제 19 회 한국정보처리학회 춘계학술발표대회 논문집 제 10 권 제 1 호 (2003. 5)

리눅스 커널에서 VoIP 트래픽 제어에 관한 연구

강경철*, 임성욱**, 류연승**
*한림대학교 정보통신공학부
**명지대학교 컴퓨터소프트웨어학과
e-mail : ysryu@mju.ac.kr

A Study on Control of VoIP Traffic in Linux Kernel

Gyong-Cheol Kang*, Seong-Uk Lim**, Yeon-Seung Ryu**
*Div. of Information and Comm. Engineering, Hallym University
**Dept. of Computer Science, Myongji University

요 약

본 논문에서는 리눅스 커널에서 VoIP(Voice over IP) 트래픽을 탐지하고 제어하는 네트워크 모듈을 설계하고 구현하는 연구를 소개한다. 구현하는 리눅스 네트워크 모듈은 실시간으로 VoIP 세션 연결 요청 패킷을 탐지하고 정해진 정책(policy)에 따라 서비스 품질(QoS)을 제어할 수 있다.

1. 서론

컴퓨터와 네트워크 기술의 발전으로 인터넷으로 음성 및 동영상을 전송하는 멀티미디어 통신 응용분야가 확대되고 있다. 특히, VoIP(Voice over IP) 서비스는 인터넷을 이용한 전화 서비스로서 비용 절감 및 다양한 응용과의 통합이 가능하므로 많은 장점을 가진다. 보안 및 통화품질 등 문제가 남아있으나 해결방안이 개발되고 있어 VoIP 서비스는 점차 확산될 것으로 예측되고 있다.

VoIP 를 위한 국제 표준 통신 프로토콜로는 90 년대 중반부터 제안되어 왔던 ITU-T H.323[1]이나 IETF 의 SIP (Session Initiation Protocol)[2,3]과 MGCP(Media Gateway Control Protocol)[5], MEGACO[4] 규격들이 있다. 최근에는 마이크로소프트 사의 메신저 및 PDA, 3GPP 등의 단말기에서는 SIP 를 적용하는 추세에 있다. 또한, 차세대 이동 통신과 무선 랜에서도 상호간에 SIP 를 이용한 인터넷 전화가 가능하리라 예측된다.

*본 연구는 산업자원부 2002 년도 공통핵심기술개발사업의 위탁과제로서 인터콘웨어㈜의 지원으로 수행되었습니다.

한편, 인터넷 사용자 별로 또는 응용 분야 별로 정책(policy)을 설정하여 사용자 트래픽을 제어하고 서비스 품질을 보장하는 QoS(Quality of Service) 장비들이 최근 등장하고 있다[16,17]. QoS 장비는 기업체, 학교 등에서 사용이 증가하고 있는데, 보통 내부 네트워크와 외부 네트워크 사이에 배치되고, 네트워크의 트래픽을 감시하고 제어하기 위해 사용되고 있다. QoS 장비의 기능으로는 트래픽의 세션(또는 flow)을 탐지하고 세션 별로 네트워크 대역폭을 보장하는 기능, DOS(Denial of Service) 공격 등 네트워크 침입을 탐지하는 기능, 트래픽 부하를 탐지하여 부하를 균등 분배하는 기능 등이 있다.

본 논문에서는 리눅스 커널의 네트워크 모듈을 분석하고 커널 수준에서 VoIP 트래픽을 제어하는 기능을 설계하고 구현하였다. VoIP 프로토콜로는 사용이 확대되고 있는 SIP 프로토콜을 선택하였다. 본 논문의 구성은 다음과 같다. 2 장에서는 리눅스 커널의 IP 네트워킹에 대해서 살펴본다. 3 장에서는 SIP 프로토콜과 특징에 대해서 살펴보고, 4 장에서 트래픽 제어 방안을 살펴본다. 마지막으로 5 장에서 결론 및 향후 연구 방

향에 대해 기술한다.

2. 리눅스 IP 네트워킹

2.1 패킷 처리 개요

리눅스 커널에서 IP 네트워킹 관련 소스 코드는 커널 소스 디렉토리의 /net 디렉토리에 위치하고 있다. 여기서, core 디렉토리는 프로토콜과는 독립적인 소스가 있고, ipv4 디렉토리는 IP 버전 4 관련 소스, ipv6 디렉토리에는 IP 버전 6 관련 소스가 있다. 또, sched 디렉토리에는 패킷 스케줄링 관련 소스가 있다.

네트워크 메시지가 IP 계층에서 어떻게 처리되는지에 대해 그림 1 에 나와있다. 커널은 두 경로에서 네트워크 메시지를 수신할 수 있다. 첫째는 응용 프로그램에서 소켓을 사용하여 보낸 것이고 둘째는 네트워크 장치(LAN 카드)를 통해 외부로부터 받은 것이다.

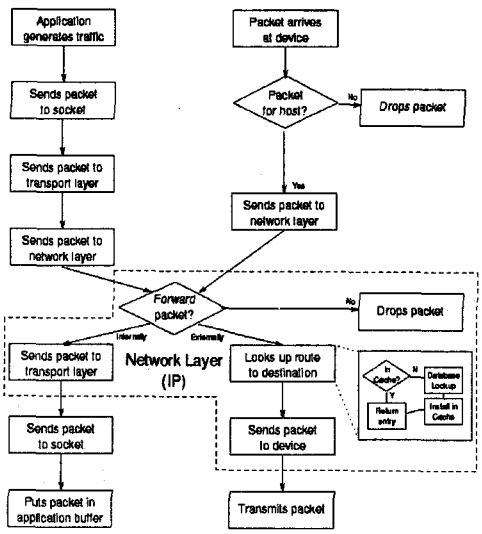

그림 1. 리눅스 커널의 IP 메시지 처리

응용 프로그램의 소켓 인터페이스를 통해 전달된 경우에는 패킷을 IP 계층으로 보낸다. 외부 네트워크에서 수신한 경우에는 목적지가 이 호스트가 맞는 지를 검사한다. 목적지가 맞다면 수신한 패킷을 네트워크 계층인 IP 계층으로 보내고, 아니면 무시한다.

패킷을 받은 IP 계층에서는 패킷의 다음 목적지를 결정하고 송신한다. 다음 목적지가 내부, 즉 같은 호스트의 응용 프로그램이 대기하고 있는 소켓 인터페이스인 경우에는 해당 소켓으로 보낸다. 다음 목적지가 외부인 경우에는 라우팅 테이블을 검색하여 목적지를 정하고 패킷을 네트워크 장치 계층으로 전달한다.

본 논문에서는 외부로부터 받은 패킷을 외부로 전달하는 과정에서 VoIP 패킷을 제어할 것이므로 패킷의 송수신과정을 좀 더 자세히 살펴본다.

(1) 네트워크 장치 인터럽트 루틴: 패킷의 수신
- 네트워크 장치에 패킷이 도착하면 인터럽트에 의해 실행된다. (수행되던 프로세스는 중지)
- 네트워크 장치는 링크의 헤더를 수신한다.
- 패킷을 저장할 버퍼 공간을 만든다.
- 패킷을 버퍼 공간에 저장한다.
- 패킷을 백로그(backlog) 큐에 연결한다.
- 디바이스 드라이버의 bottom half 가 수행될 수 있도록 플래그를 설정한다.
- 수행되던 프로세스에 복귀한다.

(2) 디바이스 드라이버 : Bottom Half 루틴
- 스케줄러에 의해 디바이스 드라이버의 bottom half 루틴이 실행된다.
- 백로그 큐에 있는 모든 패킷을 IP 계층 루틴이 처리할 수 있도록 전달한다.
- 송신 큐의 패킷을 처리한다.

(3) IP 계층: 패킷의 검사 및 목적지 결정
- 패킷의 무결성을 검사한다.(길이, 버전, 체크썸 등)
- 절단된 패킷의 결합이 필요하다면 결합한다.
- 패킷의 목적지를 결정한다.
- TTL 필드의 값을 검사하고, 감소시킨다.
- 패킷의 부적절한 라우팅을 검사한다.
- 만약 문제가 있다면 패킷의 송신측에 ICMP 패킷을 보낸다.
- 패킷을 새 버퍼에 복사하고 이전 버퍼는 제거한다.
- 필요한 IP 옵션을 설정한다.
- 패킷의 길이가 크다면 절단한다.
- 목적지 경로 상의 네트워크 장치의 출력 큐에 패킷을 연결한다.

(4) 디바이스 드라이버의 루틴
- 링크 헤더를 송신한다.
- 패킷을 송신한다.

2.2 패킷 구조체

트래픽을 제어하기 위해서는 리눅스 커널의 패킷 구조체를 알아야 한다. 그림 2 는 패킷의 관리를 위한 sk_buff 구조체를 보여주고 있다. 이 구조체는 한 패킷의 각종 정보를 참조하는 포인터들로 구성되어 있다.

이런 포인터를 사용하는 방법을 통해 패킷 내 사용자 데이터(payload)의 복사 작업을 최소화하고 있다. 사용자 데이터의 복사 작업은 두 번 수행된다. 송신하는 경우를 예를 들면, 사용자 프로세스 공간에서 커널 공간으로 한번, 커널 공간에서 네트워크 매체로 한번씩 두 번 복사된다. 네트워크의 각 계층인 데이터 링크, 네트워크, 전송(transport) 계층은 한 패킷에 대해 이 구조체를 공유하면서 필요한 작업을 수행한다.

sk	pointer to owning socket
stamp	arrival time
dev	pointer to receiving/transmitting device
h	pointer to transport layer header
nh	pointer to network layer header
mac	pointer to link layer header
dst	pointer to dst_entry
cb	TCP per-packet control information
len	actual data length
csum	checksum
protocol	packet network protocol
truesize	buffer size
head	pointer to head of buffer
data	pointer to data head
tail	pointer to tail
end	pointer to end
destructor	pointer to destruct function

그림 2 리눅스 패킷 구조체 (sk_buff)

3. VoIP 프로토콜

SIP 는 인터넷에서 멀티미디어 세션(session)을 개시하고 세션 안에서 음성, 영상, 메시지 등의 전송을 위해 IETF 에서 정의하고 있는 표준 프로토콜이다. SIP 는 TCP/UDP 에 정의되어 있어 응용 계층 프로토콜이며, 사용하는 문법(syntax)은 HTTP 1.1 에서 유래되었고 텍스트 기반으로 되어 있다. 그림 3 은 SIP 프로토콜 스택을 보여주고 있다.

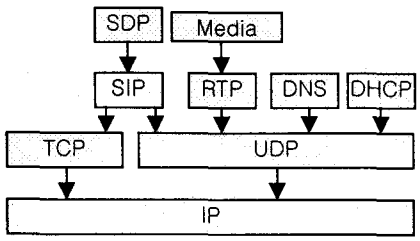

그림 3. SIP 프로토콜 스택

SIP 는 세션 설정 시에 SDP(Session Description Protocol)을 사용하고, 오디오 및 비디오 데이터의 전송 시에 RTP(Real-Time Transport Protocol)를 사용한다.

3.1 구성요소

SIP 시스템은 사용자 에이전트(UA: User Agent)와 서버로 구성된 클라이언트 서버 시스템이다. 사용자 에이전트는 SIP 세션 요청을 개시하기도 하고 세션 요청을 받기도 한다. 각각의 경우에 UAC(User Agent Client)와 UAS(User Agent Server)라고 부른다. 서버에는 프록시 서버와 리다이렉트 서버가 있다. 프록시 서버는 SIP 메시지를 다음 서버로 전달한다. 메시지가 여러 서버를 경유하여 UA 로 전달되면 그에 대한 응답 메시지는 역 순으로 서버를 경유하여 전달된다. 리다이렉트 서버는 수신한 메시지를 다음 서버로 전달하지 않으며 대신 메시지 송신 측에 다음 서버의 주소를 알려준다. 이 외에 등록(registrar) 서버가 있을 수 있다. 등록 서버는 UA 로부터 등록 요청 메시지를 받고 등록 정보를 관리하는 서버로서 사용자의 위치 서비스를 제공할 수 있다.

3.2 메시지

SIP 메시지는 크게 요청(request) 메시지와 응답(response) 메시지로 되어있다. 모든 메시지는 헤더를 가진다. 헤더에는 caller, callee, 메시지의 경로, 유형, 메시지 몸체의 길이 등의 정보를 담고있다.

(1) 요청 메시지

- INVITE : 세션의 참여를 요청함
- ACK : INVITE 에 대한 최종 응답 메시지
- BYE : 세션을 종료함
- CANCEL : 세션을 취소함
- OPTION : 상대방의 부가 능력을 알아봄
- REGISTER : 사용자의 위치를 등록함

(2) 응답 메시지

응답 메시지는 헤더에 응답 코드를 보낸다.

- 1XX : 요청메시지를 수신하고 계속 처리 중임
- 2XX : 성공
- 3XX : redirection
- 4XX : 클라이언트 오류
- 5XX : 서버의 오류
- 6XX : 일반적인 오류

그림 4 는 사용자 A 와 사용자-B 가 등록 서버에 등

록되어 있는 상태를 가정으로 상호간 세션 설정을 하는 절차를 보이고 있다.

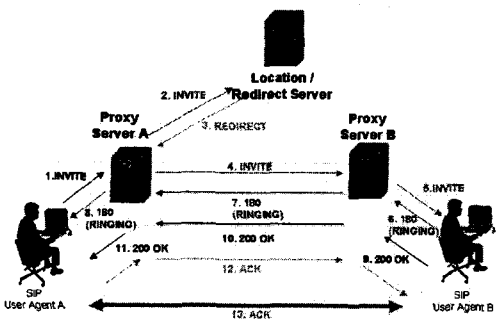

그림 4. SIP 세션 연결 절차

4. 트래픽 제어 방법

SIP 는 요청 메시지를 TCP 또는 UDP 를 사용하여 보낼 때 상대방의 5060 포트를 사용한다. 따라서, TCP/UDP 포트 5060 의 패킷을 분석하면 SIP 메시지를 캡처할 수 있다. 또한, SIP 는 텍스트 기반의 메시지를 사용하고 있으므로 SIP 메시지를 파싱(parsing)하면 VoIP 트래픽 제어를 위해 필요한 정보를 쉽게 구할 수 있다. 특히, 세션 설정을 위해 필요한 정보는 INVITE 와 ACK 메시지에 대부분 포함되어 있다. 그림 5 는 INVITE 메시지의 예를 보여주고 있다.

```
INVITE sip:userB@hostB.com SIP/2.0
Via: SIP/2.0/TCP hostA.com:2054
CSeq: 1 INVITE
Contact: sip:userA@hostA.com:5060
Expires: 3600
From: sip:userA@hostA.com
To: sip:userB@hostB.com
Call-ID: 460414147@hostA.com
Content-Type: application/sdp
Content-Length: 206

v=0
o=userA 103141879711 1006526069 IN IP4 hostA.com
s=Untitled
c=IN IP4 hostA.com
t=0
m=audio 10000 RTP/AVP 0
m=video 20000 RTP/AVP 0
m=wb 30000 RTP wb
m=text 40000 UDP chat
```

그림 5. INVITE 메시지의 예

SIP 메시지는 빈 줄로 구분되어 윗부분은 헤더이고 아래부분은 몸체이다. 몸체는 SDP 를 사용한다. 이 메시지에서 헤더를 분석하면 userA@hostA.com 가 userB@hostB.com 에게 세션 참가를 요청한 것을 알 수 있으며 Via 헤더를 통해 이 메시지에 대한 응답을 TCP 포트 2054 에서 받기를 원하고 있음을 알 수 있다. 또한, 몸체의 SDP 내용을 분석하면 오디오는 RTP 포트 10000 을 사용할 것이며, 비디오는 RTP 포트 20000 을 사용할 것임을 알 수 있다. 또한, 백색칠판(white board)은 RTP 포트 30000 을, 문자 채팅은 UDP 포트 40000 을 사용할 것임을 알 수 있다.

본 논문에서는 리눅스 커널의 패킷 송수신 모듈에서 SIP 트래픽 제어에 대한 방안을 설계하였다. 설계한 모듈은 SIP 포트를 통해 SIP 메시지를 가로채고 트래픽 제어를 위한 정보를 수집한 후, 정해진 정책에 따라 트래픽을 제어하게 된다.

5. 결론

본 논문에서는 리눅스 커널의 네트워크 모듈을 분석하고 커널 수준에서 VoIP 트래픽을 제어하는 방법을 연구하였다. 본 연구는 리눅스 라우터 기능을 탑재한 QoS 장비에서 VoIP 트래픽의 실시간 제어 기능으로 개발될 예정이다.

참고문헌

[1] ITU-T Recommandation H.323 Version 4, "Packet Based Multimedia Communications System", Nov. 2000.
[2] Hendley, M., H. Shulzrinne, E. Schooler and J. Rosenberg, "SIP:Session Initiation Protocol", IETF RFC 3261, 2002.
[3] http://www.cs.columbia.edu/sip/
[4] ITU-T Recommandation H.248 Version 1,Jun. 2000.
[5] IETF RFC 2705, "Media Gateway Control Protocol (MGCP)," Oct. 1999.
[9] http://www.ethereal.org
[10] http://www.vovida.org/vocal
[11] Vern Paxson, "Automated Packet Trace Analysis of TCP Implementation", SIGCOMM, pp. 167-179, 1997
[12] Vern Paxson, "Bro: A System for Detecting Network Intruders in Real-time", Computer Networks, Vol. 31, No.23-24, 1999.
[13] Marcus Ranum, et al, "Implementing A Generalized Tool For Network Monitoring", Proceedings of the Eleventh Systems Administration Conference (LISA '97), 1997
[14] R. Caceres, et al, "mmdump - A Tool for Monitoring Multimedia Usage on the Internet", ACM Computer Communication Review, 30(4), Oct. 2000
[15] IETF RFC 2327, "Session Description Protocol (SDP)," April. 1998.
[16] http://www.allot.com
[17] http://www.packeteer.com