임베디드 시스템을 위한 IPv6 프로토콜의 설계 및 구현

이학구, 신상현, 김선우, 김영근
삼성전자(주) 디지털미디어 연구소
전화 : 031-200-9309 / 핸드폰 : 018-423-0497

Design and Implementation of IPv6 Protocol for Embedded Systems

Hak Goo Lee, Sang Hyun Shin, Sun Woo Kim, Young Keun Kim
Digital Media R&D Center, Samsung Electronics CO., Ltd.
E-mail : goodguy.lee@samsung.com

Abstract

This paper has been studied on an IPv6 Protocol for Embedded Systems such as Informational Appliances and Mobile Terminals. IPv6 Protocol has 128-bit IP address space. Therefore, all devices come to have a chance of acquiring IPv6 address.

This IPv6 Protocol Stack is very compact and reliable because the targer of the stack is Embedded Systems, which have typically characteristics of low cost and low performance. Hence, we designed and implemented this Stack for This Embedded Systems.

I. 서론

현재 인터넷 프로토콜의 표준인 IPv4는 32 bit 주소 체계를 복잡한 클래스로 나누어 사용함으로 인해 주소 사용의 한계에 다다르고 있다. 또한 헤더가 옵션의 유무에 따라 길이가 가변적이어서 전송 도중 경유 라우터의 프로세싱이 복잡해지는 문제를 야기하고 있다. 게다가 QoS를 지원하기 위한 헤더 자체에 기능이 거의 사장되어 있는 상황이다.

본 논문은 위와 같은 IPv4의 문제점에 대한 대안으로서 IETF에서 제안하고 있는 차세대 인터넷 프로토콜인 IPv6를 가정용 정보기기 또는 휴대 이동 단말 같은 소형 임베디드 시스템에 적합하게 설계하고 구현하였다. 본 논문의 IPv6 프로토콜 스택은 Core Spec.인 Internet Protocol, Version 6 Spec., Neighbor Discovery, IPv6 Stateless Address Autoconfiguration, ICMPv6, Path MTU Discovery등의 표준을 따르고 있다. 그리고 라우터가 아닌, 소형 임베디드 시스템만을 위한 기능을 더욱 특화하여 경량화된 스택을 설계하였다. 그러면서도 다른 IPv6 Node들과 호환성을 유지하는 IPv6 프로토콜을 설계, 구현하였다.

II. IPv6 프로토콜의 구성

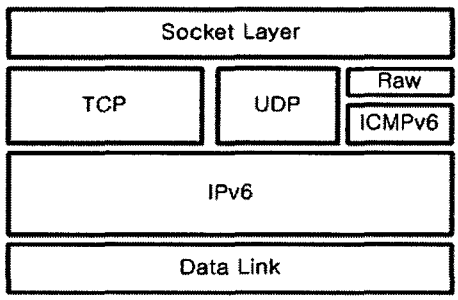

그림 1. 본 논문에서 구현한 임베디드 시스템을 위한 IPv6 프로토콜 스택

2.1 Internet Protocol, Version 6 Specification (RFC 2460)

IPv6와 현재의 IPv4의 가장 큰 차이는 헤더 포맷의 변화이다. 이는 현재 32 bit IPv4 주소의 길이를 128 bit로 대폭 늘리면서 필연적으로 발생하게 되었다. 이로 인해서 헤더 자체의 크기는 이전 기본 IPv4 헤더가 20 byte였던 것이 IPv6는 40 byte로 증가했지만, 그림 2 같이 헤더 포맷 자체는 더욱 간결해졌다. 이는 IPv4의 Fragment에 관련된 항목이 새롭게 만들어진 확장 헤더로 이동하여, IPv6의 기본헤더에서는 사라지게 되었다. 그 외에 IPv4의 TOS필드는 IPv6에서는 Traffic Class와 Flow Label로 세분하여 더 많은 기능을 구현할 수 있는 기반을 제공하고 있다. 현재 IPv4에서 헤더의 길이가 가변적이기 때문에 필요했던 Header Length는 IPv6의 기본헤더가 고정길이이기 때문에 없어졌다. 그리고 IPv4에서 헤더와 데이터의 길이를 모두 더한 값인 Total Length는 IPv6에서 확장헤더와 데이터의 합인 Payload Length로 변화했다. 이외에 TTL은 Hop Limit로, Protocol은 Next Header로 변화했다.

IPv6의 확장헤더의 종류로는 다음과 같다. 경유하는 노드인 라우터의 특정 동작을 요구하는 Hop-by-Hop

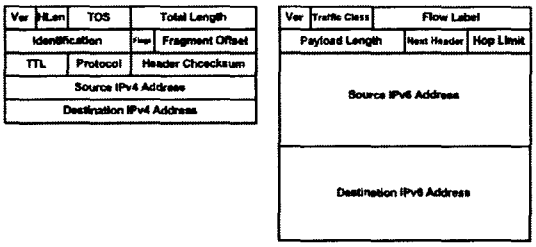

그림 2. . IPv4와 IPv6 헤더의 비교

Option Header, 특정 라우터를 필히 거치도록 요구하는 Routing Header, 현재 IPv4의 헤더에 포함되어 있던 Fragment에 관련된 내용이 확장헤더로 이동하면서 만들어진 Fragment Header, 도착하는 Destination Node에게 특정 동작을 요구하는 Destination Option Header가 있다. Security를 위한 확장헤더 중 인증을 위한 Athentication Header와 Encryption을 위한 Encapsulation Security Payload가 있다.

2.2 Neighbor Discovery for IPv6 (RFC 2461)

(1) Neighbor Solicitation & Neighbor Advertisement

ND6는 현재 IPv4에서의 ARP와 IPv6에서 새롭게 추가된 Stateless Address Autoconfiguration을 위한 기반을 제공한다. 현재 ARP는 IPv4 하단에 존재한다. 그러나 IPv6에서 ND6는 ICMPv6를 서브 프로토콜로 사용하고 있다.

ARP의 주된 목적은 3계층의 IPv4 주소를 사용하는 Node의 2계층 주소를 알고자 하는 목적으로 사용하는데, 이를 대체하는 ND6의 프로토콜로는 Neighbor Solicitation과 Neighbor Advertisement가 있다. NS는 동일 링크상에 존재하는 Node의 2계층 주소를 알기 위해서 그림 3과 같이 멀티캐스트를 이용하여 전송한다. 그러면 이를 받은 타겟 Node는 자신의 2계층 주소를 기입하여 해당 Node에게 NA를 전송하여 통신이 가능하게 해준다.

(2) Router Solicitation & Router Advertisement

현재의 IPv4에서는 IP 주소를 일일이 직접 입력해야 했다. 이를 피하기 위해 DHCP 같은 Stateful Address Autoconfiguration을 사용하기도 하지만, 이는 임시적인 주소를 일정 시간동안 임대하는 방식이므로 완벽한

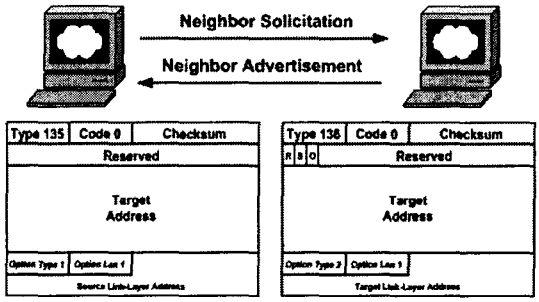

그림 3. Neighbor Solicitation 과 Neighbor Advertisement

Autoconfiguration이라고 할 수 없다.

IPv6에서는 Stateless Address Autoconfiguration을 사용하여 완벽한 Autoconfiguration을 달성했다. 이를 위해서 ND6에서 Router Solicitation과 Router Advertisement라는 프로토콜을 사용하였다. 즉 그림 4와 같이 RA에 삽입되어 있는 Prefix Information Option의 Network Prefix와 Network Interface ID를 결합하여 IPv6 주소를 생성한다. 그리고 RS는 이와 같은 RA정보를 필요로 할 때 해당 호스트가 이 메시지를 생성하여 멀티캐스트로 전송함으로서 해당 링크에 존재하는 라우터가 RA메시지를 생성하도록 요청하는 메시지이다.

(3) Redirect

동일 링크상에 다수의 라우터가 존재할 때, 특정 Destination Node로 가고자 하는 패킷에 대해서 이 메시지를 전송하는 현재의 라우터보다 더 나은 라우터가

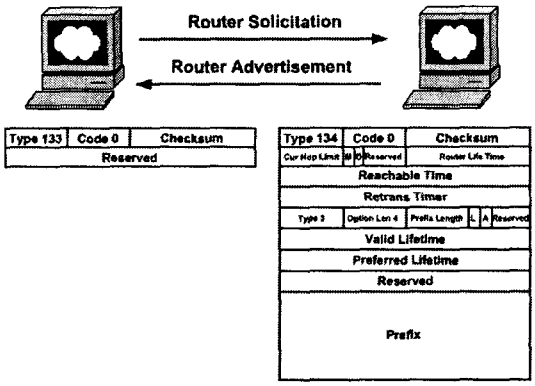

그림 4. Router Solicitation & Router Advertisement

존재할 때 이 메시지를 전송한다. 이 메시지를 받은 호스트는 해당 Destination Node로 패킷을 전송하고자 할 때, 해당 Redirect된 라우터로 패킷을 전송한다.

2.3 IPv6 Stateless Address Autoconfiguration (RFC 2462)

IPv6의 가장 큰 장점중의 하나가 바로 Stateless Address Autoconfiguration이다. 즉 사용자가 직접 IP 주소를 입력하지 않아도, 그림 5와 같이 자동적으로 RA 메시지에 포함되어 있는 Prefix Information Option의 Network Prefix와 자신의 Interface ID를 결합하여 유일한 IPv6 주소를 생성하기 때문이다.

이는 특히 사용자 인터페이스가 없는 임베디드 시스템에 있어서 큰 장점이 된다. IP 주소를 직접 입력하지 않아도 인터넷을 사용할 수 있기 때문이다. 그래서 IPv6에서는 일반 사용자가 단지 RA기능을 할 수 있는

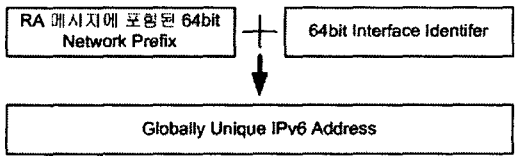

그림 5. Stateless Address Autoconfiguration 과정

라우터인 홈 게이트웨이가 존재하는 링크에 단지 네트워크 케이블을 연결하는 것만으로 세계 어느 곳에서도 접근이 가능하게 되는 것이다.

2.4 ICMPv6 (RFC 2463)

ICMP는 IP의 오류 제어를 할 수 있는 메커니즘을 제공하기 위해서 만들어졌다. 물론 IPv6에서도 이와

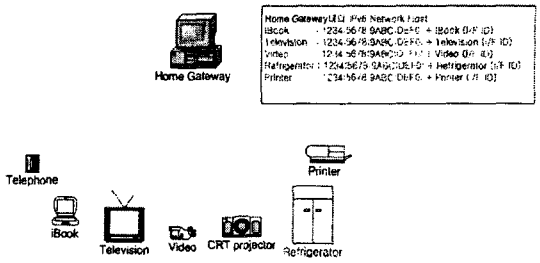

그림 6. IPv6 환경에서의 홈 네트워크 구조

같은 기능을 하고 있다. 거기에 더해져 현재의 ARP, IGMP기능을 모두 ICMPv6로 통합시켰다.

ARP는 Neighbor Discovery로 IGMP는 Multicast Listener Discovery로 변하면서 모두 ICMPv6 프로토콜을 서브 프로토콜로 사용하고 있다. 이들 다양한 프로토콜의 구분은 ICMPv6의 Type 필드로 구분한다. Type 0~127번은 에러 보고를 위한 메시지를 의미한다. 이는 현재의 IPv4의 ICMPv4와 같은 기능을 하고 있다. 그리고 Type 128~255번까지는 정보전달 메시지들이다. 이 부분에 ND6, MLD등이 사용하고 있다.

2.5 Path MTU Discovery for IP version 6 (RFC 1981)

현재의 IPv4에서 패킷이 전송되던 중 패킷의 크기보다 작은 홉간 MTU를 만나게 되면 해당 라우터에 의해서 Fragment가 된다. 그러나 IPv6에서는 라우터의 부하를 경감시키기 위해서 라우터에 의해서 Fragment되는 경우가 없어졌다. 대신, 패킷이 전송되던 중에 패킷의 크기보다 작은 홉간 MTU를 만나게 되면 패킷은 해당 라우터에 의해 폐기되고 라우터는 MTU를 기입한 ICMP Packet Too Big메시지를 패킷을 보낸 Node에게 전송하게 된다. 이 메시지를 받은 Node는 ICMP Packet Too Big메시지 안에 있는 MTU필드 바탕으로 해서 패킷을 Fragment해서 전송하게 된다.

III. 임베디드 시스템을 위한 IPv6 프로토콜의 구현 및 검증

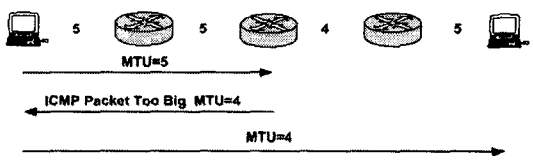

그림 7. Path MTU Discovery 과정

본 논문에서 구현한 IPv6 프로토콜은 IPv6 뿐만 아니라, TCP/UDP/RawIP를 포함하고 있다. 또한 Socket도 포함되어 기본적인 Network 프로그램을 제작할 수 있는 API도 제공하고 있다. 그리고 경량화를 위해 IPv6만을 지원하고 있다. 그리고 라우터 또는 게이트웨이를 위한 IPv6 표준에서 제안하는 기능들은 구현하지 않았다.

이렇게 구현된 IPv6 프로토콜 스택은 RTOS 분야에서 가장 많이 사용되고 있는 VxWorks에 포팅 하였다. 동작하는 플랫폼은 Intel PXA 250 CPU를 사용하는 개발 보드에서 작동한다.

3.1 테스트베드에서의 상호 연동

IPv6 프로토콜 스택의 연동 테스트는 Linux, BSD, Windows XP등의 IPv6가 지원되는 PC Platform들로 구성되어 있는 테스트베드에서 실험하였다. 그림 8의 내용은 개발 보드가 라우터머신으로부터 RA를 받아 IPv6 주소를 자동 생성하고 Ping6 Test를 하는 그림이다.

3.2 Conformance Test Suite을 통한 결과

본 논문에서 설계하고 구현한 IPv6 스택에 대한

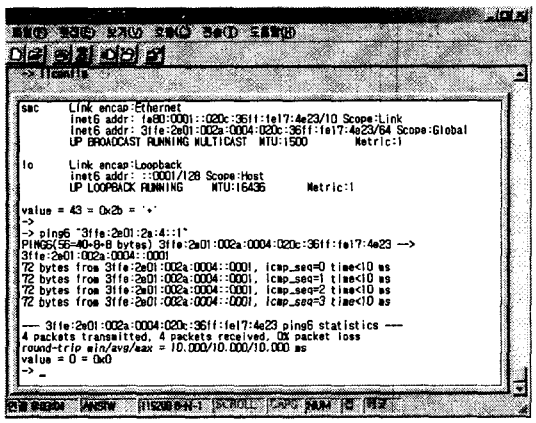

그림 8. 본 논문에서 제안하는 IPv6 스택을 지닌 보드가 직렬통신을 통해서 출력한 내용 (주소생성, Ping6를 동작)

Conformance Test는 오픈된 테스트 슈트이지만 가장 공신력이 있는 Tahi Test Suite를 사용했다. 테스트의 결과 중 Autoconf.의 경우 rlogin을 지원하지 못해 테스트를 하지 못했다.

표 1. Tahi Conformance Test의 결과

테스트 종류	결과			
	총 항목	Pass	Fail	Warn
IPv6 Spec.	50 [80]	49	0	1
ICMPv6	16 [23]	16	0	0
ND6	52 [81]	49	0	3
AutoConf.	*[57]	*	*	*
Path MTU	3 [5]	2	0	1
Robustness	2 [4]	2	0	0

(*) rlogin이 지원이 안되서 테스트를 못했음
[ ] 안의 값은 라우터를 테스트하기 위한 항목을 포함한 실제 총 항목

Ⅳ. 결론

본 논문은 가정용 정보기기 또는 이동 휴대 단말과 같은 소형 임베디드 시스템에 탑재될 수 있는 IPv6 스택을 설계하고 구현하였다. 또 IPv6관련 다양한 RFC에서 제안하는 기능 중에서 라우터에 관련된 기능들은 제외하고, 단지 호스트만 위한 스펙들을 설계 구현하였다. 그리고 테스트베드에 구성된 신뢰할 수 있는 IPv6 Node들과의 호환성 테스트와 Conformance Test를 통해 IPv6 프로토콜의 신뢰성을 얻었다. 이는 조만간 도래할 IPv6 네트워크 환경에 능동적으로 대처할 수 있는 핵심 기술을 얻었다는데 중요한 의의가 있다.

본 논문은 차후에 활성화될 IPv6 프로토콜을 구현하는 System에 있어 좋은 참고자료가 될 수 있을 것이다. 그리고 이후의 과제로서 IPsec 프로토콜과 Mobile IPv6 프로토콜을 구현하고자 한다.

참고문헌(또는 Reference)

[1] S.Deering and R.Hinden, "Internet Protocol, Version 6(IPv6) Specification", IETF RFC 2460, December 1998.

[2] T.Narten, E.Normark and W.Simpson, "Neighbor Discovery for IPv6", IETF RFC 2461, December 1998.

[3] S.Tomson and T.Narten, "IPv6 Stateless Address Autoconfiguration", IETF RFC 2462, December 1998

[4] A.Conta and S.Deering, "Internet Control Message Protocol for IPv6", IETF RFC, December 1998.

[5] J.McCann, S.Deering and J.Mogul "Path MTU Discovery for IPv6", IETF RFC, August 1996.

[6] B.A. Forouzan "TCP/IP Protocol Suite", McGraw-Hill, June 1999.