

# SIP 프록시 큐의 비선형적 과부하 제어 방법

이종민<sup>1</sup> · 전흥진<sup>2</sup> · 권오준<sup>1†</sup>

## A Non-Linear Overload Control Scheme for SIP Proxy Queues

Jong Min Lee · Heung-Jin Jeon · Oh-Jun Kwon

### ABSTRACT

Recently, the Internet telephony has been used rather than the traditional telephony by many Internet users, with low cost. Session Initiation Protocol(SIP) is the standard of application layer protocol for establishment and disconnection of the session for Internet telephony. SIP mainly runs over the UDP for transport. So in case of the loss of the INVITE request message, the message is retransmitted by an appropriate timer for reliable transmission of the UDP message. Though the retransmission is useful for ensuring the reliability of SIP messages sent by the users, it may cause the overload traffic in the SIP proxy server. The overload in SIP proxy servers results in the loss of many input messages. This paper presents a non-linear overload control algorithm to resolve the overload condition of the server. we simulate our proposed algorithm using the network simulator ns-2. The simulation results show that the throughput of the server with the proposed algorithm have been improved about 12% compared to the existing linear control algorithm.

**Key words** : SIP(Session Initiation Protocol), Overload Control, Rejection Ratio, Non-linear Control

### 요약

최근 인터넷 전화는 기존 유선 전화에 비해 적은 비용으로 통화를 할 수 있다는 장점으로 인해 많은 인터넷 사용자들에 의해 사용되어지고 있다. Session Initiation Protocol(SIP)은 인터넷 전화의 세션 설정 및 해제를 위한 응용계층 프로토콜 표준이다. SIP는 주로 전송을 위해 UDP를 사용하고 있다. 따라서 INVITE 요청 메시지가 유실되는 경우, UDP 메시지의 신뢰성 있는 전송을 위하여 그 메시지는 적절한 타이머에 의해 재전송된다. 재전송은 사용자에 의해 보내진 메시지의 신뢰성 보장을 위해 유용할 지라도, SIP 프록시(Proxy) 서버의 과부하를 초래하는 원인이 되기도 한다. SIP 프록시 서버의 과부하로 인하여 입력되는 메시지가 많이 유실될 수 있다. 본 논문에서는 프록시 서버의 과부하를 효율적으로 해결하기 위한 비선형 과부하제어 알고리즘을 제안한다. 제안된 알고리즘은 ns-2 시뮬레이터를 이용하여 성능을 시뮬레이션하였다. 시뮬레이션 결과는 기존의 선형 과부하 제어 알고리즘에 비해 제안된 알고리즘의 프록시 서버의 처리율이 약 12%정도 향상됨을 알 수 있었다.

**주요어** : SIP 프로토콜, 과부하 제어, 거부율, 비선형 제어

## 1. 서론

인터넷 전화는 통화시간당 비용이 기존의 유선 전화보

이 논문은 2009학년도 동의대학교 교내연구비에 의하여 연구되었음(2009AA180).

접수일(2010년 7월 2일), 심사일(1차 : 2010년 9월 7일, 2차 : 2010년 10월 19일), 게재 확정일(2010년 10월 27일)

<sup>1)</sup> 동의대학교 컴퓨터소프트웨어공학과

<sup>2)</sup> 동의대학교 일반대학원 컴퓨터응용공학과의

주 저자 : 이종민

교신저자 : 권오준

E-mail; ojkwon@deu.ac.kr

다 저렴하여 매년 사용자 수가 증가하고 있다. SIP는 인터넷 기반의 멀티미디어와 IP 전화를 위한 호(Session)의 설정, 변경 및 종료를 위한 텍스트 기반의 응용 계층(Application Layer) 프로토콜로써 RFC 3261로 표준화 되어 있다<sup>[1]</sup>. 인터넷 전화의 기본 동작은 송신 UA(User Agent)와 SIP 프록시 서버, 수신 UA 사이에서 이루어지며 호 설정은 UA와 SIP 프록시 서버 간의 요청/응답 형태의 3-way 핸드셰이크 방법에 의해 이루어진다. SIP 메시지는 특정 전송 프로토콜을 지정하고 있지 않기에 TCP<sup>[2]</sup>와 UDP<sup>[3]</sup> 모두 사용될 수 있다. TCP에 의한 SIP 메시지 전송은 신뢰성은 보장 될 수 있으나, 실시간 멀티

미디어 통신에는 적합하지 않다. 따라서 Ekiga, LinPhone 등 IP 기반의 통신 소프트웨어들은 호의 설정 및 사용자 메시지 전송을 위하여 UDP를 사용하고 있다. 그러나 UDP는 기본적으로 신뢰성을 보장하지 않는다. 이에 대한 신뢰성 보장을 위한 노력으로 SIP 프록시 서버의 처리 지연 또는 네트워크의 이상 발생시 UA는 재전송 방법에 의해 요청 메시지를 재전송하는 방법을 사용한다. 그러나 SIP 프록시 서버가 현재 과부하 상태라면 메시지의 재전송은 트래픽의 증가로 인해 SIP 신호(signaling) 네트워크의 성능을 심각하게 저하시키는 원인이 된다<sup>4-5)</sup>.

RFC 3261은 UA의 요청메시지에 대한 여러 가지 응답메시지들을 설정하고 있다. 그 중에서 503(Server Unavailable) 메시지는 SIP 프록시 서버가 UA의 요청메시지를 처리할 수 없을 경우에 전송하는 응답 메시지이다. 그러나 현재 서버가 과부하 상태일 경우 UA의 요청 메시지는 버퍼 큐에서 유실될 것이며, UA는 재전송 타이머에 의해 요청 메시지를 정해진 시간과 구간동안 지속적으로 보낼 것이다.

그 동안 인터넷 전화 사용자 수의 증가로 인해 프록시 서버의 과부하 해결을 위한 여러 연구가 있었다.

Ohta의 연구에서는 큐의 80%를 상한 임계치  $h$ 로 설정하고 큐에 입력되는 메시지가 이  $h$  값에 도달하면 503 메시지를 UA에게 전송하여 큐의 과부하를 제어하였다<sup>6)</sup>. Yang 등의 연구에서는 Ohta의 알고리즘을 개선하여 큐에 대한 하한 임계치  $l$ 과 상한 임계치  $h$ 를 두고서 큐의 메시지가  $l$ 과  $h$  사이에 있는 경우에 입력된 큐의 길이에 선형적으로 비례하여 호 설정 메시지를 거부하는 과부하 제어 알고리즘을 제안하였다<sup>7)</sup>. 그러나 이 연구의 선형 과부하 제어 알고리즘은 자원 이용률 측면에서 여전히 개선의 여지가 있다.

본 논문에서는 Yang 등의 연구에서 제안한 선형 과부하 제어 알고리즘을 개선하여 입력 큐가  $l$ 과  $h$  사이에 있을 때 새롭게 입력되는 호 설정 메시지들에 대하여 큐의 길이에 비선형적으로 비례하여 과부하를 제어하는 알고리즘을 제안한다. 제안된 알고리즘은 기존의 선형 과부하 제어 알고리즘과 비교할 때 서버의 처리율이 보다 효율적으로 개선될 것으로 기대된다.

본 논문의 2절에서는 관련연구로 호 설정을 위한 SIP의 기본적인 흐름과 503 응답메시지<sup>1)</sup>를 이용한 과부하 제어를 소개한다. 3절에서는 과부하 제어를 위한 프록시 서버의 큐 구조와 비선형 제어 알고리즘을 제안하였으며 4절에서는 제안 알고리즘에 대한 성능을 시뮬레이션 하였다. 5절에서는 결론과 향후 연구에 대하여 논하였다.

## 2. 관련 연구

### 2.1 SIP의 신호처리

SIP는 인터넷 기반의 호 설정 및 해제를 담당하는 응용계층 프로토콜이다. 그림 1은 호 설정을 위한 일반적인 메시지 전송 과정을 나타낸 것이다. SIP 프록시 서버는 UA1의 호 설정 요청 메시지 INVITE를 수신하고 메시지 내의 URI를 확인한 후 UA2에게 INVITE 메시지를 전달하고 UA1에게는 100 Trying 응답 메시지를 통하여 전달 사실을 통보한다. UA2는 INVITE에 대한 응답으로 180 Ringing 메시지와 200 OK 메시지를 SIP 프록시 서버를 통해 UA1에게 전달하고 UA1의 ACK 메시지에 의해 UA1과 UA2 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 요청/응답 메시지를 처리한다<sup>8)</sup>.

RFC 3261에서는 SIP을 위한 특정 전송계층 프로토콜을 규정하고 있지 않다. 그러나 SIP는 실시간 통신을 위한 프로토콜이기에 Ekiga, LinPhone과 같은 IP 응용 소프트웨어에서는 UDP를 사용하고 있다. UDP가 전송 프로토콜로 적용되는 경우 SIP는 메시지들에 대한 신뢰성을 위하여 요청 메시지에 대한 응답 메시지가 없을 시 요청메시지의 재전송을 위한 타이머들을 정의하고 있다. INVITE 메시지의 경우 타이머 T1에 의하여 재전송되어진다. 이 재전송 타이머 T1의 초기 설정 값은 500ms이며 요청 메시지에 대한 응답이 없을 경우 이 초기 설정값의 2, 4, 8, 16, 32, 64배의 시간 간격으로 재전송되어진다<sup>1)</sup>.

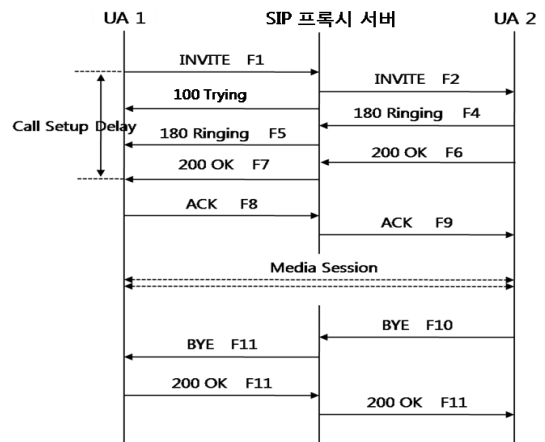


그림 1. SIP 메시지 흐름도



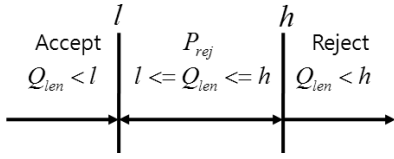


그림 4. 과부하 제어를 위한 메시지 제어

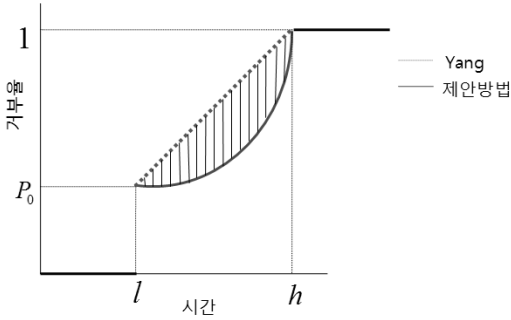


그림 5. 호 요청 메시지 거부율

정하였다.  $l$  과  $h$  사이의 control state 영역은 큐에 들어오는 입력 트래픽을 조절하여 오버로드 상태로 들어가지 않도록 하면서 서버의 자원을 충분히 이용할 수 있도록 하기 위한 영역으로 큐에 신규로 입력되는 호 설정 요청 메시지들에 대한 수락 여부는 그림 4와 같이 결정된다.

입력된 메시지의 수가  $Q_{len}$  일 때,  $Q_{len}$  이 하한 임계값  $l$  보다 적을 경우 모든 메시지는 큐에 입력되고,  $Q_{len}$  이 상한 임계값  $h$  보다 클 경우 모든 메시지는 큐에 입력되지 못하고 프록시 서버에 의해 거부될 것이다.  $l \leq Q_{len} \leq h$  일 경우 식 (1)의 거부율( $P_{rej}$ )에 따라 거부되거나 큐에 입력되도록 하였다.

$$P_{rej} = \left[ \frac{(Q_{len} - l)}{h - l} \right]^2 \times (1 - P_0) + P_0 \quad (1)$$

$P_0$ 는  $Q_{len}$  값이 하한 임계값  $l$ 에 도달했을 때의 초기 거부율이다.

그림 5는 큐의 입력된 메시지의 수  $Q_{len}$ 의 값에 따른 INVITE에 대한 Yang 등의 선형적 제어 알고리즘과 제안 알고리즘의 메시지 거부율을 나타내었다.

점선은 Yang 등의 연구에서 사용된 선형적 과부하 제어 알고리즘의 호 거부율을 나타내며, 실선은 제안된 비선형적 과부하 제어 알고리즘에 적용된 호 거부율이다. SIP 프록시 서버의 과부하 상태에서 제안 알고리즘이 선형 제어 알고리즘과 비교하여 빗금 친 부분만큼 UA의

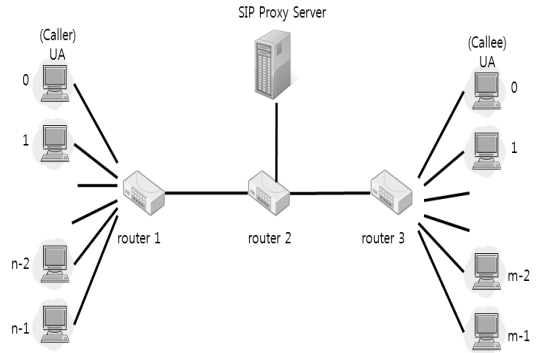


그림 6. 네트워크 모델

INVITE 메시지에 대한 호 설정 메시지를 더 많이 수락함으로써 서버의 처리율을 개선하는 효과가 있을 것으로 판단된다.

## 4. 시뮬레이션

### 4.1 네트워크 모델

시뮬레이션을 위한 네트워크 모델은 그림 6의 형태로 가정하였다.

그림 6의 좌측 UA(Caller) 0~(n-1)은 SIP 프록시 서버를 통하여 우측 UA(Callee) 0~(m-1)과의 일대일 호 설정을 요청하는 것으로 가정하였다. 송신 UA와 라우터 R1, R2, R3, SIP 프록시 서버, 수신 UA의 링크 간 대역폭은 1Mbps, 링크 지연 시간은 1ms로 가정하였다. 호 설정을 위한 UA n과 m의 수는 700개로 설정 하였으며 프록시 서버의 큐는 100개의 유한 큐로 설정하였으며 큐의 용량을 초과하여 추가로 입력되는 SIP 메시지는 유실되는 것으로 가정하였다.

### 4.2 시뮬레이션 및 평가

시뮬레이터는 ns-2 2.33<sup>[10]</sup>을 사용하였으며 SIP 프로토콜은 Rui Prior<sup>[11]</sup>의 SIP모듈을 일부 수정하여 사용하였다. 그림 7은 큐의 용량을 확인하여 입력되는 메시지의 유실 여부를 결정하기 위한 Rui Prior의 DropTail 모듈을 상속하여 임계치  $l$ 과  $h$ 를 이용하여 호 설정을 요청하는 UA의 INVITE 메시지의 거부를 결정하기 위한 의사코드이다.

시뮬레이션을 위한 파라미터는 다음과 같이 가정하였다. 큐의 하한 임계치  $l$ 과 상한 임계값  $h$ 의 값은 0.3과 0.7로 설정하였으며, 큐에 입력되는 메시지가 하한 임계

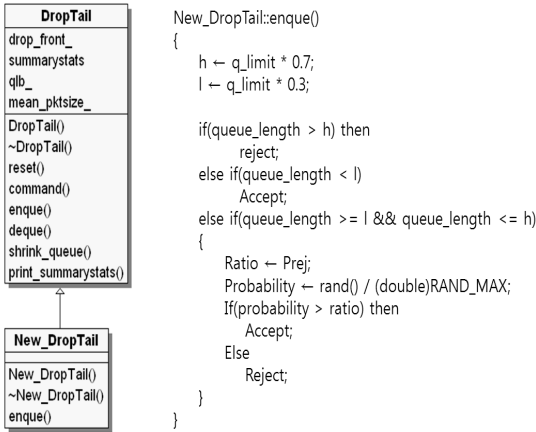


그림 7. 신규 호 요청 메시지의 수락 여부를 결정하기 위한 의사코드

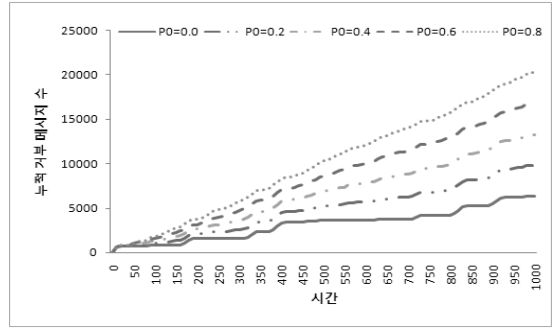
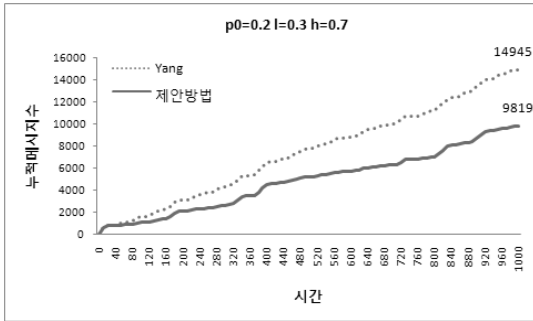


그림 9.  $P_0$ 의 변화에 따른 누적 거부 메시지

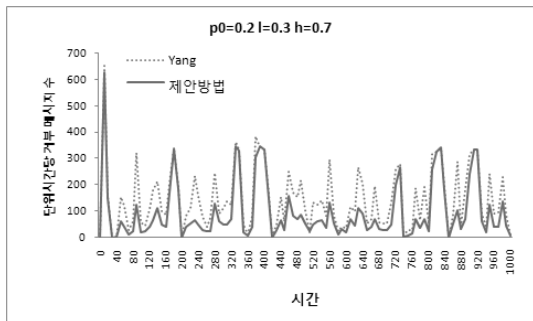
과부하를 유도하였다. SIP 프록시 서버의 과부하 상태에서 503 메시지에 의해 호 설정이 거부된 UA는 20초 동안 INVITE 메시지를 생성하지 못하도록 하였다. 시뮬레이션은 1,000초 동안 UA에 의해 생성된 43,000개의 INVITE 메시지에 대해 SIP 프록시 서버에 의해 거부된 INVITE 메시지 수를 측정하였다. INVITE 메시지의 거부는 UA의 호 설정이 거부되었음을 의미하며 잦은 호 설정 거부는 UA로 하여금 인터넷 전화에 좋지 못한 평가요소로 작용될 것이다.

그림 8의 (a)는 선형 제어 알고리즘의 경우 14,945개의 INVITE 메시지가 거부된 반면에 제안된 비선형 제어 알고리즘의 경우 9,819개의 INVITE 메시지가 거부되었음을 보여주고 있다. 이 따라서 제안된 알고리즘이 선형 제어 알고리즘에 비해 대략 12% 정도 더 많은 호를 수락(accept)했음을 알 수 있다. 그림 8의 (b)는 시간 경과에 따라 단위 시간당(40초) 거부된 메시지 수를 보여주고 있다. 이 결과에서 초기의 10초 사이에 두 알고리즘 모두에서 거부된 INVITE 수가 급증하는 것은 호 요청을 시작하고 10초 사이에 700개의 UA가 100개의 제한된 큐에 짧은 시간 내에 지수분포로 INVITE 메시지를 한꺼번에 전송하기 때문이다. 1,000초 시간동안 전체적으로 보면 제안된 알고리즘이 선형 제어 알고리즘과 비교하여 많은 구간에서 거부된 메시지의 수가 감소하였음을 알 수 있다. 또한 그림 8의 실험 결과에서 거부된 메시지의 수가 제안 알고리즘이 선형적 제어 알고리즘과 비교하여 12% 정도 적다는 것은 동일한 조건의 큐에 대하여 자원 이용률이 12% 가량 개선되었음을 의미한다.

본 연구에서는 제안된 알고리즘에 사용된 파라미터들이 거부율에 미치는 영향을 알아보고 최적값을 찾아내기 위하여 이들 파라미터들의 값을 달리하여 시뮬레이션하여 보았다. 그림 9은  $P_0$ 값의 변화에 따른 거부된 INVITE



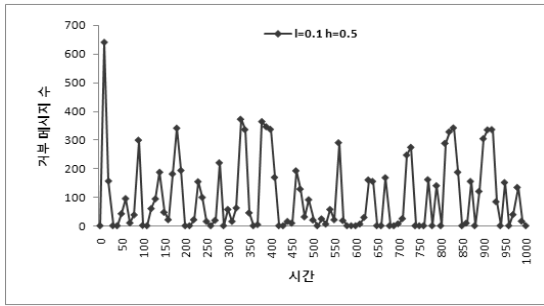
(a) 거부된 누적 메시지 수



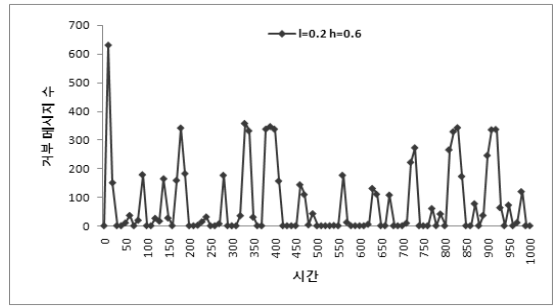
(b) 단위 시간당 거부된 메시지 수

그림 8. 과부하 상태에서 거부된 메시지 수

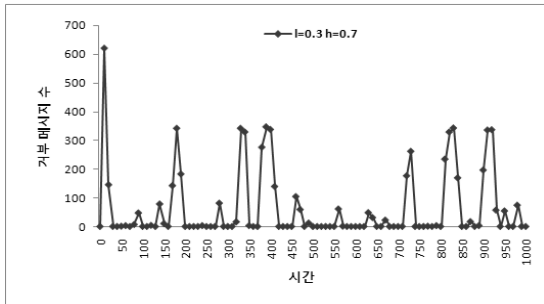
치  $l$ 에 도달 했을 시의 초기 거부율  $P_0$ 는 0.2로 설정하였다. 프록시 서버에 입력되는 UA의 INVITE 메시지는 지수 분포에 의해 생성되도록 하였으며 그 값은 10초로 설정하였으며 100개의 유한 큐를 가진 SIP 프록시 서버의



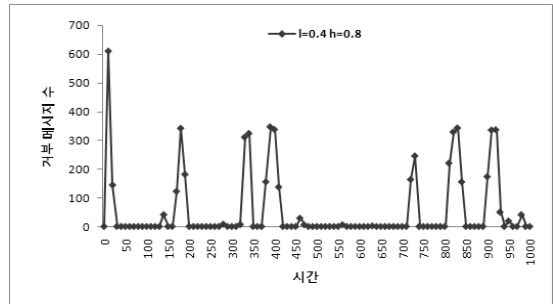
(a)  $l=0.1$   $h=0.5$



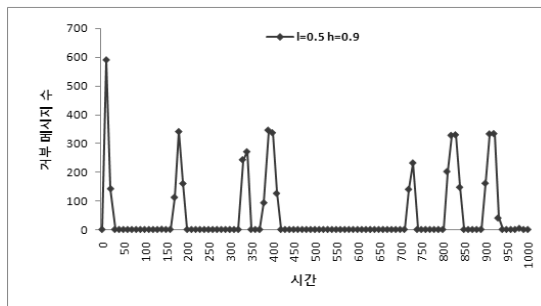
(b)  $l=0.2$   $h=0.6$



(c)  $l=0.3$   $h=0.7$



(d)  $l=0.4$   $h=0.8$



(e)  $l=0.5$   $h=0.9$

그림 10.  $l$ 과  $h$ 의 변화에 따른 거부 메시지의 변화

메시지의 누적 수를 나타낸 것이다.

$P_0$ 의 값이 0.8일 때 거부된 누적 메시지 수는 20,241 개이며  $P_0$ 의 값이 0.0일 경우 6,372의 메시지가 거부되었다. 프록시 서버에 INVITE 메시지가 거부되었다는 것은 거부된 수만큼 UA의 호 설정이 이루어지지 못했음을 의미한다. 그림 8의 실험 결과에서는  $P_0$ 의 값이 0.8인 경우가  $P_0$ 의 값이 0.0일 때보다 프록시 서버에 의해 13,869 개의 호 설정이 더 많이 거부되었음을 알 수 있다. 따라서 그림 9의 실험 결과를 통해  $P_0$ 의 값에 의해 프록시 서버의 자원 이용률이 다르게 나타남을 알 수 있으며,  $P_0$ 의

값이 적을수록 프록시 서버의 자원 이용률이 높음을 알 수 있다.

이어서 임계값  $l$ 과  $h$ 의 값이 서버의 자원 이용률에 미치는 영향을 알아보기 위하여 앞의 시뮬레이션 결과에서 가장 적은 거부된 메시지 수를 보여준  $P_0$ 의 값 0.0에 대하여 control state의 크기를 0.4로 고정된 상태에서  $l$ ,  $h$  값의 변화에 따른 거부 메시지 수를 알아보았다.

그림 10은 식 (1)의  $l$ 과  $h$ 값을 달리하여 1,000초 동안 프록시 서버에 의해 거부되는 UA의 INVITE 메시지를 나타낸 그래프이다.  $l$ 의 값이 0.1,  $h$ 의 값이 0.5인 (a)의

경우 모든 구간에서 많은 메시지가 거부되어짐을 알 수 있다. 반면  $l$ 의 값이 0.5,  $h$ 의 값이 0.9인 (e)의 경우 (a), (b), (c), (d)와 비교하여 많은 구간에서 거부된 INVITE 메시지의 수가 감소하였음을 알 수 있다.

## 5. 결 론

본 연구에서는 인터넷 멀티미디어 통신에서 송신 UA와 수신 UA 사이에서 호 설정을 위한 SIP 신호 처리를 수행하는 프록시 서버의 과부하 상태하에서의 자원 이용률을 증가시키기 위한 비선형 과부하 제어 알고리즘을 제시하였다. 시뮬레이션은 ns-2 시뮬레이터를 사용하였으며 제안 알고리즘의 평가를 위하여 앞서 Yang 등이 제안한 선형 과부하 제어 알고리즘과 비교하였다. 시뮬레이션 내용은 호 설정을 위해 SIP 프록시 서버에 요청된 INVITE 메시지 수와 과부하 제어를 위해 거부된 INVITE 메시지 수를 측정하여 평가하였다. 또한 시간 경과에 따른 단위 시간당 거부된 메시지 수를 측정하여 평가하였다. 그림 8의 실험 결과 (a)를 통해  $P_0$ ,  $l$ ,  $h$ 의 값이 동일할 때 제안된 비선형 제어 알고리즘이 선형 제어 알고리즘과 비교하여 호 요청 거부율에서 평균 12% 정도의 개선 효과를 보였으며 (b)의 시간 경과에 따른 거부 메시지의 변화를 통해 거부된 메시지의 수가 제안 알고리즘이 비교 알고리즘과 비교하여 많은 구간에서 적게 나타남을 알 수 있었다. 또한 과부하 상태에서의 파라미터 변화에 따른 성능을 비교하기 위하여  $P_0$ 의 값을 달리하여 실험하였다. 그림 9의 실험 결과를 통해  $P_0$ 의 값이 0일 때 가장 적은 메시지가 거부되었음을 알 수 있었으며  $P_0$ 의 값의 변화를 통해 프록시 서버의 자원 이용률을 높일 수 있음을 실험을 통해 알 수 있었다. 그리고 그림 9의 실험을 통해 가장 적은 거부 메시지를 보인  $P_0$ 의 값 0에 대한  $l$ 과  $h$ 의 변화에 따른 거부 메시지의 변화를 측정하기 위한 실험 10에서는  $l$ 이 0.5,  $h$ 가 0.9인 경우가 INVITE 메시지에 대한 거부율이 가장 적게 나타남을 확인하였다. 본 연구에서는 단일 큐에 대한 과부하 제어 방법만을 실험하였으나 향후 다중

큐에 대한 제안 알고리즘의 적용방법과 다양한 실험을 통하여 최적의  $l$ 값과  $h$ 값을 찾아야 할 것이다. 또한 평균 호 설정 시간과 같은 다양한 평가 측도들에 의한 제안 알고리즘의 성능을 평가하는 것이 필요하다.

## 참 고 문 헌

1. J.Rosenberg, et.al.: "SIP: Session Initiation Protocol," RFC3261, <http://www.ietf.org/rfc/rfc3261.txt>, June (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. M.Ohta: "Simulation study of SIP Signaling in an Overload Condition", 3rd Int'l Conf. on communications, Internet, and Information Technology, Nov. (2004), pp. 321-326.
5. 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, May (2003).
6. M. Ohta, "Overload Control in a SIP Signaling Network," Proc. of world academy of science, engineering and technology, vol. 12, 2006.
7. 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, 2009, pp. 3813-3816.
8. A. Johnston, Steve Donovan, Robert Sparks, Chris Cunningham, and Kevin Summers, "SIP Basic Call Flow Examples," RFC 3665, Internet Engineering Task Force (IETF), Dec. 2003.
9. V. Hilt, I. Widjaja, H. Schulzrinne "Session Initiation Protocol (SIP) Overload Control," IETF, Internet\_Draft, Draft-hilt-sipping-overload-06, 2009.
10. Network Research Group, Lawrence Berkeley National Laboratory. Network Simulator version 2 (ns-2). URL : <http://www.isi.edu/nsnam/ns/>
11. <http://www.dcc.fc.up.pt/~rprior/ns/index-en.html>



**이 종 민** (jongmin@deu.ac.kr)

1992 경북대학교 컴퓨터공학과 공학사  
1994 한국과학기술원 전산학과 공학석사  
2000 한국과학기술원 전자전산학과 공학박사  
1997~2002 삼성전자 무선사업부 책임연구원  
2005 Research associate, University of California, Santa Cruz  
2002~현재 동의대학교 컴퓨터소프트웨어공학과 부교수

관심분야 : 모바일 컴퓨팅, 라우팅, 센서네트워크, 소프트웨어공학



**전 흥 진** (dwit@deu.ac.kr)

1995 지산전문대학 전자계산학과  
1998 동서대학교 컴퓨터공학과 공학사  
2000 부경대학교 소프트웨어공학과 교육학석사  
2009~현재 동의대학교 컴퓨터소프트웨어공학과 박사과정

관심분야 : 컴퓨터 네트워크, 정보보호



**권 오 준** (ojkwon@deu.ac.kr)

1986 경북대학교 전자공학과 공학사  
1992 충남대학교 전산학과 이학석사  
1998 포항공대 전자계산학과 공학박사  
1986~2002 한국전자통신연구원 선임연구원  
2000~현재 동의대학교 컴퓨터소프트웨어공학과 부교수

관심분야 : 컴퓨터네트워크, 정보보호, 패턴인식, 인공지능경망