

논문 2006-43TC-1-9

애드혹 네트워크를 위한 DYMO 프로토콜 구현 및 적합성 검증

(Implementation and Conformance Test of DYMO Protocol for
Ad-Hoc Networks)

박 일 균*, 곽 정 남*, 김 영 한**

(Ilkyun Park, Jungnam Kwak, and Younghan Kim)

요 약

MANET 네트워크 모델을 위한 라우팅 기법은 네트워크 인프라를 요구하지 않는 무선 네트워크 기능, 이동성 지원, 그리고 동적인 라우팅 정보 관리 기능을 필요로 하며, 따라서 USN에 적용 가능한 라우팅 기법 중의 하나가 되고 있다. 제안된 MANET 라우팅 기법들 중에서는 DYMO 라우팅 프로토콜이 네트워크 부하가 작고 간단하며 확장성이 향상되어 표준화가 진행 중에 있다. 본 논문에서는 다양한 운영체제 환경에서 동작되도록 DYMO 라우팅 프로토콜을 설계 구현하였으며, 구현된 DYMO의 적합성을 검증하는 적합성 테스트 도구를 개발하여 구현 내용을 검증하였다.

Abstract

MANET routing protocols must support not only wireless networking without any relaying on network infrastructure, but also dynamic management of routing information caused by node mobility. Hence, they are one of important routing protocols for USN. Recently DYMO is regarded as a promising routing protocol for MANET because it is simple and easy to extend as well as it requires less networking load than others. In this paper, we design and implement DYMO routing protocol into various operation systems. Also, we develop a DYMO conformance test tool to evaluate our implementations.

Keywords: 무선 이동 애드혹 네트워크, DYMO 라우팅 프로토콜, 적합성 검사

I. 서 론

MANET(mobile ad-hoc network)은 사용자 노드들만으로 구성할 수 있는 무선 네트워크이다. MANET의 각 사용자 노드들은 유선 네트워크에서의 라우터처럼 다른 노드의 패킷을 수신해서 이웃 노드에게 전달해줄 수 있다. 이 때 노드들의 이동성으로 인해 전달 경로가 동적으로 변화하고, 유선 네트워크에 비해 네트워크 자

원이 상대적으로 한정되어 있기 때문에 기존 유선 네트워크에서의 라우팅 프로토콜 대신에 MANET 네트워크 모델에 적합한 라우팅 프로토콜들이 제안되었다.

MANET 라우팅 기법 중에서 reactive 라우팅 방식은 패킷을 보내기 전에 경로 정보 요청 메시지를 이용해 라우팅 정보를 획득하기 때문에 사전에 라우팅 정보 교환을 통해 미리 준비해놓고 있는 proactive 방식에 비해 라우팅 정보 오버헤드가 적고, 필요할 때에만 라우팅 메시지를 교환하여 네트워크 부하도 상대적으로 작다는 이점이 있다^[1]. IETF((internet engineering task force))에서는 reactive 라우팅 방식 중에서 AODV(ad-hoc on-demand vector) 프로토콜^[2]을 experimental RFC의 형태로 표준화를 완료했으나 standard RFC를 위해

* 학생회원, ** 평생회원, 숭실대학교 정보통신전자공학부
(School of electronic engineering, Soongsil University, Korea)

※ 본 연구는 일부 숭실대학교 교내 연구비 지원에 의해 이루어졌다.

접수일자: 2005년12월15일, 수정완료일: 2005년1월19일

AODV 프로토콜의 기능을 간소화하고 확장성을 향상시킨 DYMO(dynamic manet on-demand) 라우팅 프로토콜^[3]의 표준화를 진행하고 있다.

본 논문에서는 현재 표준화가 진행 중인 DYMO 라우팅 프로토콜을 구현하여 동작을 검증하였으며 표준과의 적합성 시험을 수행하였다. 본 논문의 구성은 다음과 같다. II장에서는 DYMO 라우팅 프로토콜에 대해 설명한다. III장에서는 DYMO 라우팅 프로토콜의 구현 구조에 대해 설명하고, III장에서 DYMO 라우팅 프로토콜을 위한 적합성 검증 기법에 대해 다룬다. IV장에서 DYMO 라우팅 프로토콜의 동작 실험 및 적합성 실험을 다루고, 마지막 V장에서 결론을 내린다.

II. DYMO 라우팅 프로토콜

DYMO 라우팅 프로토콜은 AODV에 비해 메시지의 종류가 간소화되어 경로 요청(RREQ, route request) 및 응답(RREP, route response) 메시지들이 RE (routing element) 메시지로 통합되었다. 또한 이웃 노드의 정보를 얻어오는데 필요한 Hello 메시지를 삭제하였다. DYMO 메시지 내의 각 정보 영역을 TLV(type-length-value) 형식으로 바꾸고, 지원하지 않는 메시지 형식 등 예외 상황에 대한 처리 방법을 구체적으로 정의함으로써 라우팅 기능의 확장성을 향상시켰다.

DYMO는 라우팅 정보 오류로서 전달해야 할 데이터 패킷의 목적지에 대한 라우팅 정보가 없을 때 송신자에게 RERR(routing error) 메시지를 전송한다. 이 메시지를 수신한 중간 노드들은 자신의 라우팅 테이블에서 해당 노드에 대한 라우팅 정보를 삭제한다.

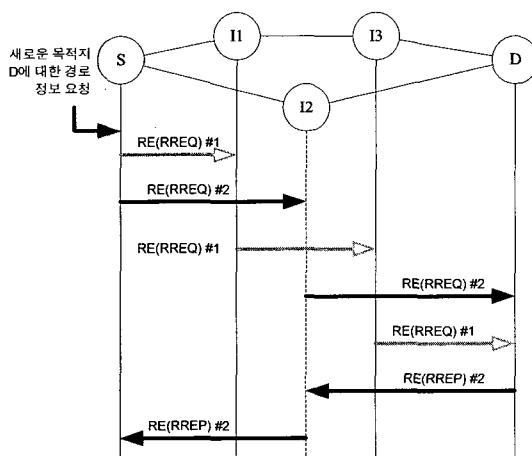


그림 1. Reactive 기반의 경로 설정 방법
Fig. 1. Reactive-based path setup.

```

for each element[i] in received_dymo_packet:
    element.TTL = element.TTL - 1           // pre-processing
    element.type:                           // element-processing
        case RE:
            call re_handler()
        case RERR:
            call rerr_handler()
        case UERR:
            call uerr_handler()
        if element[i].TTL == 0:                // post-processing
            if element[i] is first_element_of_dymo_packet:
                drop dymo_packet
            else
                drop element[i]
        end;
    end;
    retransmit dymo_packet

```

그림 2. 수신된 DYMO 메시지의 처리 과정

Fig. 2. Handling procedure for received DYMO message.

그림 1에서와 같이, S 노드에서 새로운 목적지 D에 대한 경로 정보를 필요로 하는 트래픽이 발생하면 경로 요청 메시지를 브로드캐스트 패킷에 실어 전송한다. 이 메시지는 이웃 노드 I1, I2에 각각 전송되고 이후로도 한 흡을 거쳐갈 때마다 경로 정보가 누적된다. 이 메시지들 중 하나를 수신한 D 노드는 경로 응답 메시지를 통해 이 경로 정보를 S 노드로 돌려보낸다. 그림 1의 경우 I2 노드를 거쳐 전송된 RE #2 메시지가 최적의 경로를 거쳐 전송되었기 때문에, D 노드는 RE #2에 대한 응답 메시지만을 전송한다.

다음 그림 2는 임의의 노드에서 DYMO 메시지를 수신했을 때의 처리 알고리즘을 나타낸 것이다. DYMO 메시지의 각 요소를 처리하기 전에 TTL 값을 감소시킨다. 이후에 메시지 타입 별로 처리를 한 후, TTL 값이 0인지 0이 아닌 값인지 확인 후 패킷의 삭제 및 재전송 여부를 결정한다.

III. DYMO 라우팅 프로토콜의 설계 및 구현

현재 구현된 DYMO 라우팅 프로토콜의 중 하나인 dymo-nist^[4]는 미국 NIST에서 구현 중이며 linux 운영 체제를 지원한다. DYMO의 모든 기능이 커널 모듈의 형태로 구현되어 있다. 커널 내부에서 모든 처리를 수행하므로 패킷을 가로채거나 생성하는데 필요한 오버헤드가 최소화된다는 장점이 있으나, 모든 기능을 커널 내의 복잡한 네트워크 코드 사이에 넣어야 하므로 구현, 설치 및 실험이 어렵다는 단점이 있다^[5].

또 다른 구현인 dymoum^[6]은 University of Murcia에서 작업이 진행되고 있다. 마찬가지로 linux 운영체제를 지원하며 본 논문의 구현과 유사하게 데몬 프로세스와 커널 모듈 구조로 구현되어 있다. 그러나 linux 운영체제를 위한 커널 모듈 구조와 netfilter 라이브러리만 지원하기 때문에 이식성이 떨어지고 타 운영체제에서의 동작이 검증되지 않았다는 단점이 있다.

타 기관에서의 구현과 달리 본 논문에서 구현된 DYMO 라우팅 프로토콜인 DYMO-DCN은 Linux, Windows 및 Windows CE 운영체제에서 동작할 수 있도록 개발되었다. 라우팅 정보 관리 및 목적지 별 경로 결정을 위한 사용자 프로세스와 패킷 획득 및 커널 라우팅 테이블 조작을 위한 커널 모듈 또는 네트워크 장치 드라이버로 구성되어, 커널 내에서의 수정을 최소화하고 운영체제 별로 상이한 패킷 처리 모듈을 사용하기 위해 공통 인터페이스를 정의함으로써 이식성을 높였다. 그림 3은 Linux 2.4.20 버전에 구현된 DYMO-DCN의 구현 구조도이다.

커널 공간에서는 DYMO 프로토콜의 동작을 위한 커널 모듈이 실행된다. IP 라우팅 테이블을 제어하고 패킷 송수신 이벤트를 인식하여 가로챈다. 이를 위해 커널 모듈에서는 netfilter를 이용한다. netfilter를 사용할 경우 사용자 프로세스의 작업 부하를 감소시키면서 신속한 패킷 처리가 가능해진다^[5]. netfilter에서는 패킷을 가로챌 수 있는 총 다섯 지점을 정의하며, 사용자는 이 중에서 필요한 지점 별로 처리 함수를 등록하여 사용할 수 있다. 그림 4는 커널 내부에서의 netfilter의 구조와 netfilter에서 사용하는 지점을 보여준다.

DYMO-DCN은 reactive 라우팅 프로토콜에서 요구되는 이벤트를 정의하기 위해 다음과 같이 세 군데에서 패킷을 가로챈다.

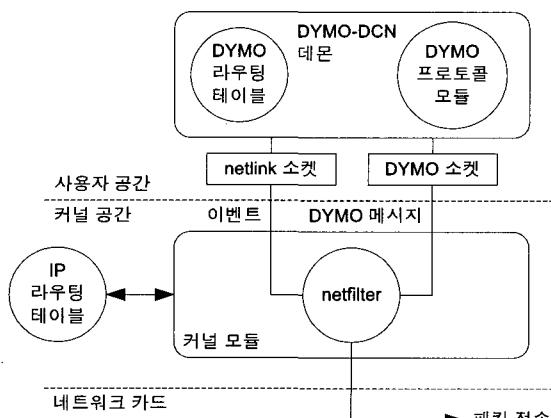


그림 3. DYMO-DCN의 구현 구조도
Fig. 3. Implementation structure of DYMO-DCN.

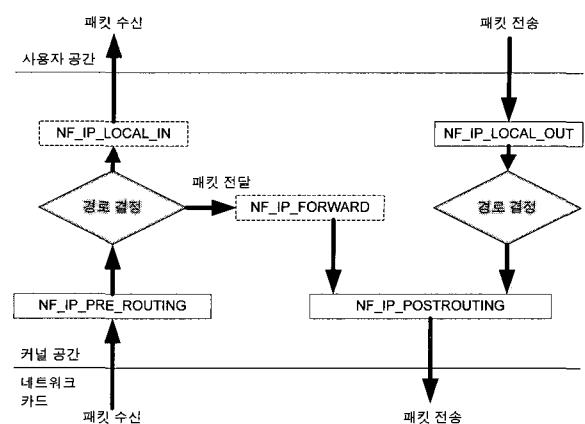


그림 4. DYMO-DCN의 구현 구조도
Fig. 4. Implementation structure of DYMO-DCN.

위와 같이 커널 내부에서 생성된 이벤트는 사용자 라우팅 프로세스로 전달되어 처리된다. DYMO-DCN은 커널 모듈과 사용자 프로세스의 이벤트 통신을 위해 커널과 응용 간의 통신을 제공하는 netlink 소켓을 이용한다. 반면 DYMO 소켓은 RE, RERR, UERR (unsupported-element error)과 같은 DYMO 메시지를 송수신하는데 사용한다.

DYMO-DCN은 DYMO 라우팅 테이블을 관리하면서 커널이 관리하는 IP 라우팅 테이블과 동기를 맞춘다. 따라서 경로 탐색 절차를 거쳐 알게 된 경로로 패킷을 전송할 때, 데몬에게 추가적인 질의 없이 커널에서 곧바로 보낼 수 있도록 한다. 또한 경로 탐색 과정 중에 있는 데이터 패킷들을 커널 모듈에서 관리하기 때문에 사용자 프로세스의 처리 부하를 경감시킬 수 있고 사용자 및 커널 공간 사이의 패킷 전달 절차를 간소화하기 때문에 처리 속도를 향상시킨다.

MANET의 토폴로지는 노드들의 이동성 때문에 동적으로 변화한다. 그러므로 DYMO 경로탐색 절차를 통해 파악된 경로 정보는 각 목적지 노드에 따라 한시적으로만 관리된다. 라우팅 데몬은 자신이 관리하는 라우팅 테이블을 갱신하고, netlink 소켓과 시스템 제어 함수인 ioctl()을 사용하여 커널 내의 IP 라우팅 테이블을 같이 갱신한다.

표 1. DYMO-DCN에서 사용하는 netfilter 이벤트
Table 1. Events of netfilter used in DYMO-DCN.

NF_IP_LOCAL_OUT	목적지에 대한 경로 정보가 없을 경우 발생하는 경로 탐색 시작 이벤트
NF_IP_PRE_ROUTING	데이터 패킷의 경로가 없을 경우 발생하는 RERR 시작 이벤트
NF_IP_POSTROUTING	테이블 갱신 이벤트

그림 5는 DYMO-DCN에서의 패킷 처리 과정을 보여 준다. 데이터 패킷과 DYMO 제어 메시지 두 가지로 구분하여, 커널 모듈에서 모든 패킷을 가로채어 검사한 결과 제어 메시지인 경우에는 커널의 기본 기능에 따라 처리한다.

DYMO 제어 메시지는 DYMO 데몬용 UDP 포트를 사용하므로 DYMO 소켓으로 전달된다. 각 DYMO 제어 메시지는 메시지의 타입에 따라 필요한 처리 함수를 호출한다. 데이터 패킷일 경우 적당한 이벤트를 생성하기 위해 이용한다. 만일 경로 탐색이 필요한 경우 패킷을 대기 버퍼에 저장하고 netlink 소켓을 통해 데몬에 경로 탐색 이벤트를 알린다. 사용자 데몬은 이벤트를 수신한 뒤, 경로 탐색을 위한 제어 메시지를 생성하고 전송한다.

Windows 및 Windows CE 환경을 위한 DYMO 구현에서 DYMO 프로토콜 모듈은 linux 기반의 DYMO-DCN을 이용하고, 하부에서 패킷 흐름을 제어 하는 모듈은 linux의 netfilter 및 커널 모듈 대신 WinPcap 및 자체 구현한 PCapMod 라이브러리를 사용 한다. 따라서 dyomoum 구현보다 이식성이 높고, 단일 DYMO 프로토콜 모듈을 다양한 운영 체제 상에서 사용 함으로써 운영체제 별로 구현 작업을 간소화 할 수 있다. 다음 그림 6은 Windows 및 Windows CE 환경에서의 DYMO 구현 구조도를 나타낸다.

다음 그림 7은 WinPcap 및 PCapMod 디바이스 드라이버에서의 패킷 처리 과정을 나타낸다. 감지된 패킷이 전송 패킷인 경우 패킷의 목적지 주소에 대한 IP 라우팅 정보가 있는지 검색한다. 있을 경우 해당 라우팅 정

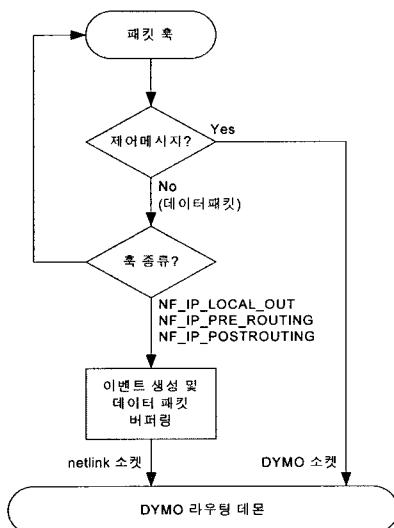


그림 5. DYMO-DCN 데몬에서의 패킷 처리
Fig. 5. Packet handling procedure in DYMO-DCN daemon process.

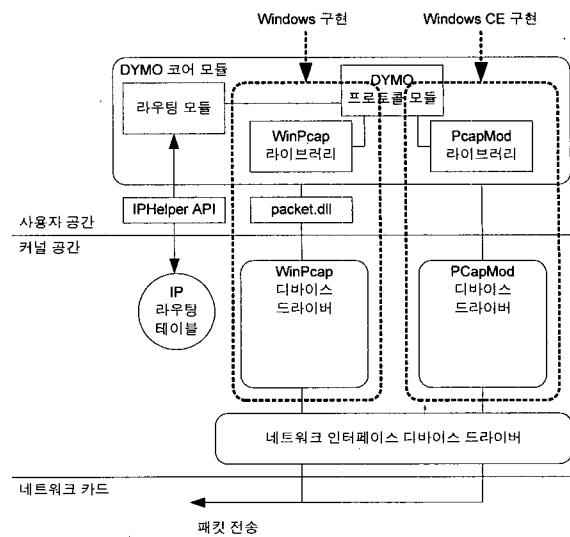


그림 6. Windows 및 Windows CE 환경에서의 DYMO 모듈 구조
Fig. 6. DYMO modules in Windows and Windows CE.

보에 따라 전송하고, 없을 경우 DYMO 라우팅 코어 모듈을 호출하여 DYMO 메시지 전송 절차를 시작한다. 만약 감지된 패킷이 수신 패킷인 경우 DYMO 라우팅 정보를 확인하고 타임스탬프 값을 갱신한다. 그리고 추가 IP 라우팅 관련 정보를 처리하기 위해 DYMO 라우팅 코어 모듈을 호출한다.

WinPCap 라이브러리는 Windows NT 계열 운영체제 전용의 라이브러리이기 때문에, Windows CE 환경을 위한 PCapMod를 자체 구현하였다. 표 2는 PCapMod 모듈에 구현된 주요 기능을 나열한다.

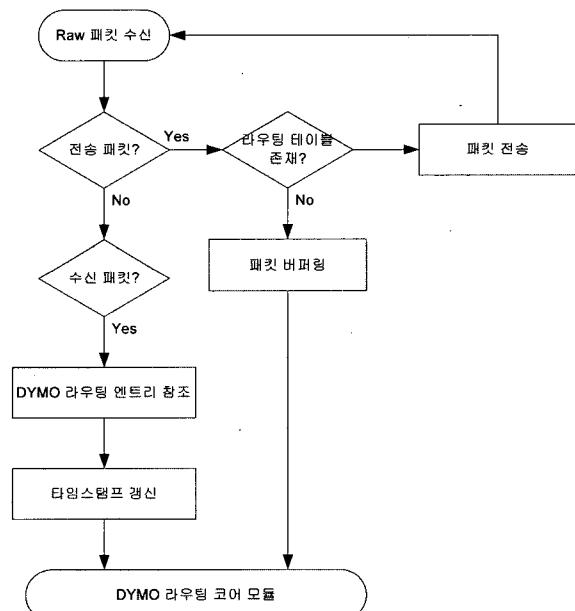


그림 7. pcap 모듈에서의 패킷 처리 과정
Fig. 7. Packet handling procedure in pcap module.

표 2. pCapMod에서 구현된 주요 라이브러리 기능
Table 2. Functions of pCapMod implementation.

LoadPCapDriver	PCapMod 드라이버를 활성화한다
UnloadPCapDriver	패킷 캡쳐를 종료하고 PCapMod 드라이버를 비활성화한다
PCapControl::newInstance	PCapMod 인스턴스를 생성한다
PCapControl::SetPacketListener	패킷 수신 메소드를 연결한다
PCapControl::start	패킷을 캡쳐하기 시작한다

IV. DYMO 라우팅 프로토콜의 기능 적합성 검사

적합성 검사(conformance test)는 구현한 기술이 표준에서 제시하는 조건을 만족하는지 여부를 검증하기 위한 것이다. 본 논문에서는 DYMO-DCN이 DYMO 라우팅 프로토콜 표준에 따라 구현되었는지와 또한 다른 DYMO 구현의 적합성 등을 검사했다.

다음 그림 8은 DYMO-DCN 모듈과 검증을 위한 검사 프로그램의 구조를 나타낸다. DYMO-DCN 라우팅 데몬 및 커널 모듈은 구현한 바와 같이 DYMO 이벤트 및 제어 메시지를 주고 받는다. DYMO-DCN 검사 프로그램은 라우팅 데몬에 의해 발생하거나 데몬으로 수신되는 DYMO 제어 메시지를 가로챈다. 또한 상대편 장비의 DYMO 동작을 검증하기 위해 직접 DYMO 제어 메시지를 생성하는 기능을 넣는다. 이외에도 DYMO 이벤트의 발생을 유도하도록 임의로 데이터 트래픽을 발생하도록 한다.

주요 검사 항목은 주고 받는 DYMO 제어 메시지의 포맷을 검증하는 형식 검증과, 메시지 헤더의 각 영역

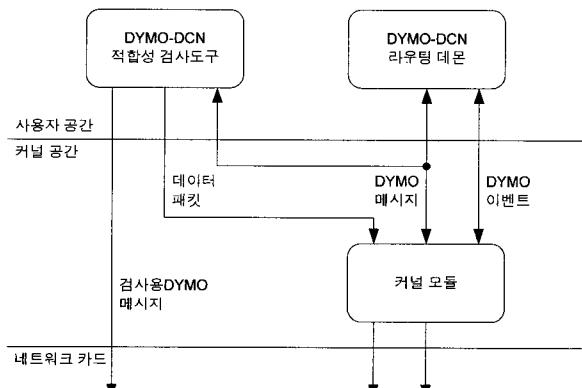


그림 8. DYMO-DCN 및 검사 프로그램의 구조
Fig. 8. Structure of DYMO-DCN and conformance test tool.

표 3. 메시지 형식 검증을 위한 적합성 검사 항목
Table 3. Conformance test list to check the format of DYMO message.

검사 항목	설명
RE 메시지 형식 검증 #1	경로 요청 메시지로서의 RE 메시지 형식을 검증한다
RE 메시지 형식 검증 #2	경로 응답 메시지로서의 RE 메시지 형식을 검증한다
RERR 메시지 형식 검증	RERR 메시지의 형식을 검증한다
UERR 메시지 형식 검증	UERR 메시지의 형식을 검증한다

내 값에 따른 기능을 검증하는 기능 검증으로 나뉜다. 주고 받는 메시지 형태의 기본 형식을 검증한 다음에, 영역 별 값 조작에 따른 DYMO 라우팅 모듈의 동작 기능을 검증한다.

(1) DYMO 메시지 형식 검증

표 3은 적합성 검사 항목 중 메시지 형식 검증에 관련된 것이다. RE, RERR, UERR 등 세 가지 종류의 DYMO 제어 메시지에 대해 검사를 수행하며, 해당 메시지의 UDP 및 IP 헤더 내 영역까지 검사 대상에 포함된다.

(2) DYMO 기능 검증

표 4는 기능 검증에 관련된 적합성 검사 항목을 나열한 것이다. 일부는 앞의 메시지 형식 검증과 중복되는 요소가 있으나, 기능 검증 단계에서는 해당 메시지의 각 영역 값들이 표준에서 요구하는 수준에 적합하다는 가정 하에 각 값에 따른 DYMO 동작을 확인한다. 각 항목별 테스트 방법 설계는 표 5에 준하는 방식으로 한다.

다음 표 5는 DYMO 기능 검사 중의 한 예로, DYMO 라우팅 프로토콜에 의해 생성된 라우팅 정보를 미리 정의된 시간 동안 데이터 패킷이 사용하지 않을 경우 해당 라우팅 정보가 삭제되는지 검증한다.

V. DYMO-DCN의 동작 실험 및 적합성 검증

DYMO-DCN 구현의 동작을 실험하기 위해 그림 9와 같은 실험환경을 구축하고, A 노드와 C 노드 사이의 펑(ping) 실험을 수행하였다. A 노드에서 C 노드에게 ICMP 메시지를 보내려 할 경우 C 노드에 대한 라우팅 정보가 없기 때문에 ICMP 메시지를 잠시 버퍼에 저장하고 DYMO RE 메시지를 브로드캐스트(broadcast) 메시지로 뿐리게 된다.

표 4. 기능 검증을 위한 적합성 검사 항목

Table 4. Conformance test list to check the function of DYMO protocol.

검사 항목	설명
M/H 비트 영역 값에 따른 기능 검증	지원하지 않는 DYMO 메시지에 대해 M/H 비트 영역에 따른 처리 기능을 검증한다.
I 비트 영역 값에 따른 기능 검증	I=1로 설정되었을 경우 지원하는 DYMO 메시지가 삭제되는지 검증한다.
A 비트 영역 값에 따른 기능 검증	A=1인 경우 메시지의 발신자로 자신의 응답 메시지를 전송하는지 검증한다.
G 비트 영역 값에 따른 기능 검증	G=1인 경우 해당 주소 값을 게이트웨이 주소로 이용하는지 검증한다.
RE 메시지의 sequence 값 증감 검증 #1	경로 요청 메시지로서의 sequence 번호의 증가 여부를 확인한다.
RE 메시지의 sequence 값 증감 검증 #2	경로 응답 메시지로서의 sequence 번호의 증가 여부를 확인한다.
RE 메시지의 sequence 값 증감 검증 #3	경로 응답 메시지로서의 sequence 번호의 증가 여부를 확인한다.
RBHopCnt 값에 따른 기능 검증	헤더 내 RBlock의 RBHopCnt 영역 값의 증가 여부를 확인한다.
경로 재탐색 기능 검증	일정 시간 동안 경로 응답이 없을 경우 경로 요청 메시지의 재전송 여부를 확인한다.
경로 삭제 기능 검증	일정 시간 동안 데이터 패킷이 없는 경로 정보에 대한 삭제 여부를 확인한다.
메시지 생성 제한 기능 검증	단위 시간 내에 DYMO 제어 메시지의 발생률이 제어되는지 확인한다.

표 5. DYMO 기능 검사 예

Table 5. Example of DYMO function test.

목적	일정 시간 동안 데이터 패킷이 이용하지 않은 경로 정보가 삭제되는지 확인한다.
실험 구성	원격 노드 테스트
실험 방법	<ol style="list-style-type: none"> 테스트 대상을 D 노드에 탑재한다. S 및 D 노드의 라우팅 테이블을 초기화한다. S와 D 노드 사이에 DYMO RE 메시지가 교환되도록 하여 라우팅 정보를 생성하도록 한다. 미리 정의된 시간 동안 패킷을 발생시키지 않고 대기한다. ROUTE_TIME (3000 msec) + ROUTE_DELETE_TIMEOUT (15000 msec) D 노드에서 S 노드로 데이터 패킷을 발생시킨다. D 노드에서 RE 메시지가 전송되는지 확인한다.
만족 요건	<p>다음 조건을 모두 만족해야 한다:</p> <ul style="list-style-type: none"> - D 노드에서 RE 메시지가 전송된다. - RE 메시지의 A-bit 값은 1이다.

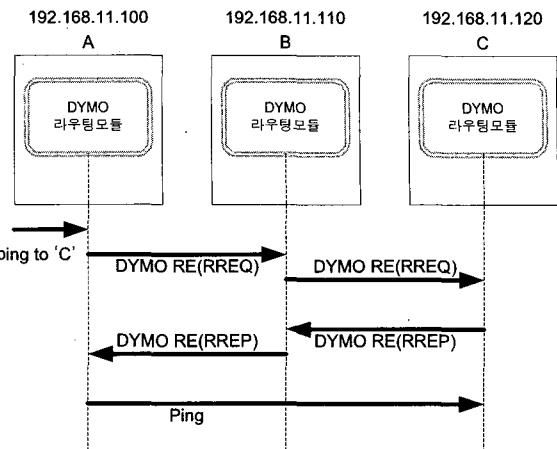


그림 9. 실험 환경 구성

Fig. 9. DYMO operation test.

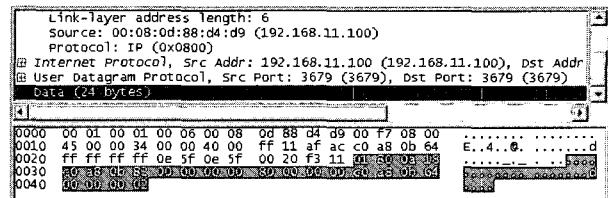


그림 10. A 노드에서 전송한 DYMO RE 메시지

Fig. 10. DYMO RE message from node A.

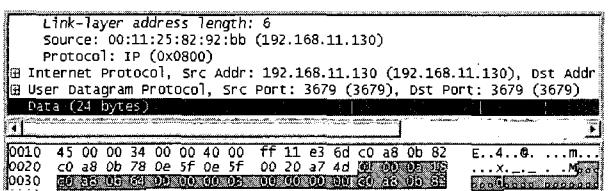
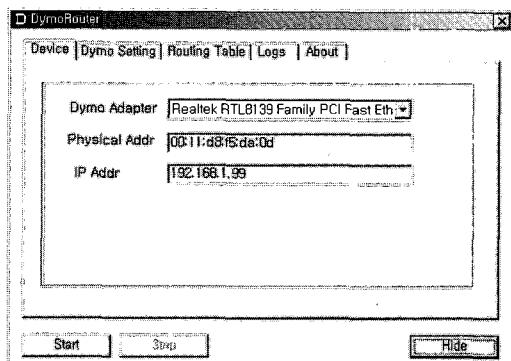


그림 11. C 노드에서 전송한 유니캐스트 DYMO RE 메시지

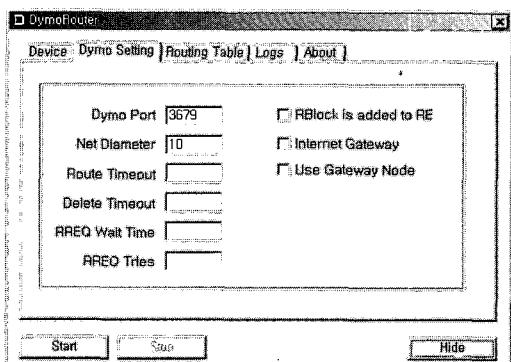
Fig. 11. Unicast DYMO RE message from node C.

DYMO RE 메시지는 B 노드를 거쳐 C 노드에 수신된다. 수신된 RE 메시지의 A 비트는 1로 설정되어 있기 때문에 C 노드는 자신의 정보를 담아서 DYMO RE 메시지로 응답해주어야 한다. 다음 그림은 C 노드에서 전송하는 메시지를 보여준다.

다음은 Windows로 구현된 DYMO 프로그램의 GUI 화면이다. 그림 12 (a)는 장치 설정 화면이다. WinPcap 라이브러리를 사용하여 패킷을 가로챈 네트워크 인터페이스를 여기에서 선택한다. 다음 그림 12 (b)는 DYMO 프로토콜과 관련된 인수 값을 설정하는 화면으로서, DYMO 라우팅 프로토콜의 UDP 포트 번호, 최대 TTL 값, 경로 정보의 유효 시간, RREP 메시지의 수신 대기 시간 및 재시도 횟수를 지정해줄 수 있다.



(a) 장치 설정 화면



(b) DYMO 프로토콜 인수 설정 화면

그림 12. Windows 용 DYMO 프로그램의 GUI 화면
Fig. 12. GUI of DYMO process for Windows.

응답을 수신하여 분석한 결과 예가 다음 그림 13에 표시되어 있다. RREP 메시지의 IP 발신지 및 목적지 주소는 각각 실험 대상 노드 및 검사도구 노드이다. RE 헤더의 ‘target address’ 영역에는 본 메시지의 수신자의 주소 정보가, 그리고 RE 헤더 다음의 routing block 내에 있는 ‘node address’ 영역에는 발신자의 주소 정보가 포함되어 있다. ‘target sequence number’ 영역 값도 표준에 의거하여 0이 아니면서 RREQ 메시지의 값보다 같거나

```
U:\parked\admit>/dev/dymo
*** PCH-DYMO CONFORMANCE TEST ***
20:56:08.451 : dymo conformance test start: scenario #11
NVM: RREQ message has been sent.
Waiting for RREP message from 192.168.2.111...
DYMO: Message is received:
IP source address: 192.168.2.111 ... ok
IP destination address: 192.168.2.90 ... ok

DYMO Routing Element:
length of message: 24 ... ok
t-bit: ... off
r-bit: ... off, RREP message
target address: 192.168.2.111
target seqno: 1
hopcount: 1
# of routing block(s): 1
  G-bit: off
  prefix: 0
  hopcount: 0
  node address: 192.168.2.111
  node seqno: 2

*** DYMO PE (with R-bit = 0) message format is verified
20:56:09.780 : dymo conformance test end.
```

그림 13. DYMO 적합성 시험 결과 예
Fig. 13. The result of conformance test.

나 크기 때문에 정상으로 검증되었다. 최종적으로 검사 대상인 RREP 메시지의 형식은 정상으로 검증되었다.

VI. 결 론

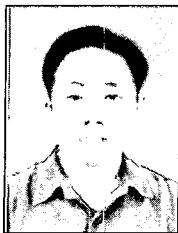
본 논문에서는 USN 라우팅 기법 중의 하나로서의 MANET 라우팅 기법인 DYMO 라우팅 프로토콜을 구현, 실험하였다. Linux 환경 위주의 다른 구현들과 달리, 본 논문에서는 Linux, Windows, 그리고 Windows CE 용으로 구현하였다. 커널 영역에서 netfilter 및 pcap 라이브러리를 이용하여 전송하고자 하는 패킷을 가로챈 후, 사용자 영역의 응용 프로그램에 보내어 DYMO 관련 작업을 수행하도록 하였다.

또한 DYMO 표준을 위한 적합성 검사 도구를 개발하여, 구현된 프로그램의 동작이 표준을 준수하는지 여부를 검사하였다. 적합성 검사는 본 논문에서 구현한 세 개의 프로그램에 대해서 수행하였다.

참 고 문 헌

- [1] A. Iwata et al., "Scalable Routing Strategies for Ad-hoc Wireless Networks," IEEE JSAC, pp. 1369-79, Aug. 1999
- [2] Charles E. Perkins, Elizabeth M. Belding-Royer, and Samir Das. "Ad Hoc On Demand Distance Vector (AODV) Routing," IETF RFC 3561
- [3] Ian D. Chakeres, Elizabeth M. Royer, and Charles E. Perkins, "Dynamic MANET On-demand Routing Protocol," IETF Internet draft, draft-ietf-manet-dymo-02.txt, June 2005
- [4] Luke Klien-Bernt, <http://sourceforge.net/projects/nist-dymo/nist-dymo> implementation homepage, NIST
- [5] Ian D. Chakeres, and E. Belding-Royer, "AODV Routing Protocol Implementation Design," Proceedings of the International Workshop on Wireless Ad Hoc Networking (WWAN), Mar. 2004
- [6] Pedro M. Ruiz and Francisco Ros, <http://masimum.dif.um.es/?Software:DYMOUM> dymoum implementation homepage, University of Murcia

저자소개

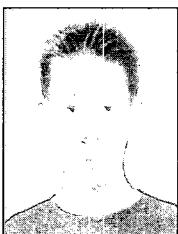


박 일 균(학생회원)
 1997년 송실대학교 정보통신공학
 과 학사
 1999년 송실대학교 대학원 정보통
 신공학과 석사
 1999년 3월~현재 송실대학교
 대학원 정보통신공학과
 박사 과정

<주관심분야 : 컴퓨터네트워크, 인터넷 네트워킹,
 인터넷 QoS, 무선 네트워크, MPLS, 차세대 인
 터넷 프로토콜>



김 영 한(평생회원)
 1984년 서울대학교 전자공학과
 공학사
 1986년 한국과학기술원 전기 및
 전자공학과 공학석사
 1990년 8월 한국과학기술원 전기
 및 전자공학과 공학박사
 1987년 1월~1994년 8월 디지콤정보통신연구소
 데이터통신연구부장
 1994년 9월~현재 송실대학교 정보통신전자공학부
 부교수, VoIP포럼 차세대기술분과위원장
<주관심분야 : 컴퓨터네트워크, 인터넷 네트워킹,
 이동 데이터 통신망>



곽 정 남(학생회원)
 2004년 송실대학교 정보통신전자
 공학부 학사
 2004년 3월~현재 송실대학교
 대학원 정보통신공학과
 석사 과정

<주관심분야 : 컴퓨터네트워크, 인터넷 네트워킹,
 인터넷 QoS, 무선 네트워크, MPLS>