

차별화된 호 처리 기법을 지원하는 SIP6

(SIP6 supporting the Differentiated Call Processing Scheme)

김진철[†] 최병욱^{**} 장천현^{***} 김기천^{***} 한선영^{***}
 (Chinchul-Kim) (Byounguk-Choi) (Chunhyun-Chang) (Keechoen-Kim) (Sunyoung-Han)

요약 본 논문에서는 NGN(Next Generation Network)을 고려하여 IPv6와 차별화된 호 처리 기법을 지원하는 SIP(Session Initiation Protocol) 프로토콜을 설계하고 구현하였다. NGN에서 다양한 응용 서비스들은 SIP를 통하여 호를 설정한다. 따라서, 소프트스위치(Softswitch) 및 SIP 서버는 호를 처리 할 때 팩스(Fax), 망 관리(Network Management), 홈 네트워킹(Home Networking)과 같이 호 설정 시간에 아주 민감한 서비스에 우선순위를 부여할 수 있어야 한다. 또한, ALL-IP를 고려하여 IPv6를 지원하여야 한다. 이러한 문제들을 해결하기 위해, 본 논문에서는 차별화된 호 처리 기법을 제안하였다. 차별화된 호 처리 기법은 우선순위에 따라 세 단계의 서비스품질을 사용하여, 호 처리에 대한 차별화 정책을 지원한다. 또한, 본 논문에서 정의된 서비스품질은 IPv6 헤더의 Flow Label 필드를 통하여 값을 설정한다. 성능분석 결과, SIP 서버 내에서 호가 처리 될 때 높은 우선순위를 가지는 호에 대한 처리율이 현저히 빠르게 나타났다. 이러한 결과는 NGN에서 호 설정 시간에 민감한 응용 서비스에 적용 할 경우, 호 설정 시간을 줄임으로써 응용 서비스의 성능을 향상 시킬 수 있음을 증명한다.

키워드 : VoIP, SIP, IPv6, QoS

Abstract In this paper, we implemented SIP protocol that supports IPv6 and differentiated call processing scheme for NGN(Next Generation Network). In NGN, SIP processes call signaling among various application services. A softswitch and SIP server must give priority to sensitive services such as Fax, network management and home networking that require a fast call setup time. Also, the support of IPv6 is needed under consideration of All-IP. We proposed differentiated call processing scheme. The differentiated call processing scheme supports differentiated call processing as priority of service class on call processing in SIP server. We defined three service classes and use the Flow Label field of the IPv6 header for setting service class. Through the performance analysis, we proved that it improves throughput for call message with the high priority. The result of performance analysis demonstrates that differentiated call processing scheme gives better performance for the service requiring a fast session establishment in NGN.

Key words : VoIP, SIP, IPv6, QoS

1. 서론

망의 고속화와 인터넷 사용자들의 빠른 증가로 인하여 현재의 네트워크에 대한 사용자들의 요구사항은 다

양해지고 있다. 특히, 서비스 사용자와 ISP(Internet Service Provider)들의 입장을 고려해 볼 때 망 및 응용 서비스의 통합 환경은 현 시점에서 가장 큰 이슈로 대두되고 있다. 이러한 요구사항들을 충족시키기 위해, NGN(Next Generation Network), 3GPP(3rd Generation Partnership Project), 3GPP2 등의 표준화 그룹에서는 VoIP기술을 중심으로 망 및 응용 서비스의 통합을 위한 표준화가 활발히 진행 중이다. 이러한 표준 기술 중 가장 핵심적인 프로토콜이 바로 SIP(Session Initiation Protocol)와 IPv6 프로토콜이다.

현재 SIP 프로토콜은 IETF의 MMUSIC(Multiparty Multimedia Session Control) 워킹그룹에서 표준화되어, VoIP 및 멀티미디어 서비스 환경에서 세션을 제어

· 이 논문은 2003년도 한국과학재단 연구비[과제번호 R01-2001-000-00349-0(2003)]지원에 의한 결과임

[†] 학생회원 : 건국대학교 컴퓨터공학부
 cckim@cclab.konkuk.ac.kr

^{**} 비회원 : 건국대학교 컴퓨터공학부
 buchoi@cclab.konkuk.ac.kr

^{***} 종신회원 : 건국대학교 컴퓨터공학부 교수
 chchang@konkuk.ac.kr
 kekim@kkucc.konkuk.ac.kr
 syhan@cclab.konkuk.ac.kr

논문접수 : 2002년 6월 21일

심사완료 : 2003년 6월 23일

하는 프로토콜로 사용되고 있다. 또한, 그 적용이 매우 광범위하여 다양한 분야에서 많은 연구가 진행 중이다 [1-4]. 3GPP와 3GPP2의 경우, SIP는 멀티미디어 서비스를 위한 표준 프로토콜로 채택되었다[5]. NGN의 경우, SIP는 다양한 응용 서비스들의 세션을 제어하며, 세션 요청을 처리하는 소프트스위치(Softswitch)의 핵심 프로토콜로 사용된다. 결국, SIP 프로토콜을 기반으로 하여 멀티미디어 서비스(Multimedia Service), 팩스(Fax), 홈 네트워킹(Home Networking), 인스턴트 메시징(Instant Messaging), 망 관리(Network Management), 웹(World Wide Web) 및 SMTP(Send Mail Transport Protocol) 등의 응용 서비스들이 통합되는 구조를 보이고 있다[6].

이렇게 SIP를 통하여 다양한 응용 서비스들이 통합될 경우, SIP 서버와 소프트스위치는 응용 서비스가 요구하는 시그널링을 처리하는 역할을 담당한다. 특히, 소프트스위치와 SIP 서버는 호를 처리할 때 호 설정 시간에 따라 서비스의 질이 결정되는 팩스, 망 관리 및 홈 네트워킹 등의 응용 서비스에 대하여 호 설정 시간에 민감하지 않은 응용 서비스 보다 높은 우선순위를 보장할 수 있어야 한다. 하지만, 현재의 SIP 프로토콜은 우선순위를 기반으로 하는 호 처리 기법을 지원하지 못하는 실정이다.

호 설정 시간에 민감한 응용 서비스들은 네트워크상에서의 호 설정 지연(Call Setup Delay)과 소프트스위치 및 SIP 서버 내에서의 호 설정 지연을 모두 보장해야 한다. 일반적으로 하나의 호는 네트워크상에서 여러 홉(Hop)의 라우터와 소수의 SIP 서버를 경유하여 설정된다[4]. 소프트스위치나 SIP 서버는 호를 처리 할 때, 사용자 위치 추적, 인증, 도메인 네임 서비스 및 기타 정보 관리 등 다양한 처리 과정을 수행한다. 따라서, 라우터에서 호가 처리되는 시간보다 수십 배의 시간을 요구한다. 결과적으로, NGN에서 빠른 호 설정을 요구하는 응용 서비스들의 서비스의 질을 보장하기 위해서는 라우터에서의 호 설정 지연 못지않게 서버 내에서의 호 설정 지연 역시 중요하다.

현재 라우터에서의 호 설정 지연을 보장하기 위해 IETF(Internet Engineering Task Force)의 IntServ(Integrated Service), RSVP(Resource Reservation Protocol)[7], DiffServ(Differentiated Services)[8][9], YESSIR[10] 등의 프로토콜이 사용된다. 하지만, 소프트스위치 및 SIP 서버 내에서 호 설정 지연을 보장하기 위한 표준 기술은 부재한 실정이다.

이러한 문제를 해결하기 위해, 본 논문에서는 NGN을 고려하여 IPv6와 차별화된 호 처리 기법을 지원하는 SIP 프로토콜을 설계하고 구현한다. 차별화된 호 처리

기법은 SIP 서버 내에서 우선순위에 따라 세 단계의 서비스 품질을 사용하여, 호 처리에 대한 차별화 정책을 지원한다. 또한, 본 논문에서 정의된 서비스 품질은 IPv6 헤더의 Flow Label 필드를 통하여 값을 설정한다. 성능 분석을 통하여, SIP 서버 내에서 호가 처리 될 때 높은 우선순위를 가지는 호에 대한 처리율이 낮은 우선순위를 가지는 호에 대한 처리율에 비하여 상대적으로 빠름을 증명한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구에 대해 살펴보고, 3장은 시스템 설계에 필요한 고려 사항을 언급하고, 4장에서는 시스템 설계에 대해 기술하며 5장에서는 시스템 구현 결과에 대해 기술하였다. 6장에서는 시뮬레이션을 통해 다른 SIP 서버와의 성능을 비교하였으며, 끝으로 7장에서는 결론 및 향후 연구 과제에 대해 제시하였다.

2. 관련 연구

2.1 시그널링 프로토콜 SIP

VoIP를 위한 대표적인 시그널링 프로토콜에는 IETF의 SIP와 ITU-T의 H.323[11]이 있다. H.323은 오랜 연구 기간을 거쳐 안정성은 보장되지만, 바이너리 메시지 포맷을 기반으로 하며 시그널링 기능 이외의 많은 요소들을 포함하고 있어 복잡한 구조를 가진다. 또한, 여러 프로토콜들이 융합 되어 있어 모듈성이 부족하며 확장 시 많은 어려움을 가진다. 하지만, SIP 프로토콜은 프로토콜의 단순성, 확장성, 다른 프로토콜들과의 상호연동성, 프락시/리다이렉트 서버를 통한 사용자의 이동성 지원 등 많은 장점을 가진다[4]. 이러한 장점들을 기반으로 현재 인터넷상에서 SIP를 통해 유무선 통합 환경을 구축하려는 많은 연구가 진행 중이다.

SIP 서버는 호 전달(Call Forwarding) 기능을 수행하는 프락시 서버(Proxy Server)와 호 재 전달(Call Redirection) 기능을 수행하는 리다이렉트 서버(Redirect Server), 사용자에게 대한 위치 서비스를 제공하는 로케이션 서버(Location Server), 사용자 등록을 위한 레지스트라 서버(Registrar Server)로 구성된다. SIP 프로토콜의 메시지 포맷은 텍스트 기반이며, 구조는 그림 1과 같다[2].

현재 SIP 프로토콜은 차세대 통신기술로 관심이 집중되고 있으나, 실제 인터넷상에서 적용되기 위해서는 여러 가지 문제들이 선결되어야 한다. 특히, 상업적 특성이 강한 VoIP나 응용 서비스들이 통합되는 ALL-IP 환경에서 호 및 메시지 처리에 있어 QoS[12]의 적용은 필수적이다. 하지만, 현재 SIP 프로토콜은 자원 예약 및 차별화된 호 처리 등을 위한 QoS 기능을 지원하지 못하는 실정이다.

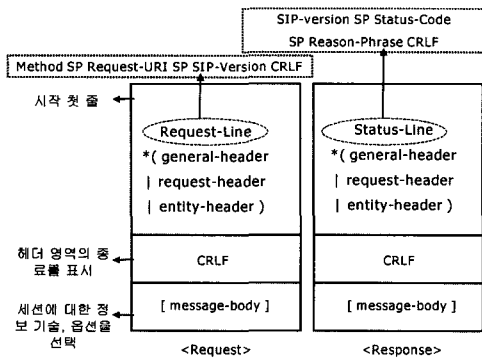


그림 1 SIP 프로토콜 메시지 형식

```

struct sockaddr_in6 {
    uint8_t    sin6_len;
    sa_family_t sin6_family;
    in_port_t  sin6_port;
    uint32_t   sin6_flowinfo;
    struct     in6_addr sin6_addr;
}
    
```

그림 2 sockaddr_in6 구조체

2.2 IPv6 QoS 구조

IPv6(Internet Protocol version 6)는 기존 IPv4의 복잡성에 따른 확장성 결여 및 무분별한 주소 할당에 따른 라우팅 구조의 비합리성을 극복하고, 현재 네트워크의 진화에 따른 주소 부족의 문제를 원천적으로 해결할 수 있는 유일한 대안으로 여겨지고 있다.

또한, IPv6는 헤더 내에 QoS을 위한 Flow Label 필드를 포함하고 있다. 현재 Flow Label 필드는 총 24비트 길이로 미래의 사용을 위해 정의되어 있다[13]. 본 논문에서는 이 필드를 이용하여 차별화된 호 처리 기법을 구현하였다.

Flow Label 필드는 IPv6 소켓 구조체 sockaddr_in6 안에 정의되어 있는 sin6_flowinfo 변수와 매핑되고, 실제 구현에 있어 QoS 파라미터는 이 변수를 통해 표시된다. 그림 2는 sockaddr_in6 구조체를 보여준다[14].

3. 시스템 설계 시 고려 사항

본 논문에서 제시하는 차별화된 호 처리 기법은 NGN의 소프트웨어 스위치 및 SIP 서버 내에서 빠른 호 설정을 요구하는 응용 서비스들의 호 설정 지연을 줄이기 위한 QoS 기능을 지칭한다.

현재 네트워크상에서 서비스가 요구하는 호 설정 지연을 줄이기 위해, RSVP와 DiffServ가 적용되고 있다. RSVP와 DiffServ는 응용프로그램이 요구하는 대역폭(Bandwidth)이나 지연(Delay) 등을 보장할 수 있는 전

송 계층의 프로토콜로 라우터에서 구현된다. 하지만, SIP 서버는 라우터가 아닌 호스트 환경에서 실행되는 응용 계층의 어플리케이션 서버로서 특정 호에 대한 호 설정 지연을 보장할 수 있는 기능을 지원하지 못한다. 이러한 기능을 지원하기 위해 두 가지 접근 방식을 고려할 수 있다. 우선, 전송 계층의 RSVP, DiffServ 프로토콜을 SIP 서버에 적용하는 방식으로 이러한 방식은 다음과 같은 문제를 가진다.

첫째, SIP 서버가 실행 중인 호스트의 전송 계층에서 모든 패킷이 제어됨으로 많은 오버헤드가 발생한다. 즉, SIP 서버는 SIP 메시지에 제한적인 차별화된 호 처리를 요구한다. 둘째, 네트워크상에 존재하는 모든 라우터의 도움을 받아야 하므로 서비스의 확장성 및 유연성을 가지지 못한다. 셋째, 자원 예약 방법 및 시스템 관리 측면에서 복잡성을 가진다. 즉, 응용 프로그램을 개발할 때 RSVP나 Bandwidth Broker 기능을 별도로 또는 통합적으로 구현하여야 하며, 하나의 호스트에서 SIP 서버와 RSVP, DiffServ 등 여러 개의 프로토콜들을 관리하여야 한다.

이러한 문제들을 해결하기 위해, 본 논문에서는 두 번째 방식으로 응용 계층의 어플리케이션 큐잉(Application Queuing) 기법을 사용하여 SIP 서버 내에서 특정 호에 대한 호 설정 지연을 줄일 수 있는 차별화된 호 처리 기법을 제안한다. 이 기법은 프로토콜의 단순성, 확장성 및 관리상의 편리성 등을 지원할 수 있다. 또한, 응용 계층에서 차별화된 호 처리 모듈을 SIP 프로토콜 모듈 아래 삽입함으로써 프로토콜의 독립성을 지원할 수 있다. 결과적으로, 이 기법은 라우터의 도움 없이 응용 계층에서 서비스가 이루어지므로 서비스의 빠른 전개를 이끌어 낼 수 있다는 장점 또한 가진다.

4. 시스템 설계

본 논문에서는 3절에서 소개된 시스템 설계상의 고려 사항을 반영하여 IPv6를 지원하고 RFC 2543을 따르는 SIP 서버를 설계하였다. 또한, 응용 계층에서 SIP 서버 모듈 아래에 차별화된 호 처리 기법에 관한 모듈을 삽입하여 상호 연동 구조를 가질 수 있도록 설계하여 SIP 프로토콜 모듈과의 독립성을 유지하도록 하였다. 그림 3은 차별화된 호 처리 기법을 포함하는 SIP6¹⁾ 시스템의 모듈 구성도를 보여주고 있다.

SIP6d는 SIP 서버 데몬으로 프락시 서버(Proxy Server), 리다이렉트 서버(Redirect Server), 로케이션 서버(Location Server), 웹레지스트라(Web Registrar) 모듈을 포함하며, 메시지 파서(Message Parser)를 통하

1) SIP6는 IPv6를 지원하는 SIP 프로토콜을 지칭한다.

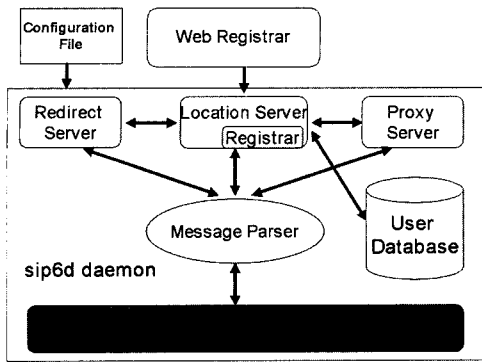


그림 3 SIP6 서버 시스템 구성도

여 수신되는 SIP 메시지들을 파싱(Parsing)하고 SIP 트랜잭션(Transaction)을 시작한다. QoS 처리기(QoS Processor)는 확장된 모듈로서 클라이언트로부터 메시지를 수신하고 서비스품질에 따라 차별화된 호 처리 기법을 지원하는 기능을 한다. 차별화된 호 처리 기법이 적용된 SIP 메시지는 메시지 파서에 전달된다. 웹레지스트라(Web registrar)는 웹 인터페이스를 통해 사용자에 대한 정보를 입력(Input), 수정(Update) 및 삭제(Delete) 하는 기능을 담당하며 사용자 데이터베이스(User Database)와 상호 연동된다. 사용자 데이터베이스는 SIP 사용자와 관련된 신원, 위치, 협상된 서비스품질 정보 등의 테이블을 유지한다.

4.1 SIP6d 기능 설계

4.1.1 QoS 처리기(QoS Processor)

QoS 처리기는 수신된 메시지에 대해 차별화된 호 처리 기법을 적용한다. 차별화된 호 처리 기법에 대한 설명은 4.2절에서 기술하도록 한다.

4.1.2 메시지 파서(Message Parser)

메시지 파서는 수신된 SIP Request 및 Response 메시지에 대한 헤더 및 헤더 옵션들을 파싱하고 SIP 트랜잭션을 시작하는 모듈이다. SIP 메시지는 문자열로 구성되어 있다. SIP 트랜잭션을 시작하기 위해 메시지 파서는 문자열을 분류하여 Request Line, Response Line 과 헤더 및 헤더 옵션을 파싱 한 후, SIP 메시지의 종류에 따라 다음과 같은 처리 과정을 따른다.

SIP Request 메시지의 경우, Provisional Response 및 Request 메시지 처리를 위해 Cseq, Call-ID, Via, Contact, Expire, From, To 등의 헤더를 생성한다. SIP는 하위 프로토콜로서 UDP를 사용한다. UDP는 신뢰성을 지원하지 않는 프로토콜이다. 이러한 문제를 해결하기 위해 RFC2543에 기술되어 있는 신뢰적인 메시지 전달 방식을 따랐다. SIP UA는 SIP Request 메시지를 0.5s, 1s, 2s, 4s, 4s, 4s 시간 간격으로 서버에 전송

한다. 단, 서버로부터 200 OK Response 메시지를 받을 때까지 유효하다. 만약 모든 메시지 전달에도 불구하고 서버로부터 응답이 없을 경우 에러(Error)로 처리한다. SIP 서버는 이렇게 전송된 SIP Request 메시지에 대한 중복성을 검사하여야 한다. 이러한 기능을 지원하기 위해, SIP6 서버는 수신된 SIP Request 메시지의 Call-ID, Via, Cseq, To, From 헤더를 사용하여 Hash 함수를 통해 Hash Key를 할당하고 그 정보를 Hash Table에 유지하여 메시지 중복을 해결한다. 수신된 문자열로부터 메소드를 파싱하여 SIP 트랜잭션에 필요한 모든 정보를 생성한 후, 생성된 Hash Key를 기반으로 Request 메시지 중복검사를 수행한다. 만약 유일한 Request 메시지이면 메시지 처리 버퍼(Message Processing Buffer)에 그 정보를 저장하고 그렇지 않으면 해당 메시지를 삭제한다. 메시지 처리 버퍼에 저장된 메시지들은 SIP6 서버 내에서 처리가 완료될 때까지 유지되며, 처리가 완료되는 즉시 버퍼에서 삭제된다. 버퍼에 저장된 메시지는 메소드 타입에 따라 INVITE, CANCEL, ACK 메소드의 경우 서버동작 모드에 따라 프락시 서버 및 리다이렉트 서버를 호출한다. REGISTER 메소드의 경우 사용자 정보 등록을 위해 로케이션 서버를 호출한다.

SIP Response 메시지의 경우, 헤더를 파싱하여 Response 메시지를 생성 한 후, 해당 클라이언트에 Response 메시지를 그대로 전달한다.

4.1.3 프락시 서버(Proxy Server)

수신된 호에 대한 전달 기능 및 그 외 응답 메시지에 대한 관리 기능을 수행한다. 수신된 메시지의 메소드 타입에 따라 INVITE 메소드일 경우, 로케이션 서버를 통해 호 전달을 위한 사용자 위치 정보를 획득한 후 INVITE 메소드를 해당 목적지에 전달하여 세션을 초대한다. 수신된 모든 메시지는 프락시 서버가 처리하기 위한 서비스 유효시간(timeout)이 설정된다. 유효시간이 만료되면 Request를 취소하고 487 응답 메시지를 전송한다. ACK 메소드 일 경우 목적지 클라이언트에 ACK 메소드를 전달한다.

4.1.4 리다이렉트 서버(Redirect Server)

수신된 호에 대한 리다이렉트 서비스 기능을 수행한다. INVITE 메소드를 수신하면 응답을 위한 접속 헤더를 초기화하고 서비스 유효시간을 설정한 후 로케이션 서버를 통해 목적지 사용자에 대한 위치 정보를 얻는다. 얻어낸 주소가 도메인 일 경우 리졸버를 통해 IP로 매핑 후 세션 초대를 요청한 클라이언트에 사용자 위치정보 URI(User ID, Host, Port Number)를 전달한다.

4.1.5 로케이션 서버(Location Server)

SIP 로케이션 서버 모듈은 크게 두 가지 기능을 수행

한다. 사용자에 대한 정보를 등록 및 관리하는 레지스트라(Registrar) 서버 기능과 SIP 프락시 서버 및 리다이렉트 서버 모듈을 위한 사용자 위치 서비스 기능을 지원한다. 표 1은 사용자 데이터베이스의 테이블 구성을 나타내며, 서비스 처리 절차는 다음과 같다.

표 1 사용자 데이터베이스의 테이블 구성

테이블 이름	설 명
User Input Table	시스템을 사용하는 인증된 모든 사용자 ID를 관리하며, 사용자는 하나의 사용자 ID와 여러 개의 별명이 등록 가능
User Identity Table	세션 설정 시 사용하는 유일한 ID를 관리하는 테이블로, 복수개의 별명과 매핑되는 유일한 사용자 ID를 포함
User Contact Table	사용자의 현재 위치에 대한 정보를 관리

- 웹레지스트라 및 SIP UA로부터 사용자 정보(User ID, SIP UA Port Number, 현재의 URL, Server Action, QoS Service Level)를 수신하면 차 후 서비스 처리 시 사용을 위해 사용자 입력 테이블(User Input Table)에 그 정보를 등록한다. 또한, 사용자 아이디(User ID)는 시스템에서 사용되는 유일한 식별자로 사용자 확인 테이블(User Identity Table)에 등록된다.
- SIP 사용자로부터 이동한 현재의 위치가 등록되면 사용자 확인 테이블에 존재하는 유일한 아이디와 현재의 위치를 매핑 한 후 사용자의 현재 위치 정보를 사용자 접속 테이블(User Contact Table)에 등록한다.
- 프락시 서버 및 리다이렉트 서버로부터 사용자에 대한 위치 정보 요청이 있을 경우 사용자 접속 테이블을 참조하여 해당 사용자 위치 정보를 서비스한다.

4.2 차별화된 호 처리 기법 설계

본 논문에서는 응용 계층에서 어플리케이션 큐잉 기법과 IPv6 헤더의 Flow Label 필드를 사용하여 차별화된 호 처리 기법을 제안한다. 차별화된 호 처리 기법은 우선순위에 기반을 두어 다음과 같이 세 단계로 정의된 서비스품질에 따르는 QoS 파라미터를 사용한다.

- *High Quality* : 가장 높은 우선순위를 가지는 서비스품질로 빠른 세션 생성 및 메시지 전달을 지원하며 High Quality Buffer를 이용한다.
- *Medium Quality* : 중간 수준의 서비스품질로 두 번째 우선순위를 가지며 Medium Quality Buffer를 이용한다.
- *Normal Quality* : 가장 하위 우선순위를 가지는 서비스품질로 Normal Quality Buffer를 이용한다.

이러한 서비스품질은 IPv6의 Flow Label 필드를 통해 설정되며, 본 논문에서는 Flow Label 필드 중 하위

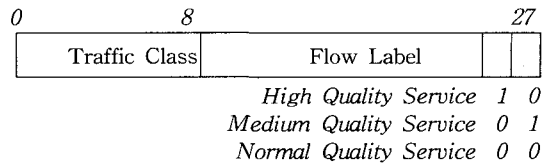


그림 4 서비스품질에 따른 코드비트 정의

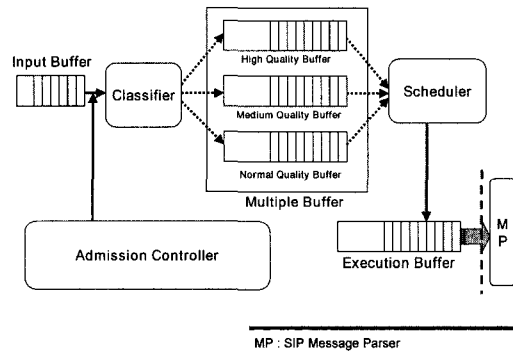


그림 5 차별화된 호 처리 기법 동작

2Bit를 사용하여 그림 4와 같이 서비스품질에 해당하는 코드 비트를 설계하였다.

그림 5는 QoS 처리기에서 차별화된 호 처리 기법을 바탕으로 호 메시지 처리 절차를 보여준다. 이러한 동작은 호를 설정하기 전에 사용자와 SIP 서버사이에 서비스품질에 대한 협상이 이미 이루어졌음을 가정한다. 서비스품질에 대한 협상은 사용자 등록 시에 이루어지며 협상된 서비스품질은 사용자 데이터베이스의 사용자 입력 테이블에 유지된다.

수락제어기(Admission Controller)는 SIP 메시지를 포함하고 있는 IPv6 헤더의 Flow Label 필드에 설정되어 있는 서비스 품질을 참조하여 차별화된 호 처리 기법의 적용여부를 결정한다. 수락제어기는 이러한 결정을 위하여 사용자 입력 테이블에 저장되어 있는 서비스 품질에 대한 정보를 참조하게 된다. 분류기(Classifier)는 수락제어기에 의해 처리된 SIP 메시지를 서비스 품질에 따라 다중 버퍼(Multiple Buffer)에 각각 분류한다. 스케줄러(Scheduler)는 다중 버퍼에 접근하여 우선순위가 높은 버퍼를 중심으로 SIP 메시지들을 실행 버퍼(Execution Buffer)에 정렬시킨다. 다중 버퍼에서 메시지를 처리할 때, 스케줄러는 다음과 같은 조건을 만족한다. 우선순위가 낮은 하위 버퍼에서 메시지를 처리하고 있는 동안, 우선순위가 높은 상위 버퍼에 새로운 SIP 메시지가 입력되면 스케줄러는 현재의 작업을 중단하고 우선순위가 높은 상위 버퍼의 메시지를 먼저 처리할 수 있는 능력을 가져야 한다. 이러한 기능을 지원하기 위하

여, 본 논문에서는 Simple Priority Scheduling 알고리즘을 제안하여 사용한다.

실행 버퍼에 정렬된 SIP 메시지들은 메시지 파서에 의해 순차적으로 읽혀져 SIP 트랜잭션이 시작된다.

5. 구현 및 결과

5.1 SIP6d 구현

SIP6d는 IPv6을 지원하는 Linux 커널 2.2.x 시스템 환경에서 C 언어를 사용하여 구현하였다. 사용자 정보를 관리하기 위해 Mysql 서버를 사용하였고, 메시지 동시 처리율을 고려하여 각각의 주요 모듈은 Posix 스레드 기법을 이용하여 구현하였다. 또한, IPv6를 지원하기 위해 IPv6 소켓 구조체 및 소켓 API들을 사용한다. SIP 트랜잭션에 필요한 모든 정보는 request_t 구조체를 사용하여 정의하였다.

프락시 서버, 리다이렉트 서버, 로케이션 서버는 SIP6d라는 데몬 프로세서로 실행되며 sipd.conf 파일 (Server Name, Domain, Port Number, Process ID, Log File Path 등을 포함)을 참조하여 구동 된다. 구동된 SIP6d는 IPv6 UDP 소켓을 개설하여 메시지를 수신 후 IPv6 헤더의 Flow Label 필드에 명세된 서비스품질에 따라 메시지를 분류하고 각각 분류된 메시지는 우선순위가 높은 메시지 순서로 처리되어 프락시 서버, 리다이렉트 서버, 로케이션 서버, 레지스트라 서버를 선택적으로 호출한다.

SIP6d의 핵심 모듈은 SIP 트랜잭션이 시작되는 메시지 파서이다. 메시지 파서는 SIP 메시지의 메소드 및

```

/* Message Parsing Algorithm */
parse Protocol, Version, Status Code, Reason Phrase
and
Request-URI
set Message type to request_t
read Message type from request_t
if(Message type is Request){
    create Cseq, Call-ID, Via, Contact, Expire, To, From
header make hash key by using From, To, Cseq,
Call-ID header field search duplicated hash key in
hash table
    if(hash key is unique){
        save the Request Message to Message
Processing Buffer
        call RequestProcessThread()
    }
    else
        erase the Request Message
}
else
    call ResponseProcessThread()
    
```

그림 6 메시지 파싱 알고리즘

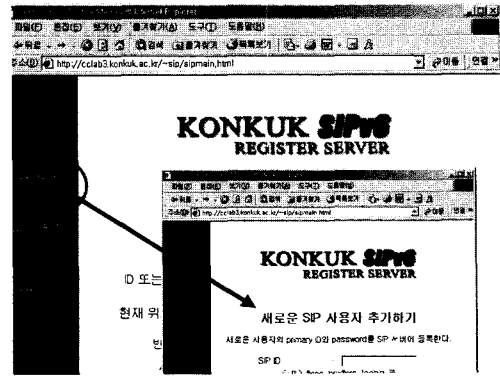


그림 7 웹 기반 레지스트라 실행 윈도우

헤더를 파싱하여 SIP 트랜잭션에 사용 될 request_t 구조체 정보들을 생성하고, 메시지 중복성 검사를 통하여 신뢰성 있는 메시지 전달 기능을 지원한다. 그림 6은 본 논문에서 구현된 메시지 파싱 알고리즘들을 보여준다.

그림 7은 웹 기반 레지스트라의 실행 모습으로 PHP Version 4.0을 이용하여 구현하였다. 주요 기능으로 사용자 추가, 사용자 확인 테이블 등록 및 변경, 사용자의 현재 위치 등록 기능 등을 지원한다.

5.2 차별화된 호 처리 기법 구현

본 논문에서는 차별화된 호 처리 기법을 지원하기 위해 IPv6 헤더의 Flow Label 필드를 이용하였다. 현재 IPv6 헤더의 Flow Label 필드는 QoS 서비스 설정을 위한 목적으로 총 24 비트의 공간을 제공하고 있다. 하지만, 미래의 사용을 위해 정의만 되어 있을 뿐 현재 사용되고 있지 않은 실정이다. 그림 8은 본 논문에서 qos.h 헤더파일에 정의한 서비스품질 및 QoS 파라미터에 대한 코드 값을 보여준다.

현재 IPv6 소켓 구조체(struct sockaddr_in6) 내의 sin6_flowinfo 변수는 IPv6 헤더의 Flow Label 필드에 대한 정보를 표현한다. 본 논문에서는 sockaddr_in6 구조체를 활용하여 qos.h 파일에 정의된 QoS 파라미터를 인식한다. 그림 9는 IPv6 헤더의 Flow Label 필드에 명세된 QoS 파라미터 코드 값에 따라 메시지를 분류하

```

/* qos.h Header File */
...
/* define the High Quality Code Value */
#define HighQuality 2
/* define the Medium Quality Code Value */
#define MediumQuality 1
/* define the Normal Quality Code Value */
#define NormalQuality 0
    
```

그림 8 QoS 파라미터 및 코드 값

```

/* Classify Processor */
if (client_sock.sin6_flowinfo is 2 )
    Save message in High quality buffer
elseif (client_sock.sin6_flowinfo is 1)
    Save message in Medium quality Buffer
else
    Save message in Normal quality buffer
    
```

그림 9 메시지 분류 알고리즘

```

/* Priority Scheduling Processor */
...
while(1)
{
    /*Message processing in high quality buffer */
    Label 1:
    while (High quality buffer have request message)
        Save message in Execution Buffer
    /*Message processing in medium quality buffer */
    Label 2:
    while (Medium quality buffer have request message)
    {
        if (High quality buffer have request message)
            Goto Label 1;
        else
            Save message in Execution Buffer
    }

    /*Message processing in normal quality buffer */
    Label 3
    while (Normal quality buffer have request message)
    {
        if (High quality buffer have request message)
            Goto Label 1;
        elseif (Medium quality buffer have request message)
            Goto Label 2;
        else
            Save message in Execution Buffer
    }
}
    
```

그림 10 Simple Priority Scheduling 알고리즘

는 알고리즘에 대한 슈도우 코드이다.

그림 10은 본 논문에서 제안된 Simple Priority Scheduling 알고리즘에 대한 슈도우 코드이다.

Simple Priority Scheduling 알고리즘은 세 단계로 동작한다. 1단계(Label 1)에서 쓰레드는 우선순위가 가장 높은 High Quality Buffer에 메시지가 존재하는지를 검사 한 후, 메시지가 존재하면 입력된 메시지를 실행 버퍼에 저장한다. 만약 존재하지 않을 경우 2단계로 분기한다. 2단계(Label 2)에서 쓰레드는 다시 우선순위가 가장 높은 High Quality Buffer를 검사하고 메시지가 존재하면 Label 1으로 분기하여 우선순위가 가장 높은

메시지를 먼저 처리하게 된다. 만약 입력된 메시지가 없을 경우, Medium Quality Buffer의 메시지를 처리 한 후 3단계로 분기한다. 3단계(Label 3)에서 쓰레드는 우선순위가 가장 낮은 Normal Quality Buffer에 입력된 메시지를 처리한다. 단, 3단계 역시 먼저 상위 우선순위 버퍼들을 차례로 검사 한 후 입력된 메시지가 존재하면 버퍼에 따라 Label 1 또는 Label 2로 분기하여 상위 우선순위 버퍼에 존재하는 메시지를 처리한다. 만약, 상위 두 개의 버퍼에 메시지가 존재하지 않을 경우 Normal Quality Buffer에 존재하는 메시지는 처리된다. 이렇게 쓰레드는 하위 우선순위 버퍼를 처리 할 때, 항상 상위 우선순위 버퍼들을 먼저 검사함으로써 우선순위가 높은 버퍼에 입력되는 메시지에게 메시지 처리의 우선순위를 부여한다.

6. 성능 분석

본 논문에서는 SIP6d에서 호 처리 및 메시지 처리 시 차별화된 호 처리 기법에 대한 성능을 실험하고 결과를 분석하였다. 이러한 결과에 대한 비교 분석을 위해 RFC 2543을 충실히 따르는 공개용 소프트웨어 중 가장 안정적이며 우수한 성능을 인정받은 콜롬비아 대학교의 SIP 서버를 사용하였다.

실험환경으로 건국대학교에 구축되어있는 IPv6 로컬 테스트 망을 사용하였다. 사용된 시스템은 FreeBSD 기반 IPv6 라우터 1대, SIP6d 구동을 위한 Linux Server 1대, 콜롬비아대학 SIP 서버 구동을 위한 Linux Server 1대, 시뮬레이션 프로그램 구동을 위한 Window 2000 Server 1대로 구성된다.

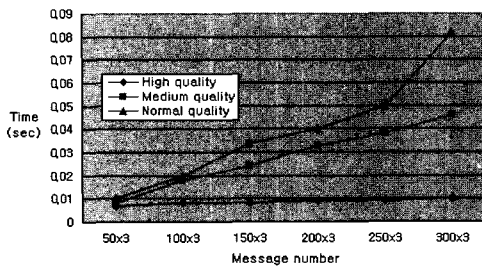
시뮬레이션 프로그램은 Posix 쓰레드 기법을 사용하여 동시에 여러 개의 메시지를 SIP6d에 전송하고 해당 메시지가 처리되어 응답이 오는데 걸리는 시간을 측정한다. 실험은 다음과 같은 방법을 사용하였다. 첫째로, 시뮬레이션 프로그램은 3개의 서로 다른 서비스품질을 가지는 REGISTER, INVITE 메시지를 50, 100, 150, 200, 250, 300개씩 증가시키면서 동시에 SIP6d에 전송한다. 단, 콜롬비아대학교의 SIP 서버는 차별화된 호 처리 기법이 지원되지 않으므로 서비스품질이 설정되지 않은 동일한 메시지를 150, 300, 450, 600, 750, 900개씩 증가시키면서 동시에 SIP 서버로 전송한다. 둘째로, SIP6d는 수신된 메시지를 차별화된 호 처리 기법을 적용하여 처리하고 200 OK 메시지를 전송한다. 단, 콜롬비아대학교의 SIP 서버는 차별화된 호 처리 기법이 적용되지 않는다. 마지막으로 시뮬레이션 프로그램은 SIP6d, SIP 서버로부터 수신된 200 OK 응답 메시지를 기반으로 각각의 메시지 처리 시간을 측정한다. SIP6d에서의 메시지 처리시간은 3개의 서로 다른 서비스품질

을 가지는 메시지 별로 평균시간을 측정한다. SIP 서버에서의 메시지 처리시간은 전송된 모든 메시지에 대한 처리시간을 평균으로 측정한다.

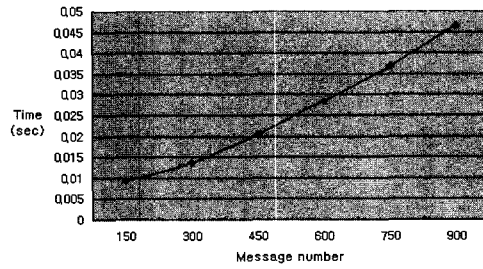
그림 11, 그림 12, 그림 13은 본 논문에서 제안된 SIP6d와 콜롬비아대학교의 SIP 서버에서의 메시지 처리율에 대한 결과를 보여준다. 그래프를 보면 알 수 있듯이, 차별화된 호 처리 기법을 가지는 SIP6d에서는 메시지 처리 시 차별성을 가짐을 볼 수 있다. 특히, High Quality와 같이 높은 서비스 우선순위를 가지는 메시지는 평균 메시지 처리 시간이 매우 빠른 결과를 보여주었다. 다음으로 Medium Quality 서비스 우선순위를 가지는 메시지에 대한 평균 처리 시간이 Normal Quality 서비스 품질을 가지는 메시지에 비해 빠른 결과를 보였

다. 하지만, 차별화된 호 처리 기법을 지원하지 않는 콜롬비아대학교의 SIP 서버에서는 메시지 처리에 대한 차별화를 지원하지 못한다.

다음으로 부하에 따른 메시지 처리율을 측정하기 위해, SIP6d와 SIP 서버가 설치되어있는 Linux 시스템에 부하를 생성한 후 메시지 처리 시간을 측정하였다. 시뮬레이션 프로그램을 이용하여 SIP6d, SIP 서버 내에 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%에 해당하는 부하를 단계적으로 생성한 후 900개의 INVITE 메시지를 전송하여 프락시 서버 내에서의 메시지 처리율을 측정하였다. 그림 14의 그래프에서 보여주듯이, CPU의 부하가 가중됨에 따라 메시지 처리 시간은 현저히 차이가 있음을 알 수 있다. 그러나, 이러한 상태에서

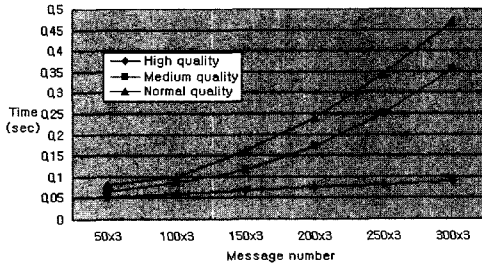


(a) SIP6d의 처리 결과

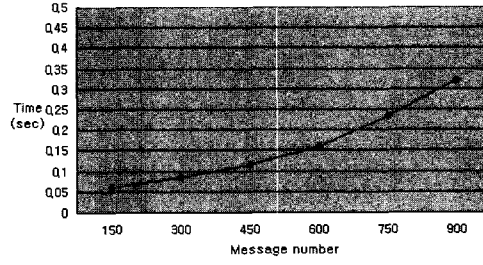


(b) SIP 서버의 처리 결과

그림 11 REGISTER 메시지 처리율

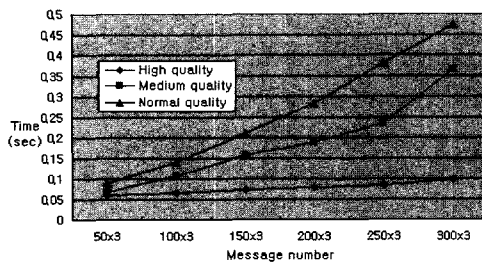


(a) SIP6d의 처리 결과

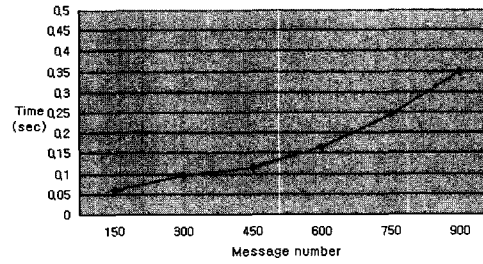


(b) SIP 서버의 처리 결과

그림 12 리다이렉트 서버에서의 INVITE 메시지 처리율

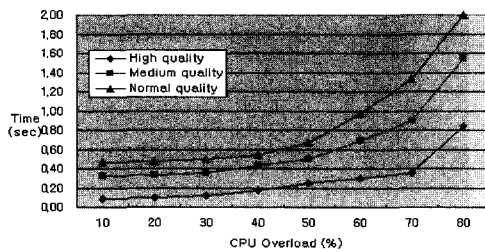


(a) SIP6d의 처리 결과

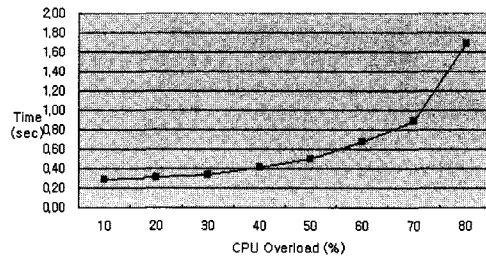


(b) SIP 서버의 처리 결과

그림 13 프락시 서버에서의 INVITE 메시지 처리율



(a) SIP6d의 프락시 서버 내에서의 처리 결과



(b) SIP 서버의 프락시 서버 내에서의 처리 결과

그림 14 CPU 부하에 따른 INVITE메시지 처리율

도 SIP6d는 High Quality 서비스 우선순위를 가지는 메시지에 대한 평균 처리시간이 낮은 우선순위를 가지는 메시지들보다 현저히 빠른 결과를 보여주었다. 하지만, SIP 서버에서는 특정한 차별이 없이 메시지 처리에 대한 평균 시간이 서버의 부하가 가중됨에 따라 현저히 증가함을 볼 수 있다.

위와 같은 결과를 분석해 볼 때, 본 논문에서 구현된 SIP6d는 네트워크 트래픽이 많은 상황이나 서버에 많은 부하가 생성되는 환경에서도, 메시지 처리에 대한 차별화 정책을 통해 높은 우선순위를 가지는 메시지에 대하여 서비스의 질을 보장할 수 있음을 증명하고 있다. 이러한 결과는 NGN에 적용 될 경우, 호 설정 시간에 민감한 응용 서비스에 우선순위를 부여하여 서버 내에서의 호 설정 지연을 최대한 줄임으로서 응용 서비스의 성능을 높일 수 있을 것이다.

7. 결론 및 향후 과제

본 논문에서는 NGN, 3GPP, 3GPP2에서 제시하는 ALL-IP 환경을 고려하여 IPv6를 지원하고 호 처리 및 메시지 처리 시 차별화된 호 처리 기법을 지원하는 SIP 프로토콜을 설계 및 구현하였다. 본 논문에서 제안한 차별화된 호 처리 기법은 세 단계의 서비스품질을 사용하며, IPv6 헤더의 Flow Label 필드를 이용하여 서비스 품질을 설정한다. 또한, 응용 계층을 기반으로 하기 때문에 프로토콜의 확장성, 단순성을 지원하고 빠른 서비스 전개를 이끌어 낼 수 있다. 이러한 장점들은 응용 계층에서 동작하는 SIP 서버가 요구하는 QoS 기능과 잘 부합되며, 빠른 호 설정을 요구하는 응용 서비스들을 위해 호 설정 지연을 줄임으로서 좋은 성능을 지원할 수 있다.

성능 분석 결과, SIP6d는 기존 SIP 서버들과는 달리 호 처리 및 메시지 처리 시 차별화된 호 처리 기법을 통해 높은 우선순위를 가지는 메시지 그룹에 대한 메시지 처리율이 현저히 빠른 결과를 보여주었다. 서버의 부하가

발생할 경우, 높은 우선순위를 가지는 메시지 그룹과 그렇지 않은 메시지 그룹들의 메시지 처리율을 비교 해 볼 때 상대적으로 높은 우선순위를 가지는 메시지에 대한 처리율이 더욱 빠른 결과를 보였다. 이러한 결과는 네트워크 및 서버 부하가 과중 되는 어려운 서비스 환경 하에서도 높은 서비스 품질을 사용하는 메시지 그룹은 서비스의 질을 어느 정도 보장 받을 수 있음을 증명하는 것이다. 결국, SIP를 통해 서비스가 통합되는 ALL-IP 환경에서 호 설정 시간에 민감한 응용 서비스에 적용할 경우, 좋은 성능을 지원 할 수 있을 것으로 기대된다.

본 논문에서 제안된 SIP6는 IPv6 프로토콜을 기반으로 하기 때문에 주소공간에 대한 문제를 해결 할 수 있고 IPv6가 가지는 여러 장점들 또한 포함한다. 향후 연구과제로서, IPv6가 지원하는 IPSec을 적용하여 세션 설정 시에 인증 및 암호화를 지원하는 방안 에 대해 연구하고자 한다.

참고 문헌

- [1] Henning Schulzrinne and Jonathan Rosenberg, "Signaling for Internet Telephony," 1998.
- [2] Handley, H.Schulzrinne, E.Schooler, and J.Rosenberg, "SIP: session initiation protocol," RFC 2543, 1999.
- [3] A.Johnston, S.Donovan, et al, "SIP Telephony Call Flow Examples," Internet Draft, 2000.
- [4] I. Dalgic and H. Fang, "Comparison of H.323 and SIP for IP Telephony Signaling," Photonics East, Proceeding of SPIE'99, Boston, Massachusetts, 1999.
- [5] 3rd Generation Partnership Project, "IP Multimedia (IM) Subsystem-Stage 2(Release 5)," 3GPP TS 23.228 V5.1.0, 2001.
- [6] International Softswitch Consortium, "ISC Reference Architecture V 1.2," 2002.
- [7] R. Braden, Ed, L. Zhang, S. Berson, S. Herzog, S.Jamin, "Resource ReSerVation Protocol (RSVP)," RFC 2205, 1997.
- [8] K. Nichols, S. Blake, F. Baker, D. Black, "Defi-

inition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," RFC 2474, 1998.

- [9] S.Blake, M.Carlson, E.Davies, Z.wang, and W. Weiss, "An architecture for differentiated service," Request for Comments (Proposed Standard) 2475, Internet Engineering Task Force, 1998.
- [10] P.P.Pan and H.Schulzrinne "YESSIR: A simple reservation mechanism for the Internet," in Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV), (Cambridge, England), 1998.
- [11] "Visual Telephone Systems and Equipment for Local Area Networks Which Provide A Non-Guaranteed Quality of Service," ITU-T Recommendation H.323, 1996.
- [12] P. Ferguson and G. Huston, "Quality of Service," John Wiley & Sons, 1998.
- [13] R. Hinden, S. Deering, "Internet Protocol version 6(IPv6) specification," RFC 2460, 1998.
- [14] R.Gilligan, S.Thomson, J.Bound, W.Stevens, "Basic Socket Interface Extensions for IPv6," RFC 2553, 1999.
- [15] Y. Bernet, R. Yavatkar, P.Ford, F.Baker, L. Zhang, M. Speer, and R. Braden., "Interoperation of RSVP/Intserv and diffserv networks," Internet Draft, 1999.
- [16] D. Atkins, G. Klyne, "Common Presence and Instant Messaging," Internt Draft, 2002.
- [17] D. Crocker, A. Diacakis, F. Mazzoldi, C. Huitema, G. Klyne, M. Rose, J. Rosenberg, R. Sparks, H. Sugano, "A Common Profile for Instant Messaging (CPIM)," Inter Draft, 2001.
- [18] J. Rosenberg, D. Willis, H. Schulzrinne, C. Huitema, B. Aboba, D. Gurle, D. Oran, "Session Initiation Protocol (SIP) Extensions for Presence," Internet Draft, 2002.
- [19] B. Campbell, J. Rosenberg, "CPIM Mapping of SIMPLE Presence and Instant Messaging," Internet Draft, 2002.
- [20] W.Almesberger, T.Ferrari. and J.-Y.Le Boudec, "SRP: a scalable resource reservation protocol for the internet," Tech. Rep. SSC/1998/009, EPFL, Lausanne, Switzerland, 1998.



김진철

1995년 한밭대학교 전자계산학과 학사
1977년 건국대학교 공학 석사. 2003년
건국대학교 공학 박사. 관심분야는 IPv6,
VoIP, QoS(Quality of Service), Multi-
casting, ALL-IP



최병욱

2000년 배재대학교 컴퓨터공학과 학사
2002년 건국대학교 공학 석사. 2002년~
현재 건국대학교 공학 박사과정 재학 중
관심분야는 SIP, Multicasting, Mobile
IP, IDN(Internalized Domain Name)



장천현

1977년 서울 대학교 계산통계학과 학사
1979년 한국과학기술원 전산학 석사
1988년 한국과학기술원 전산학 박사
1984년~현재 건국대학교 컴퓨터부 교수
2001년 2월~2002년 1월 미국 UCI 객원
교수. 2002년 9월~현재 건국대학교 정
보통신대학 학장. 2002년 9월~현재 건국대학교 정보통신대
학원 원장. 관심분야는 프로그래밍 언어, 컴파일러, 객체지
향, 실시간 처리



김기천

1988년 2월 서울대학교 계산통계학과 전
산학 학사. 1992년 7월 미 North
Western Univ. 전산학 박사. 1992년 8
월~1996년 4월 한국통신기술(주) 연구
소 선임 연구원. 1996년 4월~1998년 2
월 신세기통신(주) 기술 연구소 책임 연
구원 차장. 1998년 3월~현재 건국대학교 컴퓨터공학과 부
교수. 관심분야는 ALL-IP, 차세대 이동통신 시스템, VoIP,
SIP, 차세대 인터넷 프로토콜



한선영

1977년 서울대학교 계산통계학과 학사
1979년 한국과학기술원 전산학 석사
1988년 한국과학기술원 전산학 박사
1981년~현재 건국대학교 컴퓨터 정보통
신공학과 교수. 1995년 6월~1997년 12
월 건국대학교 산업기술연구소 정보통신
연구센터 소장. 1997년 1월~현재 한국 인터넷 협회 기술위
원회 위원 1998년 1월~1999년 1월 미국 Maryland대학교
컴퓨터 과학과 객원교수. 2000년 3월~2002년 8월 건국대학
교 정보통신원 원장. 2003년 1월~현재 개방형 컴퓨터통신
연구회 회장. 관심분야는 Real-Time CORBA, Internet
Caching, 차세대 인터넷 프로토콜, Mobile IP