} = (t_{tot} / hop_{tot}) \cdot hop_{lim} \quad (7)$$

이렇게 구한 t_{rep}는 전체 경로를 재설정하는데 걸리는 시간보다는 작고, 경로를 부분적으로 재설정하는데 걸리는 시간의 최대값이다. 또한 hop_{lim}은 RDMAR 프로토콜을 사용하여 구할 수 있다. 그리고 v_{node}는 GPS로부터 얻을 수 있다.

4.1.3 노드의 움직임에 따른 핸드오프 패킷 발생

지금까지 노드 B가 GPS로부터 정보를 얻고, 인접 노드들과 정보를 교환함으로써 계산할 수 있는 ML_b와 d_{cs}에 관해 알아보았다. 이제 노드 B의 이동에 따라 어떻게 핸드오프 패킷이 발생하는지에 대해 알아본다.

그림 8은 노드 B가 움직이면서 임계영역 안으로 들어갈 때 핸드오프 패킷이 발생하는 과정을 보여준다. 노드 B는 움직이면서 지속적으로 인접 노드들과 정보를 교환한다. 이렇게 하여 구한 ML_b가 GPS로부터 얻은 정보를 토대로 계산한 d_{cs}와 비교하여 작아지면 핸드오프 패킷이 발생하고 노드 B는 핸드오프 패킷을 자신의 다음 인접 노드인 노드 C에게 전송한다. ML_b가 d_{cs}와 비교하여 작다는 것은 노드 B가 임계영역으로 진입했다는 것이므로 노드 B가 중계범위, 특히 그 중에서도 임계영역을 벗어나기 전에 다른 경로를 재설정해야 경로가 끊어지는 것을 막을 수 있다.

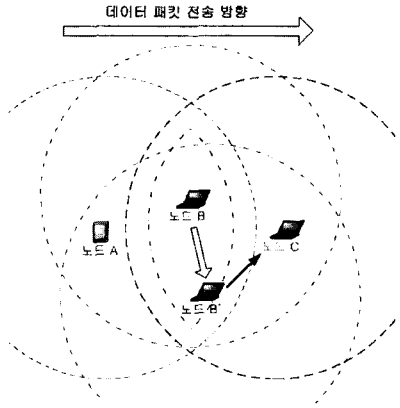


그림 8. 핸드오프 패킷 발생과정

4.2 경로 재설정 기법

노드 C는 노드 B로부터 핸드오프 패킷을 받으면 경로를 지역적으로 재설정해야한다. 만약 핸드오프 패킷을 노드 C로 보내지 않고 노드 A로 전송하여 노드 A가 경로를 재설정하게 하면, 경로 요청 패킷을 보낸 후 경로 응답 패킷을 받아야 경로가 설정이 된다. 이러한 방법은 경로를 재설정하는데 걸리는 시간이 길어지게되므로 노드들의 움직임에 따라 핸드오프 기법을 사용하고 빠른 경로 재설정 시간을 필요로 하는 본 논문의 경로 재설정 기법에는 적당하지 않다. 그러므로 본 논문에서는 노드 B가 핸드오프 패킷을 노드 C로 보내고, 노드 B로부터 핸드오프 패킷을 받은 노드 C가 지역적으로(locally) 경로를 재설정하는 방법을 제안한다.

그림 9는 노드 C가 지역적으로 경로를 재설정하는 과정을 보여준다. 핸드오프 패킷을 받은 노드 C는 메시지를 받자마자 바로 경로 요청 패킷을 브로드캐스트한다. 이때, 홉수에 제한을 두고 브로드캐스트함으로써 경로 요청 패킷이 망에 무한정으로 퍼져 나가는 것을 막고, 지역적으로 경로를 재설정한다. 홉수를 몇 개로 제한할 것인가 하는 것은 RDMAR 프로토콜을 사용하여 구한다.

노드 C가 브로드캐스트하는 경로 요청 패킷은 노드 A를 목적지 노드로 하여 브로드캐스트된다. 즉,

이동하여 경로를 벗어난 노드 B의 이웃한 이전 노드인 노드 A에게 경로 요청 패킷을 브로드캐스트하는 것이다. 경로 요청 패킷이 노드 A에 도착하면, 노드 A는 노드 C까지의 경로 중에서 최적의 경로를 선택하고 나머지 경로는 무시한다. 일단 경로가 지역적으로 재설정되면 노드 A는 새로운 경로로 데이터 패킷을 전송한다.

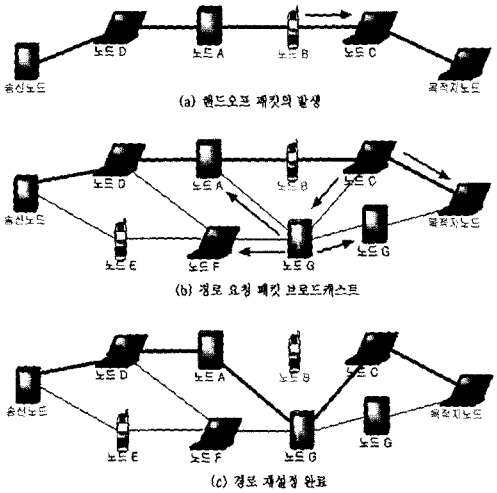


그림 9. 지역적인 경로 재설정 과정

V. 성능평가

본 장에서는 본 논문에서 제안한 경로 재설정 방법을 기존의 대표적인 프로토콜들의 경로 재설정 방법(DSR과 AODV의 경로재설정 방법, ABR과 TORA의 경로재설정 방법)과 시뮬레이션을 통해 성능을 비교하고 평가한다.

5.1 성능평가 환경

표 1은 성능평가를 위한 시뮬레이션 파라미터 값이다. 전체 시뮬레이션 공간은 1500m X 300m의 직사각형 공간이고, 이동 노드의 수는 60개로 한다. 각 노드들의 속도는 20개는 2m/sec의 속도를 가지고 20개는 4m/sec, 그리고 나머지 20개는 8m/sec의 속도를 가진다. 이동노드의 움직이는 방향은 처음 시뮬레이션을 시작할 때는 랜덤하지만 일단 정해지면 방향은 바뀌지 않고, 움직이는 속도도 변하지 않는다고 가정한다. 데이터 패킷의 사이즈는 256 bytes이고, 경로 요청 패킷과 경로 응답 패킷은 64 bytes이다. 경로 에러 패킷과 핸드오프 패킷의 크기는 32 bytes이다. 데이터 패킷의 전송율은 1 Mbps이고, 전체 시뮬레이션 시간은 300초로 한다.

표 1. 시뮬레이션 파라미터 값

파라미터	값
시뮬레이션 공간	1500*m* X 300*m* (직사각형)
이동 노드의 수	60 개
이동 노드의 속도	각 20개씩 *2m/sec, 4m/sec, 8m/sec*
데이터 패킷 사이즈	256 *bytes*
경로 요청 패킷 사이즈	64 *bytes*
경로 응답 패킷 사이즈	
경로 에러 패킷 사이즈	32 *bytes*
핸드오프 패킷 사이즈	
데이터 패킷 전송율	1 *Mbps*
시뮬레이션 시간	300 초

5.2 성능평가 결과

본 논문에서는 성능평가를 크게 두 가지로 나누어 시뮬레이션한다. 먼저 링크 오류(link failure)가 발생하는 횟수에 따라 처리량과 데이터 패킷당 전송 지연시간, 그리고 제어패킷 오버헤드가 기존의 방법들과 어떻게 다른지를 비교 평가하고, 두 번째로 이동노드의 평균속도가 변함에 따라 처리량과 데이터 패킷당 전송 지연시간, 그리고 비용이 기존의 방법들과 어떻게 다른지를 비교 평가한다. 편의를 위해 이 장에서는 DSR과 AODV의 경로 재설정 방법을 프로토콜 1, ABR과 TORA의 경로 재설정 방법을 프로토콜 2, 그리고 본 논문에서 제안하는 경로 재설정 방법을 제안 프로토콜이라고 부른다.

5.2.1 링크 오류(link failure) 발생횟수의 변화에 따른 비교평가

그림10-a는 링크 오류 발생횟수가 변함에 따라 일정 시간 동안 각 프로토콜들의 처리량이 어떻게 변하는지를 보여준다. 링크 오류가 한번도 일어나지 않는 상태에서는 모든 방법에서 처리량이 우수하게 나타나지만 링크 오류의 발생횟수가 증가 함에 따라 처리량의 변화가 심한 것을 알 수가 있다. 프로토콜 1과 프로토콜 2는 경로가 끊어졌을 경우 다시 경로를 재설정하는데 걸리는 시간이 많기 때문에 링크 오류가 발생하면 처리량이 낮아진다. 하지만 본 논문에서 제안하는 제안 프로토콜은 핸드오프

패킷을 이용하여 경로가 끊어지지 않도록 유지하기 때문에 지속적으로 데이터 패킷을 전송할 수 있게 되어 높은 처리량을 나타낸다.

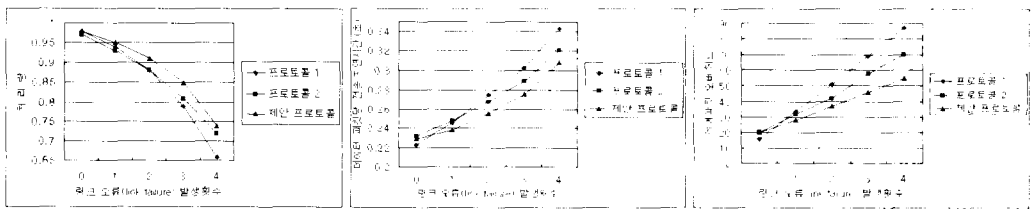
그림10-b는 링크 오류의 발생횟수가 변함에 따라 데이터 패킷당 전송 지연시간이 어떻게 변하는지를 보여준다. 링크 오류가 한번도 발생하지 않은 상태에서는 세 개의 프로토콜에서 별 차이가 나타나지 않지만 링크 오류의 발생횟수가 변함에 따라 데이터 패킷당 전송 지연시간이 크게 차이가 나는 것을 알수가 있다. 프로토콜 1과 프로토콜 2는 경로가 끊어졌을 경우 다시 경로를 재설정하는데 걸리는 시간이 제안 프로토콜보다 매우 크기 때문에 패킷당 전송 지연시간이 그만큼 늦어진다. 처음에 링크 오류가 한번도 발생하지 않았을 경우, 프로토콜 1이 가장 낮은 전송 지연시간을 나타내는데, 그 이유는 처음 경로를 설정할 때 가장 짧은 경로를 이용하여 설정하므로 재설정이 한번도 일어나지 않은 상태에서는 전송 속도가 가장 빠르기 때문이다.

그림10-c는 링크 오류의 발생횟수가 변함에 따라 제어패킷 오버헤드가 어떻게 변하는지를 보여준다. 제어패킷 오버헤드는 경로 요청 패킷, 경로 응답 패킷, 경로 에러 패킷, 핸드오프 패킷, 그리고 정기적으로 주고 받는 신호와 같은 제어 패킷들의 합을 나타낸다. 프로토콜 1과 포로토콜 2의 경우는 링크 오류가 발생했을 때 경로 요청 패킷을 브로드캐스트하고 다시 경로 응답 패킷을 수신해야 하기 때문에 제어패킷 오버헤드가 높게 나타나지만, 제안 프로토콜의 경우는 경로 응답 패킷이 필요없고 또한 경로 요청 패킷도 전체 경로 중 부분적으로 브로드캐스트 되기 때문에 제어패킷 오버헤드가 낮게 나타나는 것을 볼 수 있다.

5.2.2 이동노드의 평균속도 변화에 따른 비교평가

그림10-d는 이동노드의 평균속도가 변함에 따라 처리량이 어떻게 변하는지를 보여준다. 이동노드의 평균속도를 0m/sec부터 10m/sec까지 2m/sec씩 늘려가며 변화를 주고 그에 따른 처리량의 변화를 비교평가한다. 모든 이동노드들의 평균속도가 0m/sec일 경우는 이동 노드의 움직임이 없는 상태이므로 처리량의 값이 모두 높게 나타나고 차이가 별로 없지만, 이동노드들의 평균속도가 10m/sec일 경우는 이동 노드들의 평균 이동 속도가 매우 빠르기 때문에 처리량의 값이 차이가 나는 것을 볼 수 있다.

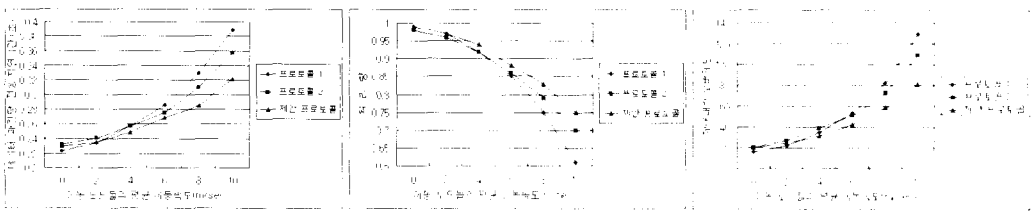
그림10-e는 이동 노드의 평균 이동속도가 변함에 따라 데이터 패킷당 전송 지연시간이 어떻게 변하는지를 보여준다. 이동 노드들의 평균 이동속도가 10m/sec라는 것은 노드들이 빠르게 이동하고 계속해서 링크 오류(link failure)가 발생한다는 뜻이므로, 경로가 끊어졌을 경우 경로를 모두 재설정하는 프로토콜 1과 프로토콜 2는 경로가 끊어지지 않도록 하는 제안 프로토콜보다 데이터 패킷당 전송 지연시간이 길어진다.



[10-a] 링크오류의 발생횟수에 따른 처리량의 변화

[10-b] 링크오류의 발생횟수에 따른 데이터 패킷당 전송 지연시간

[10-c] 링크오류의 발생횟수에 따른 제어패킷 오버헤드의 변화



[10-e] 이동노드의 평균속도에 따른 데이터 패킷당 전송 지연시간

[10-d] 이동노드의 평균속도에 따른 처리량의 변화

[10-f] 이동노드의 평균속도에 따른 비용의 변화

그림 10. 시뮬레이션 결과 그래프

그림10-f는 이동 노드의 평균 이동속도가 변함에 따라 제어패킷 오버헤드가 어떻게 변하는지를 보여준다. 이동 노드들의 평균 이동속도가 0m/sec, 2m/sec, 4m/sec로 빠르지 않을 경우에는 세 프로토콜의 제어패킷 오버헤드가 많이 차이가 나지 않는 것을 볼 수 있다. 하지만 이동 노드들의 평균 이동속도가 6m/sec, 8m/sec, 10m/sec로 높아지면 노드들이 빠르게 이동하고 계속해서 링크 오류(link failure)가 발생한다는 뜻이므로, 프로토콜 1과 프로토콜 2의 경우는 경로 요청 패킷과 경로 응답 패킷의 발생이 많아지게 된다. 그러므로 프로토콜 1과 포로토콜 2의 제어패킷 오버헤드가 급격하게 올라가게 된다.

Ⅵ. 결 론

Ad hoc 망의 on-demand 라우팅 방식에서 지금까지 수많은 연구가 이루어지고 있다. 하지만 기존의 연구들은 노드의 이동에 의하여 경로가 끊어졌을 경우, 거의 모든 경로를 다시 재설정하는 방식이었다.

본 논문에서는 ad hoc 망에서 QoS 보장을 위한 경로 재설정 기법을 제안한다. 이 라우팅 기법은 기존의 방법들이 경로가 끊어졌을 경우 다시 경로를 재설정 하는것과는 다르게 노드가 이동하면서 임의의 임계구역안으로 들어갈 경우 핸드오프 패킷을 발생시켜 다른 경로를 부분적으로 재설정함으로써 경로가 끊어지는 것을 미리 예방한다. 또한 RDM 알고리즘을 사용하여 경로를 재설정함으로써 경로 요청 패킷의 홉수를 제한하고 지역적으로 빠르게 경로를 재설정하여 QoS를 보장해야 하는 실시간 데이터 전송에 적합하다고 할 수 있다.

성능평가를 통해 본 논문에서 제안한 경로 재설정 방법이 DSR과 AODV의 경로 재설정 방법이나 ABR과 TORA의 경로 재설정 방법보다 처리량과 데이터 패킷당 전송시간에서 우수한 성능을 나타내는 것을 알 수 있다.

향후 연구 과제는 본 논문에서는 고려하지 않았던 송신 노드와 목적지 노드가 경로를 벗어났을 경우에 대한 연구가 남아 있다.

참 고 문 헌

[1] C-K Toh, Ad Hoc Mobile Wireless Networks, Prentice Hall, 2002.

[2] Elizabeth M. Royer and C-K Toh, " A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks," IEEE Personal Communications, Vol. 6., Apr. 1999, pp. 46-55.

[3] Mattew S. Gast, 802.11 Wireless Networks, O'Reilly, 2002, pp. 7-50.

[4] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing(DSDV) for Mobile Computers," Computer Communication Review, Oct 1994, pp.234-244.

[5] Boukerche A., Fabbri A. and Das S. K, "Analysis of Randomized Congestion Control in DSDV Routing," Proceedings of IEEE 8th International Symposium on, 2000.

[6] L. R. Ford Jr. and D. R. Fulkerson, Flows in Networks, Princeton Univ. Press, 1962.

[7] D. B. Johnson and D. A. Maltz. "Dynamic Source Routing in Ad-Hoc Wireless Networks," Mobile Computing, 1996, pp. 153-181.

[8] Royer E. M and Perkins C. E., "An Implementation Study of the AODV Routing Protocol," Wireless Communications and Networking Confernce, 2000. IEEE, Vol. 3, pp. 1003-1008.

[9] C-K Toh, "Associativity-Based Routing For Ad-Hoc Mobile Networks," Wireless Personal Communications Journal, Special Issue on Mobile Networking & Computing Systems, Vol. 4, No. 2, March 1997.

[10] C-K Toh, G. Guichal, and S. Bunchua, "ABAM: On-Demand Associativity-Based Multicast Routing for Ad Hoc Mobile Networks," Vehicular Technology Conference, 2000. IEEE-VTS. 52nd, Vol. 3, pp. 987-993.

[11] Toshifumi Miyagi, Masataka Iizuka and Masahiro Morikura, "A Novel Route

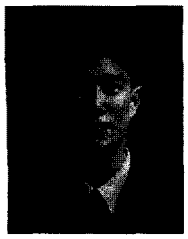
Reconstruction Protocol for P-MP Communication over Wireless Ad-hoc Network," Vehicular Technology Conference, 2000. IEEE 51st, Vol. 2, pp. 1522-1526.

[12] Chenxi Zhu and Corson M. S., "QoS Routing for Mobile Ad Hoc Networks," Proceedings of IEEE INFOCOM 2002, Vol. 2, pp. 958-967.

[13] Agglou, G. and Tafazolli, R. "Determining The Optimal Configuration for The Relative Distance Microdiscovery Ad Hoc Routing Protocol," Vehicular Technology, IEEE Transactions on, Vol. 51, March 2002, pp. 354-370.

[14] E. D. Kaplan, Understanding the GPS: Principles and Applications, Artech House, Boston, Feb. 1996.

한 정 안(Jung-Ahn Han) 준회원



1996년 2월 : 경원대학교 전자 계산학과 학사
1998년 8월 : 숭실대학교 대학원 컴퓨터학과 석사
1998년 9월~현재 : 숭실대학교 컴퓨터학과 박사과정

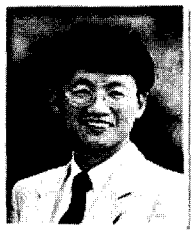
<관심분야> 이동통신 프로토콜, ad-hoc 네트워크, 무선 센서 네트워크

백 종 근(Jong-Geun Paik) 준회원

2001년 2월 : 상지대학교 응용 통계학과 학사
2003년 2월 : 숭실대학교 대학 원 컴퓨터학과 석사
2003년 10월 ~ 현재 : ㈜베타리서치&컨설팅 모바일 리서치팀

<관심분야> Ad Hoc, Mobile IP

김 병 기(Byung-Gi Kim) 정회원



1977년 2월 : 서울대학교 전자공학과 학사
1979년 2월 : 한국과학기술원 전산학과 이학석사
1997년 2월 : 한국과학기술원 전산학과 공학박사
1979년 2월 ~ 1982년 2월 : 경북대학교 전자공학과 전임강사
1982년 3월 ~ 현재 : 숭실대학교 정보과학대학 컴퓨터학부 교수

<관심분야> 병렬컴퓨터구조, 인터넷 망, 이동통신, ad-hoc 네트워크