Journal of Korea Multimedia Society Vol. 15, No. 8, August 2012(pp. 1038-1047)

SIP 시그널링 네트워크에서 지연 큐를 이용한 과부하 제어 방법

이종민†, 전흥진††, 권오준†††

요 약

SIP(Session Initation Protocol)는 IP 전화의 호 설정 및 해제를 위한 응용계층 프로토콜이다. SIP 시그널링 네트워크에서 호 설정을 요청하는 UA(User Agent)의 수가 증가하게 되면 SIP 프록시 서버가 처리해야할 메시지의 수는 증가되어 과부하가 초래될 수 있다. 본 논문에서는 SIP 프록시 서버에 일반 큐와 지연 큐를 두어 SIP 프록시 서버의 과부하를 제어하는 방안을 제안한다. 입력되는 메시지가 일반 큐의 임계치를 초과하여 과부하가 예상될 경우 SIP 프록시 서버의 부하 감소를 위하여 신규 입력되는 INVITE 메시지를 지연 큐에 배치하여 호 설정을 지연시킨다. 그 후에 일반 큐의 메시지가 하한 임계치 이하로 감소할 때 지연 큐의 INVITE 메시지를 처리한다. 시뮬레이션 결과는 일반 큐의 메시지에 대한 빠른 처리로 인해 제안된 방법이 단일 큐 방법과 비교해 재전송 메시지의 수가 약 45% 감소함을 보여주었다. 그리고 시뮬레이션 결과에 의하면 제안 방법이 일반 큐 방법보다 평균 호 성공률이 약 2% 개선되었다.

The Overload Control Scheme Using a Delay Queue in the SIP Signalling Networks

Jong Min Lee†, Heung Jin Jeon††, Oh Jun Kwon†††

ABSTRACT

The SIP(Session Initation Protocol) is an application layer protocol that is used to establish, release, and change the call session of the IP telephony. In the SIP signalling networks, when the number of the UA(User Agent) requested the call session increase, the number of messages to be processed by SIP proxy server increase. It often will be caused the overload of the SIP proxy server. In this paper, we proposed the overload control method with a normal queue and a delay queue in the SIP proxy server. When it is estimated the overload of the server by the excess of the high threshold in the normal queue, new INVITE messages will be put into the delay queue to reduce the load of the server. It results in some delay of the call session from the INVITE message. Subsequently when the number of messages in the normal queue is reduced below the low threshold, the INVITE messages in the delay queue is processed. The simulation results showed that the number of the retransmission messages by our proposed method was 45% less than the one by the method with single queue. The results also showed that the average call success rate by the proposed method was 2% higher than the one by the method with single queue.

Key words: SIP Signalling Protocol(SIP 신호 프로토콜), Overload Control(과부하 제어), Retransmission (재전송), Call Success Rate(호 성공률)

※ 교신저자(Corresponding Author) : 권오준, 주소 : 부산광역시 부산진구 엄광로 176(614-714), 전화 : 051)890-1725, FAX : 051)890-2628, E-mail : ojkwon@deu.ac.kr
접수일 : 2012년 3월 2일, 수정일 : 2012년 4월 21일
완료일 : 2012년 7월 7일

† 정회원, 동의대학교 컴퓨터소프트웨어공학과 부교수
(E-mail : jongmin@deu.ac.kr)
†† 준회원, 동의대학교 컴퓨터소프트웨어공학과 공학박사
(E-mail : dwit0212@hanmail.net)
††† 종신회원, 동의대학교 컴퓨터소프트웨어공학과 교수
※ 본 연구는 2011학년도 동의대학교 교내연구비(2011 AA179)의 지원으로 수행되었음.

1. 서 론

IP 기반의 인터넷 전화는 기존 회선교환 방식의 전화와 비교하여 통화 시간당 비용이 저렴하여 매년 그 사용자 수가 증가하고 있다. SIP는 인터넷 기반의 전화를 위한 호(Session)의 설정, 변경 및 종료를 위한 텍스트 기반의 응용 계층(Application Layer) 프로토콜로써 RFC 3261로 표준화되어 있다[1].

인터넷 전화의 호 설정은 송신 UA(User Agent)와 SIP 프록시 서버, 수신 UA간의 요청/응답 형태의 3단계 핸드쉐이크 방법에 의해 이루어진다. SIP 메시지는 특정 전송 프로토콜을 지정하고 있지 않기에 TCP[2]와 UDP[3] 모두 사용될 수 있다. TCP에 의한 SIP 메시지 전송의 경우 메시지에 대한 신뢰성은 보장될 수 있으나, 실시간 통신에는 적합하지 않다. 따라서 Ekiga[4], LinPhone[5] 등 IP 기반의 통신 소프트웨어들은 호의 설정 및 사용자 메시지 전송을 위하여 UDP를 사용하고 있다. UDP는 기본적으로 메시지에 대한 신뢰성이 보장되지 않는 프로토콜이다. SIP는 메시지에 대한 신뢰성을 보장하기 위해 프록시 서버의 처리 지연 또는 네트워크의 이상 발생 시 메시지 타이머에 의해 요청 메시지를 재전송하는 방법을 사용한다. 그러나 SIP 프록시 서버가 현재 과부하 상태라면 타이머에 의해 재전송되는 메시지는 트래픽의 증가로 인해 SIP 신호(signaling) 네트워크의 성능을 심각하게 저하시키는 원인이 된다[6,7].

RFC 3261에서는 타이머에 의한 과부하 증가를 해결하기 위하여 503(Service Unavailable) 응답 코드를 정의하고 있으나 이는 제한적인 과부하 제어 방법으로 일시적인 과부하 제어 기능만을 수행할 뿐이지 완전한 과부하 해결효과를 기대할 수 없다[8]. 그 동안 인터넷 전화 사용자 수의 증가로 인해 프록시 서버의 과부하 해결을 위한 여러 논문이 있었다. Ohta의 연구에서는 큐의 80%를 상한 임계치 h로 설정하고 큐에 입력되는 메시지가 이 h값에 도달하면 503 메시지를 UA에게 전송하여 큐의 과부하를 제어하였다[9]. Yang 등의 연구에서는 Ohta의 알고리즘을 개선하여 큐에 대한 하한 임계치 l과 상한 임계치 h를 두고서 큐의 메시지가 l과 h 사이에 있는 경우에 입력된 큐의 길이에 선형적으로 비례하여 호 설정 메시지를 거부하는 과부하 제어 알고리즘을 제안하였다[10]. 그리고 이종민 등은 Yang 등의 연구에서 제안한 선형 과부하 제어 알고리즘을 개선하여 입력 큐가 l과 h 사이에 있을 때 새롭게 입력되는 호 설정 메시지들에 대하여 큐의 길이에 비선형적으로 비례하여 과부하를 제어하는 알고리즘을 제안하여 서버 자원을 보다 효율적으로 이용하도록 개선하였다[11].

한편, Ohta는 프록시 서버의 과부하 해결을 위하여 낮은 우선순위의 큐와 높은 우선순위 두 개의 FIFO 큐를 설치하여 INVITE 메시지는 낮은 우선순위의 큐에 입력시키고 non-INVITE 메시지는 높은 우선순위의 큐에 입력 시키는 방법을 사용하여 과부하를 제어하는 방법을 제안하였다[12]. Ohta의 제안 방법은 우선순위가 높은 non-INVITE 큐에 메시지가 존재하지 않을 경우 INVITE 메시지를 처리하는 방식으로 UA의 수가 증가할수록 non-INVITE 큐의 메시지가 존재할 확률이 높아 INVITE 큐의 메시지의 처리시간은 지연될 것이다.

본 논문에서는 SIP 프록시 서버의 과부하를 제어하기 위하여 INVITE 메시지를 포함한 모든 SIP 메시지가 입력되는 FIFO 방식의 일반 큐에 상한 임계치와 하한 임계치를 두고 입력되는 메시지의 수가 상한 임계치를 초과할 경우 새로 입력되는 INVITE 메시지를 지연 큐에 입력하는 방법을 제안한다. 지연 큐에 입력된 INVITE 메시지는 일반 큐가 하한 임계치 이하가 되었을 때 일반 큐에 입력되어 처리하도록 하여 SIP 프록시 서버의 과부하를 제어하도록 하였다. 제안 방법은 INVITE 메시지의 처리를 지연시킴으로써 해당 UA의 호 설정은 지연될 수 있으나 일반 큐에 있는 메시지를 재전송 타이머가 작동하기 전에 처리하여 재전송 메시지에 의한 과부하를 감소시키는 효과를 보였다.

본 논문의 2절에서는 기본 SIP 신호(signaling) 방법과 과부하 제어 방법을 소개하였다. 3절에서는 지연 큐를 이용한 SIP 시그널링 네트워크의 과부하 제어 방법을 기술한다. 4절에서는 제안 알고리즘에 대한 성능을 시뮬레이션하였으며, 5절에서는 결론과 향후 논문과제에 대하여 기술하였다.

2. 관련 연구

2.1 SIP 신호처리

SIP는 인터넷 기반의 호 설정 및 해제를 담당하는 응용계층 프로토콜이다. 그림 1은 호 설정을 위한 일

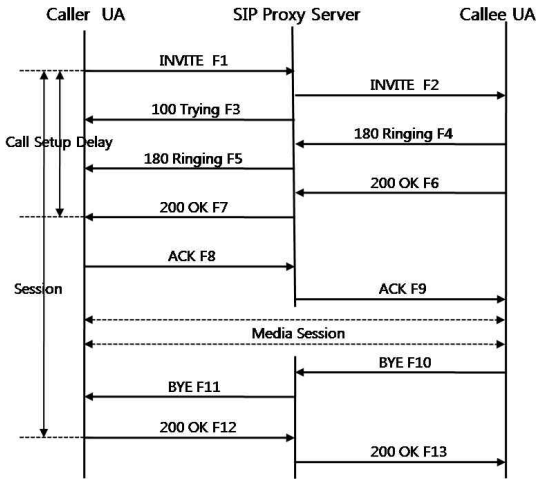

그림 1. SIP 메시지 흐름도

반적인 메시지 전송 과정을 나타낸 그림이다. SIP 프록시 서버는 Caller의 호 설정 요청 메시지 INVITE를 수신하고 메시지 내의 URI를 확인한 후 Callee에게 INVITE 메시지를 전달하고 Caller에게는 100 Trying 응답 메시지를 통하여 전달 사실을 통보한다. Callee는 INVITE에 대한 응답으로 180 Ringing 메시지와 200 OK 메시지를 SIP 프록시 서버를 통해 UA1에게 전달하고 Caller의 ACK 메시지에 의해 Caller와 Callee 사이의 호가 설정된다. 위와 같은 요청과 응답에 의한 동작 과정을 통하여 SIP 프록시 서버는 INVITE를 포함하여 100 Trying, 180 Ringing, 200 OK, ACK, BYE, 200 OK 등 7개의 메시지를 처리한다[13]. 따라서 UA의 수가 증가 할수록 SIP 프록시 서버의 부하는 UA수×7만큼 증가할 것이며 과부하 상태를 초래할 수도 있다.

SIP는 요청 메시지에 대한 응답 메시지가 없을 시 요청메시지의 재전송을 위한 타이머들을 정의하고 있다[1]. INVITE 메시지의 경우 타이머 T1에 의하여 재전송되어진다. 재전송 타이머 T1의 초기 설정 값은 500ms이며 요청 메시지에 대한 응답이 없을 경우 초기 설정 값의 2, 4, 8, 16, 32, 64배의 시간 간격으로 재전송되어진다[1]. BYE 메시지는 타이머 F에 의해 재전송되며 재전송 시간은 64×T1이다.

2.2 과부하 제어 방법

SIP의 메시지 재전송은 신뢰성을 위한 필요한 메커니즘이지만 과부하 발생 시 회복을 더욱 지연시키는 요인으로 작용한다. RFC 3261에서는 로컬 네트워크의 과부하 상태를 안정적으로 제어하기 위하여 503(Service Unavailable) 메시지를 정의하고 있다. 그림 2는 503 메시지의 전송 절차이다.

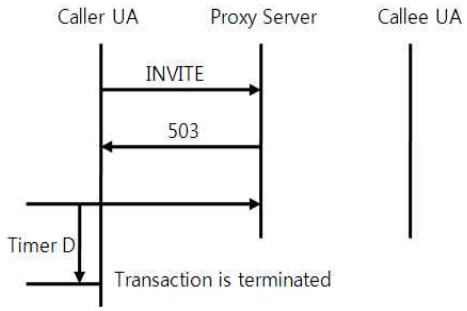

그림 2. 503 메시지 전송절차

프록시 서버는 호 설정을 요청하는 송신 UA의 INVITE 메시지에 대하여 100 Trying 응답 메시지를 전송하는 것이 일반적이다. 그러나 서버의 과부하로 인하여 호 처리 불능 시 503 메시지를 송신 UA에게 전송한다. 503 메시지를 수신한 UA는 네트워크에서 가능한 다른 대체(alternative) 프록시 서버로 INVITE 메시지를 전송하거나 타이머 D의 시간만큼 메시지 전송을 중지한다. 일반적으로 타이머 D의 기본 값은 32초이며 Retry After 필드를 통하여 값의 변경이 가능하다[1]. 503 메시지는 과부하 제어를 위한 효과적인 방법이나 이 메시지는 INVITE 메시지에 대한 하나의 응답 메시지이다. 따라서 INVITE 메시지가 프록시 서버의 큐에 입력되지 못하고 유실 되는 경우에는 503 메시지를 통한 과부하 제어는 적용되지 못하게 된다. 또한, UA의 INVITE 메시지의 재전송이 방지되어도 타이머 D에 의한 트래픽의 순간 폭주(burst) 현상이 발생하는 등 안정적이지 못한 제어 서비스를 제공한다는 문제점이 있다[8]. 따라서 503 응답코드는 제한적인 과부하 제어 방법이다.

Ohta는 프록시 서버의 과부하 해결을 위하여 낮은 우선순위의 큐와 높은 우선순위 두 개의 FIFO 큐를 두고 INVITE 메시지는 낮은 우선순위의 큐에 입력시키고 non-INVITE 메시지는 높은 우선순위의 큐에 입력 시키는 방법을 제안하였다. 낮은 우선순위 큐의 INVITE 메시지는 높은 우선순위 큐의 non-INVITE 메시지가 모두 처리되어 큐가 공백이 된 후 처리되도록 하였다[12]. 이는 과부하 상태에서

INVITE 메시지에 비해 처리 시간이 짧은 non-INVITE 메시지를 먼저 처리함으로써 메시지 처리량을 증가시키는 효과와 한 개의 INVITE 메시지의 처리는 6개의 non-INVITE 메시지를 발생시키기 때문에 INVITE 메시지의 처리를 지연시킴으로서 프록시 서버의 부하를 감소시켰다. 그러나 Ohta의 제안 방법은 non-INVITE 메시지를 우선 처리하도록 함으로써 과부하가 아닌 상황에서 UA의 호 설정 시간이 지연된다는 단점과 UA의 수가 증가할수록 INVITE 메시지가 처리될 확률이 낮아진다는 문제점이 있다.

3. 지연 큐에 의한 과부하 제어 방법

호 설정을 위한 SIP 메시지가 유실되는 경우는 크게 두 가지 경우로 분류된다. 첫째, 프록시 서버에서 메시지가 처리되어 빠져나가는 시간보다 메시지가 프록시 서버에 입력되는 시간이 더 빨라 메시지 큐가 오버플로우되는 경우이다. 둘째, 큐의 크기가 UA의 호 설정을 위한 메시지를 모두 수용할 수 없을 경우다. 만약 하나의 프록시 서버에 n개의 UA가 속해 있고 프록시 서버의 큐 크기가 m일 때 m이 $n \times 7$(하나의 호는 7개의 SIP 메시지에 의해 설정된다)보다 적을 경우 호 설정을 위한 SIP 메시지의 유실이 발생하게 된다. 인터넷의 속도는 빠르게 발전하고 있으며 인터넷 전화 가입자 수는 매년 증가하고 있는 추세이다. 따라서 프록시 서버의 한정된 자원으로 두 가지 유실 경우를 방지하기란 매우 어렵다.

본 논문에서는 한정된 프록시 서버의 자원을 효율적으로 활용하여 호 설정을 위한 SIP 메시지의 유실을 최소화하고 유실로 인해 발생되는 재전송 메시지의 발생을 방지하기 위한 과부하 제어 방법을 제안한다. 제안 방법은 SIP 메시지가 유실될 우려가 있는 과부하 상황에서 UA의 호 설정을 지연시켜 프록시 서버에 입력되는 SIP 메시지 수를 감소시켜 프록시 서버의 부하를 조절한다. 그림 3은 과부하를 제어하기 위한 제안 방법의 프록시 서버의 큐 구조를 나타낸 그림이다. 프록시 서버에는 일반 큐와 지연 큐를 두고 있다. Ohta의 제안 방법 역시 두 개의 큐를 사용하고 있으나 Ohta 방법의 경우 처리에 대한 우선순위가 다른 두 개의 큐를 설치하여 낮은 우선순위 큐에는 INVITE 메시지를 입력시키고 높은 우선순위

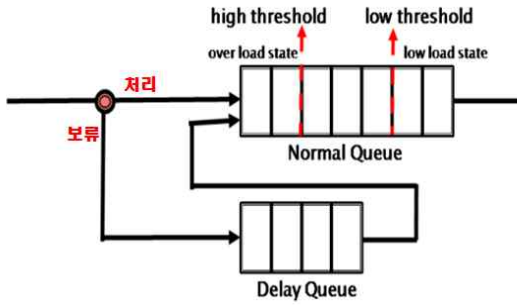

그림 3. 프록시 서버 큐 구조

큐에는 non-INVITE 메시지들을 입력시켜 높은 우선순위 큐에 있는 메시지를 먼저 처리하였으나 제안 방법에서는 두 개의 큐에 대한 우선순위를 지정하고 있지 않다.

일반 큐는 프록시 서버에 입력되는 메시지를 처리하기 위한 기억 공간으로 모든 SIP 메시지는 일반 큐를 통해 처리된다. 지연 큐는 UA의 과부하가 예상될 경우 인터넷 전화를 위한 호 설정 수를 제한하기 위해 처리가 보류된 INVITE 메시지를 보관하기 위한 기억 공간이다. 일반 큐에는 프록시 서버에 입력되는 메시지의 수를 확인하기 위한 두 개의 임계치를 설정하였다. 큐 크기에 비해 입력되어 있는 메시지의 수가 적어 호 설정이 추가로 이루어지더라도 메시지의 유실이 발생할 확률이 없는 비 과부하 지점을 하한 임계치 l로 설정하였으며, 새로운 호가 추가될 경우 메시지가 유실될 우려가 있는 과부하 예상 지점을 상한 임계치 h로 설정하였다.

입력 큐의 상한 임계치 위치는 전체 일반 큐 크기의 90% 지점으로 선정하였다. 상한 임계치를 큐 크기의 100%로 설정할 경우 호 설정이 이미 진행 중인 non-INVITE 메시지가 큐에 입력되지 못하고 유실되는 상황이 초래될 수 있다. non-INVITE 메시지는 INVITE 메시지가 처리된 이후에 발생되는 메시지다. 따라서 non-INVITE 메시지가 유실될 경우, UA 측면에서는 호 설정의 중단이나 지연을 의미하고 프록시 서버 측면에서는 자원의 낭비를 의미한다. 예를 들어 유실된 non-INVITE 메시지의 종류가 ACK 메시지라면 이미 프록시 서버가 이미 처리한 INVITE, 100 Trying, 180 Ringing, 200 OK 메시지는 프록시 서버의 자원만을 사용한 결과가 된다. 따라서 신규 호 설정은 지연시키더라도 이미 진행 중인 호의 non-INVITE 메시지는 일반 큐에 입력되어 정상적

인 호 설정이 완료될 수 있도록 큐의 10% 공간을 남겨 두었다.

그림 4는 비 과부하 상황에서 UA의 INVITE 메시지와 non-INVITE 메시지가 프록시 서버에 입력되었을 때의 일반 큐와 지연 큐의 내부를 나타낸 그림이다. ①, ②, ④번 메시지는 INVITE 메시지이며 ③번 ⑤번 메시지는 non-INVITE 메시지이다. 일반 큐에 입력되는 메시지 수가 상한 임계치 h를 초과하지 않을 경우 메시지의 종류에 관계없이 프록시 서버에 도착한 순서대로 일반 큐에 입력되어 처리되기를 기다린다. 즉 메시지 수가 상한 임계치 h를 초과하지 않는다면 지연 큐는 항상 비어있게 된다.

그림 5는 호 설정을 위한 UA의 SIP 메시지 수가 상한 임계치를 초과하여 입력될 경우의 메시지 처리 방법을 나타낸 그림이다. 상한 임계치는 신규 호 설정을 제한하기 위해 설정한 지점이다. 따라서 입력된 메시지의 종류가 INVITE 또는 non-INVITE 메시지인지를 확인하여, 일반 큐 또는 지연 큐로 입력할 것인지를 결정한다. 예를 들어 ①에서 ⑤번까지의 메시지가 일반 큐에 입력된 이후, 추가로 ⑥, ⑦, ⑧번의 INVITE, non-INVITE, INVITE 메시지가 순서대로 도착하였을 때 메시지 처리 방법은 다음과 같다. ⑥번 INVITE 메시지가 도착하였을 때 일반 큐로 넣게 된다면 과부하 예상 지점인 상한 임계치 h를 초과하게 된다. 만약 일반 큐의 크기가 10% 이하로 남은 상황에서 신규 호 설정을 시작하는 INVITE 메시지

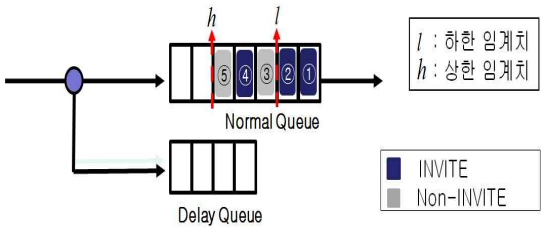

그림 4 비 과부하 상황에서의 메시지 처리 과정

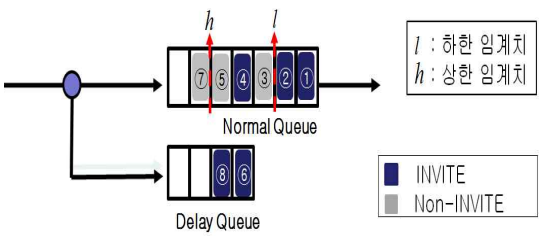

그림 5. 과부하 상황에서의 메시지 처리 과정

를 일반 큐로 넣어 처리하게 된다면 6개의 non-INVITE 메시지가 추가로 생성되어 메시지가 유실될 우려가 있으므로 신규 호 설정을 제한하기 위해 ⑥번 INVITE 메시지는 지연 큐에 입력시킨다. ⑦번 메시지는 non-INVITE 메시지로서 이미 호 설정이 진행되고 있는 메시지이다. 따라서 호 설정이 중단되거나 지연되는 것을 막기 위해 일반 큐로 입력시켜 호 설정이 완료될 수 있도록 한다. 마지막으로 도착한 ⑧번 메시지 역시 일반 큐에 입력된다면 상한 임계치를 h를 초과하게 되므로 지연 큐에 입력시켜 추가 호 설정을 제한한다.

INVITE 메시지의 처리를 지연시키는 동안 이미 호 설정을 진행 중인 UA의 메시지들은 일반 큐를 통해 호 설정이 완료되며 일반 큐의 크기는 점차 감소하게 된다. 일반 큐의 메시지가 하한 임계치 l 이하가 될 때까지 지연 큐의 INVITE 메시지는 처리가 보류된다. 만약 일반 큐의 메시지 수가 하한 임계치보다 많을 때 지연 큐의 INVITE 메시지를 일반 큐로 이동시켜 호 설정을 진행할 경우 프록시 서버에 입력되는 부하가 다시 증가하게 되어 지터(Jitter) 현상이 발생하게 된다. 그림 6은 일반 큐의 메시지 수가 하한 임계치 이하로 되었을 때 지연 큐의 INVITE 메시지가 일반 큐의 마지막 메시지 뒤로 이동했을 때 일반 큐의 내부를 나타낸 그림이다.

그림 7은 일반 큐의 메시지 수가 증가하여 과부하 상태로 천이될 경우 프록시 서버에 입력되는 메시지의 처리를 나타낸 알고리즘이다. 과부하 여부의 결정은 일반 큐의 메시지 수가 상한 임계치를 초과할 경우로 설정하였다. 만약 큐가 과부하 상태라면 입력 메시지의 종류를 확인하여 INVITE 메시지인 경우 지연 큐로 입력시키고 non-INVITE 메시지라면 일반 큐에 입력시켜 진행 중인 호 설정이 완료될 수 있도록 하였다. INVITE 메시지의 경우 URI 파싱 시간으로 인해 non-INVITE 메시지보다 처리시간이

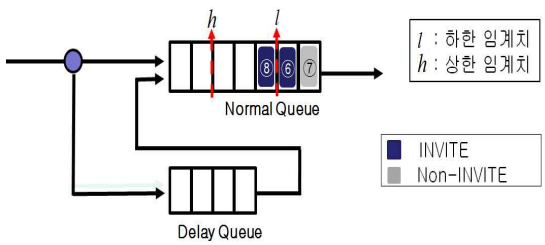

그림 6. 부하감소에 따른 지연 큐의 INVITE 메시지 처리 과정

Algorithm for SIP Message Management
Input * SIP Message (P)
if (Message counter of Normal Queue > high threshold) Overload = True; // Overload condition
if (Overload) 1. if (P->type is INVITE) enqueue P to Q_{delay}; 2. else enqueue P to Q_{normal};

그림 7. 일반 큐의 SIP 메시지 처리 동작

많이 소요된다. 따라서 INVITE 메시지의 처리를 보류시킴으로서 호 설정이 진행 중인 non-INVITE 메시지의 처리시간은 보류된 INVITE 메시지의 처리시간만큼 빠른 호 설정을 가능하게 한다.

지연 큐에 보관된 INVITE 메시지를 일반 큐로 옮길 것인지를 결정하기 위해서는 일반 큐의 메시지 수가 하한 임계치 이하로 감소하였는지를 확인해야 한다. 이를 위해 본 논문에서 지연 큐 타이머를 설치해 일정 시간 간격으로 일반 큐의 메시지 수가 하한 임계치 이하가 되었는지를 확인하도록 하였다. 그림 8의 New_DropTail 클래스 다이그램은 일반 큐로서 timeout() 모듈이 호출되면 지연 큐 타이머 New_DropTailTimer 클래스의 expire()모듈이 Δt 시간 간격으로 실행되면서 지연 큐의 메시지 수가 하한 임계치 이하로 감소하였는지를 확인하게 된다. 그림 9는 지연 큐 타이머가 Δt 시간 간격으로 호 설정이 보류된 지연 큐의 INVITE 메시지를 일반 큐로 이동 시킬지를 결정하기 위한 알고리즘이다. 일반 큐의 메시지

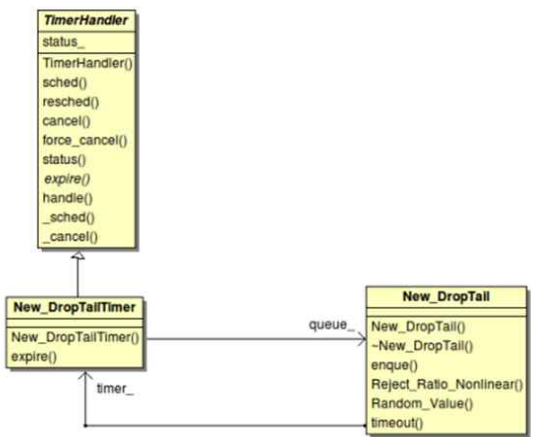

그림 8. 일반 큐와 지연 큐 타이머의 클래스 다이어그램

Algorithm for Delay Queue Management
Input * SIP Message (P)

for every Δt second {
if (Message counter of Normal Queue < low threshold)
Overload = False;
enqueue the first element in Delay Queue to Normal Queue;
else
skip;

그림 9. 일반 큐의 SIP 메시지 처리 동작

수가 하한 임계치보다 적을 경우 보류된 INVITE 메시지 중 가장 먼저 입력된 INVITE 메시지를 일반 큐로 옮겨져 호 설정이 진행 될 수 있도록 한다. 만약 일반 큐의 메시지 수가 하한 임계치 이상이라면 지연 큐의 INVITE 메시지는 계속 지연 큐에 머무르게 된다.

프록시 서버의 과부하를 방지하기 위해 본 논문에서는 지연 큐를 설치하여 INVITE 메시지의 처리를 지연시키는 방법을 제안하였다. 프록시 서버에 제안 방법을 적용할 경우 예상되는 부하 감소의 정도는 다음과 같다. 예를 들어 지연 큐에 입력 된 INVITE 메시지의 수가 q일 때 SIP 프록시 서버의 부하는 $q\times7$ 만큼 감소하게 된다. 또한 프록시 서버로 보내진 메시지에 대한 응답이 일정시간 안에 오지 않을 경우 해당 메시지의 타이머가 메시지를 재전송한다. 따라서 재전송 되는 메시지를 α라고 할 때 하나의 INVITE 메시지 처리를 지연시킴으로써 프록시 서버의 부하 감소량 T는 수식(1) 만큼 감소하게 된다.

$$T=q\times7+\alpha \quad (1)$$

제안방법을 이용한 호 설정 지연에 의한 과부하 제어방법은 해당 UA의 호 설정이 지연 된다는 단점이 있으나 과부하에 의한 메시지의 유실 방지와 일반 큐에 있는 메시지들에 대한 빠른 처리가 가능하도록 하였다. 또한 503 응답 메시지에 의한 과부하 제어 방법의 경우 프록시 서버의 과부하가 해소되어도 타이머 D시간 동안 503 응답 메시지를 수신한 UA는 호 설정이 제한된다는 문제점이 있다. 하지만 제안 방법의 경우 일반 큐의 메시지 수가 하한 임계치 이하로 감소하였을 경우에 즉시 호 설정이 가능하도록 설계하여 과부하 해소와 함께 빠른 호 설정이 가능하다.

4. 성능 평가

4.1 네트워크 모델

시뮬레이션을 위한 네트워크 모델은 그림 10의 형태로 가정하였다. 좌측 UA(Caller) 0~(n-1)는 SIP 프록시 서버를 통하여 우측 UA(Callee) 0~(n-1)과의 일 대 일 호 설정을 요청하는 것으로 가정하였다. 송신 UA와 라우터 R1, R2, R3, SIP 프록시 서버, 수신 UA의 링크 간 대역폭은 100Mbps, 각 구간별 링크 지연 시간은 10ms으로 가정하였다.

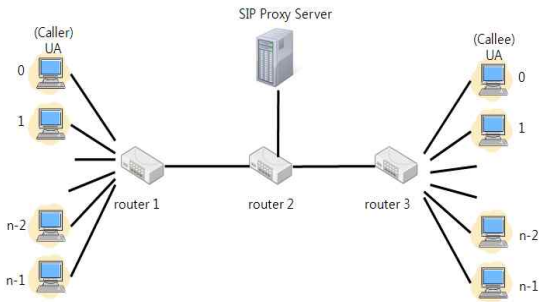

그림 10. SIP 시그널링 네트워크 모델

4.2 시뮬레이션 및 평가

시뮬레이터는 네트워크 성능평가를 위해 널리 사용되고 있는 ns-2.33[14] 버전을 사용하였으며 SIP 모듈은 Rui Prior[15]의 ns-2.27 구현을 포팅하여 사용하였다. 일반 큐는 ns-2의 DropTail 큐를 상속 확장하였으며 지연 큐는 C++의 queue 클래스로 구현하였으며 시뮬레이션을 위한 매개변수는 다음과 같이 가정하였다.

본 논문에서 제안하고 있는 과부하 제어 방안의 효과는 단일 큐만을 사용한 경우와 Ohta의 우선순위를 적용한 이중 큐와 비교 평가하였다. 하나의 메시지에 대한 처리 시간의 경우 INVITE 메시지의 URI 파싱을 위해 non-INVITE 메시지에 비해 많은 시간이 소요되나 제안 방법의 평가를 위한 실험에서는 INVITE 와 non-INVITE 메시지 모두 10ms로 가정하였다. 지연 큐의 메시지를 일반 큐로 이동시키기 위한 지연 큐 타이머의 동작 시간 Δt는 하나의 프록시 서버가 하나의 메시지를 처리하는 시간의 1/2로 설정하였다. 만약 하나의 메시지 처리 시간보다 지연 큐 타이머의 동작 시간이 클 경우 일반 큐의 메시지 수가 l보다 작아졌을 때 지연 큐의 INVITE 메시지를 일반 큐로 이동시키는 시간이 늦어지기 때문이다.

그림 11은 일반 큐의 적정 임계치를 설정하기 위한 실험결과이다. 실험은 1,000개의 UA가 지수분포에 의해 평균 3초 간격으로 호 설정을 요청하도록 설정한 후 실험하였다. 상한 임계치를 0.6으로 설정한 경우 큐의 이용률이 낮아져 지연 큐에서 유실되는 INVITE 수가 높게 나타났다. 반면 하한 임계치를 0.3으로 설정했을 경우 지연 큐의 INVITE 메시지가 보류되는 시간이 길어져 지연 큐의 오브플로우로 인해 신규 입력되는 INVITE 메시지의 유실로 인해 유실 메시지의 수가 증가하는 것을 알 수 있다.

그림 12는 전체 20개의 큐를 일반 큐와 지연 큐로 분할하기 위한 실험결과이다. 송신 UA와 수신 UA가 같은 도메인에 위치할 경우 한 개의 INVITE 메시지를 처리함으로써 100 Trying 메시지를 제외한 5개의 non-INVITE 메시지가 프록시 서버에 입력된다. 따라서 지연 큐의 INVITE 메시지를 일반 큐로 옮겨서 처리될 경우 5개의 non-INVITE 메시지가 입력

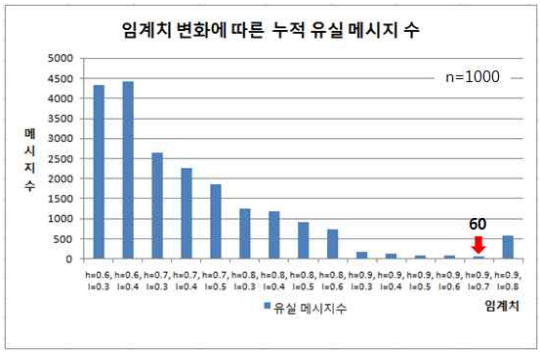

그림 11. 임계치 변화에 따른 유실 INVITE 메시지 수

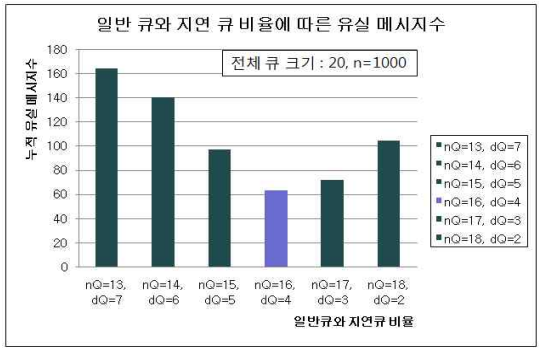

그림 12. 일반 큐와 지연 큐의 비율에 따른 유실 INVITE 메시지 수

될 수 있는 일반 큐 공간이 필요하다. 이러한 INVITE 메시지와 이로 인해 발생되는 non-INVITE 메시지 간의 관계로 인해 실험 결과와 같이 일반 큐와 지연 큐의 비율이 대략 5:1에서 가장 유실되는 메시지의 수가 적게 나타남을 알 수 있다.

그림 13은 UA에 의해 발생된 INVITE 메시지 수를 나타낸 그래프이다. 세 가지 방법 중 Ohta의 우선순위를 적용한 이중 큐 방법이 가장 많은 수의 INVITE 메시지가 발생하였다. Ohta 방법의 경우 INVITE 메시지가 처리되기 위해서는 우선순위가 높은 non-INVITE 메시지가 없을 경우에만 처리된다. 하나의 INVITE 메시지가 프록시 서버에 의해 처리되면 5개의 non-INVITE 메시지가 발생된다. 따라서 호 설정을 요청하는 UA의 수가 증가할수록 낮은 우선순위의 큐에서 유실되는 INVITE 메시지 수는 증가하게 된다. INVITE 메시지가 유실되면 T1 타이머는 500ms에 INVITE 메시지를 재전송하고 이후, 이전 전송 시간의 2배 시간 간격으로 최대 32초 동안 재전송한다. 이러한 이유로 인해 세 가지 방법 중 가장 많은 수의 INVITE 메시지가 UA에 의해 발생된 것이다. 제안 방법의 경우 세 가지 방법 중에서 가장 적은 수의 INVITE 메시지가 발생하였다. 동일한 조건하에서 INVITE 메시지가 가장 적게 발생하였다는 것은 INVITE 메시지의 유실로 인해 재전송된 INVITE 메시지의 수가 가장 적게 발생했음을 의미한다.

그림 14는 부하 증가에 따른 호 성공률을 평가하기 위해 UA의 수를 증가시켜 가면서 실험한 결과이다. Ohta의 이중 큐 방법의 경우 INVITE 메시지는 non-INVITE 메시지와 비교했을 때 처리의 우선순위가 낮으므로 non-INVITE 메시지가 큐에 존재할

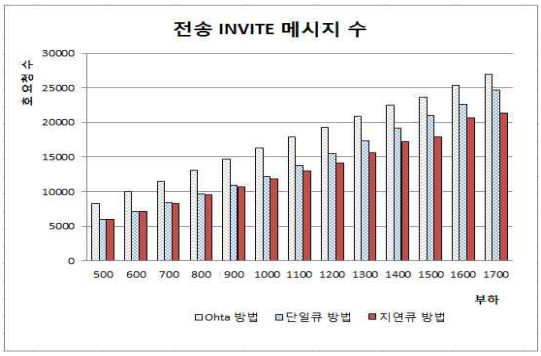

그림 13. 호 설정을 위해 발생된 INVITE 메시지 수

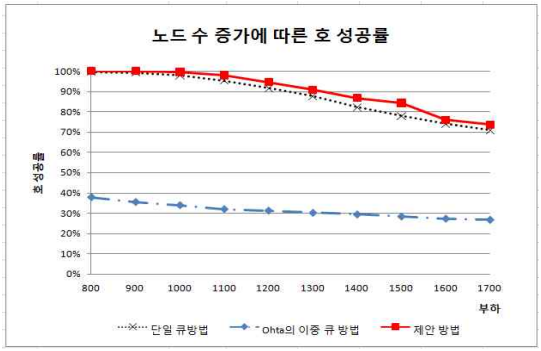

그림 14. 부하 증가에 따른 호 성공률

경우 처리되지 못한다. 따라서 UA의 수가 증가할수록 INVITE 메시지는 재전송 시간 전까지 처리되지 못하거나 대기하는 INVITE 메시지가 큐의 크기보다 많을 경우 처리되지 못하고 유실된다. 또한 UA의 수가 부하 1에 해당하는 800을 초과하면서 제안 방법이 단일 큐 방법과 호 성공률이 높게 나타나는 것을 알 수 있다. 제안 방법의 경우 일반 큐의 메시지 수가 상한 임계치 초과하기 전까지는 INVITE와 non-INVITE 메시지가 우선순위가 아닌 프록시 서버에 도착한 순서대로 처리된다. 따라서 지연 큐에 입력되는 INVITE 메시지는 일반 큐의 상한 임계치를 초과하여 입력되는 INVITE 메시지이다. 일반 큐의 메시지 수가 상한 임계치가 되더라도 non-INVITE 메시지는 일반 큐에 입력되어 처리될 수 있도록 하여 호 설정이 완료될 수 있도록 하였다. 따라서 제안 방법이 두 개의 방법과 비교해 재전송되는 메시지 수가 적게 발생하며 프록시 서버의 자원 이용률 역시 가장 높다. 실험을 통해 제안 방법의 평균 호성공률이 단일 큐 방법보다 2% 가량 높게 평가되었으며 Ohta의 우선순위를 적용한 이중 큐 방법보다 58% 가량 높은 호 성공률을 보였다.

그림 15는 부하 증가에 따라 T1 타이머에 의해 재전송된 INVITE 메시지의 수를 평가한 실험 결과다. 단일 큐 방법의 경우 1700개의 UA가 3초 간격으로 호 설정을 위한 INVITE 메시지를 프록시 서버로 전송했을 때 8,082개의 INVITE 메시지가 T1타이머에 의해 재전송되었으며 Ohta의 방법은 19,741개 제안 방법은 5,424개의 INVITE 메시지가 재전송되었다. 단일 큐 방법의 경우 호 설정 수에 제한이 없다. 따라서 INVITE 메시지의 처리는 프록시 서버에 입력되는 non-INVITE 메시지 수를 증가시키게 된다.

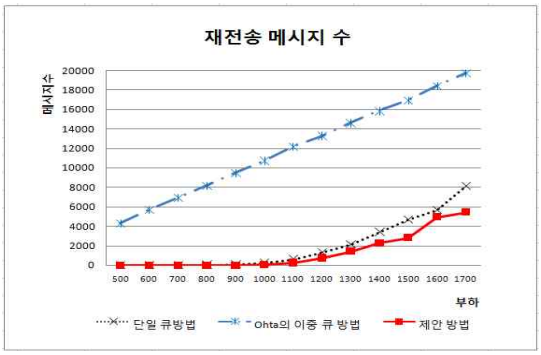

그림 15. 부하 증가에 따른 재전송 메시지 수

Ohta 방법의 경우 INVITE 메시지의 우선순위를 낮게 설정함으로써 UA가 증가할 경우 INVITE 메시지가 처리될 수 있는 기회가 낮다. 반면, 제안 방법은 부하가 상한 임계치 h를 초과 할 경우 INVITE 메시지를 지연 큐로 보내 호 설정을 제한하게 되므로 프록시 서버에 입력되는 non-INVITE 메시지의 수를 감소시킬 수 있다. 따라서 과부하 상태에서 제안 방법을 적용할 경우 재전송되는 메시지에 의한 부하 증가를 방지할 수 있음이 실험을 통해 확인되었다.

5. 결 론

IP 기반의 SIP 시그널링 네트워크에서 프록시 서버는 하나의 호 설정을 위하여 INVITE 메시지를 포함하여 7개의 SIP 메시지를 처리한다. 따라서 호 설정을 요청하는 UA의 수가 증가할수록 프록시 서버의 부하는 UA수×7배 만큼 증가되어 과부하를 초래할 수 있다. 본 논문에서는 호 설정을 요청하는 UA수의 증가로 인해 프록시 서버의 과부하가 예상될 때 일반 큐와 지연 큐에 의해 부하를 조절하는 방법을 제안하였다. 프록시 서버에 입력되는 SIP 메시지의 수가 일반 큐의 상한 임계치를 초과할 경우 신규 입력되는 메시지는 그 종류에 따라 입력되는 큐를 달리하였다. INVITE 메시지의 경우 6개의 non-INVITE 메시지가 추가로 발생되어 부하가 증가하는 것을 방지하기 위해 지연 큐에 입력되도록 하였다. non-INVITE 메시지는 INVITE 메시지가 처리된 이후에 발생되는 메시지로서 호의 설정이 진행 중임을 의미한다. 따라서 호의 설정이 완료될 수 있도록 일반 큐에 입력시켰다.

시뮬레이션은 ns-2.33 네트워크 시뮬레이터를 사용하였으며 제안 방법의 평가를 위하여 단일 큐를 사용한 방법과의 호 성공률과 메시지의 유실로 인해 타이머에 의해 재전송되는 메시지의 수를 비교하여 실험하였다. 그림 13의 실험결과를 통해 동일한 네트워크 환경에서 16개의 일반 큐와 4개의 지연 큐를 사용한 제안 방법이 20개의 단일 큐를 사용한 방법과 비교해 2%가량, Ohta의 제안방법과 비교해 58%가량 높은 호 성공률을 보였다. 메시지의 유실로 인해 타이머로부터 재전송된 메시지의 수를 평가한 그림 14의 실험결과에서는 제안 방법이 단일 큐를 사용한 방법보다 45%가량, Ohta의 방법과 비교해 768%가량 재전송되는 INVITE 메시지의 수가 감소되었다.

본 논문에서는 호 설정을 요청하는 UA의 증가로 인해 과부하가 예상될 경우 일부 UA의 INVITE 메시지에 대한 처리를 지연시키는 방법을 제안하였다. 처리가 지연된 해당 UA의 호 설정은 지연될 수 있으나 일반 큐에 있는 메시지들에 대한 빠른 처리가 가능하도록 하였다. 또한 INVITE 메시지의 처리를 지연시킴으로써 non-INVITE 메시지가 추가 발생되는 것을 방지하여 프록시 서버의 부하를 감소시켜 메시지의 유실로 인해 발생되는 재전송을 방지하였다. 실험 결과를 통해 제안 방법을 적용할 경우 과부하 상황에서 호 성공률 증가와 부하감소에 의한 프록시 서버의 자원 이용률을 개선시킬 수 있음을 확인하였다.

향후 과제로는 제안 방법의 지연 큐에 503 응답코드를 추가 적용하여 부하감소와 평균 호성공률을 평가하는 연구가 필요할 것이다.

참 고 문 헌

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler: *SIP: Session Initiation Protocol*, RFC3261, http://www.ietf.org/rfc/rfc3261.txt, 2002.

[2] J. Postel, *Transmission Control Protocol*, RFC 761, USC/Information Sciences Institute, Tanuary, 1980.

[3] J. Postel, *User Datagram Protocol*, RFC 768, USC/Information Sciences Institute, Tanuary, 1980.

[4] http://www.linphone.org. linphone: open source voip software, 2010
[5] http://www.ekiga.org. Ekiga open source software, 2011
[6] M. Ohta, "Simulation study of SIP Signaling in an Overload Condition," *3rd Int'l Conf. on Communications, Internet, and Information Technology*, pp. 321-326, 2004.
[7] M. Govind, S. Sundaragopalan, Binu K S, and Subir Saha, "Retransmission in SIP over UDP -Traffic Engineering Issues," *Proc. of International Conference on Communication and Broadband Networking, Bangalore*, pp. 573-576, 2003.
[8] V. Hilt, I. Widjaja, and H. Schulzrinne, "Session Initiation Protocol (SIP) Overload Control," *IETF, Internet_Draft, Draft-hilt-sipping-overload-06*, 2009.
[9] M. Ohta, "Overload Control in a SIP Signaling Network," *International Journal of Electrical, Computer, and Systems Engineering*, Vol. 3, No. 2, pp. 87-92, 2009.
[10] J. yang, F. Huang, S. Gou, "An Optimized Algorithm for Overload Control of SIP Signaling Network," *Proc. of the 5th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 3813-3816, 2009.
[11] 이종민, 전홍진, 권오준, "SIP 큐의 비선형적 과부하 제어 방법," 한국시뮬레이션학회논문지, 제19권, 제4호, pp. 43-50, 2010.
[12] M. Ohta, "Overload Protection in a SIP Signaling Network," *International Conference on Internet Surveillance and Protection (ICISP '06)*, pp. 26-28, 2006.
[13] A. Johnston, Steve Donovan, Robert Sparks, Chris Cunningham, and Kevin Summers, "SIP Basic Call Flow Examples," *RFC 3665, Internet Engineering Task Force (IETF)*, 2003.
[14] Network Research Group, Lawrence Berkeley National Laboratory. Network Sinulator version 2 (ns-2). URL : http://www.isi.edu/nsnam/ns/, 2011.
[15] http://www.dcc.fc.up.pt/~rprior/ns/index-en.ns-2 network simulator extensions, 2011.

이 종 민

1992년 경북대학교 컴퓨터공학과 공학사
1994년 한국과학기술원 전산학과 공학석사
2000년 한국과학기술원 전자전산학과 공학박사

1997년~2002년 삼성전자 무선사업부 책임연구원
2005년 Research Associate, University of California at Santa Cruz
2012년 Visiting Scholar, The University of Alabama
2002년~현재 동의대학교 컴퓨터소프트웨어공학과 부교수
관심분야 : 모바일컴퓨팅, 병렬컴퓨팅, 라우팅

전 흥 진

1995년 지산전문대학 전자계산학과
1998년 동서대학교 컴퓨터공학과 공학사
2000년 부경대학교 소프트웨어공학과 교육학석사

2012년 동의대학교 컴퓨터소프트웨어공학과 공학박사
관심분야 : 컴퓨터 네트워크, 정보보호

권 오 준

1986년 경북대학교 전자공학과 공학사
1992년 충남대학교 전산학과 이학석사
1998년 포항공대 전자계산학과 공학박사

1986년~2002년 한국전자통신연구원 선임연구원
2000년 현재 동의대학교 컴퓨터소프트웨어공학과 교수
관심분야 : 컴퓨터네트워크, 정보보호, 패턴인식, 인공신경망