

Ad-hoc 네트워크 테스트 베드 구현에 관한 연구

정회원 이 흥 재*, 가 순 모**, 종신회원 최 진 규**

A Study on the Implement of Test Bed for Ad-hoc Networks

Heung-jae Lee*, Soon-mo Ga** *Regular Members*, Jin-kyun Choe** *Lifelong Member*

요 약

AODV(Ad-hoc On-Demand Distance Vector) 라우팅 프로토콜은 Ad-hoc 네트워크에서 이동 노드를 사용할 수 있도록 제안된 라우팅 프로토콜이다. AODV 라우팅 프로토콜을 사용하는 Ad-hoc 네트워크에서 고속으로 이동하는 노드가 포함되어 있는 경우 항상 최적 경로를 확보할 수 없는 문제로 인하여 경로 단절과 전송 지연이 발생한다. 따라서 본 논문에서는 고속으로 변화하는 네트워크의 토폴로지에서 항상 최적의 경로를 확보 할 수 있는 메커니즘을 통하여 경로의 단절과 전송 지연을 최소화할 수 있는 AODV를 기반으로 하는 라우팅 프로토콜을 제안하였으며 ns2 시뮬레이터를 이용하여 제안 프로토콜을 평가하였다.

Ad-hoc 네트워크를 위한 여러 가지 기반 기술의 검증을 위하여 실제의 Ad-hoc 네트워크 테스트 베드를 구현하였다. 본 논문에서는 AODV 라우팅 프로토콜, NAT, Netfilter 등의 Ad-hoc을 위한 소프트웨어 검증을 위한 많은 이벤트 메시지를 성능 저하 없이 동작시킬 수 있는 고성능의 임베디드 시스템을 설계 개발하였다. 개발된 하드웨어를 이용한 Ad-hoc 네트워크 테스트 베드에서 AODV 라우팅 프로토콜의 정상 동작과 기존 인터넷 망과의 연동을 확인하였다.

Key Words : Ad-hoc, AODV, Embedded

ABSTRACT

AODV(Ad-hoc On-demand Distance Vector) routing protocol was devised for use of mobile nodes in Ad-hoc network. When we use the AODV routing protocol in Ad-hoc networks with high mobility, disturbance of optimized route path and link break occur. In order to solve the shortcomings, this paper proposes a new routing protocol in which new routing control messages are added to the existing AODV. The proposed protocol minimizes link break and transmission delay while is able to secure the optimized route path constantly in changes of network topology. The performance of the proposed routing protocol was evaluated by using ns2 network simulator.

The actual Ad-hoc network test bed provides us the most reliable experimental data for Ad-hoc networks. In order to support this experimental environment, the dissertation also developed an efficient embedded system on which AODV routing protocol, NAT, Netfilter can run and other event message can be verified without declining efficiency. The correct operation of AODV routing protocol has been verified in both the Ad-hoc network test bed in which the embedded system was used, and Ad-hoc networks linked with Ethernet backbone network.

I. 서론

Ad-hoc 네트워크는 이미 구축된 네트워크 인프라

가 유일 또는 존재하지 않는 지역에서 노드 스스로 네트워크를 구성하고 통신할 수 있는 형태의 네트워크이며, Ad-hoc 네트워크를 위한 라우팅 프로

* (주)바이오니아(stargaze@bioneer.co.kr), ** 한남대학교 전자공학과
 논문번호 : KICS2006-07-308, 접수일자 : 2006년 7월 14일, 최종논문접수일자 : 2006년 11월 15일

토폴로 제안된 AODV는 DSR(Dynamic Source Routing)의 온디맨드(On-Demand) 방식과 DSDV (Destination Sequenced Distance Vector)의 일련번호, 홉-대-홉 통신방식을 수용하여 제안된 Ad-hoc 네트워크를 위한 라우팅 프로토콜이다^[14].

Ad-hoc 네트워크는 기존 네트워크 인프라에 의존하지 않고 노드 스스로 네트워크를 구성한다는 점에서 기존 모바일 IP, 유선 네트워크 기술과 많은 차이를 가지고 있다. 따라서 Ad-hoc 네트워크에 현재 모바일 IP 또는 유선 네트워크에서 사용되고 있는 기술을 적용하는 데에는 한계가 있으며 특히 라우팅 프로토콜은 Ad-hoc 네트워크의 성능을 결정하는 중요한 기술로 많은 연구가 진행되고 있다. 유선 네트워크와 달리 무선 네트워크에서는 경로 단절이 빈번히 발생하기 때문에 경로 단절을 최소화하여 네트워크 성능을 개선할 수 있는 라우팅 프로토콜에 대한 연구가 필요하다.

경로 단절에 관련된 연구는 경로의 단절이 발생한 경우에 빠른 속도로 경로를 복구하는 연구와 경로 단절 자체를 최소화 하는 관점으로 분류할 수 있다. 경로 단절과 관련된 연구의 주류는 경로 단절 후 보다 빠른 복구에 관심을 둔 알고리즘 연구이다. 이와 같은 알고리즘은 경로 단절시 미리 확보한 대체 경로 또는 백업된 경로를 이용하여 빠르게 경로를 복구할 수 있도록 한다. 대체 경로 또는 경로의 백업에 관련된 연구의 이슈는 어떠한 방법으로 미리 경로를 확보할 것인가에 관점을 두고 있다^[15].

빠르게 네트워크의 토폴로지가 변화되고 있는 네트워크에 경로 단절이 발생한 다음 복구가 시도되는 알고리즘을 적용하는 경우에는 복구 시도 시점과 복구 완료 시점의 네트워크 토폴로지가 같을 확률이 적기 때문에 잘못된 복구가 될 것이다. 잘못된 경로 복구로 인하여 또 다시 경로 단절 메시지를 출력하고 노드는 또 다시 복구를 시도할 수밖에 없다. 최악의 경우 경로 복구만을 반복하는 경우가 발생하여 통신이 전혀 이루어지지 않는 상태가 될 수 있다. 고속으로 변화하는 노드가 포함된 Ad-hoc 네트워크에서는 경로의 단절에 따른 복구가 아닌 경로의 단절 자체를 최소화 하는 라우팅 프로토콜의 연구가 중요하다.

또한 안정적인 Ad-hoc 네트워크의 실현을 위해서는 라우팅 프로토콜과 함께 검증되어야 하는 많은 기반 기술이 있다. Ad-hoc 네트워크를 위한 기반 기술을 신뢰성 있게 검증 할 수 있는 환경은 실제 구현된 Ad-hoc 네트워크 테스트 베드이다.

Ad-hoc 네트워크 테스트 베드의 구현에 있어 중요한 작업은 노드의 역할을 수행할 하드웨어를 설계하고 개발하는 것이다. 노드의 역할을 수행할 하드웨어는 라우팅 프로토콜과 기반 기술의 모든 이벤트를 성능 저하 없이 보여 줄 수 있어야 하기 때문에 고성능의 하드웨어가 필요하다. 소프트웨어의 관점에서 라우팅 프로토콜은 네트워크 디바이스 드라이버와 상호 보완 관계를 갖기 때문에 운영체제 레벨에서의 디버깅이 필요하다. 그러나 어떠한 운영체제도 검증하고자 하는 라우팅 프로토콜에 대한 메시지와 커널 이벤트를 제공하지 않는다. 따라서 Ad-hoc 네트워크의 기반 기술을 테스트하고 평가할 수 있는 테스트 베드의 구현은 매우 중요하다.

본 논문에서는 고속으로 변화하는 Ad-hoc 네트워크 토폴로지에 대하여 노드가 빠르게 적응하여 최적의 경로를 확보하고, 경로 단절에 따른 경로의 복구 관점보다는 경로 단절 자체를 최소화 할 수 있는 AODV 기반의 라우팅 프로토콜을 제안하고 ns2 시뮬레이터를 통하여 성능을 평가하였으며, Ad-hoc 네트워크를 위한 라우팅 프로토콜과 기반 기술을 탑재하여 실험하고 평가할 수 있는 효율적이고 고성능의 임베디드 시스템을 설계하고 구현하였다.

II. 라우팅 프로토콜 제안

2.1 라우팅 프로토콜 제안

그림 2.1(a)은 AODV를 사용하는 Ad-hoc 네트워크에서 원천 노드①이 목적지 노드⑥에 대하여 설정된 경로는 ①⇔②⇔③⇔④⇔⑤⇔⑥이며, 노드⑦은 화살표 방향으로 고속 이동하고 있는 시나리오이다. 노드⑦은 화살표 방향으로 이동을 하여 그림 2.1(b)와 같은 상태가 되었을 때 최적의 경로는 점선으로 보이는 ①⇔②⇔⑦⇔⑤⇔⑥이 되지만 AODV는 RREQ에 의해서 경로가 생성된 이후에는 링크 단절과 같은 특별한 메시지가 발생하지 않는다면 점선으로 보이는 최적의 경로로 변경하지 않는다. 최초에 설정된 경로인 ①⇔②⇔③⇔④⇔⑤⇔⑥를 이용하여 통신을 하기 때문에 AODV는 항상 최적의 경로를 갖는다는 보장을 할 수 없다.

그림 2.2는 경로 단절의 관점에서 AODV를 평가하기 위한 그림으로 노드①은 목적지 노드⑤에 대하여 설정된 경로는 ①⇔②⇔③⇔④⇔⑤이며, 노드⑥은 화살표 방향으로 C 위치까지 고속 이동하고 있는 시나리오이다. 노드①이 A 위치에 도달하였을

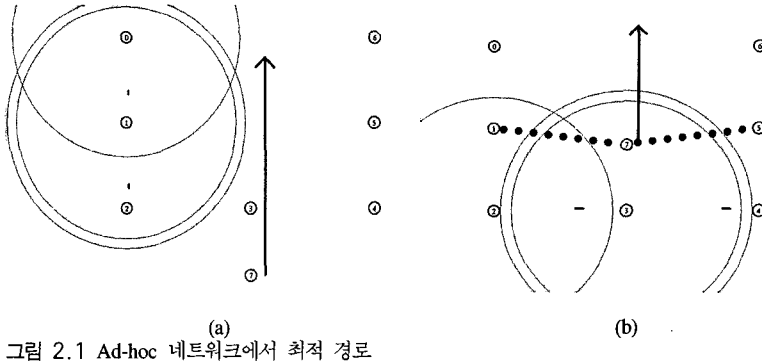


그림 2.1 Ad-hoc 네트워크에서 최적 경로

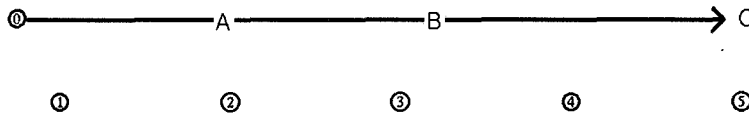


그림 2.2 Ad-hoc 네트워크에서 경로의 단절

시점에 ①⇔②의 통신 유효 반경이 벗어나게 되어 RERR 메시지가 발생하게 되고, 경로 복구 알고리즘에 따라 복구가 시도되어 노드②이 B 위치에 도달한 시점에 복구가 완료된다고 가정하면 복구된 경로는 ①⇔②⇔③⇔④⇔⑤가 된다.

그러나 B 위치에서는 ①⇔②의 통신 유효 반경이 벗어난 경로가 존재하기 때문에 잘못된 경로 복구가 된다. 따라서 또 다시 RERR이 발생하고 경로 복구 알고리즘을 수행해야 하는 악 순환을 되풀이 하는 결과를 초래한다.

이와 같은 경우에 경로의 단절이 발생한 후 빠르게 복구하는 방법은 노드의 고속 이동을 감안하면 적당한 방법이 아니다. 노드의 고속 이동시에는 경로 단절 자체를 줄이는 관점에서 접근 하는 것이 더 옳은 방법이다. 즉, 노드①이 위치 A에 도달하기 이전에 목적지 노드⑤에 대한 최적 라우팅 경로인 ①⇔②⇔③⇔④⇔⑤를 스스로 발견하여 변경하였다면 링크 단절이 발생하지 않으며 전송 지연도 감소한다.

따라서 본 논문에서는 네트워크 토폴로지가 고속으로 변화하는 네트워크에서 AODV 라우팅 프로토콜이 갖는 링크 단절과 최적화 문제에 대하여, 항상 최적 경로를 확보할 수 있으며 링크 단절을 최소화할 수 있는 AODV를 기반으로 하는 효율적인 라우팅 프로토콜을 제안하였다.

제안한 라우팅 프로토콜은 네트워크 상황을 판단할 수 있는 Info 메시지와 최적 경로 변경 요청 메시지인 Adjust 메시지를 이용하였다.

2.2 제안한 라우팅 프로토콜 메시지와 동작

그림 2.3은 Info 메시지의 형식이다. Info 메시지는 최근 성공적으로 데이터 교환이 수행된 원천 노드의 주소, 목적지 노드의 주소, 목적지에 대한 다음 홉과 홉 카운터를 포함하고 있다. 제안한 라우팅 프로토콜은 메시지 오버헤드를 최소화하기 위하여 항상 Info 메시지를 브로드캐스트 하지 않는다. 자신이 직접 데이터를 포워딩 하는 경우 또는 원천 노드인 경우에는 Info 메시지를 자발적으로 브로드캐스트 하여 최근 성공적으로 수행한 원천 노드의 주소와 목적지 노드의 정보를 주변 노드에게 통보를 한다. 자신이 통신에 참여 하고 있지 않은 노드인 경우에는 Info 메시지를 자발적으로 브로드캐스트 하지 않고 수신만을 한다. 원천 노드의 주소는 IP 패킷의 헤더로부터 알아내야 하며 목적지 노드의 주소와 다음 홉, 홉 카운트는 자신의 라우팅을 통하여 얻을 수 있다. Info 메시지를 통하여 주변의 노드는 현재의 통신 상황에 대하여 알 수 있도록 하기 때문에, Info 메시지 브로드캐스트 주기는 현재 데이터를 송수신하고 주기에 따라 가변적으로 변화한다.

0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Type				Reserved								Reserved				Hop Count							
Destination IP Address																							
Originator IP Address																							
Next Hop Address																							

그림 2.3 Info 메시지 형식

0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Type				Reserved				Reserved				Hop Count											
Destination IP Address																							
Originator IP Address																							

그림 2.4 Adjust 메시지 형식

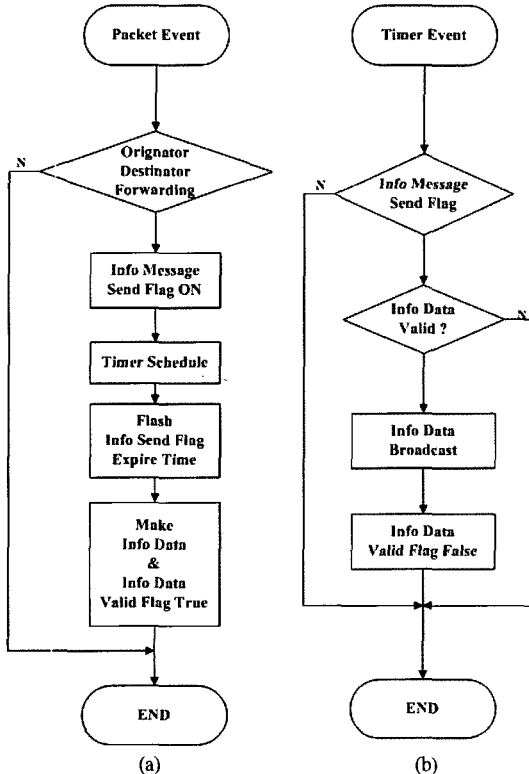


그림 2.5 Info 메시지 브로드캐스트 절차

그림 2.5(a)는 성공적으로 데이터 송수신이 완료된 후 Info 메시지를 브로드캐스트 하기 위하여 데이터를 취합하는 절차이다. 네트워크 인터페이스로부터 자신이 패킷의 포워딩 또는 목적지, 원천 노드 임은 패킷 이벤트를 통하여 판단하여 타이머 인터럽트를 스케줄링 한 후 현재 통신을 하고 있는 원천 노드의 주소, 목적지 주소, 유효시간을 Info 메시지에 기입을 한 후 브로드캐스트 플래그(Info Send Flag)를 1(True)로 한다. 데이터 취합을 마친 Info 메시지는 즉시 브로드캐스트 되지 않고 지정된 주기에 보내질 수 있도록 준비가 되어 있으며, 타이머 이벤트가 발생하게 되면 Info 메시지의 브로드캐스트 여부를 판단하여 동작한다. 브로드캐스트 플래그는 자신이 통신에 참여 하고 있지 않은 경우에는 0(False)으로 설정되어 있으며 실제 통신에 참여하고 있는 노드인 경우에는 1(True)로 설정되어 있다.

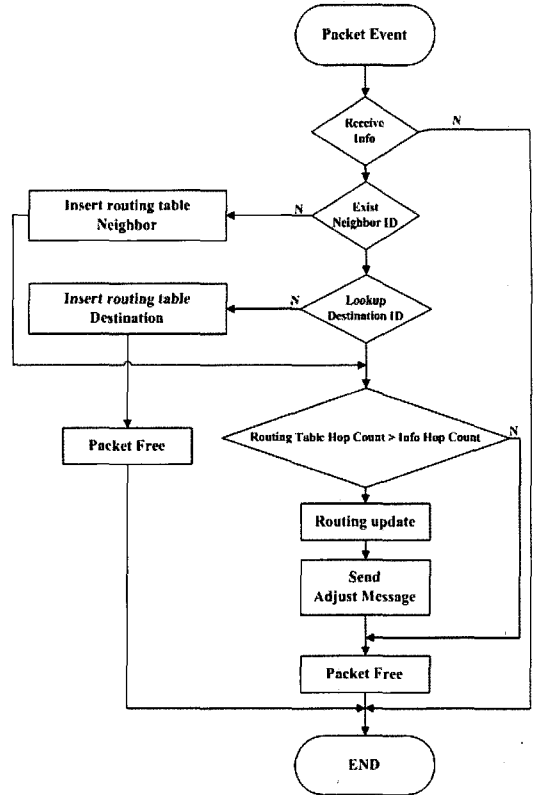


그림 2.6 Info 메시지 처리 절차

그림 2.5(b)는 설정된 타이머 이벤트를 보여주는 그림으로 브로드캐스트 플래그가 0 으로 설정되어 있는 경우에는 아무런 처리 루틴을 갖지 않는다. 브로드캐스트 플래그가 1로 설정되어 있는 경우에는 현재 가지고 있는 데이터가 유효한 데이터임을 표시하는 플래그 (Info Data Valid Flag)를 참조하여 Info 데이터의 브로드캐스트 여부를 판단한다.

그림 2.4는 Adjust 메시지 형식을 나타내며, 그림 2.6은 Adjust 메시지를 받은 노드에서의 처리 절차이다. Adjust 메시지를 받은 노드는 자신의 라우팅 테이블에서 목적지 경로를 검색 한다. 만약 목적지 경로가 존재하지 않는다면 Adjust 패킷은 폐기가 되고 경로가 존재하는 경우에는 홉 카운터 비교를 통하여 수신한 홉 카운터가 더 작다면 라우팅 테이블을 갱신하여 즉시 경로를 변경 한다. 경로를 변경한 노드는 현재 통신에 참여하고 있지 않는 노드일 확률도 있다. 이 경우에 현재 통신에 참여하고 있지 않았던 노드는 Info 메시지를 브로드캐스트 하고 있지 않았으나, 데이터 패킷의 송, 수신 또는 포워딩으로 브로드캐스트 플래그를 1로 설정되기 때문에 노드는 Info 메시지 브로드캐스트를 시작한다. 통신

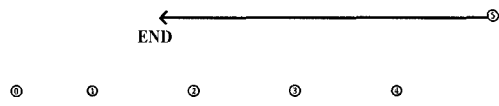


그림 2.7 링크 단절 측정을 위한 시나리오 1

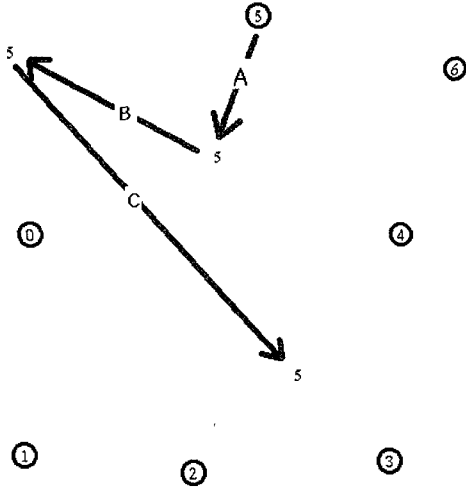


그림 2.8 경로 최적화 측정을 위한 시나리오 2

경로에서 제외된 노드는 Info 브로드캐스트 플래그 타이아웃이 발생하여 자발적인 Info 메시지 브로드캐스트를 중단한다. Adjust 메시지는 라우팅 메시지로 인한 오버헤드와 대역폭의 절약을 위하여 어떠한 경우에도 포워딩 되지 않는다.

2.3 시뮬레이션 시나리오

본 논문에서는 C++를 이용하여 제안한 라우팅 프로토콜을 작성을 하였으며 작성된 라우팅 프로토콜의 평가를 위하여 두 개의 Tcl 시나리오 스크립트 파일을 작성하였다.

그림 2.7은 제안한 라우팅 프로토콜에 대한 많은 경로 단절을 측정하기 위한 시나리오를 나타내는 그림이다. 노드⑤는 목적지 노드④와 통신을 하고 있으며, 노드⑤는 화살표 방향으로 END 위치까지 이동하도록 설정 하였다. 이동 속도는 자동차의 고속 이동을 감안하여 시속 144Km/h로 이동 한다. 시나리오 1은 노드⑤와 노드④의 통신 반경이 벗어나기 시작하면서 링크 단절이 유발되어 END 위치까지 지속된 링크 단절이 발생하도록 작성되었다.

그림 2.8은 제안한 라우팅 프로토콜의 최적 경로의 생성을 측정하고 이에 따른 패킷의 처리량 증가에 관점을 둔 시나리오를 보여주는 그림이다. 노드

⑤는 이동노드로 설정되어 있으며, 시뮬레이션 시작 후 시간 10초에 A 방향으로 이동을 한 후, 20초에 B 방향으로 이동을 한다. 마지막으로 35초에 C 방향으로 이동을 하는 시나리오이다. 제안한 라우팅 프로토콜은 노드⑤가 A 방향으로 이동하였을 경우에 최적의 경로인 ⑤⇔⑥⇔④로 변경되어야 하며 B 방향으로 이동하였을 경우에는 최초에 설정된 경로인 ⑤⇔①⇔②⇔③⇔④⇔⑥의 경로로 바뀌어야 하며, C 화살표로 이동에 따라 ⑤⇔⑤⇔④⇔⑥의 최적 경로로 변경되는 것을 확인하고자 하는 시나리오이다. 시나리오 1, 2에서 노드의 이동 속도는 고속으로 이동하는 자동차의 속도를 감안하여 144Km/h로 이동하며, 생성된 경로의 역 경로를 검증하기 위하여 양방향 통신을 해야 하는 TCP로 설정하였다. 시뮬레이션에서 사용한 MAC은 IEEE 802.11b를 사용하였다. 시나리오 1에서 노드 5는 END 위치까지 이동하는데 걸리는 시간이 불과 17초 정도밖에 되지 않기 때문에 시나리오 1의 시뮬레이션 시간은 50초간 수행하였으며 시나리오 2의 시뮬레이션 시간은 100초간 수행하였다. 제안한 라우팅 프로토콜이 정해진 스펙에 따라 정상 동작하게 된다면 데이터 처리량과 각 노드에서 포워딩 데이터가 감소할 것이다.

2.4 시뮬레이션 결과

제안한 라우팅 프로토콜은 AODV를 기반으로 하고 있기 때문에 AODV 프로토콜과 비교 하였다. 그림 2.9는 시나리오 1의 시뮬레이션 결과로 시간에 따른 원천 노드⑤에서 목적지 노드④으로 전달된 TCP 트래픽 처리량을 나타내는 그림으로 MOD_AODV는 제안한 라우팅 프로토콜이고 기존 AODV이다. 약 20초 전까지는 노드의 이동량이 떨어지기 때문에 제안한 라우팅 프로토콜의 오버헤드로 인하여 기존 AODV보다 패킷의 처리량이 떨어짐을 볼 수 있으나 노드⑤의 이동에 따라 경로의 단절이 발생한다. 시나리오 1에 의한 패킷 처리량은 기존 AODV에 비하여 제안 라우팅 프로토콜은 약 34.58% 증가하였다. 그림 2.10은 시나리오 1에서 중간 노드의 포워딩 패킷을 보여주는 그림이다. 각 노드에서 발생하는 데이터 포워딩은 시나리오에 따라 변화가 일정하지 않기 때문에 정량화가 힘들다. 그러나 기존 AODV는 END 위치에서 경로의 단절이 발생하지 않기 때문에 ⑤⇔①⇔②⇔⑤의 경로로 통신을 하고 제안한 AODV는 ⑤⇔①⇔⑤인 최적 경로로 통신을 하기 때문에 노드①과 노드②의 포워딩

개수의 차이가 큼을 알 수 있다. 또한 각 노드는 기존 AODV 보다 더 적은 데이터 포워딩의 개수를 가짐으로 전력 소모면에서도 더 나은 성능을 보임을 그림 2.10을 통하여 알 수 있다.

그림 2.11은 시나리오 2의 데이터 처리량을 나타내는 그림이다. 제안한 라우팅 프로토콜은 기존 AODV에 비하여 데이터 처리량이 23.6% 증가하였다. 시나리오 2는 경로의 단절을 발생하지 않기 때문에 기존 AODV는 최초 생성된 경로인 ①⇔②⇔③⇔④⇔⑤의 경로를 변경 없이 이용한다. 그림 2.8에서 화살표 A에 의해서 이동이 되었을 때 이용할 수 있는 최적 경로는 ①⇔⑤⇔⑥이며 그림 2.11을 통하여 데이터 처리량이 급격히 증가되는 결과로 최적 경로로 변경되었음을 알 수 있다. 그림 2.8의 화살표 B에 의한 결과로 그림 2.11의 20초 근방에서 또 다시 데이터 처리량이 증가하는 것을 볼 수 있다. 그림 2.12는 통신에 참여하고 있는 노드의 데이터 패킷 포워딩을 나타내었다. 기존 AODV는 링크의 단절과 같은 특수한 상황이 발생하지 않기 때문에 최초 설정한 경로를 변경하지 않기 때문에 노드⑤의 데이터 포워딩의 개수는 0이다. 그림 3.22에서 기존 AODV는 노드①,②,③,④의 데이터 포워딩 개수가 제안한 라우팅 프로토콜에 비하여 더 높게 표시되고 있다. 그림 2.12를 통하여 통신에 참여하고 있는 노드의 데이터 포워딩이 감소하였음에도 그림 2.11의 총 데이터 처리량은 증가함을 알 수 있다.

제안한 라우팅 프로토콜을 ns2 시뮬레이터를 이용하여 분석한 결과 데이터 처리량은 시나리오 1과 2에 대하여 각각 34.58%, 23.6% 증가되었음을 확인하였다. 또한 목적지 경로에 대한 최적의 경로를 이용하기 때문에 통신에 참여하고 있는 노드의 전력 소모를 감소시킬 수 있음을 알 수 있다.

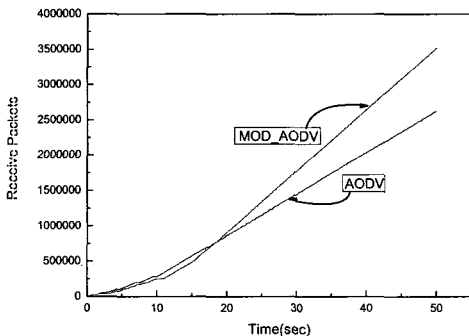


그림 2.9 TCP 패킷 처리량(시나리오 1)

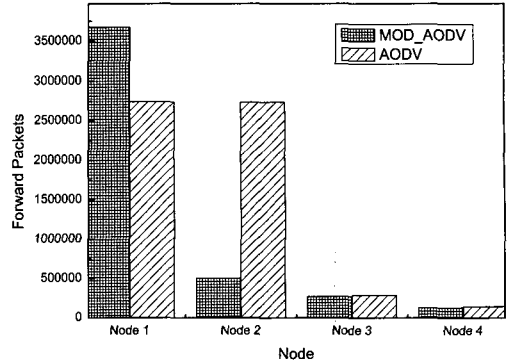


그림 2.10 데이터 포워딩 개수(시나리오 1)

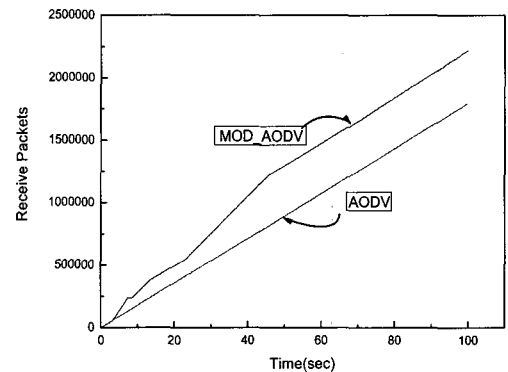


그림 2.14 TCP 패킷 처리량(시나리오 2)

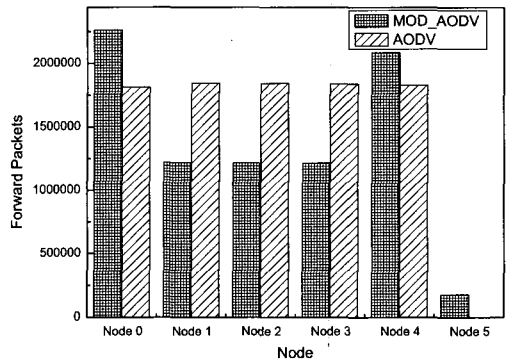


그림 2.15 데이터 포워딩 개수(시나리오 2)

III. Ad-hoc 네트워크의 테스트 베드 구현

3.1 Ad-hoc 네트워크의 테스트 베드 구현

Ad-hoc 네트워크를 실현하기 위해서는 라우팅 프로토콜과 함께 많은 기반 기술을 필요로 한다. Ad-hoc 네트워크에 필수 기반 기술의 검증은 수행할 수 있는 가장 신뢰성 있는 환경은 실제의 Ad-hoc 네트워크를 구축한 후 테스트 하는 것이다. 테스트 베드를 이용한 Ad-hoc 네트워크의 기반 기

술 검증은 시물레이션에서 미처 생각하지 못한 여러 가지 특수한 상황을 제공하여 주기 때문에 시물레이션의 결과보다 더 정확한 검증이 이루어 질 수 있다. Ad-hoc 네트워크의 테스트 베드의 구현을 위한 가장 중요하고 어려운 단계는 노드의 역할을 수행할 하드웨어의 구현이다. 노드의 역할을 수행할 하드웨어는 Ad-hoc 네트워크에 사용되는 기술의 평가를 위한 테스트 베드에 사용되므로 라우팅 프로토콜과 부가 소프트웨어의 구동에 지장을 주지 않으면서 다양한 이벤트와 여러 메시지를 출력할 수 있어야 하는 고성능의 시스템이어야 한다.

따라서 본 논문에서는 효율이 높고 고성능의 임베디드 시스템을 설계 구현하였다. 그림 3.1은 구현된 임베디드 시스템의 블록도이다. 구현된 임베디드 시스템의 OS는 커널이 공개되어 수정이 가능한 임베디드 리눅스를 포팅 하였다. Ad-hoc 네트워크와 기존의 인터넷 망과의 연동을 위한 소프트웨어로 NAT를 구현하였으며, 커널 레벨에서 프로토콜과 무선 네트워크 메시지의 디버깅을 위하여 Netfilter를 구현하여 사용하였다. Ad-hoc 라우팅 프로토콜은 AODV를 사용하였으며 커널 영역에서 동작하도록 구현되어져있다. 그림 3.2는 구현된 하드웨어를 보여주는 그림이다.

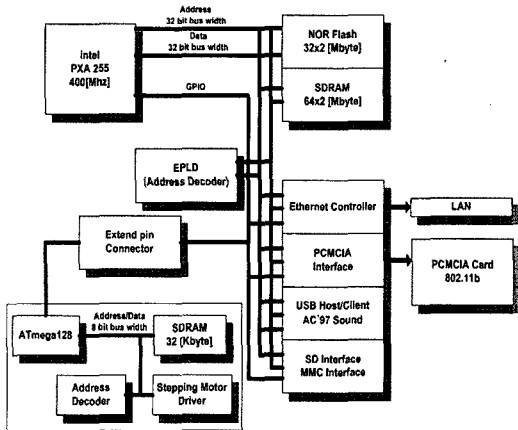


그림 3.16 구현한 임베디드 시스템 블록도

3.2 구현된 Ad-hoc 네트워크 테스트 결과

그림 3.8은 구현된 하드웨어를 이용하여 Ad-hoc 네트워크를 구성한 그림이다. 4개의 노드는 서로 데이터를 포워딩 할 수 있는 구조로 되어 있으며, 한 개의 노드는 기존 인터넷 망과의 연동을 위한 게이트웨이의 역할을 수행하는 노드로 구성되어 있다.

그림 3.9는 Ad-hoc 네트워크와 외부의 인터넷의 접속 경로를 보여주는 그림이다. 그림 3.10은 ad-hoc 네트워크와 외부의 인터넷의 접속 결과를 보여주는 그림이다. Ad-hoc 내부에 존재하는 192.168.0.81 노드에서 telnet을 통하여 외부의 네트워크에 있는 나우누리 서버에 접속을 하는 절차를 그림 10에서 보여주고 있다. 먼저 접속하고자 하는 외부의 네트워크의 IP를 알기 위하여 네임 서버의 접속을 시도하기 위한 RREQ를 브로드캐스트 한다. 브로드캐스트된 RREQ에 대한 응답을 게이트웨이로 설정되어진 노드인 192.168.0.84 노드에서 자신이 네임서버 경로를 알고 있기 때문에 RREP로 응답을 한다. 게이트웨이로 설정된 192.168.0.84는 자신의 유선 네트워크를 통하여 네임서버와 192.168.0.81의 데이터를 포워딩 해 준다. 네임 서버로부터 IP를 얻은 후에는 IP를 이용하여 나우누리 서버를 접속하여 ID를 요구하는 메시지를 보여주고 있다. 그림 10의 테스트 결과에 따라 Ad-hoc 네트워크와 외부의 인터넷과의 게이트웨이 역할을 수행하는 NAT와 AODV의 정상 동작을 확인할 수 있으며, AODV에서 기존의 인터넷 서비스인 telnet이 정상 동작하고 있음을 보여주고 있다.

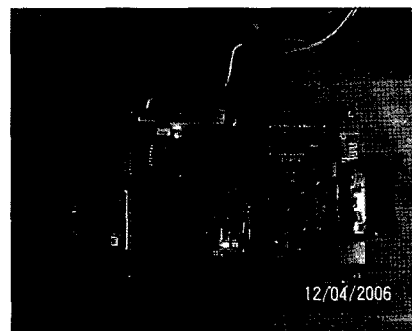


그림 3.17 구현한 임베디드 시스템 보드

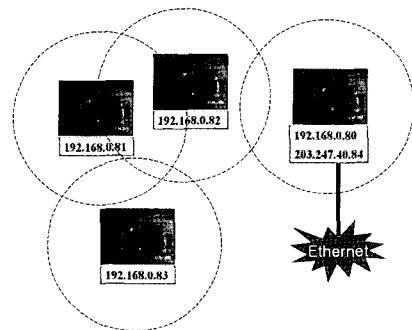


그림 3.8 Ad-hoc 네트워크 구현

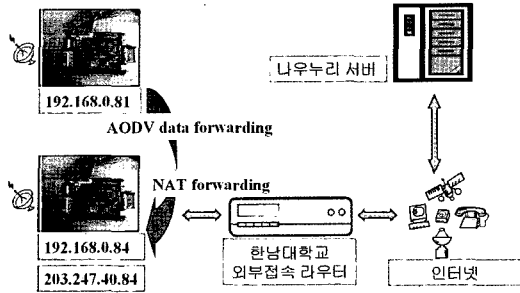


그림 3.9 Ad-hoc 네트워크와 외부 네트워크와의 연동

이스 드라이버는 임베디드 시스템의 pxa255의 GPIO 포트에 연결하여 이동체의 CPU인 Atmega128에 명령을 주어 동작하는 구조로 개발 되었다.

그림 3.13은 임베디드 시스템에서 이동체 제어를 위한 프로그램이 동작하는 그림이며, 서버 IP: 203.247.40.46, 포트:33, 모터 속도:10으로 설정하여 서버에 접속하기 위한 명령인 'vic.out 203.247.40.46 33 10'을 실행한 그림이며, AODV 라우팅 프로토콜에 의해서 경로가 설정되어 서버로부터 수신된 "Move Down" 명령을 수행한 결과이다.

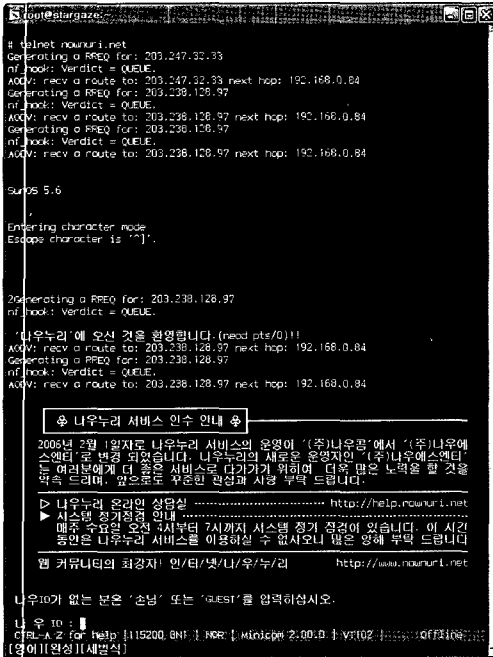


그림 3.10. Ad-hoc 네트워크와 인터넷의 연동

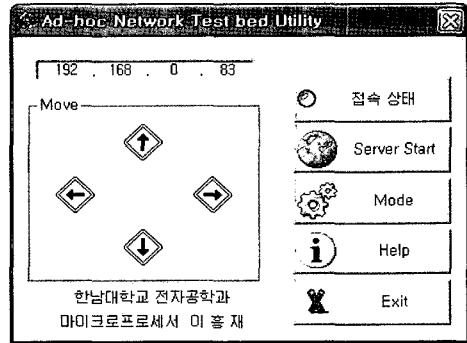


그림 3.11 이동체 제어를 위한 윈도우 어플리케이션

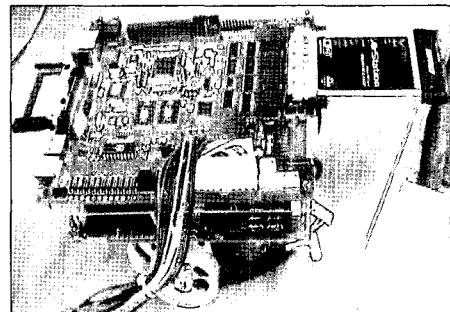


그림 3.12 개발 완료된 이동체

3.3 이동체 제어

Ad-hoc 네트워크의 완전한 검증을 위하여 실제 움직이는 이동체를 구성 하고 서버, 클라이언트의 네트워크 프로그램을 작성하였다. 그림 3.11은 윈도우에서 동작하는 이동체 제어용 프로그램이다. 프로그램은 Ad-hoc 네트워크내의 노드로 부터는 서버 모드로 동작을 하고 Ad-hoc 게이트웨이로 동작하는 노드와는 서버 또는 클라이언트 모드로 동작할 수 있도록 개발되었다. 이동체는 Atem1사의 Atmega128 CPU를 이용하여 바이폴러 방식의 스텝 모터를 동작할 수 있도록 설계 개발 하였으며 그림 3.12는 개발된 이동체를 보여주는 그림으로, 드라이버 보드와 임베디드 시스템 그리고 드라이버 제어를 위한 CPU 보드로 구성되어 있다. 개발된 디바

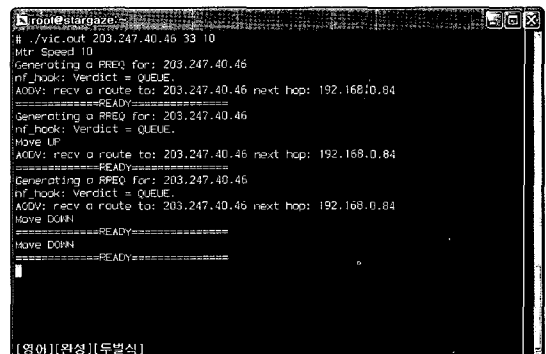


그림 3.13 이동체 제어 서버에 접속된 결과

IV. 결론

본 논문에서는 Ad-hoc 네트워크에서 노드가 고속으로 이동할 때 발생하는 경로 단절 문제를 개선할 수 있는 AODV 기반의 라우팅 프로토콜을 제안하였다. 또한 본 논문에서는 Ad-hoc 네트워크의 테스트 베드 구현을 위한 효율적이고 고성능의 시스템 구현을 위하여, 32 비트 pxa255 프로세서를 이용한 고성능 임베디드 시스템을 설계하고 구현하였다.

제안한 라우팅 프로토콜은 주변 노드의 변화를 포함하고 있는 Info 메시지와 최적 경로의 확보를 유도하는 Adjust 메시지를 이용하여 최적 경로의 확보와 경로 단절을 최소화할 수 있도록 하였다. 제안한 라우팅 프로토콜은 ns2 네트워크 시뮬레이터에 의해 분석을 하였으며, 분석된 결과 많은 링크 단절을 유도한 시나리오 1과 경로 최적화 측정을 위한 시나리오 2에 대하여 각각 34.58%, 23.6% 데이터 처리량이 증가함을 보였다. 따라서 제안한 라우팅 프로토콜은 고속으로 이동하는 노드의 링크 단절과 최적 경로 확보에 효과적이다.

구현된 임베디드 시스템은 PCMCIA, 유선 이더넷, 고용량의 메모리가 탑재된 시스템으로 Ad-hoc 네트워크 내에서 노드의 역할과 게이트웨이 역할을 수행할 수 있다. 구현된 임베디드 시스템에 탑재된 소프트웨어는 임베디드 리눅스를 운영체제로 하였으며, NAT, Netfilter/iptables 등을 탑재하여 Ad-hoc 네트워크에 최적화 되도록 하였다. 구현한 테스트 베드에서 AODV 라우팅 프로토콜의 정상 동작과 외부 인터넷과의 연동을 테스트하고 정상 동작됨을 확인하였다.

본 논문에서 구현된 임베디드 시스템은 Ad-hoc 네트워크를 위한 장비의 기반 하드웨어, 소프트웨어로 사용될 수 있을 것으로 기대되며, 또한 구현된 Ad-hoc 네트워크 테스트 베드는 본 논문에서 테스트된 것과 다른 라우팅 프로토콜 등 기반 기술의 검증에 충분히 활용될 수 있을 것이다.

참고 문헌

[1] C. E. Perkins, E. M. Royer, "Ad hoc On-Demand Distance Vector (AODV) Routing," draft-ietf-manet-aodv-13.txt, IETF, Feb. 2003.

[2] MANET, <http://www.ietf.org/html.charter/manet-charter.html>
 [3] C. E. Perkins, "AD HOC Networking," Addison Wesley, 2001.
 [4] C. K. Toh, "Ad hoc Mobile Wireless Networks(Protocol and Systems)," Prentice Hall, 2002.
 [5] S. J. Lee, M. Gerla, "AODV-BR: Backup Routing in Ad hoc Networks," IEEE WCNC 2000, Sep. 2000.

이 흥 재 (Heung-jae Lee)

정회원



1999년 2월 한남대학교 전자공학
학과 학사
2001년 2월 한남대학교 전자공
학과 석사
2006년 8월 한남대학교 전자정
보통신공학과 박사
현재 (주)바이오니아 연구원

<관심분야> ad-hoc, 라우팅프로토콜, 임베디드

가 순 모 (Soon-mo Ga)

정회원



2004년 2월 한남대학교 전자공
학과 졸업
2004년 3월~현재 한남대학교 전
자공학과 석사과정
<관심분야> 임베디드, 원격제어

최 진 규 (Jin-kyun Choe)

종신회원



1980년 고려대학교 전자공학과
학사
1982년 고려대학교 전자공학과
석사
1987년 고려대학교 전자공학과
박사
1987년 9월~1990년 8월 대전공

업대학 조교수

2005년~2006년 미국 University of Arizona 방문교수
현재 한남대학교 전자공학과 교수