

# 무선 센서 네트워크에서 NanoQplus를 이용한 DYMO 프로토콜 설계와 구현

준희원 오 수 택\*, 정희원 배 장 식\*\*, 정 흥 종\*, 김 동 균\*\*\*°, 박 정 수\*\*

## Design and Implementation of DYMO Protocol Using NanoQplus in Wireless Sensor Networks

Sutaek Oh\* *Associate Member,*

Jang Sik Bae\*\*, Hong-Jong Jeong\*, Dongkyun Kim\*\*\*°, Jungsoo Park\*\* *Regular Members*

### 요 약

한국전자통신연구원에서 개발한 NanoQplus는 센서 네트워크 운영을 위한 임베디드 운영체제로, 프로그래머에게 친숙한 다중 쓰레드 프로그래밍 환경을 제공하지만 잘 알려진 TinyOS와 같은 운영체제와 비교해볼 때 네트워크 프로토콜 스택이 빈약하다는 단점이 있다. 본 논문에서는 Internet Engineering Task Force (IETF)에서 표준화 중인 Dynamic MANET On-demand (DYMO) 라우팅 프로토콜을 NanoQplus에 적용해본다. DYMO 프로토콜은 본래 이동 애드혹 네트워크에 적합하도록 제안된 라우팅 프로토콜이며, 이동 애드혹 네트워크는 무선 센서 네트워크에 비해 자원의 제약에서 자유로우므로, 기본 DYMO 프로토콜을 아무런 수정 없이 무선 센서 네트워크에 적용하기는 힘들다. 게다가 매체 접근 제어 계층과 네트워크 계층 사이에 존재하는 중복되는 데이터를 줄이기 위해 두 계층 프로토콜간의 결합도도 고려할 필요가 있다. 따라서 우리는 DYMO 기본 명세를 수정하여 NanoQplus에 적합토록 수정하였다. 실제 센서 노드에서의 실험 결과를 통해 NanoQplus에 기반을 둔 DYMO 프로토콜이 무선 센서 네트워크 시나리오에서 효율적으로 동작함을 알 수 있었다.

**Key Words** : MANET, DYMO, NanoQplus, Sensor Networks

### ABSTRACT

NanoQplus, which is an embedded operating system for wireless sensor networks (WSNs) and developed by Electronics and Telecommunications Research Institute (ETRI), provides programmer-friendly preemptive multi-threading programming technique, but it has poor network protocol stack, as compared to TinyOS, one of the famous operating systems for WSNs. In this paper, we apply dynamic MANET on-demand routing (DYMO) protocol, which is being standardized in Internet Engineering Task Force (IETF), to NanoQplus. Since DYMO has been proposed for mobile ad-hoc networks (MANETs) and MANETs have less resource restrictions than WSNs, the basic DYMO protocol cannot be applied to WSNs without modifications. Moreover, coherence with MAC protocol should be considered in order to eliminate redundant data between MAC and network layers. Thus, we propose a modified version of the basic DYMO protocol for NanoQplus. The experimental results from a real sensor network test-bed show that the DYMO implementation using NanoQplus works efficiently in WSNs.

\* 이 논문은 2007년도 BK21 사업의 지원을 받아 수행된 연구입니다.

\* 경북대학교 컴퓨터공학과 이동인터넷망연구실(stoh@monet.knu.ac.kr), (hijeong@monet.knu.ac.kr)

\*\* 한국전자통신연구원(overset@etri.re.kr), (pjs@etri.re.kr), \*\*\* 경북대학교 컴퓨터공학과 (dongkyun@knu.ac.kr)(° : 교신저자)

논문번호 : KICS2007-12-590, 접수일자 : 2007년 12월 27일, 최종논문접수일자 : 2008년 3월 26일

## I. 서 론

최근 무선 멀티 홉 네트워크의 응용 중 하나인 센서 네트워크가 주요 연구 이슈로 관심을 모으고 있다. 센서 네트워크는 수집에서 수백 개의 센서 장치들을 넓은 지역에 배치하여 관심 있는 데이터를 수집하는 형태이므로, 각 센서 장치들은 저렴한 가격으로 생산할 수 있어야 한다. 또한 상시적인 전원 공급이 불가능하며 배터리를 사용하게 되므로 저 전력 설계도 필수적이다. 이에 따라 낮은 성능을 지닌 MCU와, 저렴하고 적은 전력을 사용하는 무선 통신을 구현하기 위한 표준인 IEEE802.15.4를 기반으로 한 RF 모듈을 사용한 센서 모트들이 많이 개발되고 있다. 따라서 제한된 자원만을 사용할 수 있는 센서 모트를 효율적으로 운용하기 위한 초소형 운영체제 기술에 대한 연구도 많이 이루어지고 있다<sup>[1][2]</sup>.

TinyOS는 UC Berkeley에서 INTEL 등의 기업과 함께 개발하고 있는 초소형 운영체제로, 이벤트 기반으로 개발되었으며 높은 수준의 모듈화 구조를 갖고 있어 응용에서 사용하는 모듈만 컴파일되도록 작성되어 있다. 또한 새로운 환경에서의 프로그램 작성을 위해 컴포넌트 기반의 nesC라는 새로운 언어를 정의하고 있다. 그러나 다중 쓰레드를 지원하지 않는 점, 쉽다고는 하지만 새로운 언어를 익혀야 하는 것과 같은 단점이 존재하였다. 한국전자통신연구원에서 순수 국내 기술로 개발된 NanoQplus는 위와 같은 TinyOS의 단점을 해결하기 위해 선점형 스케줄러를 가진 다중 쓰레드 기반의 응용 작성을 지원하며, C 언어를 그대로 사용한다. 따라서 기존의 프로그래머들이 접근하기 쉬운 것이 장점이다. 그러나 공개적으로 배포되기 시작한 기간이 짧아 많은 테스트를 거치지 못했으며, 다양한 프로토콜이 개발되어 있는 TinyOS에 비해 무선 통신을 위한 매체 접근 제어 계층과 네트워크 계층 프로토콜 지원이 빈약하다는 단점이 있다. 또 무선 통신 모듈로 2.4GHz 주파수 대역을 사용하는 CC2420 칩셋만 지원하고 있다.

본 논문에서는 Internet Engineering Task Force (IETF)<sup>[3]</sup>의 Mobile Ad hoc Network (MANET) working group에서 차세대 요구형 라우팅 프로토콜로 표준화 중인 Dynamic MANET On-demand 라우팅 프로토콜을 NanoQplus 2.3 버전에서 동작하기 적합하도록 간단하게 재설계한다. 또한 새로이 설계된 DYMO를 NanoQplus 상에서 구현하여 센서 모트에서 동작시켜 보고, NanoQplus의 네트워크 프로토콜 스택과 재설계된 DYMO의 성능 특성을 분석한다.

본 논문의 나머지는 다음과 같은 순서로 구성된다. II장에서는 NanoQplus 2.3에서 기본으로 제공하고 있는 매체 접근 제어 계층 프로토콜과 경로 설정 프로토콜인 Nano MAC과 Reactive Routing for ETRI Nano OS (RENO)에 대해서 살펴보고, 우리가 구현하려고 하는 DYMO에 대해서 간단히 살펴볼 것이다. III장에서는 DYMO를 NanoQplus에 적용하기 위해 축소된 새로운 DYMO 프로토콜을 제안하며, IV장에서는 III장에서 설계된 DYMO 프로토콜을 센서 네트워크에 적용시킨 실험 결과를 논의한다. 마지막으로 V장에서 결론을 맺는다.

## II. 관련 연구

NanoQplus는 기본적인 매체 접근 제어 계층 프로토콜로 CSMA/CA 기반의 Nano MAC을 제공하며, 경로 설정 프로토콜로는 요구형 라우팅 프로토콜의 일종인 RENO를 제공하고 있다. 이 장에서는 Nano MAC과 RENO의 기본 동작에 대해 알아보고, 우리가 구현하려 하는 DYMO에 대해 살펴본다.

### 2.1 Nano MAC / RENO

Nano MAC은 CSMA/CA를 기반으로 한 간단한 매체 접근 제어 계층 프로토콜로, 송신측에서는 유니캐스트 패킷을 전송하기 전에 clear channel assessment (CCA)를 수행하고, 채널이 유힤하다면 패킷을 전송한다. 처음 전송할 때에는 back-off를 하지 않으며, ACK 프레임을 받지 못하여 재전송을 시도할 경우 2 ~ 11 슬롯만큼 랜덤 back-off를 수행하는데, 각 슬롯의 길이는 CC2420 칩셋이 20 심볼을 전송하는데 필요한 시간으로 정의되어 있다. 재전송은 최대 3회까지 시도한다. 수신 측에서는 ACK 프레임을 명시적으로 전송하지 않는데, CC2420 칩셋의 automatic packet acknowledgement 기능에 의해 자동으로 ACK 프레임이 전송되기 때문이다. 브로드캐스트 패킷은 CCA를 수행 후 바로 전송하며, ACK 프레임이 오지 않을 것이므로 곧바로 종료한다.

RENO는 AODV<sup>[4]</sup>나 DYMO<sup>[5]</sup>와 같은 요구형 라우팅 프로토콜들과 유사하게 동작한다. 우선 경로를 알지 못하는 목적지 노드로 전송해야 하는 패킷을 받았다면, 경로 요청 메시지 (route request, RREQ)를 네트워크로 플러딩한다. RREQ 메시지를 들은 중간 노드들은 RREQ를 전송한 노드로 도달하기 위한 다음 홉 (next hop) 노드의 주소를 알 수 있

다. 마침내 RREQ 메시지가 목적지 노드에 도착하면 목적지 노드는 경로 응답 (route reply, RREP) 메시지를 유니캐스트로 전송하여 응답하며 이 메시지는 앞선 RREQ 플러딩 과정에서 설정된 경로를 통해 RREQ를 전송한 노드로 전달된다. 이 과정에서, RREP를 중계하는 노드들은 RREP를 전송한 노드, 즉 목적지 노드로의 경로 또한 설정할 수 있다. RREP가 RREQ를 전송한 노드에 도착하면 두 노드 사이에 양방향 경로가 설정된다.

비록 RENO가 간단한 시나리오에서의 라우팅을 지원하기 위해 설계되었지만, 여전히 많은 문제점들이 존재한다. RENO는 라우팅 루프를 방지하고 최신의 정보를 유지하기 위해 DYMO에서처럼 16비트의 순서 번호 (sequence number)를 사용하고 있다. 그러나 순서 번호의 넘쳐흐름 (overflow, roll over)의 처리를 정의하고 있지 않으며, 같은 순서 번호로 된 RREQ나 RREP가 오면 더 좋은 정보를 담고 있을 가능성이 있음에도 불구하고 무조건 폐기한다. 또한 노드가 재시작 되었을 때 다른 노드들이 알고 있는 순서 번호와 실제 순서 번호 간의 차이에 의해 정상적으로 라우팅에 참여할 수 없는 문제가 존재한다. 그리고 노드의 라우팅 주소를 8비트로 제한하고 있으며, 응용 계층에서의 브로드캐스팅을 위한 주소를 지원하지 않는다. 물론, 패킷을 네트워크 전체로 플러딩 하기 위한 방법 또한 지원하지 않고 있다. 만약 어떤 쓰레드가 RREQ 전송을 일으킨다면, 그 쓰레드는 RREP의 도착 여부와 무관하게 정해진 상수 시간만큼 블리킹 상태로 대기하게 되므로 아무런 일도 할 수 없다는 단점도 존재한다.

## 2.2 DYMO

DYMO는 AODV, DSR<sup>[6]</sup>과 같은 요구형 라우팅 프로토콜들의 표준화 및 구현에서 얻은 경험과 최신의 연구 결과를 바탕으로 새로이 표준화 중인 라우팅 프로토콜이다. 기존의 라우팅 프로토콜에서 핵심적인 기능만을 필수 기능으로 정의하고, 다른 복잡하거나 혹은 성능 항상 폭이 적은 기능은 옵션으로 정의하거나 설계에서 배제하여 프로토콜을 최대한 단순하게 하였다.

DYMO의 동작은 크게 경로 탐색 과정(route discovery)과 경로 유지 보수(route maintenance) 과정으로 나눌 수 있다. 경로 탐색 과정은 기본적으로 RENO와 유사하지만, 라우팅 메시지에 담긴 16비트 순서 번호와 홉 수 같은 부가적인 정보를 통해 라

우팅 정보의 유효성을 더 자세히 판단하고 4개의 부류로 분류하여 받아들일 것인지, 폐기할 것인지를 결정한다는 차이점이 있다. 경로 유지 보수 과정에서는 활성화 중인 경로들을 감시하고, 만약 경로 단절이 발생하였다면 경로 오류 (route error, RERR) 메시지를 전송하여 다른 노드들에게 알리게 된다.

DYMO가 기존의 요구형 라우팅 프로토콜과 구분되는 가장 큰 특징은 바로 경로 누적 (path accumulation) 기능이다. 경로 누적은 RREQ나 RREP를 중계하는 중간 노드가 자신의 주소와 순서 번호 등을 라우팅 메시지에 삽입하여 다른 노드에게 알리는 기능으로, 이를 통해 미래의 RREQ 시도 횟수를 줄일 수 있다.

최근 DYMO에 가해진 가장 주된 변화는 제어 메시지를 생성하기 위해 PacketBB<sup>[7]</sup> 표준을 따르게 되었다는 점이다. PacketBB는 MANET 라우팅 프로토콜들이 유사한 정보로 구성된 제어 메시지를 사용한다는 사실에 근거하여 범용적인 패킷 구조를 정의하고 있다. 그러나 PacketBB를 사용할 경우 필요 이상으로 패킷 크기가 커지고 프로세싱 오버헤드가 크다는 단점이 있어 한 번에 전송할 수 있는 패킷 크기가 매우 작은 IEEE802.15.4와 낮은 처리 능력을 가진 MCU를 사용하는 센서 모드에 적용하기 힘들다.

## III. NanoQplus를 위한 DYMO 수정

이 절에서는 NanoQplus에 적합하도록 DYMO를 수정하는 방법을 제안한다. 주소 체계는 NanoQplus에서 사용하는 IEEE802.15.4 16비트 짧은 주소를 사용한다.

### 3.1 라우팅 테이블 엔트리

현재의 DYMO 표준에서는 라우팅 테이블 엔트리를 구성하는 항목 중 NanoQplus에서는 필요하지 않은 것들이 필수 항목으로 포함되어 있다. 패킷을 다음 홉으로 전달할 때 사용할 인터페이스를 나타내는 NextHopInterface가 그 예이다. NanoQplus를 위한 DYMO는 메모리 사용량을 줄이기 위해 불필요한 항목을 모두 제거하고 다음과 같은 필드만으로 라우팅 테이블 엔트리를 구성하였다.

- 1) 목적지 주소 (Route.Address)
- 2) 목적지 순서 번호 (Route.SeqNum)
- 3) 다음 홉 주소 (Route.NextHopAddress)

4) 경로 소모비용 (Route.Dist): DYMO 표준은 경로 소모비용으로 홉 수 값(Route.Dist)을 사용하도록 권장하고 있다. NanoQplus를 위해 수정된 DYMO에서도 기본적으로 홉 수 값을 사용하되 간단한 수정을 통해 link quality indicator (LQI) 등 다른 메트릭도 사용할 수 있도록 허용하고 있다. 이를 Route.Cost라 칭한다.

라우팅 테이블 엔트리 하나의 크기는 7바이트이며, Route.Cost로 16비트 메트릭을 사용할 경우 8바이트가 된다. 본래 DYMO 표준에서는 라우팅 테이블 엔트리를 관리하기 위해 다양한 종류의 시간 경과(timeout) 값을 사용하고 있지만 NanoQplus를 위해 수정된 DYMO에서는 시간 경과 값을 유지하는데 필요한 프로세서와 메모리 사용량을 줄이기 위해 유효 시간 경과 하나만을 사용한다. 각 라우팅 테이블 엔트리는 생성된 이후 해당 엔트리가 마지막으로 사용된 이후로 유효 시간 경과 값만큼의 시간이 지나면 라우팅 테이블 엔트리에서 삭제되도록 하였다. 유효 시간 경과 값 유지에 필요한 메모리를 줄이기 위하여 이동성이 적은 센서 네트워크의 특징을 이용하여 초 단위가 아닌 5초 단위로 계산하도록 하였으며, 이를 통해 엔트리 당 1바이트의 메모리만을 사용하도록 하였다.

### 3.2 패킷 구조

DYMO 표준은 PacketBB에 따라 제어 메시지를 생성하지만, NanoQplus 상에서 PacketBB 표준을 준수하는 것은 비효율적이다. PacketBB에 의해 제어 메시지가 불필요하게 커지며, PacketBB로 작성된 패킷을 해석하기 위해 별도의 프로세싱 파워가 필요하기 때문이다. 게다가, NanoQplus는 효율성을 위하여 별도의 네트워크 계층을 두는 대신 라우팅 프로토콜이 독자적인 네트워크 계층 헤더를 사용하도록 되어 있지만, DYMO 표준은 네트워크 계층으로 IP를 사용할 것을 가정하고 있기 때문에 수정이 불가피하다.

본 논문에서 제안하는 NanoQplus를 위해 수정된 DYMO의 네트워크 계층 헤더 구조는 그림 1과 같다. Type 필드는 네트워크 계층에서 처리할 라우팅 관련 제어 메시지인지, 혹은 상위 계층의 패킷인지를 구분하기 위한 필드이다. DYMO\_DATA는 상위 계층으로 전달될 패킷이며, 나머지는 네트워크 계층에서 처리해야 하는 메시지인데, DYMO\_RREQ는 경로 요청 메시지, DYMO\_RREP는 경로 응답 메

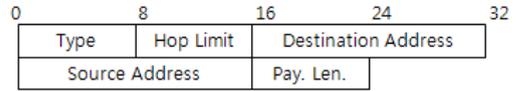


그림 1. NanoQplus를 위한 DYMO 네트워크 계층 헤더 형식.  
Fig. 1. Network header format of DYMO for NanoQplus.

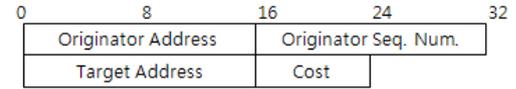


그림 2. NanoQplus를 위한 DYMO RM 형식.  
Fig. 2. An RM format of DYMO for NanoQplus.

시지, DYMO\_RERR는 경로 단절 알림 메시지를 뜻한다. Hop Limit 필드는 패킷이 전달될 수 있는 최대 홉 수를 의미하며, Destination Address 필드는 패킷의 최종 목적지 주소, Source Address 필드는 패킷을 전송한 노드의 주소이다. Pay. Len. (Payload Length) 필드는 네트워크 계층 헤더 크기를 제외한 네트워크 계층 페이로드의 크기를 뜻하며, Nano MAC에서 제공하는 매체 접근 제어 계층 헤더에서 계산이 가능하지만 패킷의 무결성을 검사하기 위한 보조적인 수단으로 포함하였다. 네트워크 계층 헤더는 패킷을 받았을 때 먼저 읽어 들이는 순서대로 정렬하였으며, 7바이트의 크기를 갖는다. CC2420 칩셋은 127바이트의 최대 패킷 크기를 제공하며, NanoQplus는 매체 접근 제어 계층 헤더로 11바이트를 사용하고 1바이트의 여분을 두고 있으므로 네트워크 계층 페이로드의 최대 크기는 108바이트가 된다.

RREQ 메시지와 RREP 메시지는 전달되는 방식이 각각 브로드캐스트, 유니캐스트로 다를 뿐 서로 같은 필드로 구성되어 있다. DYMO 표준에서는 이 두 메시지를 RM (routing message) 부르고 있다. NanoQplus를 위해 수정된 DYMO에서도 이 형식을 수용하였고, 그림 2와 같은 구조를 지니고 있다.

Originator Address 필드는 RM을 생성한 노드의 주소이며, Originator Seq. Num. 필드는 RM을 생성한 노드의 순서 번호를 뜻한다. Target Address는 RM이 도착해야 할 목적지의 주소이며, Cost는 RM이 거쳐 온 경로의 소모비용으로 현재 구현에서는 홉 카운트를 사용하고 있다. Target의 순서 번호는 포함하지 않는데, 이는 중간 노드가 RREP를 전송하는 기능을 제거하였기 때문이다. 또한 패킷 크기를 한 번에 전송할 수 있는 크기 이상으로 커지게 할 가능성이 있는 경로 누적 기능은 사용하지 않는

0	8	16	24	32
# of node				
Unreachable Address 1	Sequence Number 1			
Unreachable Address ...	Sequence Number ...			
Unreachable Address n	Sequence Number n			

그림 3. NanoQplus를 위한 DYMO RERR 형식.  
Fig. 3. An RERR format of DYMO for NanoQplus.

데, 네트워크 계층에서 메시지 분할을 지원하지 않는 데다가 통신이 주로 센서 노드와 싱크 노드 사이에서 이루어지므로, 경로 누적 기능이 오버헤드에 비해 큰 성능 향상을 가져오기 힘들기 때문이다. DYMO 표준과는 달리 NanoQplus를 위해 수정된 DYMO는 네트워크 계층 헤더를 직접 관리하므로 중복을 피하기 위해 RM에 hop limit 필드를 별도로 두지 않는다. 이는 다음에 설명할 RERR에도 공통적으로 적용된다.

경로 단절 알림 메시지(RERR)는 그림 3과 같은 구조로 되어 있다. RERR은 링크가 단절되어 현재 전송하려는 패킷의 목적지로 도달할 수 없음을 나타내는 메시지인데, DYMO 표준에서는 패킷의 목적지 외에도 해당 링크의 단절로 인해 도달 불가능해진 다른 목적지들도 포함하도록 권장하고 있다. NanoQplus를 위해 수정된 DYMO에서도 메시지의 크기가 단편화가 필요할 정도로 커지지 않는 한도 내에서 추가적인 도달 불가능 목적지들도 포함하도록 하고 있다. 이는 실제로 단절되었으나 RERR에 포함되지 않아 단절 사실을 알지 못한 링크를 사용하여 패킷을 전송하는 일을 막기 위한 것이다. 이 기능은 미래의 RERR 메시지 생성도 막을 수 있고, 단절된 링크로 패킷을 보냄으로 인해 발생하는 패킷 손실도 막을 수 있으므로 매우 효율적이다. RERR에서 순서 번호는 DYMO 표준에 따라 늦게 전송된 RERR이 최근에 생성된 경로를 무효화하는 것을 막기 위해 사용한다.

### 3.3 사용 중인 경로 감시 방법

DYMO 표준은 현재 사용 중인 경로를 구성하는 링크의 상태를 다음과 같은 방법으로 감시하고, 단절이 발생하였을 경우 이를 알아챌 수 있어야 한다고 정하고 있다.

- 1) 링크 계층으로 부터의 알림
- 2) Neighborhood discovery 프로토콜<sup>18)</sup>
- 3) 경로 시간 초과 (route timeout)
- 4) 그 외 다른 경험적 방법

NanoQplus를 위한 DYMO는 경로 시간 초과 기능과 함께, 정적인 센서 네트워크를 고려하여 링크 계층으로 부터의 통지에 의해 경로 단절을 파악한다. Nano MAC은 이웃 노드로 패킷을 전송할 때 3회 ACK 프레임을 받지 못하면 이를 DYMO에 알리고, DYMO는 해당 링크를 사용할 수 없도록 만든다. 그러나 AODV와는 달리 단절된 링크가 발견된 즉시 RERR을 전송하지 않고, 단절된 링크를 통해 전송되어야 하는 패킷을 받았을 때 RERR을 전송한다. 이는 DYMO 표준의 권장 사항이다.

## IV. 실험결과

우리는 NanoQplus를 위한 DYMO 프로토콜이 실제로 센서 네트워크에서 어떤 성능 특성을 가지고 있는지 분석하기 위하여 그림 4와 같이 36개의 노드로 구성된 6 x 6 격자형 토폴로지로 구성된 센서 네트워크를 구축하였다. 25개의 노드 중 좌측 최상단에 위치한 1번 노드는 싱크로 동작하며, 다른 24개의 노드는 센서 노드로 동작하도록 하였다. 센서 노드는 10초에 하나씩 100바이트 크기의 보고(REPORT) 패킷을 싱크 노드로 전달하며, 싱크 노드는 패킷을 성공적으로 받았다는 의미로 ACK 패킷을 전송한다. 센서 네트워크에는 이러한 통신 모델 이외에도 여러 가지 통신 모델이 있지만, 대부분 우리가 가정한 통신 모델에 비해 교환하는 데이터의 양이 적기 때문에 특징적인 성능 요소를 발견하기 어려우므로 생략하였다.

NanoQplus는 공개 릴리즈인 2.3.0 버전을 사용하였으며 센서 모드는 Cratto사의 CRM-ZRM110을 사용하였다. CRM-ZRM110은 ATmega128L MCU를 사용하며, 32KB의 외부 메모리를 제공하고, RF 모듈로 CC2420 칩셋을 사용하고 있다.

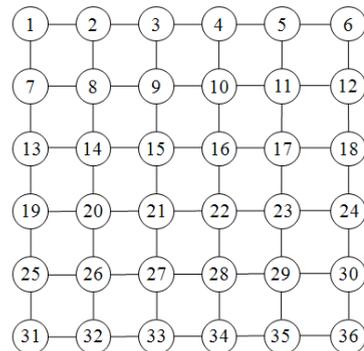


그림 4. 6 x 6 격자형 토폴로지.  
Fig. 4. 6 x 6 grid topology.

### 4.1 경로 설정 시간

먼저 홉 수에 따른 경로 설정에 걸리는 시간을 측정하였다. 결과는 그림 5와 같다. 본 논문에서 구현한 DYMO는 하드웨어 타이머 부족을 해결하기 위해 자체적으로 개발한 시분할 타이머를 사용하여, 모든 이벤트를 단 하나의 하드웨어 타이머만을 사용하여 처리하는 효율적인 구조로 되어 있으므로 RF 모듈과 낮은 MCU의 제약에도 불구하고 경로 설정이 효율적으로 이루어지고 있음을 알 수 있다. 이에 비해 NanoQplus에 기본적으로 포함되어 있는 RENO는, 관련 연구에서 살펴보았듯이 RREQ를 보낸 후 홉 수에 무관하게 500ms를 블러킹 상태로 대기하므로 불필요한 시간의 낭비가 발생한다. 즉, 본 논문에서 구현한 DYMO가 NanoQplus의 기본 프로토콜 스택인 RENO보다 더 짧은 시간에 경로를 설정함을 알 수 있었다.

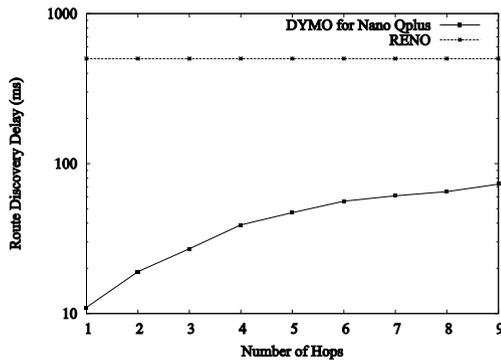


그림 5. 경로 설정에 걸린 시간.  
Fig. 5. Average time taken to perform a route discovery.

### 4.2 패킷 전달률

두 번째로는 격자형 토폴로지의 크기를 변화시키며 패킷의 단대단 (end-to-end) 전달률을 측정하였다. 싱크와 센서는 각자에게로 10개씩의 패킷을 전송하는데, MAC 계층의 충돌이나 큐 오버플로우 등의 원인으로 제대로 전송되지 못한 패킷들의 비율을 알아보기 위한 실험이다. 결과는 그림 6과 같다. 3 x 3의 격자 크기에서는 RENO와 DYMO 모두 모든 패킷이 성공적으로 전송되었으나, 격자의 크기가 더 커지면 패킷 손실이 발생하기 시작한다. 그러나 DYMO의 경우 6 x 6 크기의 토폴로지에서도 패킷 전달률은 94%에 가까웠으며, 내고장성을 갖고 구축되는 센서 네트워크의 특성 상 무시할만한 패킷 손실률이라고 할 수 있다. 그러나 네트워크의 크기가 증가할수록 패킷 손실률도 크게 증가하

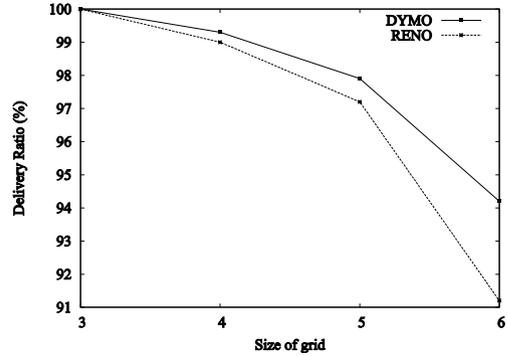


그림 6. 격자 크기에 따른 패킷 전달률.  
Fig. 6. Packet delivery ratio w. r. t. grid size.

고 있으므로, 적절한 수의 노드 배치와 트래픽 제어도 필요함을 알 수 있었다. 또한 우리가 구현한 DYMO가 제어 메시지를 효율적으로 처리하여 지연을 최소화 시켰기 때문에 함수 호출 구조가 복잡하고 블러킹 구조로 되어 있는 RENO에 비해 패킷 손실을 줄이고 있음을 알 수 있었다.

### 4.3 패킷 충돌 횟수

다음으로 Nano MAC의 충돌 방지 방법의 우수성을 살펴보기 위해 패킷의 충돌 횟수를 측정하였다. 브로드캐스트 패킷은 충돌을 감지하기 힘들기 때문에 유니캐스트 패킷을 대상으로 하였다. 각 노드에서 평균 0.55회의 MAC 계층 충돌이 발생하였는데, 우리가 가정한 트래픽 시나리오가 적은 양의 패킷을 주고받기 때문에 비교적 간단한 Nano MAC의 충돌 회피 기능으로도 충돌이 최소화 되었다고 결론내릴 수 있다.

다음으로는 브로드캐스트 패킷의 손실을 가늠해보기 위해 비효율적인 경로 설정 비율을 조사해보았다. 센서 노드가 싱크 노드에게 처음으로 REPORT 패킷을 전송할 때 RREQ 패킷을 보냄으로써 경로 탐색을 시도할 것이며 따라서 짧은 시간에 RREQ가 집중될 것임을 쉽게 예상할 수 있다. 즉, 비효율적인 경로 설정을 살펴봄으로써 대략의 RREQ 손실률을 예측해볼 수 있다. 각 센서 노드별로 설정된 경로의 길이는 그림 7과 같다.

5 x 5 토폴로지에서 실험한 결과인 그림 7에서 볼 수 있듯이, 약 33%의 노드가 비효율적인 경로를 설정하였음을 알 수 있다. 이는 RREQ 손실 비율이 높다는 것을 뜻하며, 따라서 Nano MAC을 이용하여 플러딩을 필요로 하는 프로토콜을 작성할 때에는 적절한 지연 값 사용 등을 통해 브로드캐스트 패킷의 손실을 최소화해야 한다.

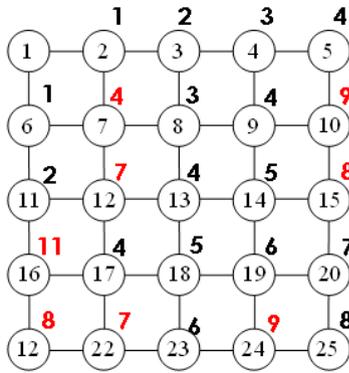


그림 7. 각 센서 노드에서 싱크로의 홉수.  
Fig. 7. Hop count from each sensor to sink.

### 4.3 제어 메시지 생성 횟수

다음으로 우리는 라우팅을 위한 제어 메시지가 어떤 노드에서 몇 회나 생성되었는지를 측정해보았다. 5 x 5 토폴로지에서 싱크 노드는 RREP를 단 2회 전송한 반면, RREQ는 22회 전송하였다. 이는 최초의 센서 노드의 경로 탐색 과정에 의해 일부 중간 센서 노드들이 싱크로의 경로를 알게 되었기 때문이다. 즉, 일부 중간 센서 노드들은 경로 탐색 없이 싱크에게로 패킷을 전송하였으며, 싱크는 그 센서 노드들로의 경로를 찾기 위해 RREQ를 플러딩하고, 이 과정에서 모든 센서 노드들이 싱크로의 경로를 설정할 수 있게 된다. 따라서 각 센서 노드들이 먼저 전송을 시작함에도 불구하고 실제로 RREQ로 경로 탐색을 시작한 노드는 2개 노드에 불과하였다. 그리고 RREQ와 RREP 생성 횟수를 합하면 24회인데, 이는 비록 RREQ 손실에 의해 비효율적인 경로가 설정되기는 하였지만, 한 번의 경로 탐색 시도로 성공적으로 경로 설정을 하였음을 뜻한다. 마찬가지로 6 x 6 토폴로지에서 비효율적인 경로 설정이 발생하기는 하였으나 RREQ와 RREP 생성 횟수가 노드 수만큼이 되어 같은 결론을 얻을 수 있었다.

## V. 결 론

본 논문에서는 NanoQplus에 적합하도록 DYMO 프로토콜을 수정한 DYMO를 구현하여 실제 테스트 베드에서 NanoQplus와 구현한 DYMO의 성능을 평가하였다. NanoQplus는 비교적 효율적인 구조를 가진 네트워크 프로토콜 스택을 갖추고 있으며, 특히 Nano MAC은 가벼운 트래픽을 사용하는 센서 네

트워크에서 충분히 사용 가능한 성능을 보이고 있다. 또한 프로그래머에게 친숙한 C언어와 다중 쓰레드 구조를 사용하고 있어 프로그래밍이 간편하고 유지보수가 용이하다. NanoQplus를 위한 DYMO 또한 부족한 자원을 효율적으로 사용하여 짧은 시간 내에 경로 설정을 마칠 수 있었다.

## 참 고 문 헌

- [1] TinyOS project web site, "http://www.tinyos.net".
- [2] Embedded S/W research division web site, Electronics & Telecommunications Research Institute (ETRI), "http://www.qplus.or.kr".
- [3] IETF Mobile Ad hoc Networks (MANET) Working Group, "http://www.ietf.org/html.charters/manet-charter.html".
- [4] C. Perkins, E. Belding-Royer and S. Das, "Ad hoc On-demand Distance Vector (AODV) routing," *IETF RFC*, RFC3561, Jul. 2003.
- [5] I. Chakeres and C. Perkins, "Dynamic MANET On-demand (DYMO) Routing," *IETF Internet draft*, draft-ietf-manet-dymo (work in progress), Nov. 2007.
- [6] D. Johnson, D. Maltz and Y. Hu, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4," *IETF RFC*, RFC4728, Feb. 2007.
- [7] T. Clausen, C. Dearlove, J. Dean and C. Adjih, "Generalized MANET Packet/Message Format (PacketBB)," *IETF Internet draft*, draft-ietf-manet-packetbb (work in progress), Nov. 2007.
- [8] T. Clausen, "MANET Neighborhood Discovery protocol (NHDP)," *IETF Internet draft*, draft-ietf-manet-nhdp (work in progress), Jul. 2007.

오 수 택 (Sutaek Oh)

준회원



2005년 2월 창원대학교 컴퓨터공학과(학사)

2008년 2월 경북대학교 컴퓨터공학과(공학석사)

2008년 2월~현재 경북대학교 이동인터넷망연구실

<관심분야> Mobile Ad Hoc Network, 무선 센서 네트워크, 라우팅 프로토콜 등

배 장 식 (Jang Sik Bae)

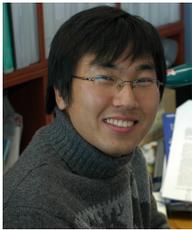
정회원



1994년 2월 충남대학교 컴퓨터공학과(학사)  
2000년 2월 충남대학교 컴퓨터공학과(공학석사)  
2006년 6월~현재 한국전자통신연구원  
<관심분야> RTOS, Embedded system, IP-Based network 등

정 흥 종 (Hong-Jong Jeong)

정회원



2004년 2월 경북대학교 컴퓨터공학과(학사)  
2006년 2월 경북대학교 컴퓨터공학과(공학석사)  
2006년 3월~현재 경북대학교 컴퓨터공학과(박사과정)  
<관심분야> Mobile Ad Hoc Network, 무선 센서 네트워크, 무선 LAN 등

김 동 균 (Dongkyun Kim)

정회원



1994년 2월 경북대학교 컴퓨터공학과(학사)  
1996년 2월 서울대학교 컴퓨터공학과(공학석사)  
2001년 2월 서울대학교 전기컴퓨터공학부(공학박사)  
1999년 미국 Georgia Institute of Technology, 방문 연구원  
2002년 미국 University of California at Santa Cruz, Post-Doc. 연구원  
2003년 3월~현재 경북대학교 컴퓨터공학과 조교수  
<관심분야> 이동인터넷, 초고속 인터넷, Mobile Ad Hoc Network, 무선 LAN 등

박 정 수 (Jungsoo Park)

정회원



1992년 2월 경북대학교 전자공학과(학사)  
1994년 2월 경북대학교 전자공학과(공학석사)  
1994년 2월~현재 한국전자통신연구원 선임연구원  
2005년 2월~현재 경북대학교 전자공학과(공학박사 수료)  
<관심분야> 차세대인터넷(IPv6), 멀티호밍, 이동인터넷, Ad Hoc 라우팅 및 보안, 무선 LAN 등