GSM Phone 상에서 Linux IPv6 프로토콜 스택 설계 및 구현

(Implementation and design of Linux IPv6 Protocol Stack on GSM Phone)

이 상 우 † 임 동 화 † 한 동 환 †† 노 순 억 ††
(Sang-Woo Lee) (Dong-Hwa Lim) (Bosco Han) (Sun-Ok Rho)

요 약 초기 인터넷 환경이 전세계적으로 급속히 확대되면서 IPv4의 32Bit의 주소공간이 얼마 되지 않아 고갈될 것으로 예상이 된다. 따라서 주소 고갈 문제를 해결하기 위해 IPv6의 효과적인 전이 방안으로의 기술 전이가 필연적으로 이루어 질 것이다. 현재, 무선인터넷의 활성화로 인한 무선인터넷 가입자의 급속한 증가와 이동통신망에서의 IP 주소 요구가 예상되기 때문에 주소자원의 부족문제는 더욱 커질 것이다. 본 논문에서는 GSM Phone 상에서 Linux kernel 2.4 기반의 IPv6 Protocol Stack 을 GSM Phone 환경에 맞게 설계하고 구현하여 GSM Phone 기반의 IPv4 Protocol Stack과 Dual Stack으로 동작하도록 하였다. 테스트환경은 GSM 망을 통한 IPv6 테스트가 불가능하기 때문에 IPv6를 지원하는 PPP연결을 통하여 GSM Phone 상에서 IPv4/IPv6 Protocol Stack이 정상적으로 동작함을 확인하였다. 그리고 GSM Phone 상에서 TCP/UDP 데이타를 전송하여 IPv4와 IPv6의 성능을 비교하였다.

키워드 : IPv6 프로토콜 스택, IPv4/IPv6, 리눅스 IPv6

Abstract It is well known that, in the near future, the lifetime of the IPv4 address space will be limited and available 32-bit IP network addresses will not be left any more. In order to solve such IPv4 address space problem in an effective way, the transition to the new version using IPv6 architecture is inevitably required. This paper presents the design and implementation of IPv4/IPv6 dual stack at the GSM Phone based on Linux Kernel 2.4 IPv6 Protocol Stack. It designs appropriately in GSM Phone environment and it is tested by a network of Linux IPv4/IPv6 dual stack on PPP. The test was processed with a test scenario and it was found that the results were successful.

Key words : IPv6 (Internet Protocol version 6), IPv4/IPv6, PPPv6, Linux IPv6

1. 서 론

현재 IPv4(Internet Protocol version 4) 주소를 이용한 인터넷 환경은 초기 인터넷 환경과는 달리 통신기술의 급진적인 발달과 인터넷 사용자 수의 기하급수적인 증가, 그리고 사용자들의 새롭고 다양한 서비스에 대한 요구로 인해 많은 변화가 발생하였다. 또한, 무선인터넷의 활성화로 인한 무선인터넷 가입자의 급속한 증가와 이동통신망에서의 IP 주소 요구가 예상되기 때문에 주소자원의 부족문제는 더욱 커질 것이다. 제 3세대 이동통신 시스템인 GPRS[1]는 현재 IPv4를 기반으로 MS (Mobile Station)에게 패킷 서비스를 제공하고 있지만, IPv4의 주소 부족 현상이 발생함에 따라 MS에게 IPv4 주소를 할당하는데 한계가 있다. 3GPP(3rd Generation Partnership Project)의 Release 2000 IM CN 서브 시스템[2]에서는 멀티미디어 서비스 지원을 위해 인터넷에 연결되어 있는 모든 MS에게 IP를 할당할 수 있도록 IPv6지원을 필수로 정하고 있다. 따라서 현재 개발되어진 GPRS에서 IPv6를 지원하기 위해 고려해야 할 사항은 다음과 같다. 첫 번째는 MS와 CN(Correspondent Node)에서 동작하고 있는 어플리케이션이 어떠한 IP모드(IPv6 모드, IPv4 모드)로 동작하는지에 대해 고려해야 한다[3]. 두 번째는 GPRS망이 어떠한 외부 인터넷망(IPv6 Internet, IPv4 Internet)과 연동하는지에 따라

† 정 회 원 : 익소로직 개발그룹1
charming@ixologic.com
dh7107@ixologic.com
†† 비 회 원 : 익소로직 개발그룹1
bosco@ixologic.com
sorho@ixologic.com
논문접수 : 2005년 2월 18일
심사완료 : 2006년 11월 16일

IP 패킷 전송 방식이 달라진다. 그러므로 본 논문에서는 기존의 IPv4로 동작하는 GSM Phone에 IPv6를 구현하여 Dual Stack으로 동작하게 한다. GSM Phone 상에서 Linux kernel 2.4 기반의 IPv6 Protocol Stack 을 Phone 환경에 맞게 설계하고 구현하여 GSM 폰 기반의 IPv4 Protocol Stack과 Dual Stack으로 동작하도록 하였다. 테스트환경은 GSM 망을 통한 IPv6 테스트가 불가능하기 때문에 IPv6를 지원하는 PPP(Point to Point tunneling Protocol)를 사용하여 GSM Phone 상에서 IPv4/IPv6 Protocol Stack이 정상적으로 동작함을 확인하였다.

본 논문은 2장에서 IPv4와 IPv6 프로토콜 구조와 Test GSM Phone Structure를 간단히 설명하고, 3장에서 테스트 환경의 토대가 되는 PPP를 GPRS망과 비교하여 생각해보고, 4장에서는 GSM Phone에 맞게 설계 및 구현 방법을 설명한다. 그리고 5장에서는 환경 구성과 테스트 결과를 보여주고 6장에서 결론을 맺는다.

2. IPv4/IPv6 Protocol과 Test GSM Phone Structure

2.1 General IPv4/IPv6 Protocol Stack

IPv4와 IPv6의 가장 큰 차이점은 그림 1,2에서 보는 봐와 같이 IP 주소 크기를 32비트에서 128비트로 확장된 것이다. 이에 따라, 주소 체계가 단계적으로 증가하고, 더 많은 노드에 체계적으로 주소를 설정할 수 있으며, 자동 주소 설정(Auto-configuration)이 가능해졌다. 패킷 처리 비용을 줄이고 IPv6 헤더의 대역폭 비용을 제한하기 위해 일부 IPv4 헤더 필드가 삭제되거나 확장 헤더로 되어 선택적으로 사용되게 되었다[4].

ICMPv6는 IPv6 노드에서 패킷 처리 시 발생한 에러를 보고하거나, 진단 등과 같은 기타 인터넷 계층의 기능을 수행하기 위해 사용된다. 따라서 IPv6 기본 프로

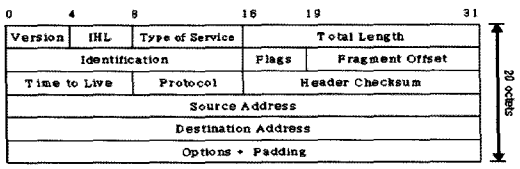


그림 1 IPv4 헤더 기본 구조

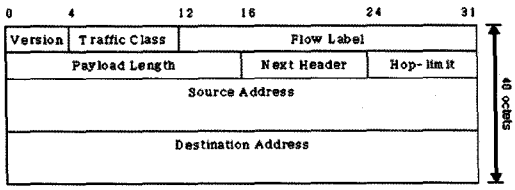


그림 2 IPv6 헤더 기본 구조

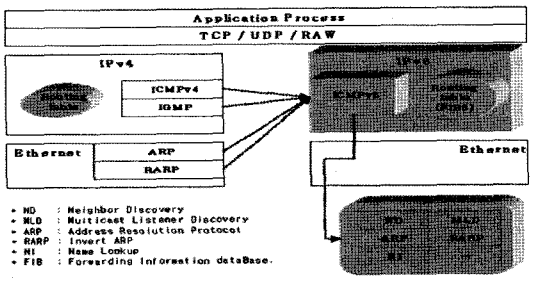


그림 3 ICMPv4/ICMPv6 비교

토콜과 한 쌍이라 할 수 있으므로, IPv6를 구현하는 노드들은 ICMPv6를 구현할 것을 요구하고 있다. 그 외, Linux에서는 IPv6에 맞는 보다 빠른 라우팅을 위하여 FIB6(Forwarding Information DataBase version 6)를 사용하고 있다[4].

2.2 Test GSM Phone Structure

본 논문의 테스트를 위한 GSM Phone은 ARM7[5] 계열의 CPU가 내장되어 있으며, ANSI-C기반의 Nucleus Plus RTOS(Real Time Operating System)[6]를 사용하고 있다.

2.2.1 Test GSM Phone Protocol Flow

그림 4는 Test GSM Phone의 전체적인 Protocol Flow를 보여주고 있다. DS Socket을 생성하게 되면 Timer가 동작해 주기적으로 TIPMAIN(test phone의 network process core)을 실행시킨다. TIPMAIN은 Current state에 따라 PPP, IPv4, Transport layer Process를 동작시킨다. 여기에 본 논문은 일반적인 RTOS하의 Phone에 Linux IPv6 모듈을 추가 구성하여 동작시키고자 한다. 현재 GSM Phone은 Linux와 달리 RTOS로 이루어져 있어 Linux IPv6를 그대로 사용하기엔 한계가 있다. 따라서, Linux IPv6를 간소화하고 RTOS기반하의 Test GSM Phone Process에 호환되도록 구현한 방법을 4절에 설명하고자 한다. 이를 위하여 PPP Process를 확장하여 PPPv6도 같이 구현되어졌다 (4.3절 참조).

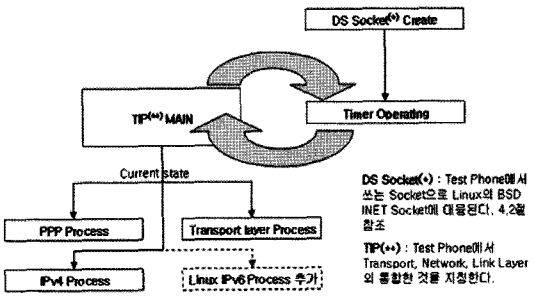


그림 4 Test GSM Phone의 Protocol Flow

3. GPRS 망과 IPv6 over PPP

GPRS 망에서의 PDP(packet data protocol) context를 살펴보고, 실제 테스트 환경의 PPP connection과 비교한다.

3.1 GPRS 망과 PDP Context

GPRS를 기존 GSM 구조에 통합하기 위해, GSN(GPRS Support Node)으로 명칭된 새로운 네트워 구성 요소가 도입되었다[7]. GSN은 이동단말과 패킷 데이타 네트웍(PDN : Packet Data Network) 사이에서 데이타를 전달하고 라우팅하는 역할을 담당한다.

SGSN(Serving GPRS Support Node)은 관리 영역 안에 들어있는 이동 단말이 보내는 또는 이동 단말로 향하는 패킷을 전달하는 역할을 담당한다[8]. SGSN이 수행하는 기능에는, 패킷 라우팅 및 전달, 이동성 관리(attach, detach, 위치 관리(location management)), 논리적 링크 관리, 그리고 인증 및 과금 등이 포함된다. 관리 영역 안에 들어있는 가입자들에 대한 위치 정보(현재 셀, 현재 VLR(Visitor Location Register)), 사용자 프로파일(IMSI(International Mobile Station Identity), PDN 에서 사용하는 주소) 등은 SGSN이 보유한 데이타베이스에 저장된다.

GGSN(Gateway GPRS Support Node)는 GPRS 백본(backbone) 네트웍과 외부(GPRS 네트웍이 아닌) 패킷 데이타 네트웍 사이에 위치한다[8]. SGSN으로부터 전달된 패킷은 GGSN을 통해 적절한 PDP(Packet Data Protocol)형식으로 바뀌어 상대방 패킷 데이타 네트웍으로 보내진다. 역으로 외부 패킷 데이타 네트웍으로부터 전달된 패킷은 그 반대의 과정을 거치게 된다. 여기서 주된 작업은 외부 패킷 데이타 네트웍에서 유효한 PDP 주소와 GSM 네트웍에서 유효한 단말의 주소 사이의 변환 작업이다. GGSN은 외부 패킷 데이타 네트웍으로부터 받은 패킷을 이동 단말에게 전달하기 위해 현재 이동 단말을 관리하는 SGSN을 알아야 한다. 따라서 GGSN이 보유한 데이타베이스는 사용자 프로파일과 함께 사용자의 현재 SGSN 정보를 같이 저장한다. SGSN과 유사하게 GGSN도 인증과 과금 기능을 수행한다.

그림 5는 GPRS망에서의 GSM Phone의 dual stack 동작을 보여주고 있다[2]. 그림 6은 GPRS망에서 GSM Phone에 효과적으로 Global address를 할당하는 방안을 제시한다[9]. 첫 번째 PDP(Packet Data Protocol) context Activation 절차를 통하여 Phone에 link-local address가 할당이 되고, GGSN에도 등록이 된다. 이어서, Phone에서는 RS(Router Solicitation) 메시지를 보내고 GGSN으로부터 RA(Router Advertisement) 메시

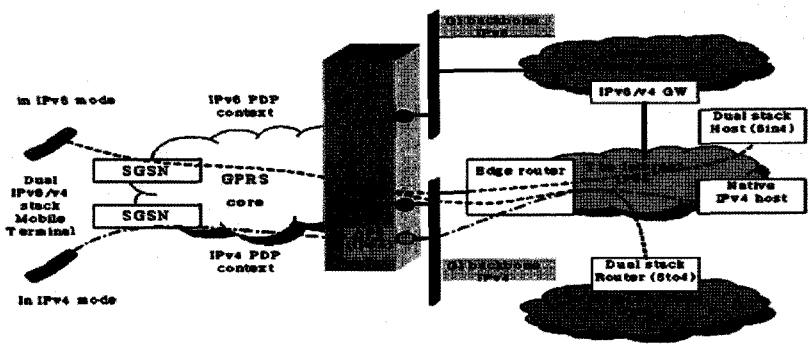


그림 5 Connections to GGSN IPv4 and IPv6 Access Point

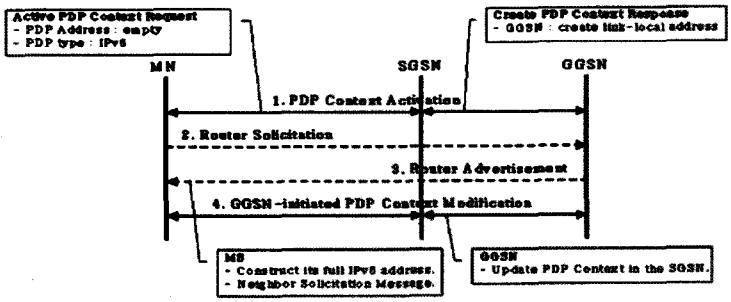


그림 6 IPv6 Address And Addressing

지를 받아 Global address를 설정하게 된다. 마지막으로 이에 대한 정보를 SGSN에 Update한다.

3.2 IPv6 over PPP (PPPv6)

PPP는 점대점 링크상에서 다중 프로토콜 데이타그램을 전송하기 위한 표준 방법을 제공한다. 그 구성요소로는 인캡슐레이션, LCP(Link control Protocol), NCP (Network control Protocol)가 있다. 인캡슐레이션은 시리얼 링크를 통하여 다중 프로토콜 데이타그램을 캡슐화하는 방법을 말한다. 데이타 링크 연결을 형상, 구성, 시험을 위한 링크 제어 프로토콜인 LCP는 다양한 환경에 적응 가능하고, 인캡슐레이션 포맷 옵션의 협상, 다양한 크기의 패킷 처리, 오류 검출 링크 종단 등을 담당한다. 또한, 여러 다른 망 계층 프로토콜의 형상, 구성을 위한 프로토콜이 네트워크 계층 프로토콜(Network Control Protocol, NCP)이다. PPP는 항상 여러 네트워크 계층 프로토콜을 동시에 사용하도록 고안되었으며, 네트워크 계층 프로토콜의 여러 가지 문제점을 해소하며 네트워크 프로토콜에 종속된다.

그림 7은 각 단계별 메시지 교환 절차를 나타낸다. 링크 설정 단계(Link Establishment Phase)는 링크 제어 프로토콜(LCP)이 패킷의 교환을 통해 연결을 설정하는데 이용된다. 이 교환이 끝나면, 일단 Configure-Ack 패킷을 모두 보내지고 받았으면, LCP 연결(connected) 상태로 들어간다. 개별 네트워크 계층 프로토콜의 협상은 네트워크 계층 프로토콜 단계 동안 별도의 네트워크 제어 프로토콜(NCP)에 의해 처리된다. 네트워크 계층 프로토콜 단계(Network-Layer Protocol Phase)에서는 Dual Stack일 경우, IPCP와 IPv6CP를 동시에 협상하게 된다. 이 협상 단계에서 IPv4 주소와 IPv6의 link-local 주소가 할당되게 된다. (그림 6의 PDP link-local 주소의 할당과 유사하다.) 이후 NCP가 연결(connected) 상태에 도달하면, PPP는 IPv4와 IPv6 프로토콜의 패킷을 전송할 수 있다. NCP가 연결 상태에 있지 않을 때 받은 어떤 네트워크 계층 프로토콜도 폐기된다. PPP는 언제라도 링크 종료 단계(Link Termination Phase)를 실행할 수 있으며, 적절한 조치를 하도록 네트워크 계층 프로토콜에게 알린다.

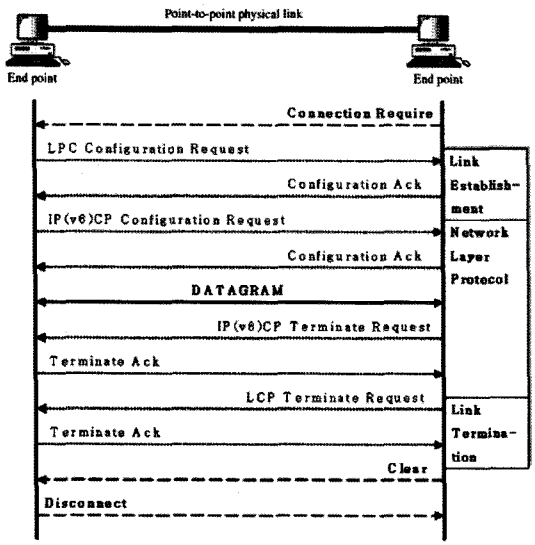


그림 7 PPP Phase Diagram

그림 8에서는 PPP Frame Format을 보여주고 있다. LCP NCP의 구별은 Payload 패킷의 type을 나타내는 Protocol Type field에서 하게 된다. Flag는 1byte로 프레임의 처음과 끝을 나타내고, Address 역시 1byte로 2진수 11111111를 포함하며, 브로드캐스트 주소의 표준이다. Control는 1byte로 링크 개설, 종료를 관리하게 되고, FCS(Frame Check Sequence)는 2bytes로 에러 발견 등을 위해 사용된다[10].

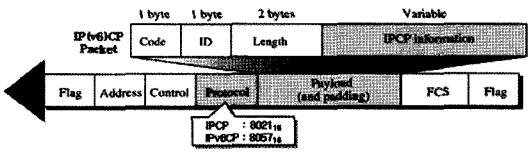


그림 8 PPP Frame Format (NCP)

또한, PPP frame에서 Datagram전송 시 protocol type이 IPv4일 경우 002116, IPv6일 경우 005716로 식별할 수 있고, 이는 IPv4/IPv6 dual stack의 분기점이 된다.

표 1 PPP protocol type

TYPE	NUMBER	Payload
LCP	0xC021	Code, ID, Length, LCP-packet
IPCP	0x8021	Code, ID, Length, IPCP-packet
IPv6CP	0x8057	Code, ID, Length, IPv6CP-packet
Datagram(IPv4)	0x0021	IPv4-original-packet
Datagram(IPv6)	0x0057	IPv6-original-packet

4. GSM Phone 상의 IPv4/IPv6 모듈 설계 및 구현

Linux 2.4 기반의 IPv6 Protocol Stack을 GSM Phone 환경에 맞게 설계하기 위하여 Linux의 IPv6를 살펴보고 그 구조를 수정하여 Phone 환경에 맞게 설계 및 구현하였다.

4.1 Linux Network Structure

Linux Network는 그림 9와 같이 기본적으로 사용자 애플리케이션에 제공하기 위해서 BSD(Berkeley Soft-

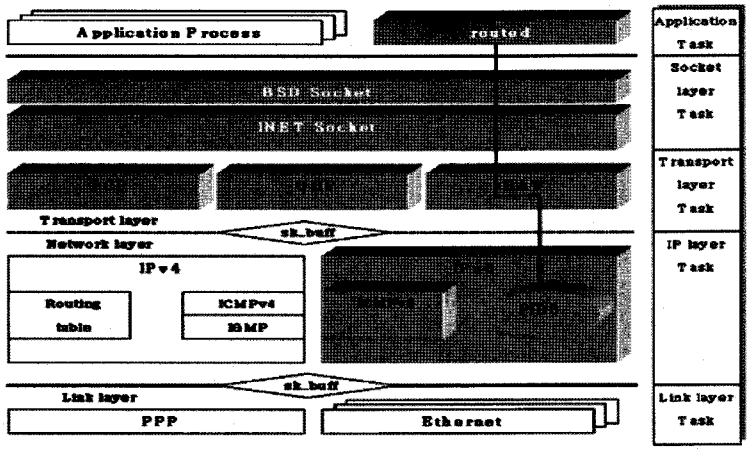


그림 9 Linux Network Structure

ware Distribution) Socket interface를 두었고 다시 이것을 INET(International NETworking) Socket을 통하여 Transport layer로 전송하고 있다[11]. 하나의 프로세스는 파일 디스크립터 테이블을 가지고 있으며, 이것을 통해서 생성된 소켓을 접근한다. File object의 파일 연산은 BSD 소켓이 제공하는 연산을 사용하게 된다. 그러므로, 생성된 BSD 소켓에 대해서 일반 파일에 대한 연산이 동일하게 적용될 수 있게 하는데 BSD 소켓의 목적이 있다고 하겠다. 또한, Link layer는 상위로부터 넘겨받은 data와는 독립적으로 동작한다. Layer별로 각각의 TASK가 동작하게 되고, User가 Routing Table에 직접 접근 가능하도록 하여 라우팅을 사용자가 직접 설정할 수 있게 되어있다.

그림 10에서 보면 하나의 BSD 소켓에 하나의 INET sock구조체를 가진다. 그리고, INET sock구조체에 대해서 들어오거나 나갈 패킷을 가리키는 sk_buff(socket buffer)구조체를 관련 지어서 보여주고 있다. 받거나 보낼 패킷들은 전부 sock구조체에 연결되어 처리를 기다리게 된다. 이곳에선 sk_buff(socket buffer)라는 단위로 데이타를 하위의 physical layer로 보내고, 다시 그곳에서 sk_buff의 단위로 데이타를 받아서, 해당하는 sock구조체에 연결 지어준다.

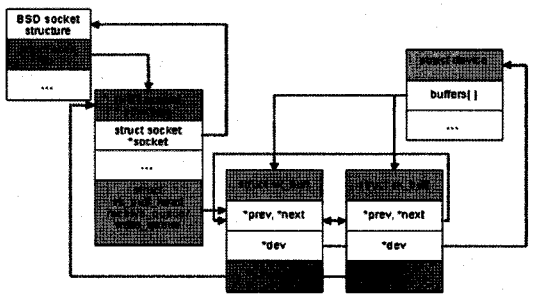


그림 10 BSD socket 및 INET sock, sk_buff의 상관관계

4.2 GSM Phone 상의 IPv4/IPv6 Structure

GSM Phone에서는 그림 11에서 보는 바와 같이 Linux의 Socket, Transport, IP, Link layer Task를 TIP Task로 단순화 시켰다. Routing 기능을 담당하는 라우터 기능을 배제(router X)시키고 Host로서의 Phone의 기능만 수행하도록 설계하여 이에 따른 라우팅 기능을 간소화 하였다. User Interface(폰 단말기)에서는 직접적으로 routing table을 접근하지 않도록 설계하였고, 기본적으로 자동화 된다. BSD Socket의 파일 디스크립터 기능을 제외시키고, INET Socket과 통합하여 DS Socket을 생성하였다.

Linux는 sock구조체를 통하여 각 layer에 전달되어지는 반면, GSM Phone에서는 PPP Buffer에 직접 Read/Write하도록 구현하였다. 패킷 Send 과정의 경우, IPv6 header의 extension header의 삽입에 따라 헤더 길이가 가변적이므로 PPP Buffer에 직접 Write하기에 문제점이 있다. 그러므로, 그림 12와 같이 IPv6 헤더가

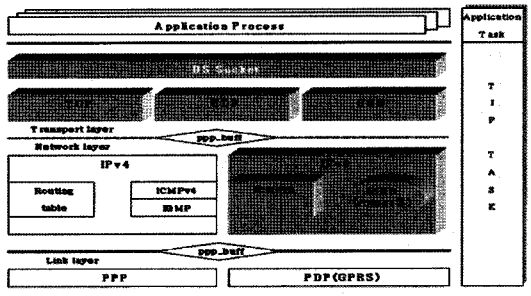


그림 11 GSM Phone 에서의 IPv4/IPv6 모듈 설계

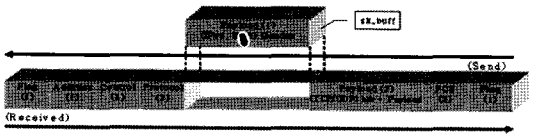
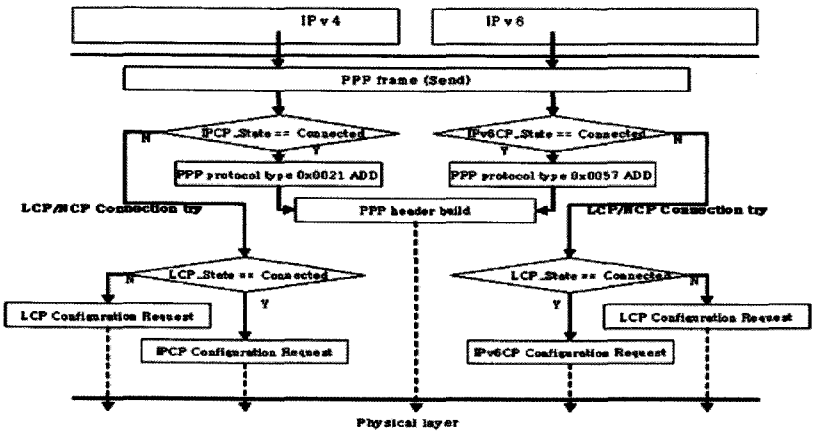


그림 12 PPP Buffer Structure

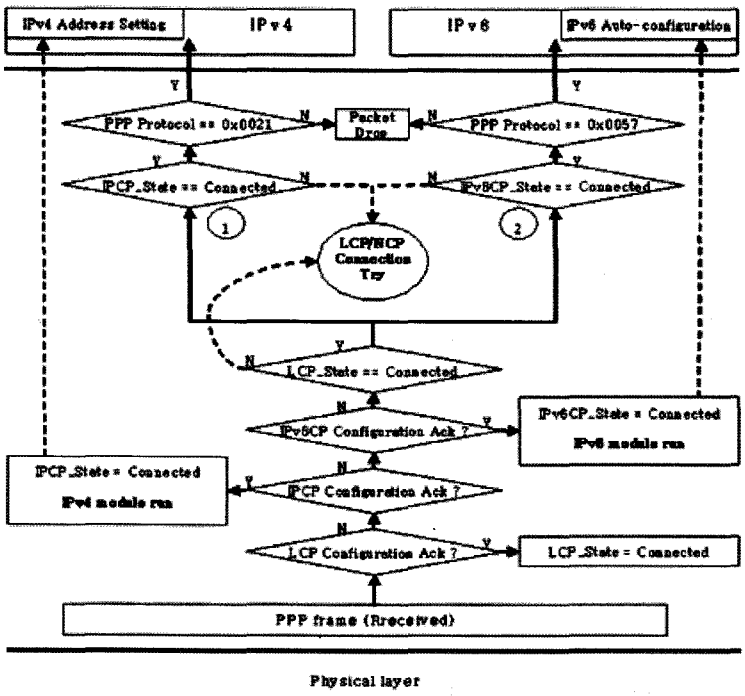
완성이 되면 PPP Buffer로 옮기게 되고, 그 후 Link layer로 보내어 PPP header를 구성하게 된다. 이 때, 만들어지는 IPv6 header는 Linux의 sk_buff와 같은 처리과정을 따르게 된다.

4.3 PPPv6 구현

그림 13은 PPP Input flow와 Output flow를 보여준다. PPP Output flow를 보면 IPv4 통신의 경우 IPCP의 연결 유무에 따라, PPP header를 구성하고 Physical layer로 전송할 것인지를 결정한다. IPCP가 연결되지 않은 상황이라면 LCP/IPCP Connection을 시도하게 된



(a) PPP Output flow



(b) PPP Input flow

그림 13 PPP 동작 절차

다. IPv6의 경우, IPv6CP의 연결 유무를 먼저 판단하게 되고, 그 유무에 따라, Physical layer로 전송할지 LCP/IPv6CP Connection을 시도할지 결정하게 된다.

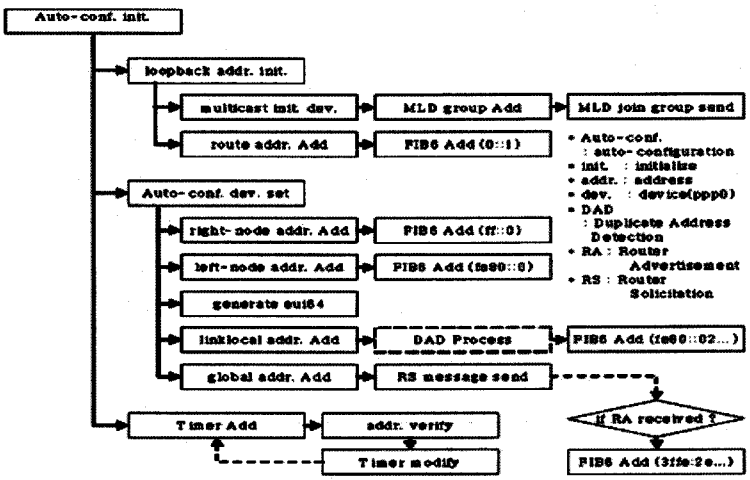
PPP Input flow에서는 LCP/IPCP/IPv6CP Configuration ACK 메시지에 따라 각각의 상태를 설정한다. IPCP가 연결상태가 되면 IPv4 Address가 GSM Phone에 할당되고 IPv4 module이 활성화 된다. IPv6CP가 연결상태가 되면 IPv6 module이 활성화 되고, Auto-configuration과정이 실행된다(4.4절 참조).

4.4 IPv6 구현

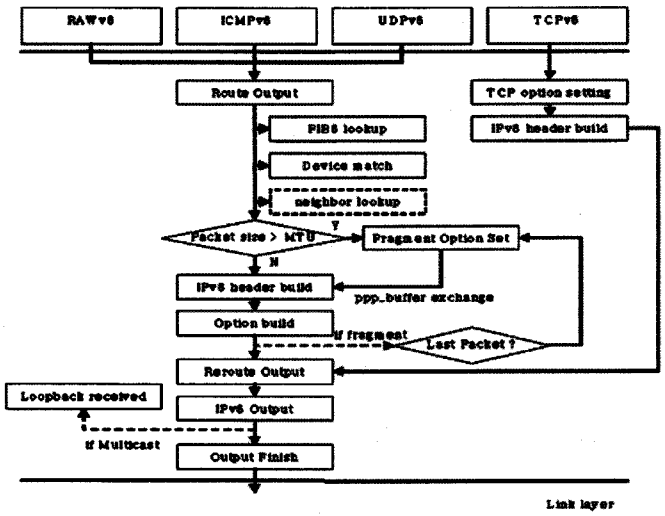
기본적인 Auto-configuration과정과 Input process, output process로 나눌 수 있으며 각 모드에 대한 동작 절차는 그림 14와 같다.

Auto-configuration은 PPP 연결과정에서 받은 128bits link-local 주소 중 prefix 64bits를 제외한 후반 64bits을 interface ID(Identifier)로 설정하게 된다. 이 interface ID는 generate-eui64[10]로 만들어지는 값과 일치되도록 구현되었고, 이후 RA(route Advertisement)메시지로부터 받은 prefix 64bits와 합쳐져 Global Address를 구성하게 된다. PPP 연결 특성상 DAD(Duplicate Address Detection) Check는 하지 않는다.

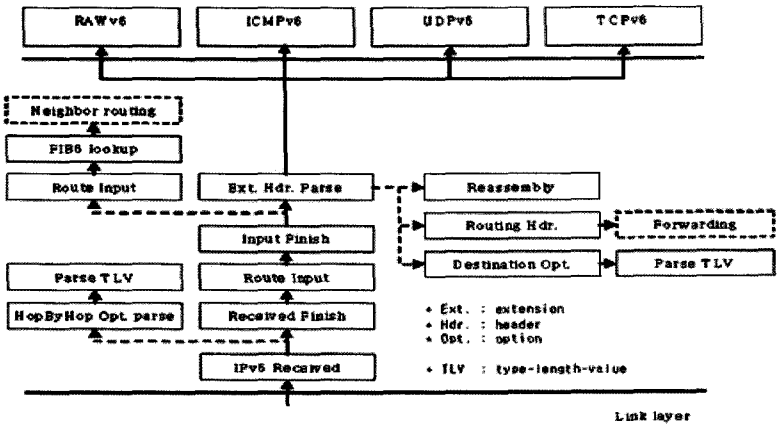
GSM Phone에서의 패킷 송수신 과정에서는 라우팅 기능과 관련된 Forwarding, Neighbor lookup 등의 절차[4]는 생략하였다. (생략된 부분은 그림 14에서 점선



(a) Auto-configuration flow



(b) Output flow



(c) Input flow

그림 14 IPv6 동작 절차

상자처리 하였다.)

5. 환경 구성 및 테스트

5.1 환경 구성

ARM7 계열의 GSM Phone 2개와 팬티엄급 PC에 Linux 2.4.21 커널을 설치하고, Dual Stack을 가진 Router 기능을 추가 시켰다. 구성과 시나리오를 GPRS 망과 유사하게 구현하여, 향후 GPRS망 테스트에 도움이 되고자 했다.

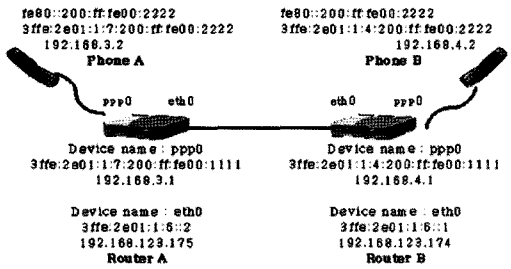


그림 15 테스트 환경 구성

Route A와 Phone A가 PPP 연결을 시도하면 그림 16과 같이 LCP/IPCP/IPv6CP의 메시지 교환이 이루어 지는 것을 볼 수 있다. 연결이 끝나게 되면 Phone A에는 IPv4주소로 192.168.3.2가 할당이 되고, IPv4 module이 활성화 된다. Auto-configuration 과정을 거치고 IPv6 주소 fe80::200:ff:fe00:2222가 할당이 되면, IPv6 module도 활성화 된다. Router B와 Phone B도 PPP 연결이 이루어 지면, Phone B에도 IPv4주소 192.168. 4.2와 IPv6 주소 fe80::200:ff:fe00:2222가 할당이 된다.

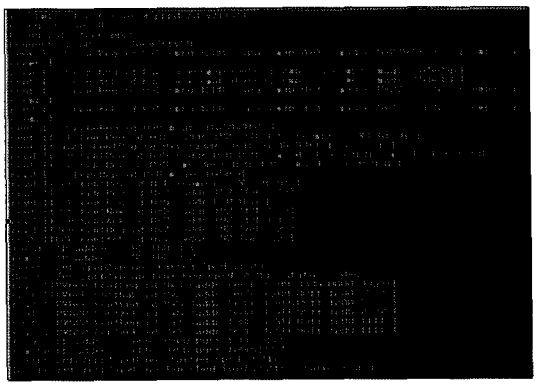


그림 16 PPP Connection

IPv6 라우팅 특성상 link-local 주소로는 Router가 패킷을 라우팅할 수 없다. 라우터를 거쳐 Phone 대 Phone으로 통신을 하기 위해서 Phone에 Global 주소를 할당해야 하고, 라우터에는 이에 따른 라우팅 테이블을 작성해야 한다. 본 테스트를 위하여 Linux 라우터에 radvd[1]모듈을 설치하고 RA prefix 정보 3ffe:2e01:1:7::/64를 Phone으로 전송 시켰다(그림 6의 GPRS망에서의 RS, RA과정과 유사하다.). 그러면 Phone A와 Phone B에 IPv6 주소 3ffe:2e01:1:7:200: ff:fe00:2222와 3ffe:2e01:1:4:200:ff:fe00:2222가 각각 할당이 된다 이렇게 하여 Phone대 Router, Phone대 Phone의 테스트가 가능하였다.

첫 번째로 Phone에 구현된 IPv6의 동작을 검증하기 위하여 Router와 Ping6 테스트를 체크하였다. 그림 17

1) Linux IPv6 Router Advertisement Daemon. IPv6를 지원하는 Linux등에서 사용된다. (by RFC2461)

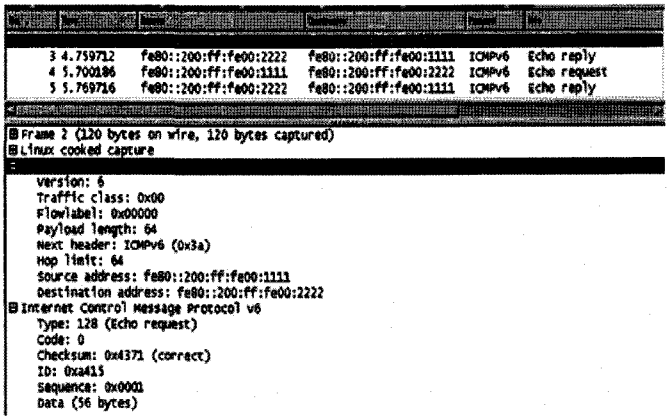


그림 17 Router A 대 Phone A의 Ping6 테스트

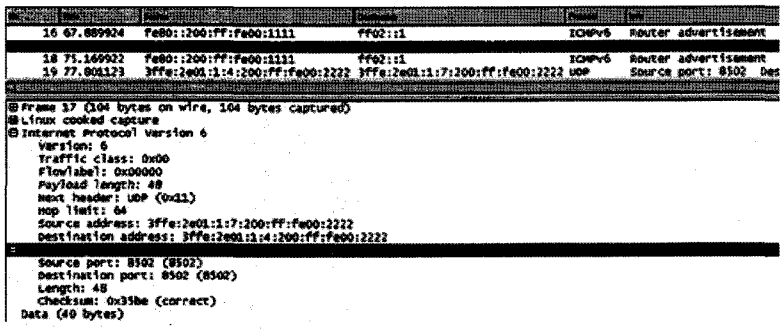


그림 18 Phone A 대 Phone B의 UDPv6 테스트

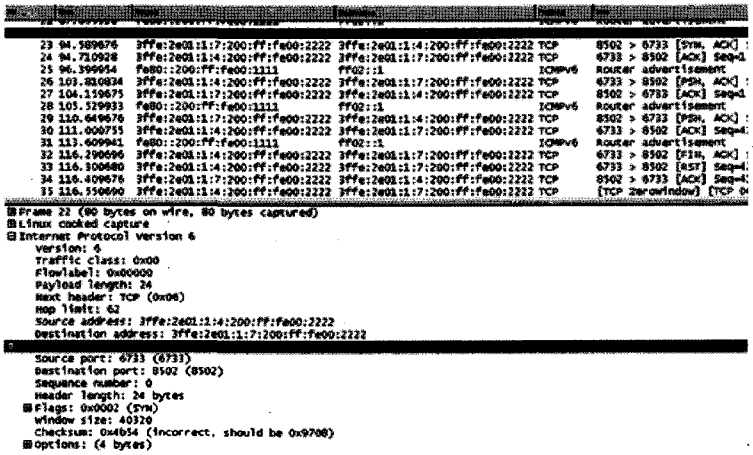


그림 19 Phone A 대 Phone B의 TCPv6 테스트

은 Router A와 Phone A의 link-local 주소의 Ping6 테스트를 ethereal[2]로 확인한 그림이다.

두 번째로 Phone A에서 Phone B로 TCPv6, UDPv6 packet을 보내 응답 받음이 확인되었고, 중간 라우터(Router A, B)에서 ethereal로 각 전달되는 Packet의 이상 유무를 체크하였다. 그림 18은 Phone A에서 Phone B로의 UDPv6데이타 "hello!"메시지를 보낼 때 Router A에서 ethereal로 확인한 그림이다. 그림 19는

2) 전세계의 네트워크전문가들이 네트워크의 문제를 해결하고, 분석하는 네트워크분석기. Windows, Linux를 포함한 모든 플랫폼에서 동작한다.

표 2 GSM Phone 테스트 내용

내 용	구 현	테스트
Auto-Configuration	o	o
TCPv6	o	o
UDPv6	o	o
PING6	o	o
MLD	o	x
RS, RA	o	o
NS, NA	o	x
Fragmentation/Reassembly	o	x
PPPv6	o	o
GPRS 망	o	x
Tunneling (6to6)	o	o
Tunneling (4to6, 6to4, 4to4)	x	x

Phone A에서 Phone B로 TCPv6데이타 "hello!"메시지를 보낼 때 Router A에서 ethereal로 확인한 그림이다. 이 그림에서는 데이타를 보내기 전 TCP Connection에 따른 패킷도 보여주고 있다.

세 번째로 IPv4와 IPv6의 성능을 비교 테스트 해보았다. IPv4의 경우 기존의 GSM Phone을 사용했으며, IPv6의 경우 본 논문에서 구현한 GSM Phone을 사용하였다. 성능 테스트 방법은 Phone A에서 Phone B로 1Kbyte의 UDP 또는 TCP 데이타를 전송시키면 Phone B에서는 받았다는 확인 메시지로 다시 1Kbytes를 보내는 방법을 사용하였다. 1Kbyte를 1024회 보내고 받는 것을 기준으로 IPv4 Phone과 IPv6 Phone의 속도를 체크하였다. 시간 측정은 Router A에서 ethereal로 패킷을 확인하고, 처음 보낸 패킷과 1024번째 받는 패킷(왕복 2048번째 패킷)까지의 시간을 측정하였다.

그림 20은 Original IPv4 GSM Phone과 Dual Stack GSM Phone의 비교 분석한 표이다. TCP/UDP 각각 총 20회를 테스트 해보았다. PPP 연결 상에서 GSM Phone을 이용한 테스트에서 왕복 1Mbyte(총 2Mbytes 송수신)데이타에서 최대 1분 미만의 차이를 보여주고 있다. IPv6 패킷이 IPv4 패킷에 비하여 패킷 구조가 단순화되었다고는 하나, IPv4 header가 기본 20bytes인데 반하여 IPv6 패킷은 기본 40bytes를 가지는 차이와 테스트 망구조가 단순하여 IPv6의 업데이트된 라우팅 기능은 고려하지 않았다는 점으로 볼 때 이 수치는 만족할 만하다고 할 수 있겠다.

5. 결 론

현재 사용 중인 IPv4는 주소 고갈 문제 해결을 포함해 새로 중요도가 부각되고 있는 라우팅의 효율성, 보안 기능, 이동성 지원, QoS 보장 등의 관점에서 볼 때 차세대 인터넷이 추구하는 응용 서비스에 문제점을 지니고 있다[4]. 이러한 환경에서 IPv6로의 전환은 필수적이라 할 수 있다.

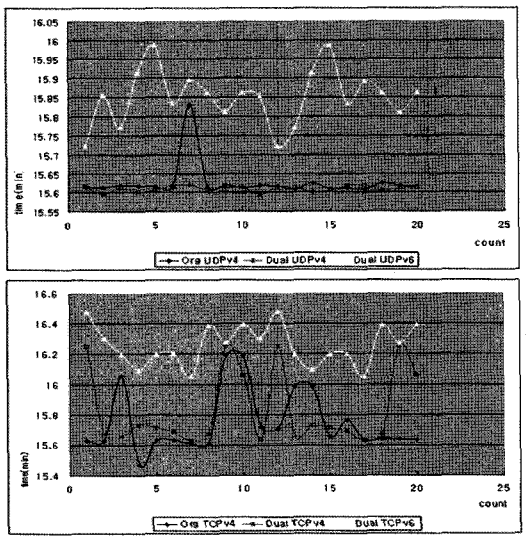


그림 20 성능 테스트 표

이에 따라 본 논문에서는 GSM Phone에 IPv6를 설계하고 구현하였다. IPv6의 설계는 Linux 2.4 Kernel을 분석하고 이를 토대로 GSM Phone에 맞게 간소화하여 설계하였다. 아직 GPRS 망에서 IPv6 지원 테스트가 어려워 PPP 연결을 통한 테스트 망을 구성하고 테스트하였다. IPv4/IPv6 dual stack이 GSM Phone에서 잘 동작함이 확인되었고, 성능의 차이도 크게 없었다. 향후 GPRS 망에서의 테스트가 이루어져야 할 것이며, 표 2에서 테스트 해보지 못한 Multicast지원과 Tunneling도 적절한 망 구성과 테스트가 뒤이어져야 되겠다.

본 논문에서 설계 및 구현된 GSM Phone IPv4/IPv6 dual stack의 결과들은 향후 Phone 및 Embedded 시스템에서의 IPv6구현에 기반 기술로 활용될 수 있을 것으로 생각되며, 인터넷 환경을 위한 효율적인 진화 전략을 수립하는데 많은 도움이 될 것으로 생각된다.

참 고 문 헌

[1] 3GPP, "General Packet Radio Service(GPRS): State 2," 3G TS 23.060 version 4.0.0, March 2001.
[2] NOKIA, "Transition to IPv6 in 2G and 3G mobile networks," White Paper.
[3] 임선화 외 1명, "The Study on based on UMTS/GPRS", 한국정보과학회, October 2002.
[4] 박정수 외 4명, "차세대 인터넷 프로토콜 소개," IPv6 포럼코리아 기술문서, 2000.
[5] ARM(Advanced RISC Machines) WWW Home-page, URL : http://www.arm.com
[6] Nucleus Plus RTOS WWW Home-page, URL : http://www.acceleratedtechnology.co.kr
[7] GSM 03.02. Network architecture.

[8] C. Bettstetter, V. Hans-Jörg, J. Eberspächer, "GSM Phase 2+ General Packet Radio Service GPRS: Architecture, Protocols, and Air Interface," IEEE Communication Surveys, 1999.
[9] 김도완, "IPv6/IPv4 Transition Mechanism & GPRS 에서의 도입방안", Netmanias White Paper, October 2001.
[10] D. Haskin, E. Allen, "IP Version 6 over PPP," IETF RFC 2460, December 1998.
[11] 권수호, "Linux Programming Bible", 글로벌 출판사, 2003.
[12] S. Deering and R, Hinden, "Internet Protocol, Version 6 (IPv6) Address Architecture," IETF RFC 3531, April 2003.
[13] R, Hinden and S. Deering, "Internet Protocol, Version 6 (IPv6) Specification," IETF RFC 2460, December 1998.

이 상 우
2002년 영남대학교 수리통계학부(이학사). 2004년 영남대학교 통계학과(이학석사). 2004년~현재 익소로직(주) 개발 1그룹 선임 연구원. 관심분야는 차세대 인터넷 프로토콜(IPv6, IP Security), VOIP, 음성인식, 등

임 동 화
2000년 대구대학교 컴퓨터공학부(공학사). 2002년 경북대학교 컴퓨터공학과(공학석사). 1999년 바로전송(주) 전산개발팀. 2002년 엠베릭스(주) 연구원. 2002년 인포네트(주) 전산개발팀. 2003년~현재 익소로직(주) 개발 1그룹 선임 연구원. 관심분야는 차세대 인터넷 프로토콜(IPv6, IP Security), 인터넷 통신망, 컴퓨터 구조, 등

한 동 환
1995년 서강대학교 컴퓨터공학과(공학사). 1995년~1998년 LG 소프트 개발팀. 1998년 DIB 개발팀. 1999년 Techna 개발팀. 2000년~2003년 휴림 개발팀. 2003년~2005년 익소로직(주) 개발 1그룹 차장. 2005년~현재 Adobe Systems Korea, 컨설팅, 차장. 관심분야는 IP Telophony, 이동통신 부가서비스 솔루션, 등

노 순 억
1999년 금오공과대학교 컴퓨터공학과(공학사). 2003년 금오공과대학교 컴퓨터공학과(공학석사). 2003년~현재 익소로직(주) 개발 1그룹 선임 연구원. 관심분야는 IPv6, 이동통신, Embedded System, 정보검색