

모바일 애드 혹 네트워크에서 링크 단절 예측을 사용한 지역 수정 라우팅 알고리즘

준회원 유 대 훈*, 정회원 최 응 철**

Local Repair Routing Algorithm using Link Breakage Prediction in Mobile Ad Hoc Networks

DaeHun Yoo* Associate Member, WoongChul Choi* Regular Member

요 약

무선 모바일 애드 혹 네트워크에서 경로를 효율적으로 관리하기 위해 많은 라우팅 알고리즘이 연구되어 왔다. 여기서 요구 기반 방식의 AODV 라우팅 알고리즘은 경로를 유지 하는데 있어서 Hello 메시지를 주기적으로 전송하여, 경로의 상태를 검사하게 된다. 그리고 경로가 단절 되었을 때, 그것을 인지한 노드가 송신지 노드에게 RERR 패킷을 전송하거나 RREQ 패킷을 지역적으로 전송함으로써 경로를 재설정하게 된다. 하지만 RREQ와 같은 제어 패킷을 플러딩 하기 때문에 대역폭의 낭비와 데이터 전송의 지연도 발생하게 된다. 본 논문에서는 이러한 문제들을 해결하기 위하여 기존의 AODV (Ad-Hoc On-Demand Distance Vector) 라우팅 알고리즘을 기반으로 링크의 상태를 예측하여 경로를 수정하는 LRRLBP 알고리즘을 제안한다. 여기서 링크의 상태는 주기적으로 수신 받는 Hello 메시지를 통한 수신 세기를 이용한다. 그리고 이러한 링크의 상태를 모니터링하여 경로가 단절되기 전에 지역적으로 경로를 수정한다. 제안된 알고리즘은 ns-2 시뮬레이터 사용하여 성능을 검증하였다.

Key Words : Routing algorithm, AODV, Link breakage prediction, Local repair, Signal strength

ABSTRACT

A number of routing algorithms have been studied for wireless mobile ad-hoc network. Among them, the AODV routing algorithm with on-demand method periodically transmits hello message and monitors link state during data transmission in order to maintain routing paths. When a path is disconnected, a node that senses it transmits a RERR packet to the transmitting node or transmits a RREQ locally so that the path could be repaired. With that, the control packet such as a RREQ is broadcast, which causes the consumption of bandwidth and incurs data latency. This paper proposes a LRRLBP algorithm that locally repairs a path by predicting link state before disconnecting the path based on the AODV routing protocol for solving such problems. Intensive simulations with the results using NS-2 simulator are shown for verifying the proposed protocol.

I. 서론

무선 모바일 애드 혹 네트워크에서 경로를 설정

하고 유지하는 라우팅 알고리즘은 데이터를 멀티홉으로 전송하는데 있어서 매우 중요한 기술이다^[1]. 모바일 애드 혹 네트워크에서 라우팅 알고리즘은

※ 본 연구는 교육인적자원부에서 지원받은 2006년 광운대학교 대학특성화사업(차세대 신성장 동력산업을 위한 실감 IT 전문인력 양성 사업)의 일환으로 진행되었으며 이에 감사를 드립니다.

* 광운대학교 컴퓨터 과학과 (yo2dh@kw.ac.kr), ** 광운대학교 컴퓨터 과학과 (wchoi@kw.ac.kr)

논문번호 : KICS2007-05-213, 접수일자 : 2007년 5월 11일, 최종논문접수일자 : 2007년 5월 11일

라우팅 테이블 업데이트 방식에 따라 테이블 기반 (Table-Driven) 방식과 요구 기반(On-Demand) 방식으로 분류된다. 테이블 기반 방식은 데이터 전송과 상관없이 주기적으로 주변 노드들과 라우팅 테이블 정보를 주고받는 구조로 되어 있다. 여기서 변경된 라우팅 테이블 정보만 전송할 수도 있으며, 네트워크에 새로 추가된 노드를 위해 전체 라우팅 테이블 정보를 전송할 수도 있다. 일반적으로 모든 노드들이 네트워크의 모든 라우팅 정보를 가지기 때문에 데이터를 전송할 필요가 있을 때 바로 전송할 수 있는 장점을 가지고 있다. 하지만 노드의 수가 많은 경우 라우팅 테이블을 완성하기까지의 많은 시간과 메모리가 필요하다. 또한 노드의 이동성이 많은 경우에 라우팅 테이블의 정보가 빈번하게 갱신되기 때문에, 데이터를 전송하지 않는 노드도 라우팅 패킷에 대한 대역폭의 낭비가 생기는 단점을 가지고 있다. 또 다른 방식인 요구 기반 방식은 데이터의 전송이 필요한 노드가 경로를 탐색하여 설정하는 구조로 되어 있다. 이 방식은 필요에 따라서 라우팅 정보를 얻기 때문에 테이블 기반 방식과 비교해서 데이터 전송의 지연이 생기는 단점을 가지고 있다. 하지만 필요에 의한 경로에 대해서만 유지를 하기 때문에, 관리하기 위한 경로의 수가 적어 경로 수정에 대한 제어 패킷 수를 최소화 할 수 있다. 또한, 네트워크 확장성에 있어서 라우팅 테이블에 대한 메모리 문제나 대역폭의 낭비를 줄일 수 있는 장점을 가지고 있다. 이러한 요구 기반 방식의 라우팅 알고리즘 중에 AODV (Ad Hoc On-Demand Distance Vector) 라우팅 알고리즘은 모바일 애드혹 네트워크 환경에서 대표적인 알고리즘이다²⁾. 하지만 AODV 라우팅 알고리즘은 경로가 단절된 이후에 경로를 재설정하는 메커니즘을 가지고 있는데, 이것은 많은 제어 패킷으로 인한 대역폭의 낭비가 발생하고, 데이터 전송 지연이 생기는 단점을 가지고 있다. 그래서 이러한 문제점들을 보완하기 위해 AODV 라우팅 알고리즘을 기반으로 많은 연구가 이루어지고 있다^{6,7,8,9)}. 이 연구들은 노드의 이동성을 고려하거나, 노드의 남은 에너지량을 고려하여 경로를 선택하거나 수정하도록 되어 있다. 그리고 경로가 단절되기 전에 이를 미리 예측하고 다른 경로로 수정하여 제어 패킷으로 인한 대역폭의 낭비와 전송 지연에 대한 문제를 완화시킬 수 있도록 하였다. 이러한 연구를 바탕으로 하여 본 논문에서는 경로가 단절되는 중요한 요인이 되는 노드의 이동성을 고려하여 경로의 상태를 예측하고, 경로가

단절되기 전에 다른 경로로 수정하는 LRRLBP (Local Repair Routing Algorithm using Link Breakage Prediction)를 제안한다. 이렇게 제안한 알고리즘은 AODV와 비교하여 중단 간 처리량의 향상과 더 적은 전송 지연 시간을 보여준다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 살펴본다. 3장에서는 문제점을 제시하고 성능 개선 방안으로 LRRLBP 을 제안한다. 4장에서는 ns-2 를 통한 실험 결과에 대한 성능을 분석한 후 5장에서 결론으로 마친다.

II. 관련 연구

AODV 라우팅 알고리즘과 같은 요구 기반방식은 경로 탐색과 경로 유지 단계로 구성된다. 데이터 전송이 필요한 노드가 목적지 노드까지 라우팅 테이블에 정보를 가지고 있지 않다면, RREQ 패킷을 플러딩(flooding)하여 목적지 노드를 탐색한다. 그래서 목적지 노드가 RREQ 패킷을 수신하게 되면 송신 노드에게 RREP 패킷을 유니캐스트(unicast)로 전송하여 경로를 설정한다. 하지만 이러한 경우 설정된 경로가 최단 홉이라는 보장도 없고, 경로의 상태를 검사할 수 있는 방법은 없다. 그래서 다수의 RREQ 패킷을 수신하는 특성을 활용하여 보다 효율적인 경로를 선택하는 알고리즘이 제안되었다.

LPR 같은 경우 DSR을 기반으로 만들어진 알고리즘이다^{3,4)}. 이것은 경로를 선택하기 위해 예측된 생존 시간 값을 매트릭으로 사용하였다. DSR 프로토콜은 요구 기반 방식으로 AODV와 비슷하지만 목적지 노드까지 경로의 모든 중간 노드를 송신 노드가 가지고 있다는 것이 다르다. 이것은 RREQ 패킷을 플러딩 하면서 중간 노드들의 남아있는 에너지 양과 에너지 소비율을 이용하여 경로의 생존시간을 계산한다. 그리고 경로를 선택할 때 예측된 생존 시간이 가장 긴 경로를 선택한다.

PEER은 노드의 에너지 소비를 매트릭으로 사용하였다⁵⁾. 이 알고리즘에서도 홉 마다 전송할 때 에너지 소비를 계산하여 RREQ 패킷에 설정한다. 그리고 목적지 노드는 RREQ 패킷을 수신하면 일정 시간 동안 다른 RREQ 패킷을 기다린 후 가장 에너지 소비가 적은 노드로 경로를 선택하여 RREP 패킷을 전송한다. 하지만 RREP 패킷을 전송한 후에 에너지 소비가 적은 RREQ 패킷을 수신 할 수도 있기 때문에 RREP 패킷을 오버히어링(overhearing)하여 병렬적으로 빠른 경로의 수정이 가능하도록

제안하였다. 또한 PEER은 경로를 유지하기 위하여 RTS, CTS 프레임 혹은 브로드캐스트 패킷을 통하여 전송 파워와 링크 비용을 계산하여 경로를 삽입, 삭제, 대체 메커니즘을 제안하였다.

Link-Quality Aware AODV는 경로를 선정하기 위한 매트릭으로 홉의 수와 SNR(Signal to Noise Ratio)를 사용하여 하였다⁶⁾. Router Handoff 알고리즘은 신호 세기를 사용하여 중간 노드의 링크 상태를 검사하고, 경로가 단절되기 전에 다른 노드로 핸드오프 하는 메커니즘을 제안하였다⁷⁾. 이를 위해 주기적으로 수신한 Hello 메시지의 신호 세기를 NPL (Neighbor Power List)과 PDT(Power Difference Table)에 저장하였다.

PLRR 알고리즘은 GPS를 사용하여 노드의 위치와 이동 속도를 가지고 링크 상태를 예측하여 경로를 수정하는 방법을 사용하였다⁸⁾. 하지만 GPS의 사용은 비용, 크기의 문제점과 더불어 실내에서 사용할 수 없다는 단점을 가지고 있다.

AODV-LRQ(Local Repair Quota)와 AODV-LRT(Local Repair TTL) 알고리즘은 각각 지역적 경로 수정 과정에서 RREQ 패킷을 재전송하는 수와 TTL의 값을 적응성 있게 결정하는 방안을 제안하였다⁹⁾.

III. 문제 제시

LPR, PEER과 같은 경로를 선택하는 알고리즘은 패킷 전송 지연에 대한 문제점을 가지고 있다. 기존의 요구 기반 방식은 데이터를 전송하기 전에 경로를 탐색하기 때문에 패킷 전송 지연이 더 발생하는 문제가 발생한다. 또한 경로가 하나 밖에 없다면 일정 시간 동안 기다리는 것은 무의미한 지연이 된다. 그림 1은 송신 노드 S가 목적지 노드 D에게 데이터를 전송하고자 할 때 경로를 설정하는 과정이다. 여기서 링크의 매트릭을 숫자로 표시하였다. 먼저

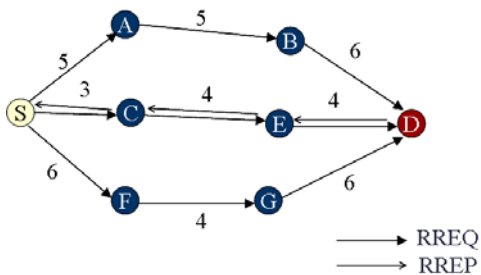


그림 1. 경로 선택 메커니즘
Fig. 1 Route selection mechanism

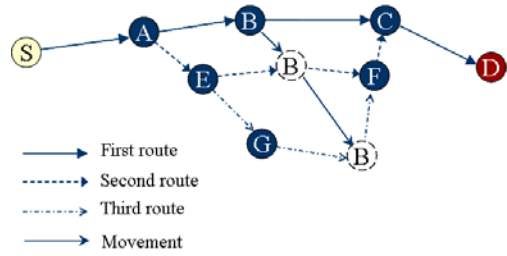


그림 2. 경로 삽입 작업
Fig. 2 Route insertion operation

송신 노드는 RREQ 패킷을 플러딩 한다. 그러면 목적지 노드 D는 처음의 RREQ 패킷을 수신 받는다. 기존의 AODV 라우팅 알고리즘은 처음에 받은 RREQ 패킷을 수신한 경로를 선택하고 RREP 패킷을 즉시 응답한다. 하지만 경로를 선택하기 위해서 목적지 노드는 다른 경로의 RREQ 패킷을 기다리게 된다. 그 이후 또 다른 RREQ 패킷을 전송 받으면 기존의 RREQ 패킷의 매트릭 값과 현재 받은 RREQ 패킷의 매트릭 값을 비교하여 경로를 선택한다. 일정 시간이 지나면 현재 경로 중 가장 좋은 매트릭이라고 선택된 경로로 RREP 패킷을 전송한다. 이 때, 목적지 노드가 기다리는 시간은 송신 노드의 데이터 전송의 지연을 가져오는 문제의 원인이 된다. 나중에 수신한 RREQ 패킷의 매트릭이 먼저 수신한 RREQ 패킷의 매트릭 보다 좋다는 보장도 없기 때문에 불필요한 지연이 생길 확률이 높아진다.

또한 PEER은 경로의 삽입, 삭제, 대체 메커니즘을 제공하는데, 이 메커니즘은 경로가 설정된 중간 노드에서 경로가 수정된다. 이것은 다음과 같은 문제점을 가지고 있다. 그림 2를 살펴보면, 먼저 송신 노드 S에서 목적지 노드 D까지 S->A->B->C->D로 경로가 설정되어 있다. 여기서 노드 B가 이동함에 따라 A->B에서 A->E->B로 경로 삽입 과정이

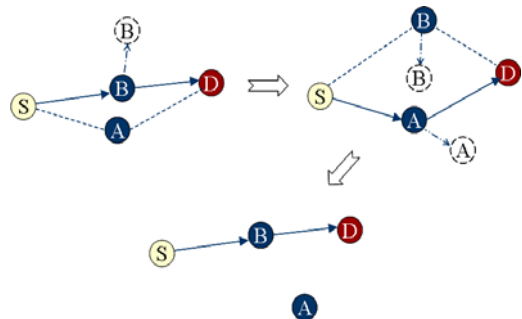


그림 3. 지그재그한 이동
Fig. 3 Zigzag movement

수행된다. 또한 경로 B->C에서 B->F->C로 경로가 수정된다. 이렇게 되면 경로가 S->A->B->C->D에서 S->A->E->B->F->C->D로 홉의 수가 4에서 6으로 늘어나게 된다. 그 이후에 노드 B가 다시 이동함에 따라 경로 E->B가 경로 E->G->B로 수정된다. 이렇게 되면 처음에 홉의 수가 4에서 7로 늘어나게 된다. PEER 알고리즘은 경로를 수정할 수 있는 홉의 수를 2로 제한하였다. 그 이유는 경로를 수정하는 영역이 넓어지면, 경로 수정 과정이 너무 많이 수행되어 성능 저하가 발생하기 때문이다. 하지만 이 때문에 홉의 수가 7로 늘어난 상황에서 원래 노드 B의 자리에 새로운 노드가 나타났다면 하더라도 경로를 수정할 수가 없게 된다.

Router Handoff 알고리즘은 제안된 LRRLBP와 같이 신호 세기를 사용하여 핸드오프를 수행한다. 이것은 PDT와 NPL를 통해서 이루어지는데 이 값들은 패킷을 통해 얻은 최신의 값과 바로 전 패킷에서 얻은 값을 통해서 구성된다. 하지만 노드의 움직임은 정해진 규칙이 없기 때문에 거리의 값은 선형적이지 않다. 그림 3의 송신 노드 S는 노드 B를 경유하여 목적지 노드 D까지 경로가 설정되어 있는 상황이다. 다음에 노드 B는 그림과 같이 위로 이동한다. 그렇게 되면 신호 세기는 약해지게 되고, 노드 A로 핸드오프 하게 된다. 하지만 노드 B는 지그재그하게 이동하기 때문에 원래의 자리로 돌아오고, 노드 A는 다른 곳으로 이동한다. 그러면 다시 노드 A에서 노드 B로 핸드오프 과정이 수행된다. 이것은 NPL과 PDT가 현재의 신호 세기 값과 이전의 신호 세기 값을 사용하기 때문에 민감하게 경로가 수정될 수 있음을 보여준다. 노드 B는 경로가 단절되기 전에 원래 자리로 돌아옴으로써 경로 상태는 좋아지게 된다. 만약 이 경우에 노드 B로 경로를 계속 유지 했다면, 두 번의 경로 수정 과정은 수행될 필요가 없다.

이러한 문제들을 해결하기 위해, 본 논문에서는 먼저 경로를 선택하는 과정에서 매트릭을 사용하여 목적지 노드에서 선택하지 않았다. 그 대신 RREQ를 수신하여 플러딩을 수행할 때, 신호 세기에 따라 전송하지 않는 메커니즘과 경로가 단절되지 않도록 링크의 상태를 예측하여 효율적으로 경로를 수정하는 방안을 제안한다. 또한 경로를 삽입하는 과정이 중간 노드에서 수행하는 것이 아니라 목적지 노드 주변에서만 수행하도록 제한하여 발생할 수 있는 문제를 예방하였다. 그리고 노드의 지그재그한 움직임과 같이 이동성에 너무 민감하게 경로가 수정되지

않도록 SSS(Smoothed signal strength)를 제안 한다.

IV. LRRLBP 제안

LRRLBP는 AODV 기반 라우팅 알고리즘으로써 경로 탐색 과정과 경로 유지 단계로 구성된다. 그리고 경로 유지 단계에서 노드의 이동성을 고려하여 경로의 상태를 예측하고, 경로가 단절되기 전에 이웃 노드들이 경로를 수정하는 라우팅 알고리즘이다. 이것은 경로의 재탐색의 수를 줄이고, 제어 패킷의 감소를 가져온다. 또한 경로 유지 시간을 연장하여 데이터 전송 지연시간을 줄이는 효과를 가져 올 수가 있다.

4.1 패킷의 구조

AODV에서 이미 사용되었던 RREQ, RREP, RERR 패킷들은 동일하게 사용된다. 기존의 Hello 메시지는 RREP 패킷에서 값만 수정하여 사용하였는데, 그림 4와 같이 필요한 정보만으로 수정하였다. 이것은 경로가 설정되었을 때 주기적으로 전송하여 노드의 존재를 알릴 때 사용된다. 여기서 제안된

Type	Hop Count	Dst Address	Dst Sequence Number
------	-----------	-------------	---------------------

그림 4. Hello 메시지
Fig. 4 Hello message

Type	Hop Count	Dst Address	Dst Sequence Number	Node Address
------	-----------	-------------	---------------------	--------------

그림 5. 확장된 Hello 메시지
Fig. 5 Extended Hello message

Type	Subtype	Operation	Neighbor Src Address	Neighbor Dst Address
------	---------	-----------	----------------------	----------------------

그림 6. RREPAIR_REQ 패킷
Fig. 6 RREPAIR_REQ packet

Type	Subtype	Oper.	Dst Address	Dst Hop Count	Dst Seqno	Neigh Src Address	Neigh Dst Address
------	---------	-------	-------------	---------------	-----------	-------------------	-------------------

그림 7. RREPAIR_ACK 패킷
Fig. 7 RREPAIR_ACK packet

ID	SSS	T
Node ID	The Smoothed Signal Strength	Expire Time

그림 8. NSL
Fig. 8 NSL

LRRLBP는 추가로 확장된 Hello 메시지, RREPAIR_REQ, RREPAIR_ACK 가 사용된다. 그림 5의 확장된 Hello 메시지는 기존의 Hello 메시지에서 노드 주소(Node address) 필드만 추가된다. 노드 주소는 송신 노드 혹은 목적지 노드에서 경로 삽입이 필요한 노드가 발견되었을 때, 기존의 Hello 메시지 대신 확장된 Hello 메시지에 발견된 노드의 주소를 입력하여 전송한다. 그림 6의 RREPAIR_REQ 패킷은 경로 수정 요청 패킷으로 확장된 Hello 메시지를 수신한 노드가 경로 삽입이 가능하면 경로 수정을 요구하는 노드에게 전송한다. 즉, 해당 노드 주소 필드의 주소를 가지고 경로가 설정된 노드의 이웃 노드들이 경로의 상태를 검사하여 능동적으로 경로의 수정을 요청하는 패킷이다. 타입(Type)과 서브타입(Subtype) 필드는 패킷의 종류를 구분하기 위하여 사용되고, 오퍼레이션(Operation) 필드는 삽입, 삭제와 같은 작업의 종류를 구분한다. 이웃 송신 주소(Neighbor Src Address) 필드는 경로를 수정할 노드의 이전 노드 주소를 의미하고, 이웃 목적지 주소(Neighbor Dst Address) 필드는 수정할 노드의 다음 노드 주소를 의미한다. 그림 7의 RREPAIR_ACK 패킷은 경로를 실질적으로 수정하기 위해 사용되는 패킷이다. 타입, 서브타입, 오퍼레이션, 이웃 송신 노드, 이웃 목적지노드 필드는 RREPAIR_REQ 패킷과 동일하고, 목적지 주소(Dst Address), 목적지 홉의 수(Dst Hop Count), 목적지 시퀀스 번호(Dst Seqno) 필드는 라우팅 테이블을 적용하기 위해 사용된다.

4.2 테이블 구조

LRRLBP에서는 AODV 라우팅 알고리즘과 마찬가지로 라우팅 테이블과 이웃 노드의 정보를 담은 테이블을 사용한다. 여기서 이웃 노드의 정보를 담은 테이블을 확장한 NSL(Neighbor Smoothed Signal Strength List)을 사용하며, 이것은 그림 8과 같이 구성된다. Hello 메시지를 전송 받을 때 송신한 노드의 ID를 입력하고, 링크 계층에서 측정된 신호 세기 값을 사용하여 SSS 값을 계산하여 입력한다. 그리고 T 값으로 Hello 메시지를 받은 시간 기준으로 적당한 시간을 입력하여, 현재 시간이 T보다 클 경우 링크가 단절되었다고 판단하게 된다. 이것은 기본적으로 Hello 메시지 전송 주기 시간*3으로 설정된다. SSS 값은 Router Handoff 알고리즘과 같이 신호 세기를 사용하는데, 이 알고리즘은 현재와 이전의 신호 세기만을 사용하기 때문에 노드

의 이동성에 민감한 문제점을 가지고 있다. 그래서 LRRLBP에서는 이러한 문제점을 완화하기 위해 누적된 신호 세기를 사용하여 보다 이동성에 민감하지 않도록 SSS 값을 계산하여 사용한다. SSS 값을 구하는 수식은 다음과 같다.

$$SSS_n = \alpha SSS_{n-1} + (1-\alpha)S_n \quad (1)$$

여기서 n은 패킷을 전송 받은 수이고, S_n은 n번째 수신된 패킷의 신호 세기이다. 그리고 α는 0과 1 사이의 값으로써 0에 가까워지면 현재의 신호 세기의 값이 SSS 값에 영향을 덜 주게 되고, 1에 가까워질수록 현재 값이 SSS 값에 더 많은 영향을 주게 된다.

4.3 경로 탐색

송신 노드가 목적지 노드로 데이터 전송이 필요할 때, RREQ 패킷을 플러딩하여 경로를 탐색한다. RREQ 패킷을 수신한 노드는 패킷을 송신한 노드 주소를 라우팅 테이블에 입력한다. 만약 이전에 받은 RREQ 패킷이면 시퀀스 넘버를 비교하여 같거나 낮으면 폐기한다. 그리고 송신 노드의 주소로 NSL를 검사하여 해당되는 SSS 값을 Th_{limit_sss}과 비교한다. NSL에 송신 노드가 없다면, 현재 들어온 신호 SSS 값에 입력하여 비교한다. Th_{limit_sss}는 신호 세기가 약한 노드의 기준을 정하는 임계치 값이다. 만약 신호 세기가 약한 노드로 경로가 설정되었다면, 경로가 설정되었다 하더라도 경로가 단절될 확률이 높기 때문에 RREQ 패킷을 플러딩 하는 과정에서 경로가 설정되는 것을 막기 위함이다. 그래서 SSS 값보다 작다면, RREQ 패킷을 전송하지 못하도록 한다. 이렇게 함으로써 경로가 단절되는 확률을 줄인다. 여기서 송신 노드의 SSS 값이 작지 않다면, 계속 플러딩을 수행한다. 그리고 해당되는 목적지 노드가 RREQ 패킷을 받으면 라우팅 테이블에 입력하고 RREQ 패킷을 송신한 노드로 RREP 패킷을 전송한다. 여기서 RREP 패킷을 받은 노드는 라우

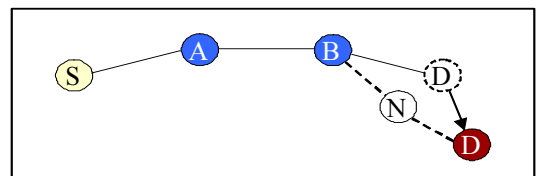


그림 9. 경로 삽입
Fig. 9 Route insertion

팅 테이블에 입력하고 RREP 패킷을 전달한다. 송신 노드가 RREP 패킷을 수신하면 라우팅 테이블에 정보를 저장하고 데이터를 전송할 수 있도록 한다.

4.4 경로 유지

경로가 설정이 되면, 각각의 경로가 설정된 노드들은 Hello 메시지를 주변 노드들에게 브로드캐스트한다. 이것을 수신한 주변 노드들은 NSL의 SSS 값과 T 필드 값을 갱신 한다. 그리고 주기적으로 NSL를 검사하여 만약 NSL의 T 필드 값이 현재 시간보다 작다면 NSL에서 삭제하고, 삭제된 노드의 주소가 라우팅 테이블의 다음 홉의 주소와 같은 것이 있으면 경로를 사용하지 못하도록 플래그의 값을 RTF_DOWN으로 설정한 후에 RERR 패킷을 프리커서(Precursor) 노드들에게로 브로드캐스트 한다. RERR 패킷을 수신한 노드는 라우팅 테이블에서 해당 노드의 정보를 삭제하고, 그 노드로 전송할 데이터가 있다면 경로 재탐색을 위해 RREQ 패킷을 다시 플러딩 한다. 추가적으로 LRRBP에서는 경로를 삽입, 삭제하는 메커니즘을 제공한다.

4.5 경로 삽입

경로 삽입은 라우팅 테이블의 다음 홉의 주소와 목적지 노드 주소가 같은 라우팅 정보를 가진 노드만이 수행할 수 있다. 이 조건이 성립한 노드는 목적지 노드의 SSS가 Th_{repair_sss} 보다 작는지 검사한다. Th_{repair_sss} 은 경로를 삽입하는 기준을 제시하는 임계치 값이다. 여기서 SSS 값이 Th_{repair_sss} 보다 작다면, Hello 메시지 대신에 확장된 Hello 메시지에 목적지 노드의 주소를 입력하여 브로드캐스트 한다. 이것을 해당 목적지 노드로 경로가 설정 되지 않은 주변 노드가 수신하면 확장된 Hello 메시지를 전송한 노드의 SSS 값과 NSL에서 목적지 노드의 SSS 값을 검색하여 $Th_{properly_sss}$ 보다 크지 검사한다. $Th_{properly_sss}$ 는 안전하게 통신할 수 있는 적절한 SSS 값을 의미한다. 두 노드의 SSS 값이 $Th_{properly_sss}$ 보다 크면, RREPAIR_REQ 패킷을 확장된 Hello 메시지를 송신한 노드에게 전송한다. RREPAIR_REQ 패킷의 이웃 송신 주소 필드에는 확장된 Hello 메시지를 전송한 노드의 주소를 입력하고, 이웃 목적지 주소 필드에는 확장된 Hello 메시지의 노드 주소 필드의 값을 입력한다. 이것을 수신한 노드는 해당 경로가 라우팅 테이블의 다음 홉 주소를 RREPAIR_REQ 패킷의 전송한 노드 주소로 수정하고 홉의 수를 증가한 후에 RREPAIR_ACK에 라우팅 테이블 정보를

입력하여 응답한다. 만약 다른 주변 노드의 RREPAIR_REQ 패킷을 수신하더라도 라우팅 테이블의 다음 홉 주소와 목적지 노드의 주소가 같지 않기 때문에 무시된다. RREPAIR_REQ 패킷을 전송한 노드가 RREPAIR_ACK를 수신하면 경로 정보를 라우팅 테이블에 저장한다. 예를 들어 그림 9는 S->A->B->D로 경로가 설정되어 있을 때, 목적지 노드 D가 이동하는 상황이다. 목적지 노드 D가 이동함에 따라 노드 D의 Hello 메시지를 수신한 노드 B는 SSS 값이 Th_{repair_sss} 보다 작아지게 되고 확장된 Hello 메시지에 노드 D의 주소를 입력하여 브로드캐스트 하게 된다. 그리고 이웃 노드 N이 이 패킷을 수신하게 되고, 노드 B와 노드 D의 SSS 값이 $Th_{properly_sss}$ 보다 크지 비교하게 된다. 이것이 성립되면 노드 N은 노드 B에게 RREPAIR_REQ 패킷을 전송하게 되고 노드 B는 라우팅 테이블의 목적지 노드 D에 해당하는 다음 홉의 주소를 노드 N의 주소로 설정하게 된다. 그리고 노드 N에게 RREPAIR_ACK 패킷에 해당되는 라우팅 정보를 입력하여 유니캐스트로 전송한다. RREPAIR_ACK를 수신한 노드 N은 라우팅 테이블 정보를 수정하게 되고, S->A->B->N->D로 경로가 수정되게 된다.

4.6 경로 삭제

경로가 설정된 노드가 Hello 메시지를 수신할 때 라우팅 테이블의 목적지 노드의 주소와 Hello 메시지를 전송한 노드의 주소가 같은지를 검사한다. 만약 주소가 같으면 해당되는 주소로 NSL의 SSS 값을 검색한다. 만약 목적지 노드의 라우팅 정보에서 홉의 수가 2보다 크고, SSS 값이 $Th_{properly_sss}$ 보다 크면 RREPAIR_ACK 패킷을 목적지 노드로 브로드캐스트하고, 라우팅 테이블의 홉의 수를 1로 설정하고 다음 홉의 주소를 목적지 노드 주소로 수정한다. RREPAIR_ACK 패킷을 수신한 노드 중에 같은 목적지의 라우팅 정보를 가지고 있다면, 해당 프리커서의 노드 주소를 검색하여 삭제한다. 이것은 경로 삭제를 한 노드에서 목적지 노드까지의 중간노드들이 RERR 패킷이 발생하였을 때, 경로 삭제를 한 노

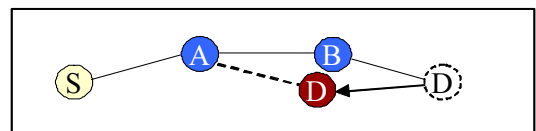


그림 10. 경로 삭제
Fig. 10 Route deletion

드의 주소가 오류로 처리됨을 막기 위함이다. 다음 그림 10은 S->A->B->D 로 경로가 설정되어 있을 때 목적지 노드 D가 이동하는 상황이다. 이 때 목적지 노드 D는 주기적으로 Hello 메시지를 브로드캐스트하게 되고, 이것을 수신 받은 노드 A는 라우팅 정보의 홉의 수가 2 이상이고 목적지 노드 주소가 같은지 검사하게 된다. 이것이 성립하기 때문에 노드 A는 라우팅 테이블에서 목적지 노드 D의 다음 홉을 노드 B에서 노드 D로 수정하고 홉의 수를 1로 설정한다. 그래서 최종적으로 S->A->D 로 경로가 수정됨으로써 홉의 수를 줄이고, 대역폭의 낭비와 데이터 전송 지연을 줄일 수가 있게 된다.

V. 성능 평가

5.1 실험 환경

본 논문에서 제안한 LRRLBP의 성능에 대하여 실험하기 위해 ns-2 시뮬레이터를 사용하였다[10]. 물리 계층은 DSSS (Direct-Sequence Spread -Spectrum) 를 사용하였고, 안테나는 전방향 안테나로 설정하였다. mac 계층은 IEEE 802.11 로 설정하였고, UDP의 CBR (Constant Bit Rate) 트래픽을 사용하였다. 전송 패킷의 크기는 512 bytes로 설정하였고, 전송 비율은 0.25 초로 설정하였다. 그리고 1200m x 1200m 범위 안에서 노드를 위치시켰다. 여기서 노드의 수를 30~80개 범위에서 10개 단위로 증가시키면서 랜덤하게 위치시키고, 접속 수는 노드의 수의 20%로 설정하였다. 노드의 최대 이동 속도는 6 m/s 로 설정하였고, 이동 대기 시간을 약 10초로 설정하였다. 그리고 실험 시간은 총 600초로 설정하였고, ns-2의 버전은 2.28를 사용하였다.

AODV 라우팅 알고리즘과 제안된 LRRLBP에서 SSS 값을 계산하는데 사용하는 α 값을 0.7, 0.8, 0.9, 1.0으로 설정하여 비교하였다. 환경 변수의 설정은 표 1과 같다. $Th_{properly_sss}$ 는 거리가 150m 되었을 때의 신호 세기의 값이고, Th_{repair_sss} 는 노드 간의 거리가 200m 되었을 때의 신호 세기의 값이고, Th_{remove_sss} 는 노드 간의 거리가 125m 되

표 1. 시뮬레이션 파라미터
Table 1. Simulation parameter

Item	Value
$Th_{properly_sss}$	$2.818 * 10^{-9}$
Th_{repair_sss}	$1.359 * 10^{-9}$
Th_{remove_sss}	$8.92 * 10^{-10}$
Th_{limit_sss}	$4.0 * 10^{-10}$

었을 때의 신호 세기의 값이다. 마지막으로 Th_{limit_sss} 는 거리가 약 240m 정도였을 때의 신호 세기의 값이다.

5.2 실험 결과

그림 11을 보면, 전반적으로 제안된 LRRLBP의 패킷 처리량이 더 높은 것을 알 수 있다. α 값이 1.0인 경우 노드의 움직임에 매우 민감하게 신호 세기가 변하기 때문에, 이것이 빈번한 경로 수정으로 이어져 처리량이 AODV 보다 좋지 못한 것을 볼 수 있다. 하지만 α 값이 0.7, 0.8, 0.9인 경우 보다 우수한 처리량을 보여준다.

그림 12는 패킷 드롭 수를 보여준다. LRRLBP는 RREQ 패킷을 SSS에 따라 필터링하여 전송하고, 경로가 단절되기 전에 경로를 수정하기 때문에 그만큼 패킷 드롭 수가 AODV 보다 적은 것을 확인할 수 있다.

그림 13에서 LRRLBP가 전반적으로 AODV 보다 평균적으로 종단 간 패킷 지연 시간이 짧다는 것을 보여준다. α 값이 0.7 일 때, 다소 좋지 않은 결과를 보여주는 것은 SSS 값의 변화량이 작기 때

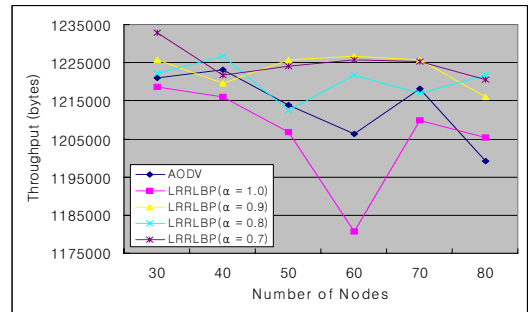


그림 11. 패킷 처리량 측정
Fig. 11 Packet throughput evaluation

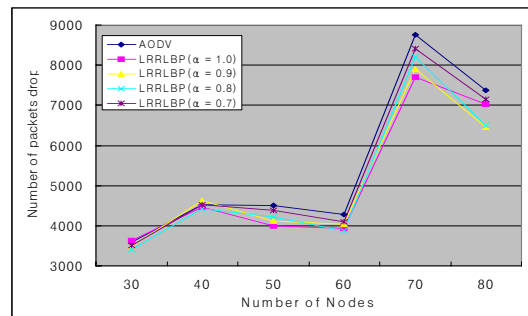


그림 12. 패킷 드롭 수 측정
Fig. 12 Number of packet drop evaluation

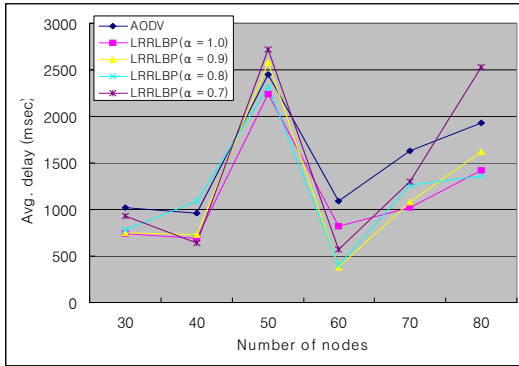


그림 13. 패킷 평균 지연 시간 측정
Fig. 13 Average packet delay time evaluation

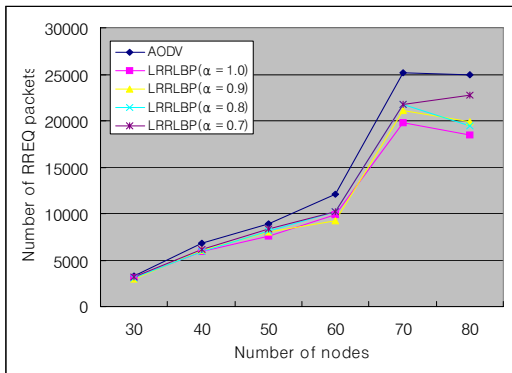


그림 14. RREQ 패킷 수 측정
Fig. 14 Number of RREQ packet evaluation

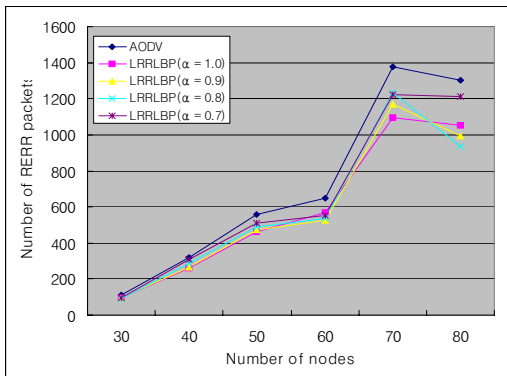


그림 15. RERR 패킷 수 측정
Fig. 15 Number of RERR packet evaluation

문에 보다 빠르게 경로를 수정하지 못해서 나오는 것으로 판단된다. 하지만 α 값이 0.8, 0.9 인 경우에는 AODV 보다 짧은 지연 시간을 보여준다.

그림 14와 15에서 알 수 있듯이, LRRLBP가

AODV보다 적은 제어 패킷의 수로 효율적으로 경로를 관리 한다는 것을 확인할 수 있다. 경로 탐색 과정에서 RREQ 패킷을 SSS 값으로 제한하기 때문에 RREQ 패킷이 AODV 보다 적은 수로 목적지를 발견한다는 것을 확인할 수 있다. 또한 경로가 단절되기 전에 경로를 수정하기 때문에 경로의 단절 횟수가 적어지게 되고, 이것은 보다 적은 RERR 패킷 전송량으로 이어지게 된다는 것을 확인할 수 있다.

VI. 결론

본 논문에서는 모바일 애드 혹 네트워크 환경에서 성능 향상을 위해 개선된 라우팅 알고리즘인 LRRLBP를 제안하였다. LRRLBP는 요구 기반 방식의 AODV 라우팅 알고리즘을 기반으로 한다. AODV 라우팅 알고리즘은 경로가 단절된 이후에 경로로 재탐색하는 과정을 가지는데 이것은 제어 패킷의 낭비로 인한 처리량의 저하를 가져온다. 또한 경로 재설정 과정 동안 패킷 지연이 발생하게 된다. 그래서 이를 완화하기 위해 제안된 LRRLBP에서는 신호 세기가 약한 노드에서 수신한 RREQ 패킷을 플러딩하지 않도록 제한하고, 경로의 상태를 예측하여 경로가 단절되기 전에 경로를 수정하는 방안을 제안하였다. 링크의 상태는 링크 계층에서 얻은 신호 세기를 사용한다. 신호 세기를 그대로 사용하면 이동성에 민감해 지기 때문에 경로가 빈번하게 수정되는 현상이 발생할 수 있다. 그러면 마찬가지로 패킷 지연이 발생하기 때문에 누적된 결과를 사용하여 계산한 SSS를 적용하도록 하였다. 본 논문에서는 제안된 LRRLBP의 성능을 증명하기 위해서 ns-2 시뮬레이터를 사용하여 성능을 검증하였다. 결과적으로 모바일 애드 혹 네트워크 환경에서 중단 간 처리량의 향상과 패킷 지연과 패킷 드롭 수의 감소를 확인 하였다. 이것은 본 논문에서 제안하고 있는 LRRLBP가 AODV 라우팅 알고리즘보다 개선된 방법이라는 것을 보여준다.

향후 보완해야할 연구과제로서는 SSS을 구하는 α 의 값을 네트워크 환경에 따라 적응성 있게 설정하는 연구를 통하여 모바일 애드 혹 네트워크의 성능 향상에 대한 방안을 제시할 수 있을 것이다.

참고 문헌

- [1] IETF MANET Working Group, <http://www.ietf.org/html.charters/manet-charter.html>.

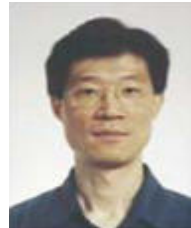
- [2] C. Perkins, E. Belding-Royer and S. Das, "Ad hoc On-demand Distance Vector(AODV) Routing," *RFC3561*, July 2003.
- [3] M. Maleki, K. Dantu, M. Pedram, "Lifetime prediction routing in mobile ad hoc networks," *WCNC 2003*, pp. 1185-1190, March 2003.
- [4] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad-hoc wireless networks," *Mobile Computing*, pp. 153-181, 1996.
- [5] Jinhua Zhu, Xin Wang, "PEER: A Progressive Energy Efficient Routing Protocol for Wireless Ad Hoc Networks," *IEEE INFOCOM 2005*, pp. 1887-1896, March 2005.
- [6] Hsin-Mu Tsai, N. Wisitpongphan, O.K. Tonguz, "Link-Quality Aware Ad Hoc On-Demand Distance Vector Routing Protocol," *Wireless Pervasive Computing, 2006 1st International Symposium on*, pp. 16-18, Jan. 2006.
- [7] P. Srinath, P. Abhilash, I. Sridhar, "Router handoff: A Preemptive Route Repair Strategy for AODV," *2002 IEEE International Conference on*, pp. 168-171, Dec. 2002.
- [8] S. Crisostomo, S. Sargento, P. Brandao, R. Prior, "Improving AODV with Preemptive Local Route Repair," *Wireless Ad-Hoc Networks, 2004 International Workshop on*, pp. 223-227, June 2004.
- [9] M. Pan, Sheng-Yan Chuang, Sheng-De Wang, "Local Repair Mechanisms for On-Demand Routing in Mobile Ad hoc Networks," *WIMOB.2005*, pp. 192-196, Aug. 2005.
- [10] <http://www.isi.edu/nsnam/ns>.

유 대 훈 (DaeHun Yoo) 준회원



2005년 광운대학교 수학과 학사
2007년 광운대학교 컴퓨터과학과 석사
2007년 3월~현재 광운대학교 컴퓨터과학과 박사과정
<관심분야> 무선 네트워크, 교환 및 라우팅

최 응 철 (WoongChul Choi) 정회원



1989년 서울대학교 컴퓨터공학과 학사
1991년 서울대학교 컴퓨터공학과 석사
2001년 University of Illinois, Urbana-Champaign, IL., USA., Ph.D.

2002년 9월~현재 광운대학교 컴퓨터공학부 조교수
<관심분야> Wireless/Mobile networks, QoS