

논문 2008-45CI-6-16

이동 ad hoc 네트워크의 이동성을 지원하기 위한 향상된 라우팅 프로토콜

(An Enhanced Routing Protocol for Support Mobility in Mobile Ad hoc Networks)

김 관웅*, 김 대익**

(Kwan-Woong Kim and Dae-Ik Kim)

요약

이동 ad hoc 네트워크는 기지국 또는 중앙 관리국 없이 주어진 영역 내에서 다중 노드들 사이에 무선 데이터 통신이 이루어지는 자체적인 네트워크이다. 노드의 이동성과 제한된 배터리 수명에 의해 네트워크 토플로지는 자주 바뀌게 된다. 따라서 경로 탐색 절차에 있어서 가장 신뢰성 있는 경로를 선택하는 것은 ad hoc 네트워크의 성능을 향상시키기 위해 중요하다. 본 논문에서는 수신 신호 세기의 변화를 측정함으로써 AODV (Ad hoc On-demand Distance Vector routing)에 기반을 둔 향상된 라우팅 프로토콜을 제안한다. 라우팅을 결정하기 위해 노드 이동성과 경로의 흡으로 이루어진 새로운 측정 함수가 사용된다. 또한 데이터 전송 중 노드의 움직임에 의한 경로의 단절을 방지하기 위해 LRC (Local Route Change)라는 새로운 라우트 관리 기법을 소개한다. 노드의 움직임을 감지하면, 경로가 단절되기 전에 라우팅 에이전트는 다음 흡 노드를 이웃노드로 변경한다. 시뮬레이션 결과를 통해 제안한 라우팅 기법의 성능이 기존의 AODV 방식보다 우수함을 보인다.

Abstract

Mobile Ad hoc NETworks (MANETs) refer to autonomous networks in which wireless data communications are established between multiple nodes in a given coverage area without a base station or centralized administration. Because of node mobility and limited battery life, the network topology may changes frequently. Selecting the most reliable path during route discovery process is important to improve performance in ad hoc networks. In this paper, an enhanced routing protocol based on AODV(Ad hoc On-demand Distance Vector routing) by monitoring variation of receiving signal strength is proposed. New metric function that consists of node mobility and hops of path is used for routing decision. For preventing route failure by node movement during data transmission, a new route maintenance named as LRC (Local Route Change) is presented. If the node movement is detected, the routing agent switches path to its neighbor node in LRC. Simulation results show that the performance of the proposed routing scheme is superior to previous AODV protocol.

Keywords : Ad hoc network, Route discovery, Route Maintenance, Node Mobility function

I. 서론

이동 ad hoc 네트워크 (MANETs)는 기존의 통신 인

프라(기지국, AP) 또는 중앙 제어장치를 통하지 않고 네트워크상의 모든 무선 노드들이 서로 통신할 수 있는 이동 노드들로 이루어져 있다^[1~2]. 따라서 MANETs는 일반적인 무선 LAN 시스템과는 달리 멀티 흡 네트워크에서 기지국과 이동 사용자 사이에 마스터-슬레이브 관계를 갖고 있지 않다. 노드들 사이의 통신은 직접적인 연결 또는 다중 흡 (hop) 릴레이를 통해 제공되어 진다. 이 노드들은 무선 환경 내에서 노드의 이동성과 신호의 변동 등으로 인해 발생되는 네트워크 토플로지

* 정회원, (주)썬더테크놀로지
(THUNDER TECHNOLOGY)

** 종신회원, 전남대학교 전기전자통신컴퓨터공학부
(School of Electrical, Electronic Communication,
and Computer Engineering, Chonnam National
University)

접수일자: 2008년10월10일, 수정완료일: 2008년11월14일

의 변화에 강건할 (robust) 수 있도록 자체구성 (self-organizing)을 해야 한다. 이러한 특성들로 인해 MANETs는 임시 구성용 네트워크나 지진, 태풍, 테러 등에 의한 재해/재난 복구 또는 구조, 전쟁터와 같은 군사용 네트워크, 그리고 LAN(Local Area Network)에서 블루투스, HomeRF 등에 적용할 수 있다.

MANETs는 정적인 노드를 갖는 유선 네트워크와는 달리 자주 발생하는 네트워크 토폴로지의 변화를 잘 처리하기 위해 높은 적응력을 갖춘 라우팅 프로토콜을 요구한다. 따라서 라우팅 프로토콜은 네트워크 토폴로지의 변화를 빠르게 파악하여 목적지까지의 생신된 경로를 제공해 줘야 한다. 라우팅 프로토콜은 크게 table-driven 방식과 on-demand 방식으로 나눌 수 있다.

Table-driven 방식은 네트워크 내의 루트를 지속적으로 갱신하여 패킷을 전송할 때 경로를 미리 알고 있어서 바로 사용할 수 있으며, 이 방식으로는 Destination-Sequence Distance Vector (DSDV), Wireless Routing Protocol (WRP), 그리고 Clusterhead Gateway Switch Routing (CGSR) 등이 있다^[5]. 반면에, on-demand 방식은 경로가 요구되어질 때만 경로 검색 절차를 수행하며, Dynamic Source Routing (DSR), Ad hoc On-demand Distance Vector routing (AODV)^[3], Lightweight Mobile Routing (LMR), Temporally Ordered Routing Algorithm (TORA), Associativity-Based Routing (ABR)^[6], Signal Stability Routing (SSR)^[5] 등이 포함된다.

MANETs에서 노드들은 자유롭고 예측불가능하게 움직이기 때문에 목적지까지의 패킷 전송 경로는 자주 끊어지게 되며 이는 네트워크의 전체적인 성능을 감소시킨다. 이러한 문제를 해결하기 위해 라우팅 프로토콜에서 이동성에 대한 연구가 지속적으로 이루어지고 있다. 그러나 지금까지의 대부분의 연구는 MANETs에 속한 모든 노드들이 GPS (Global Positioning System)와 같은 위치 인식 장치를 장착하고 있다는 가정을 하였고, 이는 모든 노드들이 이동성 검출을 위해 필요로 하는 모든 정보를 갖도록 하였다. 그러나 일반적으로 GPS는 실내에서 사용 불가능하고, 많은 전력이 소모되는 이유로 소규모의 작은 장치에는 적합하지 않은 제약 사항이 있다^[11].

Tom Goff는 AODV와 DSR을 향상시킨 preemptive-routing 프로토콜을 제안하였다^[12]. 이 프로토콜은 노드 이동성에 의해 링크가 끊어지기 전에 경로를 재탐색하기 위한 결정을 하기 위해 수신 신호 전력을 측정한다.

경로를 회복시켜주는 방식은 셀룰러 네트워크의 핸드오프 (hand-off) 방식과 유사하다. GPS를 사용하지 않은 또 다른 방식으로는 신호 대 잡음비 (SNR)를 측정하여 전송 노드가 접근하는지 또는 멀어지는지를 해당 노드에게 알려주는 것이다. 기존에 제안된 모든 방안들은 오직 한 노드만을 고려한 협소적인 방안이며, 전체적인 경로를 해결하기 위한 방안이 제시되지 못한 문제점을 갖고 있다.

본 논문에서는 노드 이동성을 고려한 향상된 라우팅 프로토콜을 제안하며, 노드의 움직임으로 링크가 단절되기 전에 경로가 이웃 노드로 전환되는 셀룰러 네트워크의 핸드오프 방식에 기반을 둔 새로운 경로 관리 기법을 소개한다. II장에서는 AODV에 기반을 둔 새로운 라우팅 프로토콜과 LRC 경로 관리 기법을 제안한다. III장에서는 NS2 시뮬레이션을 통한 성능 평가 결과를 논의하고, 마지막으로 IV장에서 결론을 맺는다.

II. 제안한 라우팅 알고리즘

여러 개의 가능한 다중 경로로부터 신뢰성 있는 경로를 검색해 주는 라우팅 프로토콜은 end-to-end 연결이 좀 더 길게 지속될 수 있도록 해준다. 무선 네트워크에서 두 노드 사이의 거리는 수신노드가 받는 신호 세기 RxP 와 반비례 한다^[8]. RxP 가 낮아지면 송신노드는 멀어지고, 반대로 RxP 가 증가하면 송신노드는 접근하고 있음을 의미한다. 노드 B의 움직임에 따른 거리와 수신 신호 전력과의 관계를 그림 1에 보여주고 있다. 노드 B는 속도 v 로 p_1 에서 p_2 로 움직이고 있으며, 이에 따라 노드 A에서의 수신신호 전력은 RxP_1 에서 RxP_2 로 변화된다. 따라서 RxP 의 변화를 측정함으로써 노드 이동성을 얻어낼 수 있다.

그림 1로부터 식 (1)을 유도할 수 있다.

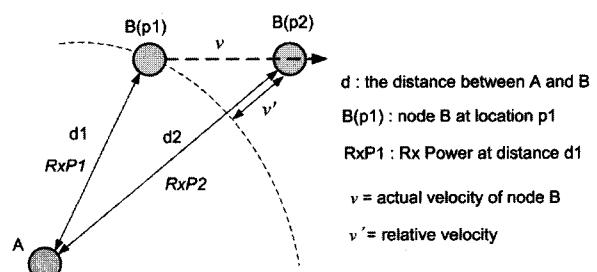


그림 1. 노드의 이동성에 따른 수신신호 전력 RxP 와 거리와의 관계

Fig. 1. The relationship between RxP and distance according to node mobility.

$$M = \frac{d(t_1) - d(t_0)}{t_1 - t_0} \simeq \frac{v'}{k},$$

where $d(t) = RxP(t)^{-\frac{1}{n}}$ (1)

여기에서 k 는 라디오 채널 조건이며, 시뮬레이션에서 적용할 two-ray ground 모델에서 n 은 일반적으로 4가 된다. M 이 양수이면 두 노드가 서로 멀어지는 것을 의미하고, M 이 '0'이면 두 노드가 움직이지 않거나 또는 같은 속력을 갖고 같은 방향으로 동시에 움직이는 것을 의미한다. 또한 M 이 음수이면 두 노드가 서로 가까워짐을 의미한다. 식 (1)로부터 함수 M 은 상대속도 v' 과 채널 조건 k 에 따라 결정됨을 알 수 있다.

1. 경로 탐색

제안한 라우팅 프로토콜은 경로 탐색 동안 낮은 이동성을 갖는 경로를 찾으려는 시도를 한다. 경로요청 (RREQ: route request) 패킷을 플로딩 (flooding)할 경우에는 노드의 이동성 정보를 저장하기 위해 4바이트 M 필드가 RREQ 헤더에 필요하다. 각 노드는 최상의 경로를 검색하기 위해 흡 카운트를 고려하여 다중 경로 중에서 가장 적은 M 을 갖는 경로를 선택한다. 경로 탐색 절차에서 경로 요청 노드는 RREQ 패킷의 M 필드를 '0'으로 세팅한다. 중간 노드들 중에서 노드 j 가 RREQ 패킷을 노드 i 로부터 수신하면, 노드 j 는 식 (2)와 같이 최대 M 값을 찾아 RREQ 패킷의 M 을 갱신한다.

$$M_{RREQ} = \max(M_{RREQ}, M(i)) \quad (2)$$

여기에서 $M(i)$ 는 식 (1)로 계산한 노드 j 와 i 사이의 mobility cost function이고, M_{RREQ} 는 RREQ 패킷의 mobility cost function이다.

또한 경로 정보를 갱신하기 위해 식 (3)의 route cost function R_{cost} 를 사용한다. 한 노드가 RREQ 패킷을 수신할 때마다 라우팅 에이전트는 RREQ 패킷의 새로운 경로를 위한 경로 엔트리를 갱신하기 위해 route cost function을 계산한다.

$$R_{cost} = h_{RREQ} - h_{RT} + F_m \quad (3)$$

여기에서 h_{RREQ} 는 RREQ 패킷의 흡 카운트이고, h_{RT} 는 현재 경로 정보의 흡 카운트이다. 그리고 F_m 은 식 (4)와 같이 정의되는 상대적 mobility cost function이다.

$$F_m = k \frac{M_{RREQ} - M_{RT}}{M_{RREQ} + M_{RT}}, \text{ if } M_{RREQ} + M_{RT} \neq 0$$

$$F_m = 0, \text{ otherwise} \quad (4)$$

여기에서 M_{RT} 는 노드의 라우팅 테이블에 저장된 경로의 mobility cost function이다. 식 (4)에서 F_m 의 범위는 $-k$ 에서 k 이고, 본 논문에서는 k 를 2로 설정하였다. 한 노드가 RREQ 패킷을 수신하면 R_{cost} 를 계산하여 그 값이 0보다 작으면 경로 정보는 새로운 경로로 갱신된다. 그러므로 $[h, h+k]$ 범위에서 작은 이동성을 갖는 경로가 선택된다. 여기에서 h 는 h_{RT} 와 h_{RREQ} 중에서 최소 값이다.

목적지 노드의 동작

목적지 노드가 첫 번째 경로 요청 패킷을 수신할 때 라우팅 에이전트는 임의의 시간으로 타이머를 세팅하고 다른 경로들로부터 또 다른 경로 요청 패킷을 기다리기 위해 타이머를 동작시킨다. 따라서 목적지 노드는 다중 경로 중에서 최선의 경로를 선택할 수 있다. 타이머의 시간이 만기되면 목적지 노드는 송신 노드에 즉시 경로 응답 메시지 (RREP)를 보낸다.

채널 페이딩 (fading)과 일시적인 방해요소들

무선네트워크의 실제 환경에서 채널 페이딩과 다중 경로 효과가 갑작스런 신호 전력 변동을 발생시킬 수 있다. 이는 라우팅 에이전트가 잘못된 정보를 참조하여 경로를 잘못 선택할 수 있게 만든다. 이러한 문제를 해결하기 위해 MIMO (multiple input multiple output)와 소수 패킷의 수신 신호 전력 평균과 같은 두 가지 방안을 생각할 수 있다. 다중 송수신 안테나를 사용하는 MIMO 기법은 채널 대역폭과 전송 범위를 넓히기 위해 사용된다^[7]. 따라서 이 기술은 빠른 전송 속도 뿐만 아니라 페이딩과 다중 경로 효과에 의한 성능 저하를 줄일 수 있다. 또 다른 해결 방안은 두, 세 개 패킷의 신호 전력 평균값을 이용하는 것이다. IEEE 802.11 MAC 프로토콜에서 데이터 패킷을 전송하기 위해 RTS-CTS-DATA-ACK 방법을 사용한다^[9]. 따라서 노드는 하나의 데이터 패킷을 전송하기 위해 적어도 두 개의 패킷을 수신해야만 한다. 최근에 수신한 두 패킷의 평균 M 값을 계산함으로써 채널 페이딩에 의한 갑작스러운 신호 변화를 감소시킬 수 있다.

2. 경로 관리

라우팅 프로토콜에서 경로 관리는 경로 연결을 관리하고 연결의 끊김을 검색하는 역할을 한다. AODV에서 로컬 경로 정정 알고리즘은 경로를 빠르게 회복시키는 데 사용 된다^[10]. 그러나 대부분의 라우팅 알고리즘들은 링크가 끊어지기 전에 경로를 회복시키는 능력이 결여되어 있다. 본 논문에서는 노드 이동성에 의해 발생되는 경로 단절을 방지해 주고, 패킷을 전송하는 동안 라우팅 프로토콜의 효율성을 개선시키는데 초점을 맞췄다. 이러한 목적을 달성하기 위해 라우팅 프로토콜은 움직이는 노드로부터 대체 가능한 노드로 로컬 경로를 바꿔줄 수 있는 능력이 있어야 한다. LRC에 대한 기본 개념은 기존에 연구 발표한 [14]에 기반을 두고 있다.

LRC 프로세스의 동작

그림 2는 LRC 프로세스의 첫 번째 단계를 보여주고 있다. 중간 노드 'C'가 이전 흡 노드 'A'로부터 데이터 패킷을 받을 때, 수신 신호 전력 RxP 와 mobility function M 을 측정한다. RxP 가 LRC 임계치 $RxTh$ 보다 작고 $M(A)$ 가 0을 초과하면 (즉, 노드의 움직임이 검출되면), 노드는 이웃 노드들 중에서 대안이 되는 노드를 찾기 위해 one-hop 이웃 노드들에 HELP 메시지를 방송하면서 LRC 프로세스를 시작한다. 여기에서 LRC 임계치 $RxTh$ 는 식 (5)와 같이 정의된다.

$$RxTh = K \times RxP_{min} \quad (5)$$

RxP_{min} 은 노드가 수신 가능한 최소 전력이고 (IEEE 802.11b에서는 $RxP_{min} = 3.65 \times 10^{-10} \text{ Watts}$ ^[9]), K 는 수신 신호 세기 임계치를 설정하기 위한 상수이며 5로 설정하였다.

그 다음에 선위 (previous) 흡 노드 'A'와 차위 (next) 흡 노드가 'C' 노드의 이웃노드이므로 HELP 메시지를

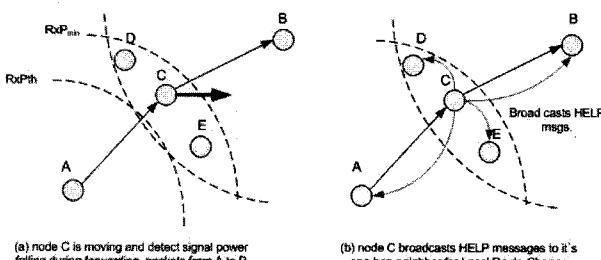


그림 2. 노드가 one-hop 이웃 노드들에 HELP 메시지를 방송하는 LRC 프로세스 예제

Fig. 2. An example of LRC process: broadcasts HELP message to its one-hop neighbors.

수신한다. Mobility cost $M(A)$ 와 $M(B)$ 가 '0'이면, 그 노드는 노드 'C'의 대체 노드가 될 수 있다. 그렇지 않으면 HELP 메시지는 무시된다. 대체 가능한 노드는

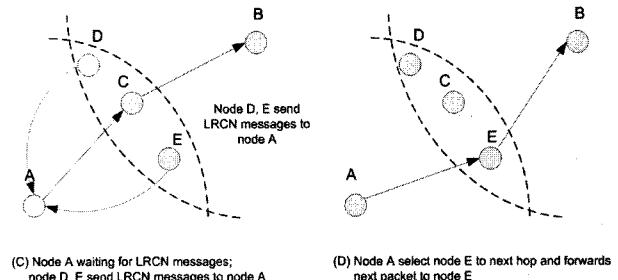


그림 3. 선위 흡 노드가 LRCN 메시지를 수신하고 차위 흡 노드 'C'를 새로운 노드로 변경하는 LRC 프로세스 예제

Fig. 3. An example of LRC process: previous-hop node receives LRCN message and switch next-hop node 'C' to new one.

표 1. LRC 프로세스의 의사 코드

Table 1. Pseudo code of LRC process.

```

On_Receive_DATApacket(pkt){
    Next_ip = rtable(pkt->destip);
    Prev_ip = rtable(pkt->srcip);
    M_prev = Get_Mfunc(Prev_ip);
    M_next = Get_Mfunc(Next_ip);
    forwards pkt to nexthop
    // check condition
    If(rxP < rxPTh && M_prev > 0 && M_next == 0)
    {
        Broadcasts HELP;
    }
}

On_receive_HELP(Helper_pkt) {
    If(Helper_pkt->next_hopip is myneighbor &&
       Helper_pkt->prev_hopip is myneighbor) {
        M_prev = GetMfunc(Helper_pkt->prev_hopip);
        M_next = GetMfunc(Helper_pkt->next_hopip);
        If(M_prev == 0.0 && M_next == 0.0){
            update_rtable(Helper_pkt);
            send LRCN to Helper_pkt->prev_hopip;
        }
    }
    /* previous hop node receives Help msg. */
    Else if(Helper_pkt->prev_hop == node_ip){
        // set timer for wait LRCN msg.
        Set_LRCN_timer(Helper_pkt->destip);
    }
}

On_receive_LRCN(LRCN_pkt) {
    next_hopip = rtable(LRCN_pkt->destip);
    M_next = GetMfunc(nexthopip);
    If (M_next > LRCN_pkt->M){
        Cancel_LRCN_timer();
        update_rtable(LRCN); // change next_hop ip
    }
}

On_LRCN_timer_expire(dest_ip){
    Local_route_repair(dest_ip);
}

```

HELP 메시지 정보를 참조해 경로 정보를 갱신하고 LRCN (Local Route Change Notification) 메시지를 HELP 메시지의 선위 흡 노드에 전달한다.

그림 3은 선위 흡 노드가 LRCN 메시지를 수신하고 차위 흡 노드 'C'를 새로운 노드로 변경하는 LRC 프로세스 예제를 보여주고 있다. 선위 흡 노드 'A'는 차위 흡 노드 'C'로부터 HELP 메시지를 수신하고, 노드 'A'는 LRCN 메시지를 전송 받기 위해 타이머를 설정한다. LRCN 메시지를 수신하면, 선위 흡 노드는 타이머를 해제하고 차위 흡 노드 'C'를 LRCN 메시지 송신 노드 'E'로 변경 한다. 타이머가 종료되면, LRC 프로세스가 경로변경이 실패된 상황이므로, 경로 단절 전에 새로운 경로를 찾기 위해 로컬 경로 복구(Local Route Repair) 프로세스를 수행한다^[10]. 표 1은 LRC 프로세스의 의사코드를 보여준다.

새로운 메시지 형식

로컬 경로 교환 (LRC) 프로세스를 위해 그림 4와 같이 AODV 프로토콜에 HELP와 LRCN 메시지를 새로 정의하였다. 여기에서 M 필드는 32 비트 부동소수점으로 표현된다. 목적지 일련번호는 수신 노드에서 새로운 경로를 관리하기 위해 사용된다.

III. 성능평가

1. 시뮬레이션 구성

제안한 라우팅 프로토콜은 AODV 프로토콜을 기반으로 기능을 개선하였으므로 그 성능을 평가하기 위해

Type (8bits)	Reserved (24bits)
Prev-hop IP Address (32bits)	
Next-hop IP Address (32bits)	
Destination IP Address (32bits)	
Destination Sequence Number (32bits)	
M (32bits)	

(a) HELP message

Type (8bits)	Reserved (24bits)
Alternative node IP Address (32bits)	
Destination IP Address (32bits)	
M (32bits)	

(b) LRCN message

그림 4. LRC 프로세스를 위한 HELP와 LRCN 메시지 형식
Fig. 4. HELP and LRCN message formats for LRC process.

표 2. NS2에서 에너지 모델의 파라메타
Table 2. Parameters of energy model in NS2.

Parameters	Description	Value
initial energy	given energy for each node	200 Joules
Grx, Gtx	antenna gain	1
txPower	transmitting power in Watt	281.8mW

AODV와 비교 시뮬레이션을 수행하였다. 시뮬레이션을 위해 NS2 시뮬레이터를 사용하였으며, ad hoc 네트워크 모델은 1.0km×1.0km 영역에 100개의 이동 노드들로 구성되었고 노드들의 초기 위치는 무작위 방식으로 발생된다. 각 노드는 IEEE802.11 MAC 프로토콜을 사용하였고, 채널 모델은 무선 채널/무선 물리 전송 모델을 이용하였다. 라디오 전송 모델로 two-ray ground 모델을 사용하였으며, 무선 노드의 전송 범위와 간접 범위는 250m와 550m로 설정하였다. 트래픽 소스는 CBR (Constant Bit Rate)이고, 15 개의 CBR 소스는 0.1 초마다 512 바이트 크기의 UDP (User Datagram Protocol) 패킷을 발생시킨다. 각 실험에서 시뮬레이션 시간은 200 초로 설정하였고, 노드의 최대 속력은 5m/sec~20m/sec 사이에서 변화된다. 편중된 난수 발생을 피하기 위해 같은 구성 조건에서 10번의 시뮬레이션을 수행하였다. 표 2는 NS2에서 에너지 모델의 파라메타를 보여주고 있다.

2. 시뮬레이션 결과 및 토의

시뮬레이션은 기존의 AODV, AODV+LRC, 제안한 프로토콜, 그리고 제안한 프로토콜+LRC에 대해 수행하였다.

그림 5에서는 노드의 최대 속력에 따른 경로 메시지의 오버헤드 비율에 대한 시뮬레이션 결과를 보였고, 그림 6은 경로 탐색의 개수를 보여주고 있다. 제안한 라우팅 알고리즘과 LRC는 기존의 AODV와 비교하여 라우팅 메시지의 오버헤드와 경로 탐색의 개수를 급격하게 줄여주었음을 확인할 수 있다. 또한 이동 노드의 최대 속력이 증가함에 따라 AODV에서는 경로 메시지 오버헤드와 경로가 끊겨지는 횟수도 비례적으로 증가되지만, 제안한 알고리즘에서는 거의 일정한 수준을 유지하고 있다. 더욱이 기존의 AODV 방식에 LRC를 혼합한 방식은 AODV 방식만을 적용한 경우에 비해 더 좋은 성능을 보여주고 있다. 시뮬레이션 결과를 통해 제안한 LRC가 노드의 움직임에 의해 야기되는 경로 실패

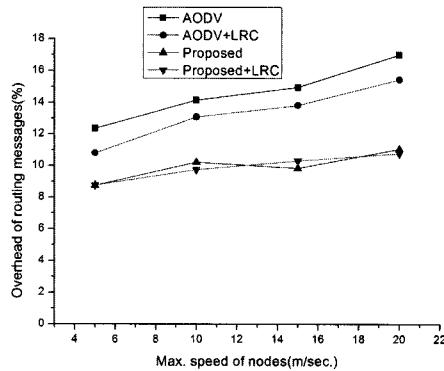


그림 5. 경로 메시지의 오버헤드 비율

Fig. 5. Overhead ratio of routing messages.

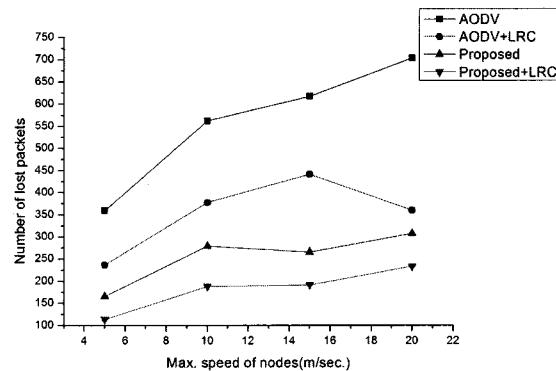


그림 8. 평균 손실 패킷 수

Fig. 8. Average number of lost packets.

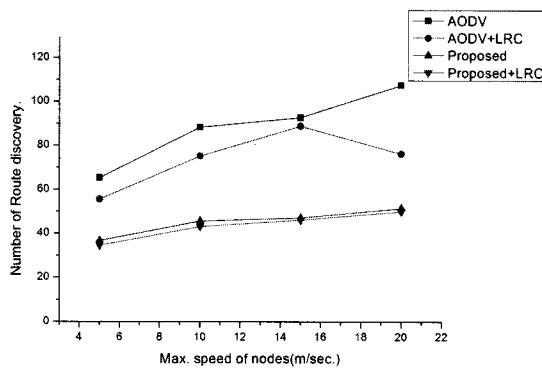


그림 6. 경로 탐색의 개수

Fig. 6. Number of routing discovery.

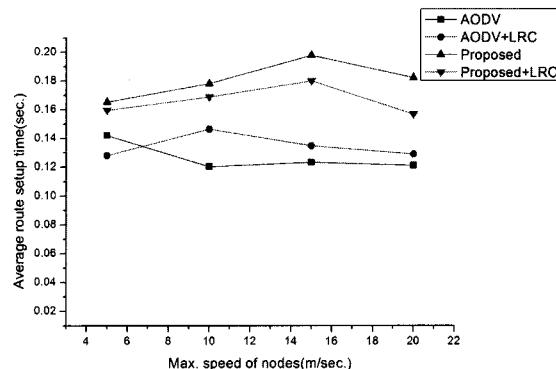


그림 9. 평균 경로 설정 시간

Fig. 9. Average setup time of routes.

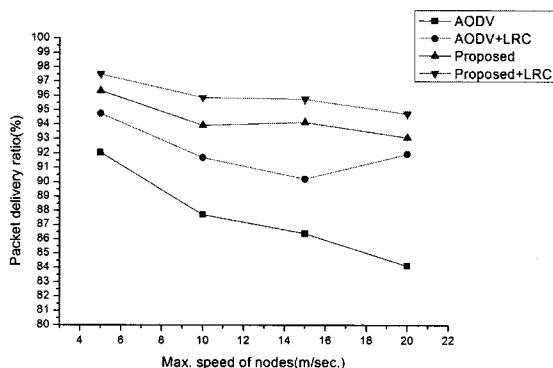


그림 7. 패킷 전달률

Fig. 7. Delivery ratio of packets.

율을 감소시켜주고 경로 탐색의 유연성을 제공하는 효과를 검증할 수 있다.

그림 7과 그림 8에서는 end-to-end 패킷 전달율과 수신측에서의 평균 손실 패킷 수에 대한 시뮬레이션 결과를 비교해 보여주고 있다. 노드의 최대 속력이 증가 할수록, 네트워크 내에서 끊어진 경로에 의해 전달되지

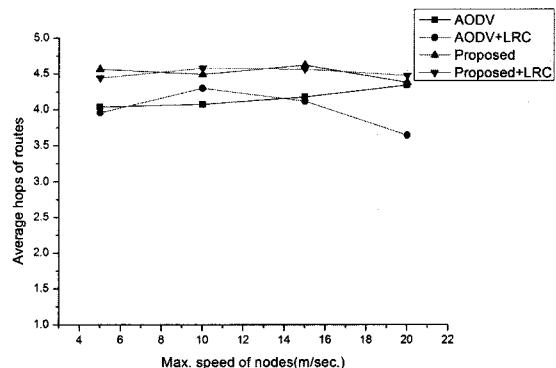


그림 10. 경로의 평균 흡 카운트

Fig. 10. Average hop count of routes.

못하는 패킷의 개수도 증가한다. 기존의 AODV의 경우 높은 이동성 조건에서 패킷 전달율은 급속하게 감소한다. 모든 이동성 조건에서 제안한 방식이 패킷 전달율을 개선시켰으며, 손실 패킷 수를 줄여줄 수 있음을 확인할 수 있다. AODV+LRC는 기존의 AODV 방식에 비해 패킷 전달율을 4~11% 정도 개선시켰다. 또한 제안

한 알고리즘+LRC는 제안한 알고리즘만을 적용한 경우에 비해 3% 정도 패킷 전달율을 개선해 주었다. 따라서 LRC와 결합된 경로 탐색 프로세스는 성능과 효율성을 향상시켜 줄 수 있음을 확인할 수 있다.

그림 9와 그림 10은 평균 경로 설정시간과 평균 흡 카운트를 보여주고 있다. 목적지 노드들은 서로 다른 방향에서 전송되는 RREQ 패킷들을 수신하기 위해 임의의 시간동안 기다려야 한다. 시뮬레이션 결과를 살펴보면 LRC를 갖는 제안한 프로토콜의 평균 경로 설정시간은 기존의 AODV에 비해 30~40% 증가되었다. 또한 LRC를 갖는 제안한 프로토콜의 경로의 평균 흡 카운트는 AODV에 비해 0.5~1 흡 정도 증가하였다. 이러한 현상은 제안한 방식이 최소 이동성을 갖는 좀 더 긴 경로를 선택하는 경향이 있기 때문이다. 비록 제안한 방식이 평균 경로 설정 시간과 경로의 평균 흡 카운트에서 좋지 못한 성능을 보이고 있지만, 이 외에 앞에서 살펴본 다른 성능 평가에서는 AODV를 압도하고 있고 노드의 이동성과 위치 정보를 검출하기 위한 별도의 장치가 필요 없는 장점이 있다.

IV. 결 론

Ad hoc 네트워크의 토플로지는 노드의 이동성으로 인해 자주 변화된다. 즉, 노드들이 각 연결 범위들 사이에서 이동함으로써 존재했던 연결이 끊어지고 새로운 연결이 설정된다.

본 논문에서는 수신 신호 전력의 변화를 이용하여 신뢰성이 가장 높은 경로를 검색할 수 있는 새로운 라우팅 프로토콜을 제안하였다. 경로 검색 단계에서는 노드의 이동성뿐만 아니라 경로의 흡을 고려하였다. 그리고 경로 결정을 단순화시키기 위해 흡 카운트와 노드 mobility function M 으로 구성된 새로운 route cost function을 제시하여 목적지 노드가 다중경로 중에서 최선의 경로를 선택할 수 있도록 하였다. 또한 경로 관리를 향상시키기 위해 LRC 프로세스를 제안하였다. LRC는 차위 흡 노드가 전송범위를 벗어나기 전에 이웃 노드들 중에 사용 가능한 노드를 차위 흡 노드로 변환시켜 네트워크 연결이 지속될 수 있게 해준다.

시뮬레이션을 통해 제안한 기법이 기존의 AODV에 비해 경로 설정시간과 흡 카운트에서 단점을 보였지만, 경로의 단절과 경로 메시지의 오버헤드를 감소시켜주고 end-to-end 패킷 전송율을 높여줌으로써 제안한 알고리즘의 우수성을 확인하였다.

참 고 문 헌

- [1] C. E. Perkins, *Ad Hoc Networking*, Addison-Wesley, Upper Saddle River, NJ, USA, Jan. 2001.
- [2] <http://www.ietf.org/>, IETF MANET Working Group.
- [3] C. Perkins, E. Belding-Royer, and S. Das, "Ad Hoc On-Demand Distance Vector Routing," *Proc. 2nd IEEE workshop on Mobile Computing Systems and Applications*, pp. 90-100, New Orleans, LA, Feb. 1999.
- [4] C. E. Perkins, E. M. Royer, and S. R. Das, "Ad-hoc on demand distance vector routing," *IETF RFC3561*, <http://www.ietf.org/rfc/rfc3561.txt>, 2003.
- [5] Elizabeth M. Royer and Chai-Keong Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks," *IEEE personal Communications*, Apr. 1999.
- [6] C. K. Toh, "Associativity Based Routing for Ad Hoc Mobile Networks," *Wireless Pers. Commun. J., Special Issue on Mobile Networking and Computing Systems*, vol. 4, no. 2, Mar. 1997.
- [7] D. Gesbert, M. Shafi, D. Shiu, P. J. Smith, and A. Naguib, "From theory to practice: An overview of MIMO space-time coded wireless systems," *IEEE J. Select. Areas Commun.*, vol. 21, no. 3, pp. 281-302, Apr. 2003.
- [8] J. B. Anderson, T.S. Rappaport, and S. Yoshida, "Propagation Measurements and Models for Wireless Communications Channels," *IEEE Communication Magazine*, vol. 33, no. 1, pp. 42-49, Jan. 1995.
- [9] Lahti Marja-Leena, "IEEE 802.11 Wireless LAN", http://www.tml.hut.fi/Opinnot/Tik-10.551/2000/papers/IEEE_802/wlan.html, 2000.
- [10] C.E. Perkins, "Ad hoc on-demand distance vector(AODV) routing," *RFC3561*, Jul. 2003.
- [11] E. D. Kaplan, *Understanding GPS: Principles and Applications*, Artech House, Boston, MA, 1996.

저자 소개



김 관 웽(정회원)
 1992년 전북대학교 전자공학과
 학사 졸업.
 2000년 전북대학교 전자공학과
 석사 졸업
 2004년 전북대학교 전자공학과
 박사 졸업.
 2002년 8월~2004년 8월 한국표준과학연구원
 Post-doc.
 2004년 9월~2005년 8월 Post-doc,
 University of Haute Alsace
 2006년 3월~2008년 2월 원광대학교
 전기전자정보공학부 교수
 2008년 11월 현재 (주)썬더테크놀로지 이사
 <주관심분야 : 정보통신, 무선네트워크, 인터넷프
 로토콜>



김 대 익(종신회원)
 1991년 전북대학교 전자공학과
 졸업(학사)
 1993년 전북대학교 대학원
 전자공학과(공학석사)
 1996년 전북대학교 대학원
 전자공학과(공학박사)
 2002년~현재 전남대학교 공학대학 전기전자통신
 컴퓨터공학부 교수
 2006년 7월~2007년 6월 미국 오레건주립대학
 교환교수
 <주관심분야 : 저전력 VLSI 설계, 테스팅, 데이터
 변환기, Ad hoc network>