

이동 애드 혹 네트워크에서의 QoS를 지원하는 라우팅 프로토콜

정희원 강경인*, 박경배**, 유충열***, 정찬혁****, 이광배****, 김현욱****

Routing Protocol with QoS Support in the Mobile Ad Hoc Networks

Kyeong-in Kang*, Gyong-bae Park**, Choong-youl You***

Chan-hyeok Jung****, Kwang-bae Lee****, Hyun-ug Kim**** *Regular Members*

요 약

최근 실시간 데이터와 멀티미디어 데이터의 수요가 급격히 증가하고 있으며, 이동 애드 혹 네트워크에서 최선형(best effort) 서비스의 지원만으로는 이러한 데이터의 전송을 효율적으로 수행하는 것이 불가능하다. 따라서 통신 자원들을 사전에 예약하는 QoS(Quality of Service) 지원형 서비스가 절대적으로 필요하다.

노드간의 링크가 양방향이라고 가정한 기존의 라우팅 프로토콜은 경로 발견과 유지가 편리한 반면에 실제적인 무선 환경에서 시간에 따라 변하는 무선 링크의 특성에 기인되는 단방향 링크에 대해서는 라우팅 기능을 지원할 수 없다. 이러한 문제점을 해결하기 위해서 본 논문에서는 단방향 링크를 지원하고, 각 노드에서 사용 가능한 통신 자원을 미리 고려하여 QoS 지원에 적합한 최단 경로를 설정함으로써 자원 소비를 줄일 수 있는 단방향 QoS 라우팅 프로토콜을 제안했다.

ABSTRACT

Recently, demand for real-time data and multimedia data is rapidly increasing, and it is impossible for mobile Ad Hoc networks with only best effort service to efficiently transfer such data. So, we absolutely need the QoS service which reserves the communication resources in advance.

The existing routing protocols, which assume that the links between nodes are bidirectional, provide the convenience for the route discovery and maintenance, but can not support the routing function for the unidirectional links due to the wireless link property easily changing with time under the real wireless environment. In order to solve such problems, in this dissertation we suggested a unidirectional QoS routing protocol that supports unidirectional links and reduces a waste of communication resources. The waste of communication resources is reduced by establishing the shortest route suitable to QoS support, considering in advance the usable communication resources at each node

1. 서 론

현재 대부분의 무선 이동 통신 분야는 기반(infra-structure) 네트워크에 기초해 있다. 예를 들어, 고정

된 기지국(base station)을 통한 종단 시스템(endpoint system)만의 이동성을 고려한 서비스 지원의 연구에 집중되어 있는 형편이다.¹⁾

그렇지만 기반 네트워크가 갖추어지지 않은 전쟁터, 재난 구조 지역, 산간 지역 등에서의 통신을 위

* 여주대학 정보통신과(kikang@mail.yeojoo.ac.kr),

*** 동아방송대학 인터넷방송과,

논문번호 : K01162-0721, 접수일자 : 2001년 7월 21일

** 여주대학 컴퓨터사이언스과,

**** 명지대학교 전자공학과 (supei@hanmail.net)

해서는 중단 시스템 사이의 직접적인 통신이 필요할 수 있다. 중단 시스템 사이의 직접 통신을 위해서 개발되어진 네트워크가 바로 이동 애드 혹 네트워크(MANET : Mobile Ad hoc NETWORKS)이다.^{2,3)}

이동 애드 혹 네트워크의 통신 환경은 각 노드들의 이동이 네트워크 도처에 편재하고 있으며 이로 인해서 네트워크 토폴로지(topology)가 변하게 되고 결과적으로 네트워크 상의 각 흐름(flow)들의 경로(route)에 에러가 발생하게 되면 통신을 지속적으로 지원하기 위해 라우팅이 다시 시도되어진다.⁴⁾

기반 네트워크로 구성되어있는 현재 인터넷에서는 VOD(Video On Demand), VoIP(Voice over Internet Protocol), 전자 상거래, 화상 회의, 네트워크 게임, 가상 현실 등과 같은 새로운 응용들이 사용되어지면서 인터넷 자원의 효율적인 사용에 대한 문제가 심각하게 대두되고 있다. 또한 이러한 응용들의 대부분이 QoS 지원을 고려하고 있기 때문에 이러한 요구에 맞추어 자원을 효율적으로 사용하는 문제는 차세대 인터넷뿐만 아니라 이동 애드 혹 네트워크에서도 가장 주요한 과제의 하나로 등장하고 있다.⁵⁾

아직까지는 이동 애드 혹 네트워크 상에서의 QoS 지원에 대한 확실한 표준은 없으며 최선형 서비스에 의해 데이터 전송이 이루어지고 있다.

따라서 본 논문은 이동 애드 혹 네트워크에서 단방향 링크를 지원하는 라우팅 프로토콜에 노드들의 이동과 변하기 쉬운 각 노드에서의 유용한 자원을 고려한 QoS 지원을 위한 전처리 과정을 삽입함으로써 신속하고 정확한 QoS 지원과 라우팅 프로토콜 수행 시 자원 소비를 줄일 수 있는 방법을 제안하였다.

II. 기존 라우팅 프로토콜

현재 이동 애드 혹 네트워크에 제안된 라우팅 프로토콜은 일반적으로 테이블 구동(table driven) 방식과 요구 기반 구동(on demand driven) 방식으로 나눌 수 있다.

테이블 구동 방식은 DSDV(Destination Sequence Distance Vector)와 DSDV에서 파생된 CGSR(Clusterhead Gateway Switch Routing)과 WRP(Wireless Routing Protocol) 등으로 구분할 수 있다.

요구 기반 구동 방식은 DSR(Dynamic Source Routing), AODV(Ad hoc On demand Distance

Vector), TORA(Temporally Ordered Routing Protocol) 및 ABR(Associativity Based Routing) 등이 있다

2.1 테이블 구동 라우팅 프로토콜

테이블 구동 라우팅 프로토콜은 각 노드로부터 네트워크 내의 다른 모든 노드로 일관되게 갱신되는 라우팅 정보를 유지한다.

이러한 프로토콜은 라우팅 정보를 저장하기 위하여 하나 또는 그 이상의 테이블을 필요로 하며, 네트워크 정보를 전송하여 모든 노드들이 일관된 네트워크 토폴로지에 대한 정보를 갖도록 테이블을 갱신한다. 이 프로토콜은 라우팅에 관련된 어떠한 테이블을 유지하는가에 따라 다시 분류될 수 있다.

2.2 요구 기반 구동 라우팅 프로토콜

요구 기반 구동 라우팅 프로토콜은 테이블 구동 라우팅 프로토콜과는 다른 접근 방식을 사용한다. 이러한 방식의 경우, 소스 노드의 필요에 따라 경로가 설정된다. 목적지 노드로의 경로가 필요하다면, 네트워크 내의 경로 발견 과정을 수행한다. 이 과정은 경로가 발견되거나, 모든 가능한 경로 가능성에 대한 검사가 끝난 후에 완료된다. 이 경로는 목적지 노드로 접근 불가능하게 되거나, 경로가 더 이상 필요 없을 때까지 유지된다.

III. DSR 프로토콜

DSR 프로토콜은 위에서 설명된 요구 기반 방식으로서 소스 노드가 패킷을 전송하기 위해서 중간 노드들의 완전한 순서를 결정하는 라우팅 기술을 말한다. 소스노드에서 전송된 패킷은 해당 경로를 저장하고 있는 경로레코드를 확인하여 다음 노드의 주소로 전송된다.

경로 요구 패킷(route request packet : RREQ)과 경로 응답 패킷(route reply packet : RREP)을 생성한 노드를 정확히 구별하기 위해서 소스 노드에서 생성되는 RREQ와 RREP에는 S(Source)를 붙이고, 목적지 노드에서 생성되는 RREQ와 RREP에는 D(Destination)를 붙인다.

3.1 경로 발견

DSR 프로토콜에서의 경로 발견은 두 노드간 통신의 필요성이 있을 경우에만 동적으로 이루어지게 된다. 소스 노드가 RREQ-S를 브로드캐스트하면서 경로 발견이 시작된다. 소스 노드는 RREQ-S를

위해 유일한 요구 식별자(request identifier)와 목적지 노드 주소를 포함한 목록을 관리하고 중간 노드와 목적지 노드는 요구 식별자와 소스 노드 주소를 포함한 목록을 관리한다.

각 노드는 최신의 RREQ-S를 확인하여 경로의 중복이나 루프를 제거하기 위해 이 목록을 관리한다. 또한 RREQ-S에는 소스 노드에서 목적지 노드까지의 경로를 기록하는 경로 레코드(route record)가 있다.^{6,7)}

RREQ-S를 브로드캐스트한 노드의 전송 영역 내에서 RREQ-S를 수신한 노드들은 다음과 같은 단계를 수행한다.

첫째, RREQ-S의 수신 노드가 RREQ-S의 목적지 노드일 경우는 RREQ-S 상의 경로 레코드에 기록된 경로에 자신의 주소를 더한 경로를 RREP-D에 복사한 후, 소스 노드로 전송한다.

둘째, RREQ-S의 수신 노드가 목적지 노드가 아닌 경우 RREQ-S가 수신 노드의 <소스 주소, 요구 식별자> 목록에 포함되어 있는지를 확인하고, 목록에서 발견될 경우는 수신된 RREQ-S는 이미 수신한 것으로 간주하여 중복된 RREQ-S를 폐기한다.

셋째, 수신 노드의 <소스 주소, 요구 식별자> 목록에 없는 경우, 수신 노드는 RREQ-S 상의 경로 레코드 내에 자신의 주소가 있는지를 확인한다. 만약, 자신의 주소가 발견될 경우는 루프로 간주해 폐기한다.

넷째, RREQ-S 상의 경로 레코드 내에 자신의 주소가 없는 경우 자신의 주소를 RREQ-S의 경로 레코드에 추가시키고 RREQ-S를 다시 브로드캐스트한다.

다섯째, 목적지에서 RREQ-S패킷을 받은후 소스 노드에서 중간노드를 거쳐 목적지까지의 순방향 경로를 RREP-D 패킷에 저장한뒤 유니캐스트 방식으로 소스노드로 전송한다.

여섯째, 소스노드는 목적지에서 온 RREP-D패킷을 받은 뒤 자신의 경로캐시에 RREP-D패킷에 담겨져 있는 순방향 경로를 저장한다.

3.2 경로 유지

DSR 프로토콜은 주기적인 경로 정보를 교환하지 않고 경로 발견 과정을 통해 얻은 경로를 사용한다. 이때, 중간 노드들은 경로 유지 과정을 수행하여 경로의 동작 상태와 에러를 탐지하고 그 사실을 소스 노드에게 알린다. 경로의 에러를 탐지하는 방법은 다음과 같다.

첫째, IEEE 802.11에서처럼 연결 계층에서 응답 신호를 사용하는 경우, 패킷을 전송한 노드가 다음 노드로부터의 응답 신호가 도착하면 자신과 다음 노드 사이의 링크에는 문제가 없음을 인식한다.^{8,9)}

둘째, 경로 에러를 탐지하기 위하여 감청 모드를 사용하는 것이다. 패킷을 전송한 노드에서 다음 노드가 자신이 전송한 패킷을 그 다음 노드로 전송하는 것을 감청 모드를 통해 수신하면 자신과 다음 노드 사이의 링크에 문제가 없는 것으로 인식한다.¹⁰⁾

셋째, 명백한 응답 신호를 요구하는 방법이 있다. 패킷을 다음 노드로 전송하기 전에 송신 노드는 확인 요구(acknowledgement request) 필드를 세팅하여 전송한다. 패킷을 수신한 노드는 반드시 이전 노드로 응답 신호를 보내주어야 한다. 만일 송신 노드에서 응답 신호를 수신하지 못한다면 송신 노드는 자신과 다음 노드 사이의 링크에 에러가 발생한 것으로 인식한다.¹¹⁾

IV. 제안한 라우팅 프로토콜

기존의 DSR 프로토콜은 소스 노드가 데이터 패킷을 전송하기 위해서 목적지 노드까지의 최단 경로를 결정한 다음 그 최단 경로를 이용해서 데이터 패킷을 전송한다.

기존의 DSR 프로토콜의 라우팅 과정에는 RREQ-S의 반복적인 브로드캐스트로 인해서 통신 자원이 굉장히 많이 소비되고 또한 선택된 최단 경로는 RREQ-S가 전송되어진 시간에서 고려된 최단 경로에 불과하다. 즉 시간에 따라 변하는 경로 상태 특성 때문에 최단 경로도 시간에 따라 변화할 수밖에 없다. 따라서, 중간 노드에서의 RREQ-S의 반복적인 브로드캐스트 시의 자원 소비도 줄이고 유용한 자원을 고려한 정확한 최단 경로를 설정할 수 있는 라우팅 프로토콜을 제안하고자 한다.

또한 경로 에러 시에 신속한 경로 복구와 재 라우팅을 위한 자원 소비를 줄이고 경로 상의 자원 변화에 신속하게 대처하기 위해 순방향 경로와 역방향 경로 모두에 대해 다중 경로를 사용하고 경로 유지를 위해서는 주기적인 헬로우 메시지를 사용한 다.

4.1 경로 발견

제안한 라우팅 프로토콜의 수행은 최선형 서비스와 QoS 지원형 서비스로 나누어서 동작한다. 최선

형 서비스를 위해 제한한 라우팅 프로토콜 동작 과정은 기존의 DSR 프로토콜 동작 과정에 유용한 자원 결정 과정만이 포함되고, QoS 지원형 서비스는 기존의 DSR 프로토콜에 유용한 자원 결정 과정과 허용 제어 과정 등이 추가된다. QoS 지원을 위해 기존의 RREQ-S에 QoS 지원 서비스와 최선형 서비스를 구분하는 RES(REServation) 필드, 현재 경로가 제공하는 유용한 자원을 저장하는 PAR(Path Available Resource) 필드, 역 방향 경로가 제공하는 유용한 통신 자원 값 RPAR(Reverse Path Available Resource) 필드, 소스 노드가 요구하는 최대 자원 요구 값 Rmax(Request max) 필드, 소스 노드가 요구하는 최소 자원 요구 값 Rmin(Request min) 필드, 노드의 제공 가능한 통신 자원 값 NAR(Node Available Resource) 필드, 순방향 경로상의 변화된 자원 값을 평가하는 RE(Resource Estimate) 필드, 역방향 경로상의 변화된 자원 값을 평가하는 RRE(Reverse Resource Estimate) 필드 등의 QoS 지원 필드를 추가한다. Rmin 이상을 제공하는 경로만을 찾는 경로 발견 과정을 수행한 결과는 Rmin 이상을 제공하는 경로와 각 경로에 대한 유용한 자원 값을 저장하는 PAR 필드를 QoS 지원형 과정에 제공한다.

4.2 최선형 서비스 경로발견

최선형 서비스 경로 발견은 DSR 에 기반한 요구 기반 방식으로 소스 노드에서 목적지 노드로 데이터를 전송하려고 할 때, 자신의 경로 캐쉬에 목적지 노드에 대한 경로가 없을 경우에만 통신이 시작됩니다. 소스노드는 자신의 경로캐쉬에 목적지 노드에 대한 경로정보가 없으면 네트워크 환경내에 있는 이웃노드에게 최선형 서비스를 의미하는 RES 필드 값을 0으로 하여 각 노드에서 경로상의 루프와 중복수신 제거를 위해 요구 식별자(request id)와 소스 주소와 목적지 주소를 포함한 RREQ-S를 브로드캐스트하여 전송합니다. 한편 RREQ-S 패킷안에는 소스 노드에서 목적지 노드까지의 경로를 기록하는 경로 레코드(route record)를 RREQ-S 안에 가지고 있습니다. RREQ-S를 수신한 노드가 목적지 노드인지 또는 목적지 노드가 아닌지에 따라 다음과 같은 단계를 수행한다.

RREQ-S를 수신한 노드가 목적지 노드가 아닌 경우에는 다음과 같이 수행한다.

첫 번째 소스 노드에서 목적지 노드로 경로 설정 패킷 RREQ-S를 브로드캐스트한다.

두 번째 RREQ-S를 수신한 중간 노드들은 RREQ-S를 자신의 이웃 노드들에게 전송한다.

각 노드는 수신된 RREQ-S가 노드의 <소스 주소, 요구 식별자> 목록에 포함되어 있는지를 확인하고, 목록에서 발견되면 RREQ-S를 폐기한다.

세 번째 수신 노드의 <소스 주소, 요구 식별자> 목록에 없는 경우, 노드는 RREQ-S 상의 경로 레코드 내에 자신의 주소가 있는지를 확인한다. 만약, 자신의 주소가 발견된 경우는 루프로 간주해 폐기한다.

네 번째 RREQ-S 상의 경로 레코드에서 자신의 주소가 발견되지 않은 경우 자신의 경로 캐쉬에 목적지 노드까지의 경로가 저장되어 있는 경우는 목적지 노드까지 RREQ-S를 유니캐스트한다. 완전한 경로의 유용한 자원 값은 현재 노드에서 목적지 노드까지의 경로의 유용한 자원 값과 현재 RREQ-S의 PAR 필드를 비교해서 적은 값이 유용한 자원 값이 된다.

다섯 번째 경로 캐쉬에 목적지 노드까지의 경로가 없는 경우에는 자신의 주소를 RREQ-S의 경로 레코드에 추가시키고 노드 유용한 자원 결정 과정을 수행한다. 자신이 제공할 수 있는 유용한 자원 값과 PAR 필드에 저장되어 있는 값을 비교해서 유용한 자원 값이 적으면 PAR 필드를 갱신하고 크거나 같으면 PAR 필드 값을 그대로 유지하고 RREQ-S를 다시 브로드캐스트한다.

RREQ-S를 수신한 노드가 목적지 노드인 경우에는 다음과 과정을 실행합니다.

첫 번째 목적지 노드가 처음으로 수신된 RREQ-S에 대해서는 RREQ-S의 경로 레코드에 기록된 경로와 PAR 필드에 저장되어 있는 경로 유용한 자원 값을 자신의 경로 캐쉬에 저장하고 RREP-D에 복사한 후 경로 캐쉬에 소스 노드까지의 경로가 있는 경우는 RREP-D를 경로를 통해 유니캐스트하고 경로가 없는 경우는 RREP-D를 RREQ-D에 피기백시켜 소스 노드로의 역방향 경로를 발견하기 위해서 브로드캐스트한다.

두 번째 RREQ-D를 수신한 소스 노드는 RREP-D의 경로 레코드를 참조하여 순방향 최단 경로와 순방향 최단 경로의 유용한 자원 값 및 역방향 최단 경로와 역방향 최단 경로의 유용한 자원 값을 자신의 경로 캐쉬에 저장하고 역방향 최단 경로의 유용한 자원 값과 역방향 최단 경로를 RREP-S에 넣어서 순방향 최단 경로를 사용해서 목적지 노드까지 유니캐스트하고, 데이터 패킷을 결정

된 순방향 최단 경로를 통해 유니캐스트한다.

4.3 QoS 지원형 서비스 경로 발견

QoS 지원형 서비스 경로 발견은 요구 기반 방식으로 소스 노드에서 목적지 노드로 데이터를 전송하려고 할 때, 자신의 경로 캐쉬에 자원 요구 최소값 R_{min} 이상을 지원하는 목적지 노드에 대한 경로가 없을 경우에만 시작된다. 소스 노드는 QoS 지원형 서비스를 표시하기 위해 RES 필드를 1로 하고, 자신이 요구하는 최소 자원 값과 최대 자원 값을 R_{min} 필드와 R_{max} 필드에 저장하고 자신이 제공할 수 있는 유용한 자원이 R_{min} 이상이면 PAR 필드에 저장하고 RREQ-S를 자신의 이웃 노드에게 브로드캐스트한다. RREQ-S를 수신한 노드가 목적지 노드인지 목적지 노드가 아닌지에 따라 구분하여 다음과 같은 단계를 수행한다. RREQ-S를 수신한 노드가 목적지 노드가 아닌 경우에는 다음과 같다.

첫 번째, 수신된 RREQ-S가 노드의 <소스 주소, 요구 식별자> 목록에 포함되어 있는지를 확인하고, 목록에서 발견되면 RREQ-S를 폐기한다.

두 번째, 수신 노드의 <소스 주소, 요구 식별자> 목록에 없는 경우, 노드는 RREQ-S 상의 경로 레코드 내에 자신의 주소가 있는지를 확인한다. 만약, 자신의 주소가 발견된 경우는 루프로 간주해 폐기한다. 이 과정은 소스 노드에서만 수행된다.

세 번째, RREQ-S 상의 경로 레코드에서 자신의 주소가 발견되지 않은 경우 자신의 경로 캐쉬에 목적지 노드까지의 경로가 저장되어 있고 그 경로의 유용한 자원 값이 R_{min} 이상인 경우는 목적지 노드까지 RREQ-S를 유니캐스트한다. 완전한 경로의 유용한 자원 값은 현재 노드에서 목적지 노드까지의 경로의 유용한 자원 값과 현재 RREQ-S의 PAR 필드를 비교해서 적은 값이 유용한 자원 값이 된다.

네 번째, 목적지 노드까지의 경로가 없고 또는 경로가 있더라도 유용한 자원 값이 R_{min} 미만인 경우에는 자신의 이웃 노드로 브로드캐스트한다.

RREQ-S를 수신한 노드가 목적지 노드인 경우에는 다음과 같다.

첫 번째, RREQ-S를 처음으로 수신한 목적지 노드는 RREQ-S 안의 순방향 경로를 RREP-D에 저장하고 역방향 경로를 발견하기 위해서 RREQ-D를 소스 노드로 브로드캐스트한다. RREQ-D에는 RREP-D가 피기백(piggyback)되어진다.

두 번째, RREQ-D를 수신한 소스 노드는

RREP-D에 저장되어 있는 순방향 경로를 자신의 경로 캐쉬에 저장하고 RREQ-D에 저장되어 있는 역방향 경로를 RREP-S에 저장하여 순방향 경로를 이용해서 목적지 노드로 유니캐스트한다. 그림 1은

소스노드에서 목적지 노드로 RREQ패킷을 전송하였을 때 중간 노드와 목적지 노드에서 이를 수신한 경우와 목적지 노드에서 소스노드로 RREP 메시지를 전송하여 소스노드가 이를 수신하여 경로설정이 이루어지는 2가지의 관점에서 나타난 순서도이다.

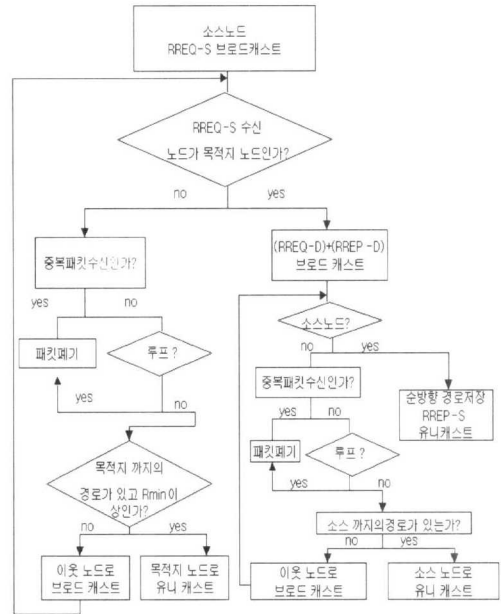


그림 1. QoS 지원형 서비스의 경로 발견

4.4 경로 유지와 통신 자원 평가

최선형 서비스와 QoS 지원형 서비스의 경로 유지를 위해서 경로 발견 과정에서 얻은 순방향 최단 경로와 역방향 최단 경로, 순방향 다중 경로 두 개와 역방향 다중 경로 두 개를 쌍으로 해서 소스 노드에서 목적지 노드로 주기적으로 헬로우 메시지를 보내고 헬로우 메시지를 수신한 목적지 노드는 다시 소스 노드로 확인 헬로우 메시지를 보낸다.

확인 헬로우 메시지가 일정 시간 안에 소스 노드로 돌아오면 경로가 유지되고 있고 돌아오지 않으면 순방향 경로와 역방향 경로가 모두 에러가 났다고 생각해서 두 경로를 모두 폐기한다.

그림 2는 헬로우 메시지에 의한 경로 유지를 개략적으로 나타낸다. 헬로우 메시지와 확인 헬로우 메시지에 의한 경로 유지와 경로 유용한 자원 값

유지 방법은 다음과 같다.

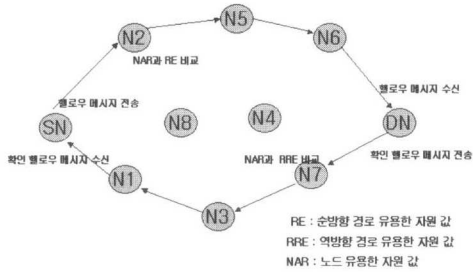


그림 2. 경로 유지

첫 번째, 소스 노드에서 목적지 노드로 헬로우 메시지를 보낼 때 헬로우 메시지의 PAR 필드에는 순방향 경로의 경로 유용한 자원 값과 RE 필드에는 소스 노드에서 유용한 자원 결정 과정을 통해 얻게 된 노드 유용한 자원 값 NAR을 저장하고 RPAR 필드와 RRE 필드에는 역방향 경로의 경로 유용한 자원 값을 저장한 다음 순방향 경로 상의 다음 노드로 유니캐스트한다.

두 번째, 헬로우 메시지를 수신한 다음 노드는 유용한 자원 결정 과정을 통해 얻은 노드 유용한 자원 값 NAR와 RE 필드에 저장되어진 값을 비교해서 NAR 값이 RE 값보다 적으면 RE 값을 NAR 값으로 대체한다. 이러한 과정은 목적지 노드에 도착할 때까지 반복된다.

세 번째, 헬로우 메시지를 수신한 목적지 노드는 헬로우 메시지의 PAR 필드와 RE 필드를 비교해서 서로 다르면 RE 필드 값을 자신의 경로 캐쉬에 있는 순방향 경로의 유용한 자원 값으로 대체하고 RPAR 필드와 RRE 필드를 비교해서 다르면 역시 경로 캐쉬의 역방향 경로 유용한 자원 값 필드를 RRE 필드 값으로 대체한다. 그리고 헬로우 메시지의 PAR 필드와 RE 필드를 확인 헬로우 메시지의 PAR 필드와 RE 필드에 복사하고 RPAR 필드에 역방향 경로의 유용한 자원 값을 저장하고 노드 유용한 자원 결정 과정을 통해 얻게 된 노드 유용한 자원 값 NAR를 RRE 필드에 저장한 후에 역방향 경로 상의 다음 노드로 유니캐스트한다.

네 번째, 확인 헬로우 메시지를 수신한 다음 노드는 노드 유용한 자원 결정 과정을 통해 얻은 NAR 값과 RRE 필드에 저장되어진 값을 비교해서 NAR 값이 RRE 값보다 적으면 RRE 값을 NAR 값으로 대체한다. 이러한 과정은 소스 노드에 도착할 때까지 반복된다.

다섯 번째, 확인 헬로우 메시지를 일정한 시간 이내에 수신한 소스 노드는 순방향 경로와 역방향 경로가 유지되고 있음을 확인하고 확인 헬로우 메시지의 PAR 필드와 RE 필드를 비교해서 서로 다르면 RE 필드 값을 자신의 경로 캐쉬에 있는 순방향 경로의 유용한 자원 값으로 대체한다. 또한 RPAR 필드와 RRE 필드를 비교해서 다르면 역시 경로 캐쉬의 역방향 경로 유용한 자원 값 필드를 RRE 필드 값으로 대체한다.

V. 결과 및 고찰

5.1 실험 환경

본 논문에서 라우팅 프로토콜에 대한 성능을 평가하기 위해서 미국 버클리 대학, 남가주 대학 및 카네기 멜론 대학에서 개발한 네트워크 시뮬레이터인 NS-2 및 CMU 확장 프로그램을 사용했으며, CMU 확장 프로그램에 본 논문에서 제시한 내용을 코드로 구현한 후, 성능평가를 수행하였다. 한편 성능평가를 위해 700MHz 펜티엄 III, 128MB RAM 및 13GB 하드디스크를 갖춘 PC가 사용되었다.

5.2 성능 평가 및 파라미터

평균 데이터 수신율은 소스 노드에서 전송되는 데이터 패킷과 목적지 노드에서 수신되는 데이터 패킷의 비율로서 정의한다. 평균 데이터 수신율에 영향을 미치는 요소는 다음과 같다.

첫째, 라우팅 시간 지연에 의해 전송 버퍼에서 데이터 패킷이 손실된다.

둘째, 패킷을 전송한 후에 패킷 충돌을 감지하는 웨이브LAN(WaveLAN)의 사용에 의해 패킷들의 충돌에 의해 데이터 패킷이 손실된다.

셋째, 노드에서의 트래픽 혼잡으로 IFQ(Inter Face Queue)에서 데이터 패킷이 손실된다.

넷째, 잘못된 경로 유지에 의해 데이터 패킷이 손실된다.

QoS 지원형 서비스는 라우팅 시간 단축과 정확한 경로 유지 문제와는 관련이 적고 대신에 통신 자원 할당과 IFQ 우선 순위를 높여 줌으로써 데이터 패킷 충돌과 트래픽 혼잡을 줄여 주는 것이다.

5.3 성능 평가

성능 평가를 위해 이동 패턴 파일과 통신 패턴 파일을 먼저 분류하였다. 이동 패턴 파일은 노드들의 평균 정지 시간이 0초와 120초로 분류하고, 통신 패턴 파일은 50개의 노드들 중에서 30개의 노드

들이 데이터 전송을 요구하는 것으로 분류했다.

성능 평가 환경은 노드가 50개, 노드 최고 이동 속도 20m/s, 네트워크 범위 1500m × 300m 및 1초에 4개 패킷이 소스 노드에 공급되는 형태를 사용했다.

성능 평가 파라미터인 평균 데이터 수신율은 최선형과 QoS 지원형 서비스를 구분하여 평가하였다. 30개 흐름의 평균 데이터 수신율과, 특정 흐름의 데이터 수신율, 30개 흐름에서 특정 흐름을 뺀 나머지 흐름의 평균 데이터 수신율을 평가하여 서로 비교하였다.

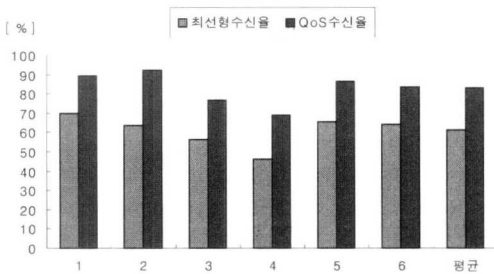


그림 3. 평균 데이터 수신율

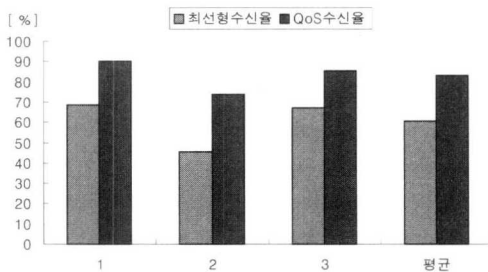


그림 4. 평균 데이터 수신율

그림 3와 그림 4는 노드 평균 정지 시간 0초, 흐름 30개 및 QoS 지원형 서비스 흐름 수 1개와 2개에 대해 900초 동안 최선형 서비스와 QoS 지원형 서비스로 나누어 성능 평가한 결과를 나타낸다.

그림 3에서 X축의 인자 1은 (1, 2) 흐름, 2는 (3, 4) 흐름, 3은 (10, 11) 흐름, 4는 (20, 21) 흐름, 5는 (27, 28) 흐름, 6은 (32, 33) 흐름을 나타내고 그림 4에서 X축의 인자 1은 (1, 2) 흐름과 (3, 4) 흐름 쌍, 2는 (10, 11) 흐름과 (20, 21) 흐름 쌍, 3은 (27, 28) 흐름과 (32, 33) 흐름의 쌍을 나타낸다. 예를 들어 (1,2)흐름은 노드 1에서 노드 2로 데이터를 전송하는 흐름을 나타낸다. 두 그래프에서 공통적으로 평균은 6개 흐름에 대한 평균을 나타낸다.

그림 3에서 QoS 지원형 서비스를 수행한 결과 최선형 평균 데이터 수신율은 급격히 떨어지고 QoS 지원형 평균 데이터 수신율은 약간 증가하는 양상을 보인다. (3, 4) 흐름이 다른 흐름들에 비해 평균 데이터 수신율이 현저하게 증가하는데 트래픽 혼잡이 많이 발생하는 (3, 4) 흐름을 제공하는 경로에 IFQ 우선 순위를 높임으로써 보다 많은 수신율의 향상을 얻을 수 있었다. 이것은 (3, 4) 흐름이 QoS 지원형 환경에 좀 더 적합한 흐름이라는 것을 알 수 있다. (20, 21) 흐름도 다른 흐름들에 비해서는 QoS 지원형 환경에 좀 더 적합한 흐름으로 평가된다.

그림 4에서 (1, 2) 흐름과 (3, 4) 흐름의 쌍이 다른 흐름들에 비해 평균 데이터 수신율이 현저하게 증가하는 데 이것은 그림 3에서와 같이 (1,2) 흐름이 다른 흐름들에 비해서는 QoS 흐름에 적합하기 때문이다.

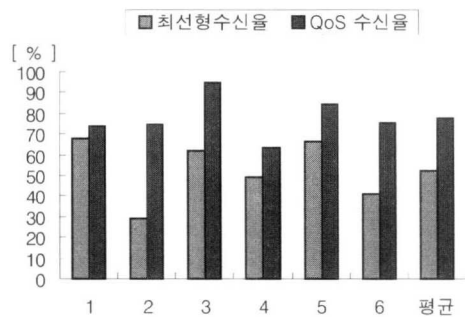


그림 5. 평균 데이터 수신율

그림 5와 그림 6은 노드 평균 정지 시간 120초, 흐름 30개 및 QoS 지원형 서비스 흐름 수 1개와 2개에 대해 900초 동안 최선형 서비스와 QoS 지원형 서비스로 성능 평가한 결과를 나타낸다.

그림 5에서는 (3, 4) 흐름이 다른 흐름들에 비해 평균 데이터 수신율이 현저하게 증가하는 데 이것은 (3, 4) 흐름이 다른 흐름보다 QoS 지원형 환경에 좀 더 적합한 흐름이기 때문이다.

그림 6에서는 (1, 2)흐름과 (3, 4) 흐름의 쌍이 다른 흐름들에 비해 평균 데이터 수신율이 현저하게 증가하는 데 트래픽 혼잡이 많이 발생하는 (1, 2)흐름과 (3, 4) 흐름을 제공하는 경로에 IFQ 우선 순위를 높임으로써 보다 많은 수신율의 향상을 얻을 수 있었다. 이것은 그림 5에서와 같이 (3, 4) 흐름이 다른 흐름들에 비해서는 QoS 지원형 환경에 좀 더 적합한 흐름이기 때문이다.

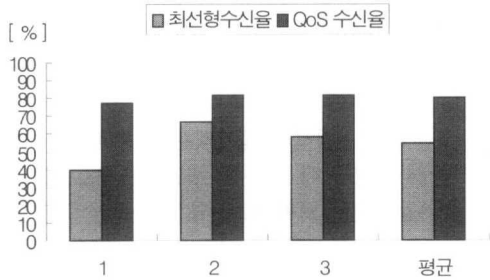


그림 6. 평균 데이터 수신율

VI. 결론

본 논문에서는 QoS를 지원하는 단방향 이동 애드혹 라우팅 프로토콜과 선택된 경로의 통신 자원 변화와 에러에 신속하게 적응할 수 있는 적응형 QoS 지원 알고리즘을 제안했다.

단방향과 양방향 링크가 혼재하는 이동 애드혹 네트워크 환경에서 모든 링크가 양방향 링크라고 가정하고 라우팅 프로토콜을 설계하는 것은 선택된 경로에 단방향 링크가 존재하는 경우에는 선택된 경로를 통해서 데이터 전송 서비스가 불가능하다. 또한, 최선형 서비스만을 지원할 수 있는 기존 라우팅 프로토콜로서 요즘 증가하고 있는 멀티미디어 데이터와 실시간 데이터의 전송을 지원하기에는 한계가 있다. 따라서, 단방향 링크와 QoS를 지원하는 제한한 라우팅 프로토콜은 대단히 유용한 프로토콜로 평가된다.

제한한 라우팅 프로토콜은 소스 노드와 목적지 노드사이의 순방향 경로와 역방향 경로를 독립적으로 설정함으로써 단방향 링크를 지원하고 노드들의 유용한 통신 자원을 고려하여 최단 경로를 설정함으로써 QoS를 지원한다. 한편, 이동 애드혹 네트워크에서 라우팅 프로토콜의 수행 결과 선택된 경로는 경로를 구성하는 노드들의 이동과 노드들 사이의 단방향 링크에 의해 에러가 발생할 수 있으며 선택된 경로의 유용한 통신 자원은 시간에 따라 변화될 수 있다. 이러한 환경에서 경로 에러와 경로의 유용한 자원 변화에 신속하고 정확하게 대처할 수 있는 헬로우 메시지를 사용하는 적응형 QoS 지원 알고리즘은 대단히 효율적인 것으로 평가된다.

참고 문헌

[1] Elizabeth M. Royer and et. al., "A Review of

Current Routing Protocols for Ad Hoc Mobile Wireless Networks," IEEE Personal Communication, pp.46-55, Apr. 1999.

- [2] Tony Larsson, and Nicklas Hedman, *Routing Protocols in Wireless Ad-hoc Networks : A Simulation Study*, Master's thesis, Lulea University of Technology, Stockholm, 1998.
- [3] R. Pandya, and et al., "IMT-2000 standard : Network aspects," IEEE Personal Communication, pp.20-29, Aug. 1997.
- [4] S.Lee and A. Campbell, "INSIGNIA: In-band Signaling Support for QOS in Mobile Ad Hoc Networks," Proc of 5th International Workshop on Mobile Multimedia Communications(MoMuC, 98), Berlin, Germany, Oct. 1998.
- [5] R. Comerford, "State of the Internet : Roundtable 4.0", IEEE Spectrum, Oct. 1998.
- [6] C. Huitema, *Routing in the Internet*, Prentice Hall, 1995.
- [7] Mc. R. Lin and M. Gerla, "A distributed architecture for multimedia in a multihop dynamic packet radio network," IEEE Globecom'95, pp.1468-1472, Nov. 1995.
- [8] C. E. Perkins, "Mobile Ad Hoc Networking Terminology," draft-ietf-manet-term-01.txt, internet draft, Nov.1998.
- [9] IEEE Computer Society LAN/MAN standards Committee, "Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) Specifications," IEEE Std. 802.11, 1997.
- [10] Chander Dhawan, *Mobile Computing*, McGraw-Hill, pp.149-160, 1997.
- [11] Stephen E. Deering and Robert M. Hinden, "Internet Protocol, version 6 (IPv6) Specification," RFC 2460, Dec. 1998.

강경인(Kyongl-In Kang)

정회원



1994년 2월 : 명지대학교 전자공학과졸업

1996년 2월 : 명지대학교 전자공학과석사

2001년 6월:명지대학교 전자공학박사

1998년 3월~현재 : 여주대학 정

보통신과 조교수

<주관심 분야> Ad hoc 네트워크, 컴퓨터시스템, IPV6

박 경 배(Kyong-Bae Park)

정회원



1994년 2월 : 명지대학교 전자공학과 졸업

1996년 2월 : 명지대학교 전자공학과 석사

2002년 2월 : 명지대학교 전자공학과 박사

1998년 3월 ~ 현재 : 여주대학 컴퓨터 사이언스와 조교수

<주관심 분야> Ad hoc 네트워크, 컴퓨터시스템, IPV6

유 충 렬(Choong-youl You)



1988년 2월 : 명지대학교 전자공학과 졸업(공학사)

1991년 2월 : 명지대학교 일 반대학원 전자공학과 졸업 -전자공학전공(공학석사)

1996년 2월 : 명지대학교 일 반대학원 전자공학과 졸업

-통신정보시스템전공(공학박사)

1994년 3월 ~ 1999년 11월 : 여주대학 정보통신과 조교수

2000년 4월 ~ 현재 : 동아방송대학 인터넷방송과 겸임교수(안성시 타운뉴스 Agent 대표)

<주관심 분야> 정보통신, 정보기술, 웹캐스팅, 멀티미디어, 컴퓨터네트워크 시스템.

정 찬 혁(Chan-Hyeok Jung)

정회원



2000년 2월 : 명지대학교 전기전자공학부 졸업

2002년 2월 : 명지대학교 전자공학 석사

2002년 3월 ~ 현재 : 명지대학교 전자공학과 박사과정

<주관심 분야> Ad Hoc Network, Mobile IP, MPLS, Routing

김 현 옥(Hyun-Ug Kim)



1978년 2월 고려대학교 전자공학과 졸업(공학사)

1980년 2월 고려대학교 전자공학과 졸업(공학석사)

1987년 2월 고려대학교 전자공학과 졸업(공학박사)

1980년 3월 ~ 1981년 2월 동양공업전문대학 전자과 전임강사

1981년 3월 ~ 1988년 1월 명지대학교 전자공학과 교수

1988년 2월 ~ 1990년 1월 Dept. of Computer Science of Arizona State University Adjunct Faculty

1990년 3월 ~ 현재 명지대학교 전자공학과 교수

<주관심 분야> 이동 무선 인터넷망, 멀티미디어, 컴퓨터 구조, 보안

이 광 배(Kwang-Bae Lee)



1979년 2월 : 고려대학교 전자공학과 졸업(공학사)

1981년 2월 : 고려대학교 전자공학과 졸업(공학석사)

1981년 3월 ~ 1982년 3월 : 삼성반도체

1982년 3월 ~ 1983년 4월 : 금성사

1984년 ~ 1986년 : Univ. of Southern California, Computer Engineering 전공(공학석사)

1986년 ~ 1991년 : Arizona state Univ., Computer Engineering 전공(공학박사)

1992년 3월 ~ 1998년 1월 : 명지대학교 전자공학과 교수

1998년 2월 ~ 1999년 1월 : Worchester Polytechnic Institute 대학 WINLAB 연구실 방문 연구교수

1999년 2월 ~ 2000년 1월 : Columbia 대학 COMET 연구실 방문 연구 교수

2000년 2월 ~ 현재 : 명지대학교 전자공학과 교수

<주관심 분야> 이동 무선 인터넷망, 멀티미디어, 컴퓨터 구조