IMT-2000을 위한 GGSN과 FA/ISP 간 패킷 전송에 대한 설계 및 구현

○임선화*, 김영진*, 이윤주*, 김현곤**
한국전자통신연구원 *핵심망기술연구부, **차세대보안응용연구부
{limsh, yjkim, yjlee, hgkim}@etri.re.kr

Design and Implementation on Data Traffic Transmission between GGSN and FA/ISP for IMT-2000

SunHwa Lim*, YeongJin Kim*, YoonJu Lee, HyunGon Kim**
*Core Network Technology Department, ETRI
**Next Generation Security Application Research Department, ETRI

요 약

제 3세대 이동통신 시스템인 UMTS는 무선 인터페이스를 통해 전달되는 IP 패킷을 인터넷 망에 전송하기 위해 GPRS 시스템을 정의하고 있다. GPRS 시스템 중 게이트웨이 기능을 수행하는 GGSN은 이동망과 인터넷 망을 연결시켜주는 역할을 한다. Mobile IP 망에 가입된 가입자 또는 ISP 망에 가입된 가입자 간에 인터넷 서비스를 지속적으로 제공하기 위해 이동통신 망과 인터넷 망에 단일화된 하나의 라우팅 메커니즘이 필요하다. 따라서 본 논문에서는 이들 망 간의 패킷을 효율적으로 라우팅할 수 있는 IP-in-IP 방법을 제안한다. 그리고 제안한 방법에 따라 패킷 전송을 위한 기능, 처리 절차 및 메시지 구조를 설계하고 구현한다. 마지막으로 GGSN과 FA 시스템 간의 패킷 전송률, 그리고 GGSN과 FA/ISP 시스템을 동시에 연결한 상태에서 패킷 전송률을 살펴본다.

1. 서 론

기존의 무선통신 서비스는 빠른 속도의 서비스를 제공하지 못하며 많은 데이터의 전송을 요구하는 멀티미디어와 같은 서비스 통신에는 적합하지 않다. 그리고 인터넷 사용의 증가와 다양한 무선 서비스에 대한 계속적인 수요 증가에 따라 용량이 크고, 데이터 전송 속도가 빠르며, 멀티미디어를 지원할 수 있는 무선 서비스가 필요하게 되었다. 이에 따라 인터넷을 무선 통신 시스템과 연동시키려는 연구들이 활발하게 진행되고 있다. 3GPP에서 표준화되고 있는 유럽의 제 3세대 이동통신 시스템인 UMTS(Universal Mobile Telecommunication System)는 무선 인터페이스를 통해 전달되는 IP 패킷을 인터넷에 보내기 위해서 GPRS 시스템을 정의하고 있다. GPRS(General Packet Radio Service) 시스템은 패킷 스위치 기능을 수행하는 SGSN(Serving GPRS Support Node)과 게이트웨이 기능을 수행하는 GGSN(Gateway GPRS Support Node)을 통하여 인터넷과 접속한다. 즉 GPRS 망에서 GGSN은 라우터의 기능을 가지고 있으며 인터넷과 같은 외부망과 연결된다[1].

Mobile IP(MIP)[2] 망에 가입된 가입자 또는 ISP(Internet Service Provider) 망에 가입된 가입자 간에 인터넷 서비스를 지속적으로 제공하기 위해서는 이동통신 망과 인터넷 망에 단일화된 하나의 라우팅 메커니즘이 적용되어야 한다. 따라서 본 논문에서는 이들 망 간의 패킷을 투명(transparent)하게 전송하기 위한 효율적인 라우팅 방법을 제안한다. 그리고 제안한 방법에 따라 GGSN을 설계하고 구현하며 이들 시스템 간의 패킷 전송률을 살펴본다.

본 논문의 구성은 다음과 같다. 2장에서는 패킷 전송에 대한 설계 및 기능과 구현된 라이브러리에 대해 기술한다. 3장에서는 GGSN과 FA, GGSN과 FA/ISP 간의 패킷 전송률을 살펴보며 마지막으로 4장에서는 결론 및 향후 연구 방향에 대해 기술한다.

2. 설계 및 구현

GGSN과 FA 및 ISP 시스템 간의 패킷 전송을 위한 프로토콜은 효율적인 IP routing을 제공해야 한다. 이를 위해 라우팅은 두 가지 기법이 적용 가능하다. 특정 호스트별로 라우팅을 수행하는 host-specific 라우팅 방법과 시스템 간의 터널링 기법이 있다. [3]에 기술되어 있는 바와 같이 GGSN과 FA 및 ISP 시스템 간의 패킷을 효율적으로 라우팅하기 위해서는 host-specific 라우팅 방법보다 터널링 방법을 이용하는 것이 더 적합하다. 따라서 터널을 형성하는 여러 프로토콜 중에서 fragmentation 계층, 구현의 용이성, 상대적인 성능 측면을 고려해 볼 때 IP-in-IP가 적합하다고 판단된다. 본 논문에서는 패킷 전송을 위해 IP-in-IP/IP/Ethernet 구조를 적용한다.

2.1 설계

패킷 송·수신을 위해 중요한 기능은 GGSN과 FA 및 ISP 간의 패킷을 인/디캡슐레이션하는 기능이다. 따라서 커널에서 인/디캡슐레이션을 구현하기 위해 유닉스 커널에서 제공하는 스트림 메커니즘[4]을 이용한다. 스트림 메커니즘은 커널 내에 구현된 모듈과 어플리케이션이 통신하고자 할 때 이용하는 방식이다. IP-in-IP 스트림 모듈을 구현한 후, 이 스트림을 커널 내에 있는 IP 계층 위에 삽입하고 인/디캡슐레이션을 수행하도록 한다.

2.2 구성 및 기능

그림 1은 본 논문에서 설계한 패킷 송·수신의 기능 구성도를 나타낸다. 어플리케이션 계층에서는 초기화 기능(IF: Initialization

Function), 시스템 구성 갱신 기능(SCUF: System Configuration Update Function), 패킷 송신 기능(PSF: Packet Send Function), 그리고 패킷 수신 기능(PRF: Packet Receive Function)이 있다. 그리고 커널 계층에서는 FA/ISP IP 어드레스를 관리하는 기능(IPMF: FA/ISP IP address Management Function), IP-in-IP 인캡슐레이션(IPIPE: IP-in-IP Encapsulation), 그리고 IP-in-IP 디캡슐레이션(IPIPD: IP-in-IP Decapsulation) 기능이 있다.

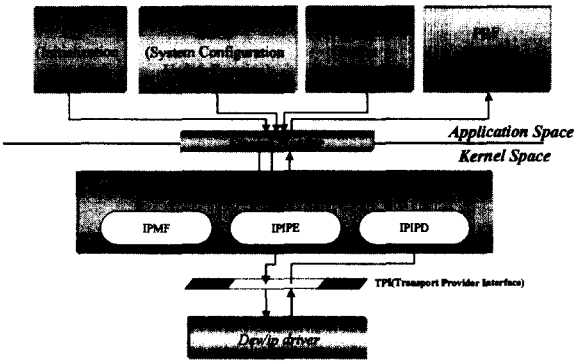


그림 1. 패킷 송·수신 기능 구성도

■ 초기화 기능(IF)

IP-in-IP 인/디캡슐레이션 기능 수행을 위해 IP-in-IP 스트림 모듈을 /dev/ip 드라이버 상에서 동작하도록 IP-in-IP 스트림 모듈에 대한 초기화 작업을 수행한다.

■ 시스템 구성 갱신 기능(SCUF)

FA 또는 ISP의 IP Address가 변경될 때마다 SCUF는 변경된 FA 또는 ISP의 IP Address를 커널에 전송한다.

■ 패킷 송신 기능(PSF)

GGSN은 이동 노드(Mobile Node)로부터 받은 패킷을 FA 또는 ISP에게 송신하기 위해 이 패킷을 커널의 인캡슐레이션으로 전송한다.

■ 패킷 수신 기능(PRF)

MIP/ISP 망에 가입되어 있는 이동 노드에게 패킷을 전송하기 위해 상대 노드로부터 패킷을 수신한 경우, PRF는 커널 계층에 있는 디캡슐레이션 기능으로부터 패킷을 받는다.

■ FA 및 ISP의 IP 어드레스 관리 기능(IPMF)

변경된 FA 또는 ISP의 IP 어드레스를 어플리케이션으로부터 받아 커널에 있는 테이블을 갱신하고 관리한다. 데이터 패킷을 인캡슐레이션할 때, 터널 입구(entry point)와 출구(exit point)를 결정하기 위해 테이블이 사용된다. 그리고 디캡슐레이션할 때, 이 테이블에 저장되어 있는 FA와 ISP 외의 다른 IP 어드레스의 패킷일 경우 디캡슐레이션을 하지 않기 위해 테이블이 사용된다.

■ IP-in-IP 인캡슐레이션(IPIPE)

IP-in-IP 인캡슐레이션은 이동 노드로부터 수신한 IP 패킷을 인캡슐레이션시켜 즉, 외부(outer) 헤더를 부착시켜 IP 계층으로 전달하는 기능을 수행한다. 터널 입구(entry point)는 GGSN이 되며 출구(exit point)는 FA 또는 ISP가 된다. 인터넷 망 상에 ISP가 여러 개 존재하기 때문에 터널 출구를 지정하기 위해 ISP ID(Identifier)가 필요하다. 외부 헤더의 포멧은 RFC2003을 따른다. 외부 헤더의 TTL은 디폴트로 정의한 값(IPIP_DEF_TTL = 32)을 사용한다. 만약 TTL=0이면 IP 패킷을 인캡슐레이션 하지 않고 제거(discard) 한다.

■ IP-in-IP 디캡슐레이션(IPIPD)

IP-in-IP 디캡슐레이션은 FA 또는 ISP로부터 수신한 IP 패킷을 디캡슐레이션시켜 즉, 외부 헤더를 제거하는 기능을 수행한다. 인캡슐레이션과 동일하게, TTL=0이면 IP 패킷을 디캡슐레이션 하지 않고 제거한다. 어플리케이션에 전송할 때, 디캡슐레이션 되어진 패킷이 FA 인지 ISP 인지를 구분할 수 있는 플래그를 지정하며, 만약 ISP인 경우 ISP ID도 함께 전송한다.

2.3 구현된 라이브러리

■ 초기화 라이브러리

Init_GGSN_FA_lib()와 Init_GGSN_ISP_lib() 라이브러리에서의 주요 소스 코드는 다음과 같다. 이 두 라이브러리 중 먼저 호출된 라이브러리에서 ipip 스트림 모듈을 커널에 삽입하고 후에 호출된 라이브러리는 삽입 성공에 대한 리턴 값만 반환한다.

```
ipip_fd = open("/dev/ip", O_RDWR) //ip 디바이스 open
ioctl(ipip_fd, I_PUSH, "ipip") // ipip 스트림 모듈을 ip 디바이스 상에 push
return(Init_OK) // 반환값 리턴
```

■ 시스템 구성 갱신 라이브러리

SCUF_FA_lib(T_FAADDRTBL *PtrFAADDRTBL)와 SCUF_ISP_lib(T_ISPADDRTBL *PtrISPADDRTBL) 라이브러가 있다. 메시지 타입인 T_FAADDRTBL와 T_ISPADDRTBL는 ID와 IP 어드레스 구조로 되어 있으며 현재 FA 시스템은 1개, ISP 시스템은 32개로 정의가 되어 있다. 메시지 구조는 다음과 같다.

```
typedef struct {
    u32 fa_id;
    u32 fa_ip_addr;
} T_FATBL;

typedef struct {
    T_FATBL fa_addr_tbl[1];
} T_FAADDRTBL;
```

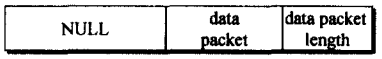
```
typedef struct {
    u32 isp_id;
    u32 isp_ip_addr;
} T_ISPTBL;

typedef struct {
    T_ISPTBL isp_addr_tbl[32];
} T_ISPADDRTBL;
```

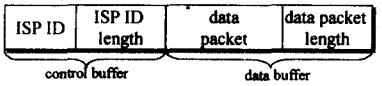
■ 패킷 송신라이브러리

FA_Data_Encap_lib(int SndLength, unsigned char *SndPkt)와 ISP_Data_Encap_lib(int SndID, int Sndlength, unsigned char *SndPkt) 라이브러리가 있다. SndLength는 FA 또는 ISP 시스템에게 전송할 패킷 길이, SndPkt는 이들 시스템에 전송할 데이터 패킷의 포인터이다. 그리고 SndID는 데이터 패킷을 받을 ISP 시스템의 식별자이다. FA_Data_Encap_lib에서는 putmsg(ipip_fd, NULL, &dat, 0) 시스템 콜을 사용한다. GGSN과 FA인 경우 FA가 하나만 존재하기 때문에 control buffer는 null이며 data buffer는 이동 노드로부터 받은 패킷이 저장된다. ISP_Data_Encap_lib에서는 putmsg(ipip_fd, &ctl, &dat, 0) 시스템 콜을 사용한다. ISP는 여러 개가 존재하기 때문에 ISP ID를 control buffer에 저장한다.

FA의 putmsg() 시스템 콜의 메시지 구조



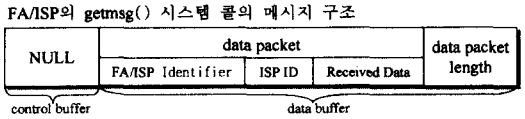
ISP의 putmsg() 시스템 콜의 메시지 구조



■ 패킷 수신 라이브러리

FA_Data_Decap_lib(int RcvLength, unsigned char *RcvPkt)와 ISP_Data_Decap_lib(int RcvID, int RcvLength, unsigned char *RcvPkt) 라이브러리가 있다. RcvLength는 FA 또는 ISP 시스템으로부터 받은 패킷 길이, RcvPkt는 이들 시스템으로부터 받은 데이터 패킷이다. 그리고 RcvID는 데이터 패킷을 보낸 ISP 시스템의 식별자이다. 이들 라이브러리는 하나의 getmsg(ipip_fd, NULL, &dat, &flagsp) 시스템 콜을 사용하여 FA와 ISP 구분 식별자를 이용하여 커널로부터 받은 메시지를 처리한다. 그리고 flagsp를 0으로 설정하여 메시지 큐에 있는 메시지의 우선 순위에 상관없이 메시지를 처리한다.

FA/ISP의 getmsg() 시스템 콜의 메시지 구조



2.4 패킷 송·수신 절차

GGSN이 FA 및 ISP에게 패킷을 송신하기 위해 인캡슐레이션 하는 처리 절차는 그림 2의 하향식 방향과 같다. 그리고 FA 및 ISP로부터 패킷을 수신하는 경우는 그림 2의 상향식 방향과 같다.

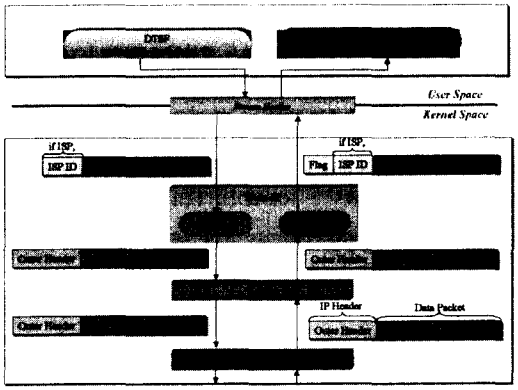


그림 2. 데이터 트래픽 송·수신 절차와 메시지 구조

3. 시뮬레이션

본 논문에서 GGSN과 FA/ISP 시스템 간의 패킷 전송률을 구하기 위해 그림 3과 같은 환경에서 시뮬레이션을 하였다. GGSN, FA, 그리고 두 개의 ISP는 서로 다른 서브넷에 각각 존재하며 SPARCengine(tm) Ultra CP 1500 (UltraSPARC-IIi 360MHz)과 Fast Ethernet (10/100Mbps) 상에서 동작한다.

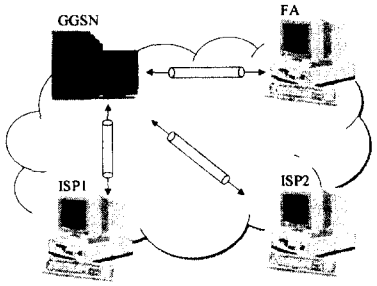


그림 3. 시뮬레이션 환경

먼저 GGSN과 FA 간의 패킷 전송률을 구하였다. 그림 4는 300, 600, 900, 그리고 1200 바이트를 전송하고 패킷 전송 횟수는 1000~5000까지 500씩 증가하면서 측정한 패킷 전송 시간을 나타낸다. 패킷 크기가 600 바이트이고 전송 횟수가 3000회 일 때, 평균 패킷 전송률은 2.5Mbps이다.

두 번째로 시뮬레이션한 환경은 GGSN과 FA, ISP1, ISP2 시스템을 동시에 연결한 상태에서 패킷 전송률을 구하였다. 전송되는 패킷 크기는 600 바이트이며, 패킷을 전송하는 횟수는 10~100까지 10회씩 증가하여 전송한다. 그림 5는 시뮬레이션 결과를 나타낸다. 전송 횟수가 50회 일 때, GGSN과 FA 간의 평균 패킷 전송률은 약 150Kbps이다. 그리고 GGSN과 ISP1, ISP2 간의 평균 패킷 전송률은 각각 49Kbps, 43Kbps이다.

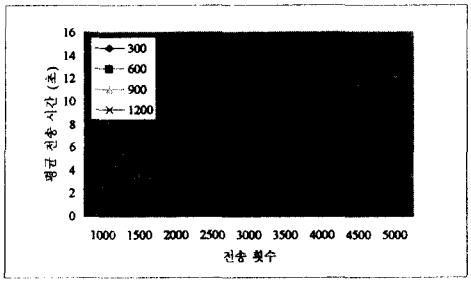


그림 4. GGSN과 FA 간의 평균 전송 시간 및 전송 횟수

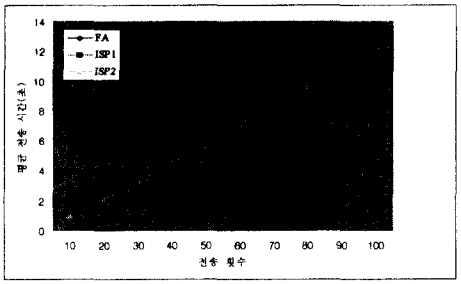


그림 5. GGSN과 FA/ISP1, ISP2 간의 평균 전송 시간 및 전송 횟수

하나의 GGSN 시스템에서 FA를 처리하는 경우와 FA/ISP를 모두 처리하는 경우, 패킷 전송률이 크게 차이가 남을 알 수 있다. GGSN 시스템이 다수인 경우, FA와의 데이터 트래픽을 위한 GGSN 시스템, 그리고 ISP와의 데이터 트래픽을 위한 GGSN 시스템을 따로 두어 성능을 높일 수 있지만 구현과 관리 측면을 고려해야 할 것이다.

4. 결론 및 향후 과제

본 논문에서는 MIP 망에 가입된 가입자 또는 ISP 망에 가입된 가입자 간에 인터넷 서비스를 지속적으로 제공하기 위해 이동통신 망과 인터넷 망간의 패킷을 효율적으로 라우팅할 수 있는 IP-in-IP 방법을 제안하였다. 그리고 GGSN과 FA/ISP 간의 패킷 전송을 위한 기능들과 처리 절차 및 메시지 구조를 설계하고 구현하였으며 이들 각각에 대해 패킷 전송률을 측정하였다.

현재 GPRS 망에서는 패킷 전송이 IPv4에서 동작하고 있다. 그러나 인터넷 망의 증가에 따라 IPv6에 대한 연구가 활발히 진행되고 있다. 따라서 GPRS 망에서도 IPv6에 대해 패킷 전송이 가능하도록 이에 대한 연구가 계속 되어져야 할 것이다.

참고 문헌

[1] 3GPP, General Packet Radio Service (GPRS); Service Description; Stage 2, 3G TS 23.060 VERSION 3.3.0, April 2000.

[2] James D. Solomon, Mobile IP: The Internet Unplugged, Prentice Hall, 1998.

[3] SunHwa Lim, et.al, "Design and Implementation of a Gi+ Interface between GGSN and FA for Supporting MIP on UMTS/GPRS," Proc. WOC '2001, pp. 27-32, June 2001.

[4] Sun Microsystems, Inc. "STREAMS Programming Guide," 1998.

[5] W.Richard Stevens, "UNIX Network Programming," Volume I, 1998.