

논문 2007-02-27

무선 메쉬 라우터에서 이종 라우팅 프로토콜의 효율적인 탑재구조

(An Efficient Embedding Structure for Heterogeneous Routing Protocols in Wireless Mesh Routers)

이명석, 김영한

(Youngsuk Lee, Younghan Kim)

Abstract: In this paper, we propose more efficient implementation architecture for realizing combination of heterogeneous routing protocols in wireless mesh routes. For realizing heterogeneous routing protocol in wireless mesh router, the following should be considered; which position in OS platform protocols should be implemented, how to define the common API for multiple routing protocols, how to provide architecture for providing high data transfer throughput and for supporting multi platform, and finally how to verify the throughput of protocol by using simulator. For solving those consideration, we propose HRPC(Heterogeneous Routing Protocol Coordinator) and describe the method for testing the protocols.

Keywords : Wireless Mesh Router, Ad-hoc, DYMO, OLSR

1. 서론

무선 메쉬 네트워크는 점대점(point to point) 또는 점대다점(point to multi-point)의 스타형 토폴로지를 기반으로 하는 단일 홉 네트워크이다. 반면 기존의 유선 네트워크는 다점대다점(multi-point to multi-point)의 메쉬형 토폴로지를 기반으로 한다.

메쉬 구조는 다중 데이터 경로를 설정할 수 있어서 네트워크의 오류에 대응하기 용이하고 데이터 전달의 신뢰성을 향상 시킨다. 무선 메쉬 네트워크는

이러한 메쉬 구조를 무선 네트워크에 적용하여 안정적인 무선 경로를 확보하고, 필요 시 저비용으로 네트워크를 확장할 수 있다.[1]

* 교신저자(Corresponding Author)

논문접수 : 2007. 12. 3 채택확정 : 2008. 1. 25.

김영한 : 숭실대학교 교수

이명석 : 숭실대학교 박사과정

※ 본 논문은 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스컴퓨팅및네트워크인천기반기술개발사업의 지원에 의한 것이다.

무선 메쉬 네트워크는 무선 ad-hoc 네트워크의 한 응용으로 자기형성(self-organization) 및 자기구성(self-configuration)이 가능한 다중 홉 기반 무선네트워크이다. 즉, 네트워크를 구성하는 메쉬 클라이언트들 간의 협업과 메쉬 라우터들을 통해 ad-hoc 네트워크를 구성하게 되고 동적으로 토폴로지를 관리하고 유지하게 된다. 이때 사용되는 것이 ad-hoc 라우팅 프로토콜이다[2].

따라서 무선 메쉬 네트워크의 가장 중요 요소인 무선 메쉬 라우터는 어떤 ad-hoc 라우팅 프로토콜을 효율적으로 탑재하느냐에 따라 그 성능이 좌우된다. 또한 메쉬 라우터간에 ad-hoc 라우팅 프로토콜이 다르다면, 메쉬 네트워크를 자기형성 할 수 없기 때문에, 메쉬 라우터는 표준화된 여러 종류의 ad-hoc 라우팅 프로토콜이 반드시 탑재되어야만 한다.

하지만 ad-hoc 라우팅 프로토콜은 MANET (Mobile Ad-hoc Network)[3] WG(Working Group)에서 표준화 진행 중인 DYMO(Dynamic On-Demand Routing Protocol)[4], OLSR(Optimized Link State Routing Protocol)[5] 이외에도 reactive, proactive를 근간으로 하는 많은 종류의 라우팅 프로토콜이 있다.

그렇기 때문에 여러 종류의 ad-hoc 라우팅 프로

토콜을 하나의 메쉬 라우터에 탑재하기 위해서는 반드시 공통의 ad-hoc 라우팅 개발 구조가 필요하다.

본 논문은 무선 메쉬 라우터에 여러 종류의 ad-hoc 라우팅 프로토콜을 효율적으로 탑재할 수 있는 HRPC(Heterogeneous Routing Protocol Coordinator) 개발 구조를 제안한다[6].

본 논문의 구성은 다음과 같다. 2장에서 이종 라우팅 프로토콜을 탑재하기 위한 HRPC의 개발 고려사항에 대해 논의 하고, 3장에서 HRPC의 개발 구조를 제안한다. 4장에서는 HRPC의 실험 환경을 설명하고, 5장에서 결론을 맺는다.

II. HRPC 개발 고려사항

HRPC는 다양한 ad-hoc 라우팅 프로토콜 개발 구조이다. 이를 만족하기 위해서는 다음과 같은 고려사항이 만족되어야만 한다.

- HRPC를 사용자 공간(User Space) 만들 사용해 구현할지, 커널 공간(Kernel Space) 만들 통해 구현할지, 두 가지 모두를 조합하여 구현할지에 대해 고려해야한다.

- Reactive, Proactive, Hybrid의 모든 ad-hoc 라우팅 프로토콜의 일반화 및 동일한 API 정의

- Hybrid 라우팅 프로토콜을 위한 응동성있는 coordination 구조와 높은 패킷 전달 성능

- 멀티 플랫폼을 위한 구현 환경과의 독립성

- 개발된 라우팅 프로토콜의 성능 검증을 위한 적절한 시뮬레이터 연계

1. HRPC의 개발위치

- 사용자 공간만을 이용한 HRPC 개발은 개발자 입장에서 손쉬운 ad-hoc 라우팅 프로토콜 개발이 가능한 장점을 갖는다. 또한 멀티 플랫폼을 지원함에 있어 좀 더 응동성있는 개발 환경이 될 수 있다. 하지만 패킷 전달이나 프로세싱에 있어 커널 공간만을 이용한 HRPC 보다 성능이 떨어지고, ad-hoc 라우팅 프로토콜이 네트워크 계층에 요구하는 정보를 얻기 어려운 단점을 갖는다.

- 커널 공간 만들 이용한 HRPC 개발은 패킷 전달이나 프로세싱에 있어 다른 어떠한 구조보다 뛰어나지만, HRPC가 추구하는 공통의 ad-hoc 라우팅 프로토콜 개발 구조는 될 수 없다. 왜냐하면 커널 공간만을 이용할 경우 멀티 플랫폼을 지원할 수 없고, 새로운 ad-hoc 라우팅 프로토콜을 개발할 경우 소스 코드 전체를 재 컴파일 해야 하는 어려

움이 따른다. 또한 리눅스의 경우 커널 버전이 업데이트 될 때마다 그에 맞는 버전을 개발해야만 하는 어려움이 따른다.

- 사용자 공간, 커널 공간을 동시에 사용한 HRPC 개발은 위 두가지 장점을 모두 수용하여, 빠른 패킷 전달과 프로세싱, 멀티 플랫폼 지원 및 개발자에게 응동성있는 개발 환경이 될 수 있다. 하지만 구조 설계를 잘못할 경우 이것도 저것도 아닌 어중간한 개발 위치가 될 수 있는 단점을 갖는다.

본 논문에서는 HRPC 구현위치를 사용자 공간과 커널 공간 모두를 사용하는 구조를 택한다. 이는 HRPC가 추구하는 공통의 ad-hoc 라우팅 개발 디자인으로서 충족되는 구조를 갖으며, 네트워크 계층에 요구할 수 있는 정보를 보다 쉽게 얻을 수 있기 때문이다. 특히 커널 공간에는 HRPC에 필요한 최소한의 모듈만 탑재시킴으로써 ad-hoc 라우팅 프로토콜 개발자가 좀 더 안정적인 개발을 할 수 있다.

2. ad-hoc 라우팅 프로토콜의 공통 API

모든 ad-hoc 라우팅 프로토콜은 보낼 패킷의 목적지 주소를 찾는 방식이 다른 뿐 라우팅 테이블의 생성, 추가, 삭제, 갱신에 있어 동일한 API를 사용할 수 있다. 또한 패킷 전달의 경우, ad-hoc 라우팅 프로토콜은 라우팅 테이블에 저장된 다음 노드 주소를 참조하여 전달하는 동일한 방식을 사용한다. 따라서 HRPC가 이를 위한 두 가지 API를 지원할 수 있다면 모든 ad-hoc 라우팅 프로토콜을 탑재할 수 있는 구조가 된다. 하지만 ad-hoc상의 QoS 라우팅 프로토콜의 경우 현재 사용중인 리소스를 참고하거나, 전달한 패킷의 개수를 파악하는 등의 추가 동작이 있을 수 있기 때문에 위 두가지 이외에 노드 정보를 일반화 시켜 보고 할 수 있는 API 또한 필요하다.

3. 응동성있는 구조와 높은 전달 성능

HRPC는 커널 공간에 최소한의 모듈만 탑재한다. 탑재되는 모듈은 그림 1과 같은 구조를 갖는다.

단말로 수신 되는 모든 패킷은 HRPC 필터를 거쳐 원하는 패킷만 사용자 공간의 HRPC PAM 인터페이스로 전달된다. HRPC PAM 인터페이스는 커널과 사용자 공간에 있는 HRPC 간의 인터페이스로 커널에서 수신되는 모든 패킷을 받아 HRPC에게 전송하는 역할을 수행한다.

HRPC에서 전송되는 모든 패킷은 ip_output을

거쳐 전송 된다. 이때 발생된 모든 패킷은 HRPC의 라우팅 테이블 lookup을 통해 경로를 결정하고 ip_output을 통해 전송된다. 그림과 같은 기능만을 탑재하는 이유는 reactive 뿐만 아니라 proactive, hybrid 방식의 모든 라우팅 프로토콜을 지원하기 위해서이며, ioctl을 통한 라우팅 테이블 생성, 추가, 삭제를 사용자 공간만을 통해 제어함으로써 그 접근 및 동작을 쉽게 하기 위함이다. 또한 수신된 패킷이 전체 네트워크 스택을 타고 올라가지 않고, HRPC에 빠르게 공급함으로써 성능 향상을 꾀함이다.

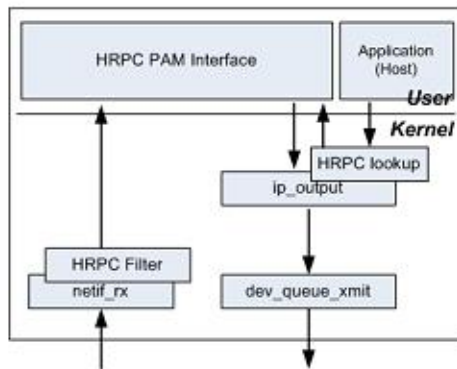


그림 1. HRPC의 커널 모듈
Fig1. Kernel module of HRPC

따라서 최소한의 모듈만 탑재한 HRPC는 패킷 전달을 사용자 공간에서 수행하게 된다. 그렇기 때문에 HRPC는 여러 ad-hoc 라우팅 프로토콜의 융통성있는 구조와 높은 패킷 전달 성능을 갖는다.

4. 멀티 플랫폼 지원

ad-hoc 라우팅 프로토콜이 멀티 플랫폼을 지원하기 위해서는 리눅스 뿐만 아니라 가장 대중적인 윈도우 기반 OS에도 같은 소스를 이용하여 개발할 수 있어야 한다. 따라서 모든 ad-hoc 라우팅 프로토콜의 개발 환경인 HRPC는 이를 위해 계층적 구조로 설계 되어야 한다.

III. HRPC 개발 구조

그림 2는 2장의 모든 고려사항을 충족하는 HRPC의 설계 구조이다. HRPC는 3계층으로 구성된다. 첫째 다양한 OS 플랫폼에 독립적인 PAL

(Platform Adaption Layer)과 둘째, 패킷 전달 및 조정 기능이 탑재된 HRPCL(HRPC Layer) 마지막으로 서로 다른 라우팅 프로토콜의 핵심 기능 소스가 되는 GRPML(Generalized Routing Protocol Module Layer)로 구성된다.

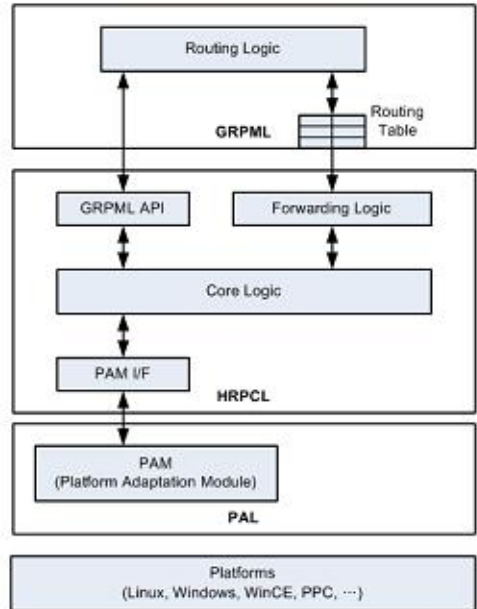


그림 2 HRPC 상세 설계
Fig2. Design of HRPC

이중 HRPC의 가장 핵심이 되는 계층은 HRPCL이다. HRPCL은 HRPC를 제어하는 핵심구조로, 모든 구성요소(PAL, GRPML)를 통합 제어한다. HRPCL의 세부 기능은 PAL을 통해 수신되는 패킷을 받아, 현재 활성화 상태(예, DYMO, OLSR, ZRP 등)의 GRPM에게 패킷을 전달하거나, 각 GRPM에서 생성한 라우팅 테이블을 참조하여 패킷을 전달하는 역할을 수행한다. 또한 Hybrid 형태의 ad-hoc 라우팅 프로토콜은 동작 우선순위를 설정하여 어떠한 프로토콜을 먼저 실행할지, 다음 프로토콜을 트리거 시키는 포인트는 어디일지를 Core Logic을 사용하여 결정, GRPM을 연결하는 동작을 수행한다. 따라서 여러개의 ad-hoc 라우팅 프로토콜이 탑재된 무선 메시 라우터는 무선 메시 네트워크의 구성 환경(라우팅 프로토콜 종류)에 따라 프로토콜 동작 순위를 결정할 수 있다.

HRPCL은 2장의 고려사항 중 ad-hoc 라우팅 프로토콜이 공통으로 요구하는 API가 탑재된다. 이

는 각 라우팅 프로토콜이 시작될 때 노드에서 얻어야 하는 기본적인 정보(무선 인터페이스 종류 및 개수, IP 주소, MAC 주소 등)를 공급해 줄 수 있으며, 패킷 전달시 공통의 API를 통해 라우팅 테이블을 참조하는 기능을 갖는다.

GRFML은 DYMO, OLSR 등의 ad-hoc 라우팅 프로토콜의 핵심기능이 탑재되는 계층이다. DYMO의 경우 RREQ를 생성하거나, RREQ 전달, RREP 생성 기능을 수행하고, RREQ 및 RREP를 통한 DYMO 라우팅 테이블을 생성하는 등 DYMO 고유의 기능이 포함된다.

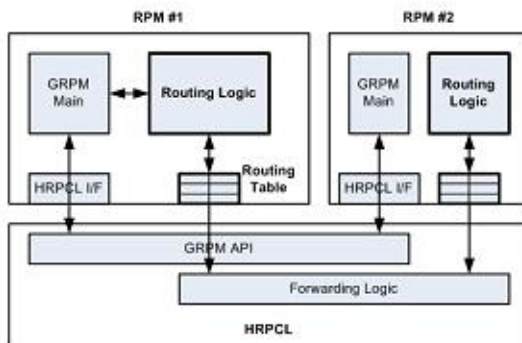


그림 3. GRFML 상세 설계
Fig.3 Design for GRFML

OLSR의 경우 HELLO 메시지를 주기적으로 생성하여 무선 메쉬 네트워크를 구성하는 기능을 수행하고, 그 결과로 OLSR 라우팅 테이블을 생성한다. GRFML에 속한 모든 라우팅 프로토콜은 HRPCL에서 제공되는 공통의 API를 통해 라우팅 테이블을 검색하거나 참조할 수 있으며 동적으로 라우팅 모듈을 시작 할 수 있다.

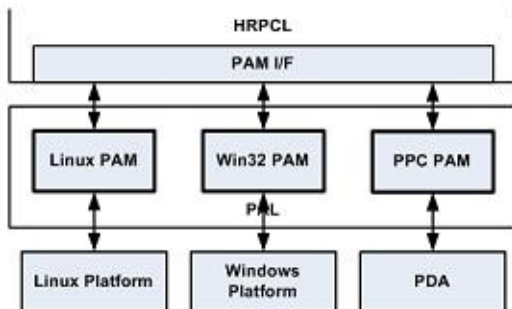


그림 4. PAL 상세 설계
Fig.4. Design for PAL

PAL은 멀티 플랫폼을 위한 것으로 리눅스, 윈도우 등 어떠한 OS에서도 HRPCL가 동작할 수 있도록 하기 위한 계층이다.

IV. HRPC 실험 환경

본 논문에서는 2장에서 HRPC의 고려사항에서 개발한 무선 메쉬라우터의 성능 평가를 위한 적절한 시뮬레이터와 강력한 개발도구가 갖추어져야 된다고 언급했다. 본 장에서는 HRPC 개발 및 실험을 위해 자체 개발한 USE(UzNet Simple Emulator)와 이를 이용한 무선 메쉬라우터 실험 방법을 설명한다.

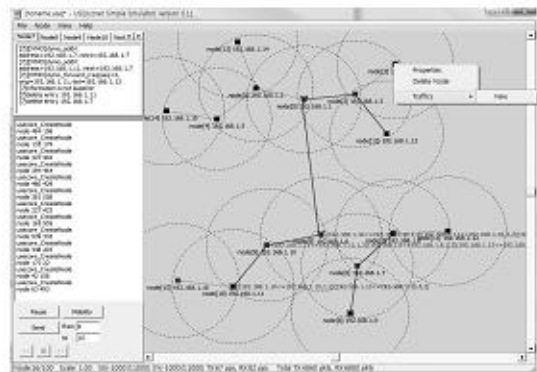


그림 5. USE를 사용한 무선 메쉬라우터 개발
Fig.5. Implementation of Wireless Mesh Router

그림 5는 HRPC의 개발에 사용된 USE이다. USE의 기능은 첫 번째, 에뮬레이터 형태의 개발 도구로 HRPC에 탑재되는 GRFML(DYMO, OLSR)을 통합 테스트 할 수 있다. 두 번째 USE에 사용된 GRFML 및 HRPC 소스는 PC에 그대로 포팅 가능하므로 실제 무선 메쉬 네트워크 환경에서 발생가능한 여러 구조를 미리 테스트할 수 있어 개발이후 발생할 수 있는 문제를 최소화 할 수 있는 장점을 갖는다.

세 번째, USE는 그림6와 같은 형태로 삼은 무선 메쉬 라우터 각각에 그림 5와 같은 무선 메쉬 네트워크 구조를 전달하여 각 장비가 하나의 무선 커버리지에 있지만 서로간에 멀티홉으로 구성된 무선 메쉬 네트워크인 것처럼 설정하여 장비들간의 통신 실험을 할 수 있다. 이때 USE는 MAC[7][8] 필터링 기능을 수행하여 각각의 장비에게 USE 화면(그

림 5)에서 보여지는 자신의 옆 노드의 MAC 주소에 해당되는 패킷은 받고, 그 이외(2 hop 이상) 노드의 MAC 주소 패킷은 드랍 시키라는 정보를 전달한다.



그림 6. 무선 메쉬 라우터 실험 방법
Fig.6. Demonstration of Wireless Mesh Router

따라서 USE를 통해 화면과 같이 무선 메쉬 네트워크를 구성하여 개발 이후 발생될 수 있는 여러 가지 시나리오를 실험할 수 있다.

V. 결 론

본 논문에서는 이종 라우팅 프로토콜의 효율적인 탑재 구조인 HRPC를 제안하였다. HRPC는 ad-hoc 라우팅 프로토콜이 탑재되는 GRPML과 여러 라우팅 프로토콜간의 실행 우선 순위나 패킷 전달을 담당하는 HRPCL, 멀티 플랫폼 지원을 위한 PAL 로 구성된다. HRPC는 무선 메쉬 라우터에 탑재될 ad-hoc 라우팅 프로토콜 개발에 있어 반드시 필요한 구조로 융통성있는 개발구조와 높은 패킷 전달성능을 갖는 이종 라우팅 프로토콜 탑재를 위한 효율적인 구조이다.

참고문헌

- [1] I. F. Akyildiz, X. Wang, W. Wang, "Wireless mesh networks: a survey," *Computer Networks Journal* (Elsevier), vol.47, 2005, pp.445-487.
- [2] E. M. Hoyer, S. Barbara and C.K. Toh, "A Review of Current Routing Protocols for Ad hoc Mobile Wireless Networks," *IEEE Personal Communications*, 1999, pp.46-55.
- [3] IETF Mobile Ad-hoc Networks (MANET) working group, URL: http://www.ietf.org/html_charters/manet-charter.html

- [4] Internet-Drafts, "Dynamic On-Demand (DYMO) Routing Protocol", URL: <http://ietf.org/internet-drafts/draft-ietf-manet-dymo-11.txt>
- [5] RFC 3626, "Optimized Link State Routing Protocol (OLSR)", URL: <http://ietf.org/rfc/rfc3626.txt>
- [6] N. Kang, I. Park and Y. Kim, "Ubiquitous Zone Networking Technologies for Multi-hop Based Wireless Communications," LNCS 4124, Sep. 2006.
- [7] W. Kiess and M. Mauve, "A Survey on Real-World Implementations of Mobile Ad-Hoc Networks," Elsevier' Ad Hoc Networks 5, Apr. 2007.
- [8] A. Kamerman and L. Monteban, "WAVELAN II: a high performance wireless LAN for unlicensed band," Bell Labs Technical Journal, Summer 1997.

저 자 소 개

이 영 석



2004년 : 송실대학교 정보통신 전자공학과 학사.
2006년 : 송실대학교 전자공학과 석사.
현재 : 송실대학교 정보통신과 박사과정중.

관심분야 임베디드 소프트웨어, IMS, Ubiquitous, MANET
Email: youngsuk@dcn.ssu.ac.kr

이 영 석



1984년 : 서울대학교 전자공학과 학사.
1986년 : 한국과학기술원 전기 전자공학 석사.
1990년 : 한국과학기술원 전기 전자공학 박사.
현재 : 송실대 정보통신공학과 교수.

관심분야: MANET, QoS, BcN, IMS
Email: yhkim@dcn.ssu.ac.kr