SIP 보안 프로토콜의 성능 분석

차 은 철[†] · 최 형 기[††]

요 약

최근 높아져 가는 VoIP에 대한 관심 뒤에는 보안에 대한 과제들이 존재한다. 인증과 메시지 암호화는 PSTN 수준의 보안을 제공하기 위해 필수적인 요소이다. SIP는 VoIP에서 안전하게 호(call)을 생성하기 위한 책임을 가지고 있으며 SIP는 TCP와 UDP 그리고 SCTP에서 사용될 수 있는 TLS와 DTLS 혹은 IPSec과 같은 보안 메커니즘들을 통해 보안 서비스를 제공한다. 이들 보안 메커니즘의 적용이 SIP 성능에 추가적인 오버헤드를 가져올 수 있음에도 불구하고 현재 이에 대한 분석은 미미한 수준이다. 본 논문에서는 보안 메커니즘이 SIP 성능에 미치는 영향을 분석하였다. 영향 분석을 위해 SIP에서 사용되는 보안 메커니즘들과 전송 프로토콜들의 다양한 조합을 시뮬레이션으로 구현하였다. 그 결과 UDP를 사용하는 보안 메커니즘들이 높은 성능을 보였다. 또한 TLS가 SCTP 상위에서 동작할 때 SCTP의 스트림 수와 같은 수의 보안 채널을 생성해야 한다는 사실이 SIP의 성능에 큰 영향을 미칠 수 있다는 것을 확인하였다.

키워드 : SIP, 세션생성지연, 네트워크 보안, 신호 프로토콜

Evaluation of Security Protocols for the Session Initiation Protocol

Eun-Chul Cha[†] · Hyoung-Kee Choi[††]

ABSTRACT

Behind the popularity of VoIP in these days, it may present significant security challenges in privacy and accounting. Authentication and message encryption are considered to be essential mechanisms in VoIP to be comparable to PSTN. SIP is responsible for setting up a secure call in VoIP. SIP employs TLS, DTLS or IPSec combined with TCP, UDP or SCTP as a security protocol in VoIP. These security mechanisms may introduce additional overheads into the SIP performance. However, this overhead has not been understood in detail by the community. In this paper we present the effect of the security protocol on the performance of SIP by comparing the call setup delays among security protocols. We implement a simulation of the various combinations of three security protocols and three transport layer protocols suggested for SIP. UDP with any combination of security protocols performs a lot better than the combination of TCP. TLS over SCTP may impose higher impact on the performance in average because TLS might have to open secure channels as the same number of streams in SCTP. The reasons for differences in the SIP performances are given.

Key Words : Session Initiation Protocol, Call setup delay, Network security, Signaling protocol

1. 서 론

음성 데이터를 IP(Internet Protocol) 망에서 전송하고자 하는 VoIP(Voice over IP)[1,2]의 개념이 처음 소개된 이 후로 이 기술을 실제로 도입하기 위한 지속적인 노력들이 있었다. VoIP가 성공적으로 도입 되기 위해서는 QoS(Quality of Service)와 보안[3] 등에서 전화망과 동일한 수준의 서비스가 제공되어야 한다. 그 동안 VoIP는 이러한 통화 품질과 보안에 대한 요구사항을 해결하지 못해 요금이 저렴하고 기반시설을 새로 구축할 필요가 없다는 장점들을 가지고 있음에도 불구하고 상업적으로 큰 성공을 거두지 못했다.

반면에 Skype는 이러한 VoIP의 당면 과제들을 해결하고 성공적인 서비스를 제공하고 있어 VoIP의 새로운 가능성을 보여 주고 있다. Skype는 P2P(Peer To Peer) 기반의 VoIP 응용 프로그램으로 2003년에 무료 인터넷 전화 서비스를 시작한 이후로 지속적으로 서비스를 확장해 전 세계에 8500만 명의 가입자를 확보하기에 이르렀다. 그러나 Skype는 표준을 사용하기 보다는 자체적인 프로토콜을 설계하고 사용하기 때문에 다른 VoIP 서비스들과 상호 연동이 어렵다. Skype와 달리 표준 프로토콜에 기반하는 VoIP 서비스들은 대부분 ITU T(ITU Telecommunication Standardization Sector)의 H.323 과 IETF(Internet Engineering Task Force)의 SIP(Session Initiation Protocol)[4][5]를 신호 프로토콜로

※ 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (IITA 2006 C1090 0603 0028)
† 정 회 원 : 성균관대학교 컴퓨터공학과 석사과정
†† 정 회 원 : 성균관대학교 정보통신공학부 컴퓨터공학과 교수
논문접수 : 2006년 9월 7일, 심사완료 : 2007년 1월 23일

채택하고 있다. 두 프로토콜 모두 VoIP 서비스에서 음성 데이터 세션의 생성을 위해 사용될 수 있지만 간편함과 유연성 때문에 SIP가 보다 더 유력한 VoIP 신호 프로토콜의 후보로 인식되고 있다. 또한 3GPP(Third Generation Partnership Program)가 SIP를 음성, 데이터 통합망을 위한 플랫폼인 IMS(IP Multimedia Subsystem)[6]의 신호 프로토콜로 선택하면서 SIP에 대한 관심은 더욱더 증가할 것으로 보인다.

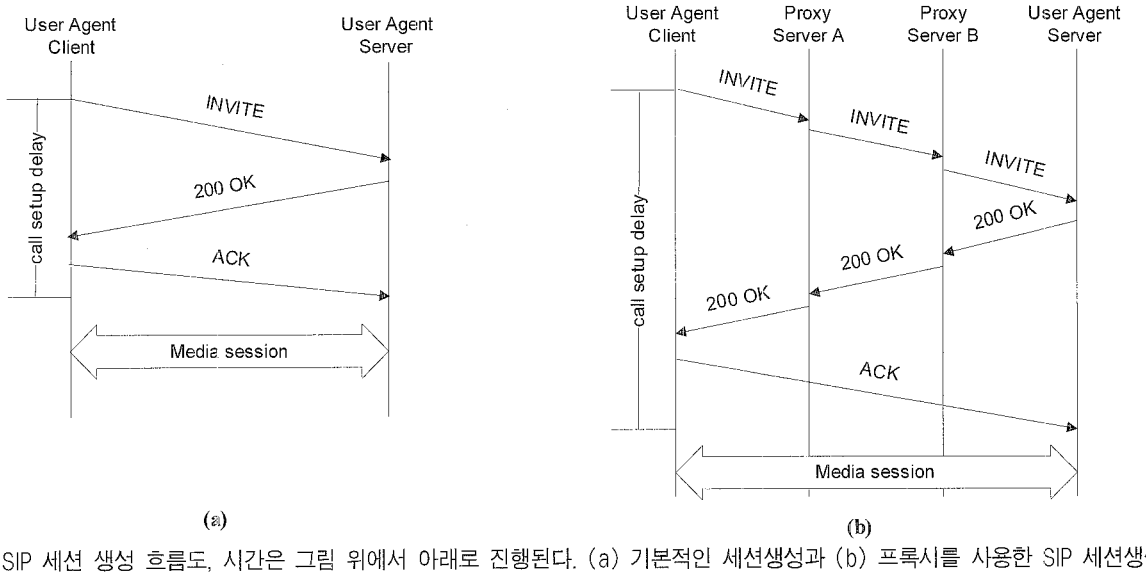
SIP는 멀티미디어 세션을 생성, 수정, 종결하는 응용 계층 프로토콜로 세션 생성을 요청하는 사용자 에이전트(UAC: User Agent Client)와 요청에 대한 응답을 수행하는 UAS(User Agent Server), 세션 요청을 다른 서버나 목적지로 중계해 주는 프록시로 구성된다. SIP는 HTTP(Hypertext Transfer Protocol) 와 유사한 텍스트 기반이며 전송계층과 독립된 응용계층 프로토콜이기 때문에 전송 프로토콜로 TCP(Transmission Control Protocol)와 SCTP(Stream Control Transmission Protocol) 같은 연결 지향(connection oriented) 프로토콜뿐 아니라 UDP(User Datagram Protocol) 와 같은 비연결지향(connectionless) 프로토콜 역시 선택할 수 있다.

공공망인 IP(Internet Protocol) 망에서 전송되는 SIP 메시지들은 포함된 정보들은 쉽게 도청과 변조가 가능하다. 또한 사용자 인증과 서버 인증을 제공하지 않을 경우 다른 사용자나 서버로 위장이 가능하여 과금과 프라이버시 보호에 문제를 야기시킬 수 있다. SIP 표준에서는 공격으로부터 SIP를 사용하는 시스템과 사용자를 보호하기 위해서 새로운 보안 프로토콜을 개발하기 보다는 기존의 보안 프로토콜들을 사용하기를 권고하고 있다[5]. 보안 프로토콜의 적용은 인증과 키 교환을 위한 추가적인 메시지 교환과 암호화로 인한 추가 지연을 통해 SIP의 성능에 영향을 미칠 수 있다. 그러므로 SIP의 성능에 대한 정확한 분석을 위해서는 SIP에 다양한 보안 프로토콜의 적용을 고려해서 분석이 되어야 한다.

본 논문에서는 TLS(Transport Layer Security)[13], DTLS (Datagram Transport Layer Security)[15] 와 IPSec(Internet Protocol Security)[16,18,19]과 같은 보안 프로토콜들을 SIP 프로토콜에 적용하고 SIP와 보안 프로토콜을 TCP/UDP/SCTP 과 같은 전송 프로토콜에서 운용했을 경우에 SIP 프로토콜의 성능을 측정 분석하는 연구를 수행한다. 보안 프로토콜들과 전송 프로토콜들의 조합을 비교하기 위해 SIP 세션을 생성할 때 걸리는 시간 즉, 세션생성 지연(call setup delay)을 사용하였다. 그림 1.a 와 1.b 는 두 개의 다른 네트워크 환경에서 SIP 세션생성 지연에 대한 정의를 그림으로 보여주고 있다. (그림 1.a) 는 SIP 클라이언트와 서버가 프록시를 사용하지 않고 직접 연결을 하는 경우이고 (그림 1.b) 는 두 개의 프록시들을 경유해서 SIP 세션을 생성하는 예를 보여주고 있다. SIP 에서의 서버와 클라이언트들은 이메일 주소와 유사한 URI 식별이 되는데, 서버의 IP 주소를 모르는 경우 클라이언트는 서버의 URI를 포함하는 메시지를 프록시에게 전송한다. 클라이언트의 프록시는 서버의 프록시를 통해 클라이언트의 메시지를 서버에게 전달한다.

SIP의 세션생성은 기본적으로 INVITE 메시지와 200 OK 메시지, ACK 메시지로 구성된 3방향 핸드쉐이크로 이루어진다. UAC가 UAS에게 INVITE로 세션 생성을 요청하면 UAS는 200 OK 메시지를 전송하여 요청에 대한 응답을 하며 UAC가 응답에 대한 확인 응답으로 ACK 메시지를 보낸 후에 음성 데이터가 전송되는 미디어 세션(Media session)이 서버와 클라이언트 간에 생성된다. 생성된 미디어 세션을 통해 음성 데이터는 프록시를 통하지 않고 클라이언트와 서버간에 직접 전송된다. 신호 트래픽과 미디어 세션이 분리되어 있으며 생성된 미디어 세션은 SIP 메시지를 통해 제어 할 수 있다. 본 논문에서 SIP의 세션생성 지연은 UAC가 INVITE 메시지를 보내는 시간에서 UAS가 ACK 메시지를 받는 시간의 차이로 정의한다.

본 논문에서 선택한 보안 프로토콜들과 전송 프로토콜의 조합은 TLS/TCP, DTLS/SCTP와 DTLS/UDP, TLS/SCTP,



(그림 1) SIP 세션 생성 흐름도, 시간은 그림 위에서 아래로 진행된다. (a) 기본적인 세션생성과 (b) 프록시를 사용한 SIP 세션생성

IPSec/UDP, IPSec/TCP, IPSec/SCTP이다. NS 2(Network Simulator 2)를 이용한 시뮬레이션을 통해 보안 프로토콜들의 적용이 세션생성 지연에 미치는 영향을 측정하고 분석한다.

시뮬레이션 결과 UDP를 사용하는 DTLS/UDP와 UDP/IPSec를 적용할 경우 네트워크 혼잡 상황에서 상대적으로 높은 성능을 보이는 것을 발견하였다. DTLS/UDP의 경우 초당 64 세션의 세션생성 요청이 발생할 때 TLS/TCP와 비해 96.1%, TLS/SCTP에 비해 97.3% 낮은 세션생성 지연을 보였다. DTLS/UDP와 IPSec/UDP가 높은 성능을 보여주지만 혼잡 제어의 부재로 인해 네트워크 혼잡상황에서 세션생성이 실패하는 확률이 높아진다. 또한 일반적으로 TLS나 IPSec에서 암호화와 메시지 인증에 의한 오버헤드는 큰 차이가 없지만 동작하는 계층의 차이로 인해 네트워크 혼잡상황에서는 TLS의 오버헤드가 더 커질 수 있다는 것을 확인하였다. 시뮬레이션 결과 세션생성 요청이 초당 42 세션 보다 작을 때는 TLS/TCP와 IPSec/TCP 사이에 거의 차이가 없지만 그 이상으로 세션생성 요청이 발생할 경우 네트워크 혼잡이 발생하면서 TLS/TCP가 더 큰 지연을 발생시켰다. 이 결과들을 통해 다양한 네트워크 환경에서 SIP에 보안을 적용하기 위한 기준점을 얻을 수 있다.

논문은 다음과 같이 구성되었다. 먼저 2장에서는 관련된 연구들을 소개하며 3장에서는 기본적인 SIP의 세션 설정과 보안 프로토콜을 적용했을 때의 세션 설정을 비교 분석한다. 4장에서 시뮬레이션의 구현과 시나리오에 대해 소개하고 5장에서는 그 결과를 분석한다. 마지막으로 6장에서 결론을 맺는다.

2. 관련연구

현재 SIP의 성능에 관련된 연구는 대부분 SIP의 메시지가 서버나 사용자의 장치에서 처리될 때 발생하는 처리 지연(processing delay)와 메시지가 네트워크에서 전송될 때 발생하는 전송 지연(transfer delay)에 관련되어 있다. 이에 반해 보안 프로토콜의 적용이 SIP의 성능에 미치는 영향을 분석하는 연구는 아직 미미한 수준이다.

2.1. 처리 지연에 대한 연구

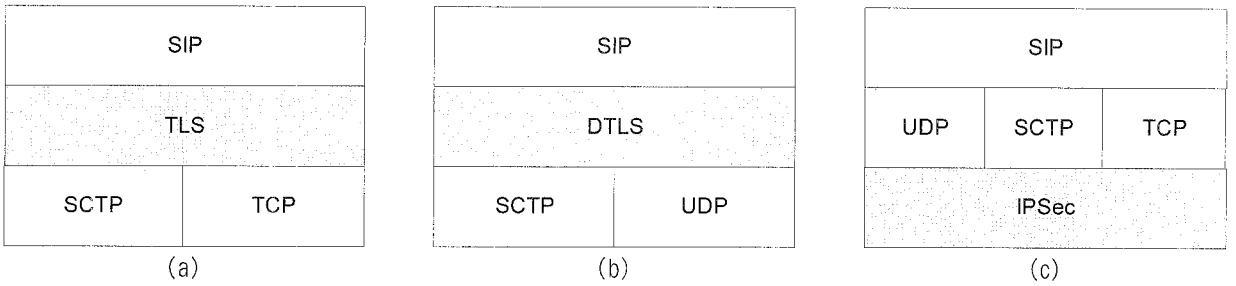
처리 지연은 메시지 파싱에 의한 지연과 같이 사용자 장치와 서버에서 발생하는 지연을 뜻한다. Swapna S. Gokhale과 Jijun Lu[11]는 LAN 환경에서의 SIP의 성능을 처리 지연에 초점을 맞춰 분석하였다. 이 논문에서는 SIP의 공개 Java API인 JAIN(Java APIs for Integrated Networks) 을 이용하여 테스트 베드를 구축하고 LAN 환경에서 사용자 장치의 프로세싱 부담과 SIP의 성능을 측정하였다. 실험 결과 두 사용자 장치 사이에 초당 12 세션 이상의 세션생성이 발생할 때 CPU의 이용률이 100%에 가까워지면서 세션생성 지연이 급격히 증가하였다. Stefano Salsano와 동료들[12]은 SIP의 인증 과정에서 나타나는 처리 지연을 분석하였다. 그들은 SIP에서 발생할 수 있는 인증 과정의 프로세싱 부담을 테스트베드를 통해 측정하였다. 결과로 인증 과정을 적용할 때 stateless 서버에서는 80%, stateful 서버에서는 45%의 프로세싱 비용이 증가하였다.

2.2. 전송 지연에 대한 연구

전송 지연는 네트워크에서 발생하는 transmission delay, propagation delay 등을 포함한다. Tony Eyers와 Henning Schulzrinne[9]은 SIP와 H.323의 세션생성 지연에 대한 연구를 전송 지연에 초점을 맞춰 진행하였다. 그들은 UDP 패킷의 지연과 손실에 대한 통계치를 제공하는 트레이스를 기반으로 SIP의 세션생성을 시뮬레이션 하였다. 시뮬레이션을 통해 대량의 에러가 발생할 경우 SIP가 H.323 보다 상대적으로 높은 성능을 보임을 발견하였다. 그러나 이 연구는 SIP가 UDP, TCP 그리고 SCTP와 같은 다양한 전송 프로토콜에서 사용가능 함에도 불구하고 SIP의 전송 프로토콜을 UDP로 국한시켜 정확한 분석이 어렵다는 문제점을 가지고 있다. Camarillo 등[10]은 UDP, TCP 그리고 SCTP에서의 SIP의 성능을 ns 2 시뮬레이터를 이용해 측정하였다. 그들은 SIP가 UDP에서 사용될 때 TCP나 SCTP와 같은 신뢰성 있는 프로토콜에서 사용될 때 보다 패킷 손실에 대한 감지가 늦다는 것을 발견하였다. 또 TCP에서 나타나는 HOL(Head Of Line) 차단현상을 SCTP의 멀티 스트림을 통해 해결할 수 있다는 사실을 시뮬레이션을 통해 증명하였다. 이 연구는 TCP, UDP 그리고 SCTP에서의 SIP의 성능을 보여주지만 프록시 사이의 트래픽에 초점을 맞추고 있으며 종단간의 지연에 대해서는 다루지 않고 있다. Hanane Fathi 등[7]은 전송 실패율이 상대적으로 높은 무선 환경에서의 SIP 세션생성 지연에 대한 연구를 수행하였다. 이 연구에서는 무선 채널을 표현하는 모델로 Gilbert Elliot 모델을 사용하였다. 이 논문에서 무선에서의 SIP의 세션생성 지연을 얻기 위해서 수치적인 계산 방법을 사용하였다. 연구 결과 TCP 대신 UDP를 사용했을 때 세션생성 지연이 30% 짧아졌으며 무선 링크에서 사용되는 ARQ(Automatic Repeat Request) 프로토콜인 RLP(Radio link protocol)를 사용하면 사용하지 않았을 때에 비해 세션생성 지연을 4초에서 5초 가량 줄일 수 있다는 것을 발견하였다.

3. SIP에 보안 프로토콜을 적용했을 때의 세션생성

(그림 2)는 SIP의 보안을 위해 사용될 수 있는 다양한 프로토콜 스택의 조합을 보여준다. (그림 2.a)에서 보듯이 TLS는 전송계층에 보안을 제공하면서 신뢰성 있는 전송 프로토콜(TCP, SCTP) 상위에서 동작한다 TLS는 TCP와 SCTP의 상위에서 동작하는 응용 프로그램에 안정적인 보안 서비스를 제공하지만 SIP에서 사용하기에 몇 가지 문제가 있다. 가장 큰 문제점 중 하나는 TLS는 UDP에서 사용할 수 없다는 것이다. SIP는 연결 생성 없이 빠른 전송이 가능한 UDP에서 동작하는 것이 유리한 경우가 많다. 또한 동시에 많은



(그림 2) SIP 보안의 프로토콜 계층구조. (a) TLS (b) DTLS (c) IPSec

사용자로부터 세션생성 요청을 받는 프록시의 경우 상대적으로 부하가 큰 TLS 연결을 동시에 유지해야 하는 문제점이 발생할 수도 있다. 또한 TLS의 전송 프로토콜로 SCTP를 선택할 경우에는 추가적인 문제들이 발생할 수도 있다. 그 중 가장 큰 문제 중 하나는 SCTP 스트림마다 TLS 연결을 유지해야 하는 것이다[14]. 예를 들어, 다수의 SIP 세션생성 메시지가 하나의 SCTP 연결에 다중화(multiplexing)되는 프록시와 프록시 사이에서는 HOL(Head Of Line) 차단현상을 방지하기 위해 각 SIP 세션의 메시지를 서로 다른 SCTP 스트림으로 전송한다. 이 경우에 수신 단에 도착한 TLS 메세지의 순서가 송신 단에서 보낸 순서와 다를 수가 있는데 TLS는 순서가 바뀐 메시지가 도착하면 공격의 한 형태로 재전송 된 것이라고 단정하여 메세지를 버린다[13]. 그러므로 TLS/SCTP를 사용할 경우 패킷 순서가 바뀌거나 패킷이 네트워크에서 유실되는 것을 허락하는 SCTP의 unordered delivery와 partial unreliability 를 사용할 수 없다. 그리고 모든 메시지를 TLS로 보호하기 위해서는 SCTP 스트림 수만큼의 TLS 연결을 생성하고 유지해야 하며[14], SIP 세션의 개수가 늘어나면 그에 비례해서 유지해야 하는 TLS 연결의 수도 증가한다. 이 제약 사항은 멀티 스트림을 사용하는 어플리케이션에 심각한 부담이 될 수 있다. TLS/SCTP 에서 발생하는 문제점에 대한 해결책으로 DTLS 를 SCTP 상에 보안 프로토콜로 운영하는 제안이 있다[16]. DTLS는 신뢰성이 없는 전송 프로토콜(UDP) 상에서 동작하도록 설계되었으므로 TLS와 달리 DTLS는 순서가 바뀐 패킷이 도착하는 것을 허락한다. DTLS는 패킷 순서가 바뀌는 것을 허가하기 때문에 스트림마다 DTLS 연결을 가질 필요가 없으며 SCTP가 가지는 unordered delivery와 partial unreliability와 같은 다양한 전송 형태를 사용할 수 있다.

(그림 2.b) 와 같이 UDP에서 동작하지 않는 TLS 대신 UDP에서 동작하는 응용 프로그램에 보안 서비스를 제공하기 위해 DTLS[15]가 제안되었다. DTLS 프로토콜은 대부분 TLS 프로토콜과 흡사하지만 UDP에서 효과적으로 동작하기 위해 수정이 되었는데 가장 큰 차이는 핸드쉐이크와 재전송 메커니즘에 있다. DTLS 핸드쉐이크에서는 TLS에서 나타나는 DoS 공격을 방어하기 위해 쿠키 (cookies) 교환기술을 사용한다. TLS에서 DoS 공격은 핸드쉐이크를 통해 상당한 수의 세션요청을 하고 공격대상이 세션요청에 따른 많은 상태정보를 유지하기 위해 자원을 고갈해야 되는 취약점을 노리는 것이다. 쿠키를 사용하게 되면 세션요청 상태정보를 시스템에서 유지할 필요 없게 된다. 클라이언트가 세션생성 요청을 하게 되면 서버는 클라이언트가 임의로 변경할 수 없는 세션상태 정보를 쿠키에 담아 클라이언트에 보낸다. 클라이언트가 받은 쿠키를 서버에 바로 전송함으로써 서버는 클라이언트의 IP 주소를 확신할 수 있고 세션상태 정보를 전송된 쿠키로부터 얻을 수 있게 된다. 이 쿠키기술을 통해 DTLS 는 TLS 보다 2개의 메시지가 더 많은 핸드쉐이크 프로토콜을 사용하나 UDP 에서 발생할 수 있는 DoS 공격을 무력화 시킬 수 있게 되었다. DTLS에서의 쿠키기술과 유사한 형태로 SCTP에서는 TCP SYN cookies 라고 하는 쿠키기술을 사용하고 있다. 또한, DTLS 에서는 핸드쉐이크 메세지가 네트워크에서 유실 될 수 있게 때문에 재전송 메커니즘이 필수적이다. 이를 위해 500ms 에서 1000ms 사이에 재전송 타이머를 두고 타이머가 만료될 때까지 응답이 없을 경우에 해당 메세지를 재전송하게 된다.

(그림 2.c)는 SIP에 IPSec[16,18,19]을 보안 프로토콜로 적용한 것이다. IPSec의 보안 서비스를 사용하기 위해서는 양단 간에 SA(Security Association)가 존재해야 한다. SA는 암호화를 위한 키와 암호화 알고리즘의 종류와 같은 정보를 포함하고 있다. SA의 생성을 위해서는 암호화와 메시지 인증을 위한 키를 교환해야 하는데 이 키들은 수동으로 분배되거나 동적으로 분배될 수 있다. 키를 동적으로 분배하기 위해서 UDP에서 동작하는 IKE(Internet Key Exchange)[20]와 같은 프로토콜이 사용된다. IPSec은 전송계층에 독립적이기 때문에 TCP, SCTP 그리고 UDP에서 모두 사용할 수 있다.

보안 프로토콜들의 적용이 SIP의 세션생성 지연에 영향을 주는 요소는 적어도 두 가지가 있다: 1) 보안 핸드쉐이크와 2) 메시지 보안이다. 첫 번째, 보안 핸드쉐이크는 보안 통신을 하기 위해 보안 채널을 생성하는 과정을 의미하며 인증과 키 생성을 위한 연산과 보안 채널을 생성하기 위한 메시지 교환으로 인한 지연이 발생한다. 두 번째, 메시지 보안은 보안 핸드쉐이크를 이용하여 보안 채널이 생성된 후 메시지에 기밀성과 메시지 인증 등을 제공하는 과정을 뜻하며 패킷 크기의 증가와 암호학적 연산으로 인한 지연이 발생한다.

보안 프로토콜들은 보안 핸드쉐이크(TLS/DTLS 핸드쉐이크, IKE)를 통해 상대방을 인증하고 암호화와 메시지 인증코드(MAC : Message Authentication Code)에 사용될 키와

〈표 1〉 보안 프로토콜들의 보안 핸드쉐이크 비교

	핸드쉐이크 메시지 수		전송 프로토콜
	완전한 핸드쉐이크	축약된 핸드쉐이크	
TLS 핸드쉐이크	4	3	TCP/SCTP
DTLS 핸드쉐이크	6	3	UDP/SCTP
IKE	9	3	UDP

보안 파라메터를 교환한다. <표 1>은 각 보안 프로토콜들의 핸드쉐이크에 대해 요약하고 있다. <표 1>에서 보듯이 각각의 보안 핸드쉐이크들은 완전한 핸드쉐이크(full handshake)와 축약된 핸드쉐이크(abbreviated handshake)의 두 종류의 핸드쉐이크가 있다. 최초에 보안 채널을 생성하기 위해서는 완전한 핸드쉐이크를 통해 인증과 키 교환을 수행해야 한다. 그 후에 전 세션의 상태 정보가 남아 있을 경우 새로 키를 생성하거나 보안 파라메터를 교환할 필요 없이 전 세션을 재개(resumption) 할 수도 있다. 이 과정에서는 완전한 핸드쉐이크 대신 축약된 핸드쉐이크를 사용한다. 예를 들어 TLS의 완전한 핸드쉐이크는 총 4개의 메시지를 교환하여 인증과 키 교환을 수행하지만 축약된 핸드쉐이크의 경우 인증 과정을 생략하고 총 3개의 메시지로 전의 보안 세션을 재개할 수 있다.

보안 핸드쉐이크를 통해 보안 채널이 생성되면 그 후에 보내지는 메시지에 기밀성 및 무결성을 보장 하기 위한 과정이 수행된다. 그 과정은 암호화 및 메시지 인증 코드 추가를 포함하며 이는 메시지 전송에 추가적인 지연를 유발시킬 수 있다. 암호화와 메시지 인증 코드 추가로 인한 세션 생성지연의 오버헤드는 메시지 크기 증가로 인해 발생될 수 있다. 예를 들어 SHA 1을 메시지 인증 알고리즘을 선택할 경우 TLS는 원래 메시지에 25 바이트(패딩 제외)를 추가하며 이는 네트워크에서 추가 지연을 유발한다. 처리 지연에 속하는 보안 핸드쉐이크에서의 인증과 키 교환에 따른 연산과 암호학적 연산에 따른 지연은 본 논문에서는 고려하지 않는다.

4. 시뮬레이션

SIP에 TLS와 DTLS 그리고 IPSec을 적용했을 때의 세션생성 지연에 대한 영향을 분석하기 위해 ns 2(Network Simulator 2)를 사용하여 시뮬레이션을 수행하였다. 시뮬레이션의 목적은 보안 메커니즘이 SIP의 세션 생성 지연에 미치는 영향을 확인하는 것이다. 3장에서 설명한 것과 같이 본 논문에서는 보안 메커니즘의 기능을 핸드 쉐이크와 메시지 보안으로 나누었다. 그러므로 시뮬레이션에서도 이 두 가지 요소에 초점을 맞추었다. 이 장에서는 시뮬레이션의 구현과 구성에 대해 살펴본다.

4.1. 구현

시뮬레이션을 위해 TLS, DTLS 그리고 IPSec의 ns 2 모듈을 C++로 구현하였다. SIP 프록시의 시뮬레이션을 위해서 기존에 존재하는 SIP 모듈을 수정하여 사용하였다. 원 모듈에는 프록시 기능을 제공하지 않아 프록시 기능을 하는 코드를 추가하였다.

TLS, DTLS 그리고 IPsec 이렇게 세 개의 보안 프로토콜 모듈은 공통적으로 3장에서 언급한 보안 핸드쉐이크와 메시지 보안의 두 가지에 중점을 두어 구현하였다. 구현된 모듈에서 상위계층(SIP 모듈)으로부터 메시지를 받으면 보안 채널이 생성되어 있는지를 확인하고 생성되어 있지 않다면 보안 핸드쉐이크를 수행한다. 보안 채널이 생성 된 후에 비로소 SIP 메시지를 전송한다. 이때 암호화와 메시지 인증 코드 삽입에 의한 메시지 처리 과정이 선행된다.

TLS 모듈은 TCP와 SCTP에서 동작하도록 구현하였다. ns 2에서는 다양한 종류의 TCP 모듈을 제공하는데 본 시뮬레이션에서는 SCTP와의 비교를 위해 SCTP와 유사한 TCP SACK 모듈을 선택하였다. 시뮬레이션에서는 현재 전달하고 있는 TLS 핸드쉐이크 메시지와 SIP 메시지의 종류를 알아야지만 프로토콜에 의한 다음 TLS 메시지와 SIP 메시지를 정확히 생성할 수 있게 된다. 이를 위해 구현된 TLS 모듈은 ns 2에서 제공되는 TcpApp 모듈을 참고하여 구현하였다. TcpApp 모듈은 TCP 모듈과 응용계층 모듈 사이에서 동작하며 응용 계층 메시지의 내용을 TCP에 전달하는 역할을 한다. TcpApp에 내재되어 있는 해당 기능을 추출해서 TLS 모듈에서 사용할 수 있도록 변환하여 구현하였다.

구현된 DTLS 모듈의 구성은 TLS 모듈과 거의 같으며 둘 사이의 차이는 핸드쉐이크 프로토콜과 재전송 메커니즘에 있다. 3장에서 언급한 것처럼 DTLS 핸드쉐이크는 UDP에서 발생할 수 있는 DoS 공격을 방지하기 위해 추가적인 2개의 메시지를 교환한다. 또한 UDP 상에서 보안 핸드쉐이크의 실패를 줄이기 위한 재전송 메커니즘의 구현이 필요하다. 이러한 모든 요소들은 구현된 DTLS 모듈에도 반영되어 있다. DTLS 핸드쉐이크의 구현은 2개의 추가 메시지를 제외하고는 TLS 핸드쉐이크의 구현을 따랐으며 각각의 핸드쉐이크 메시지를 전송할 때 재전송 타이머를 설정하도록 구현하였다.

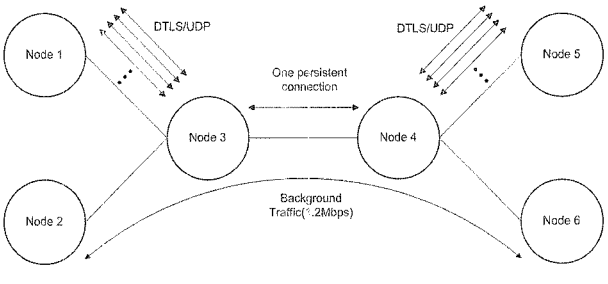
마지막으로 구현된 IPSec 모듈은 TCP, SCTP 그리고 UDP에서 동작하며 보안 핸드쉐이크를 위해 IKE의 메시지 교환을 구현하였다. UDP에서 사용하기 위해 DTLS와 마찬가지로 재전송 메커니즘을 구현하였다.

4.2. 시뮬레이션 시나리오

보안 프로토콜을 적용하는 SIP의 시뮬레이션을 위해 ns 2 상에서 UAC 역할을 하는 노드 1, UAS 역할을 하는 노드 5와 두 개의 노드를 연결하는 프록시들과 (노드 3과 4) 백그라운드 트래픽을 발생하는 노드 2과 6로 구성된 네트워크를 (그림 3)와 같이 설정하였다. 노드 1에서는 세션생성을 요청하고 노드 5에서 그에 대한 응답 메시지를 전송한다. 실제에서의 노드 1과 5는 프록시에 연결된 여러 사용자 에이전트들이 (UA) 모인 하나의 SIP 도메인을 의미하지만, 본

시뮬레이션에서는 하나의 노드가 여러 개의 DTLS/UDP 연결을 통해 각각 세션요청을 함으로서 다수의 사용자 에이전트 효과를 내고 있다. 따라서, 시뮬레이션에서는 노드 1에서 발생하는 모든 세션생성 요청과 그에 따른 응답은 각각 다른 사용자로부터 발생한다고 가정한다. 노드 2와 6은 프록시 사이의 링크의 사용률을 높은 상태로 유지하기 위해 1.2Mbps의 백그라운드 트래픽을 발생시킨다.

SIP에서 TLS와 DTLS 그리고 IPSec은 종단 간(end to end)의 보안 보다는 홉 간(hop by hop)의 보안을 위해 사용되며 시뮬레이션에서도 보안 프로토콜을 프록시 홉 단위로 적용하였다. (그림 3)에서 보듯이 시뮬레이션 상에서 노드 1과 5 사이에는 3개의 홉이 존재한다. 그 중 사용자 에이전트와 프록시 간의 홉(노드 1 3과 노드 4 5)에 적용되는 전송 프로토콜과 보안 프로토콜은 UDP/DTLS으로 고정하였다. 사용자 에이전트와 프록시 간에서 TCP와 SCTP와 같은 연결 지향의 전송 프로토콜을 사용하면 연결 설정에 따른 추가 지연이 발생하여 세션생성 지연을 증가시킨다. 또한 동시에 많은 유저가 세션 생성을 요청할 경우 모든 유저와의 연결을 유지 해야 하는 문제가 발생한다. 그러므로 사용자 에이전트와 프록시 사이에서는 빠른 세션 생성을 위해 전송계층에서 연결 설정을 하지 않는 UDP를 사용하는 경우가 대부분이다[5]. 본 시뮬레이션에서도 사용자 에이전트(노드 1과 5)와 프록시(노드 3과 4) 서버 사이의 전송 프로토콜로 UDP를 선택하였으며 보안 프로토콜은 자체적으로 사용자 인증이 가능한 DTLS를 선택하였다.



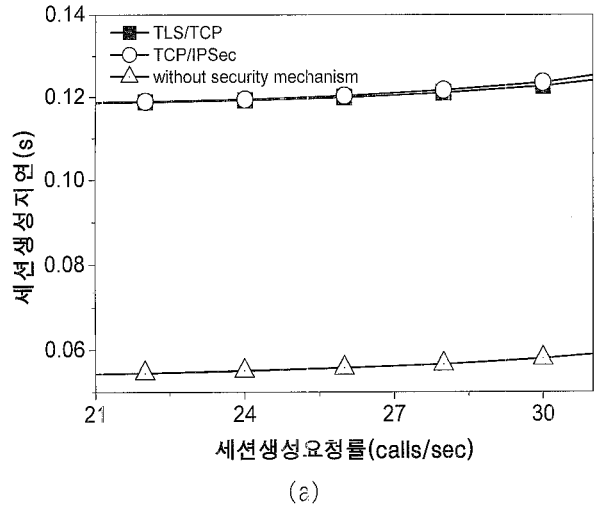
(그림 3) 시뮬레이션 시나리오의 네트워크 구성도

반면에 두 프록시 간(노드 3 4)의 홉에서는 한 개의 연결만이 존재하는 시뮬레이션을 구성하였다. 여러 사용자 에이전트로부터 발생하는 대량의 SIP 메시지는 노드 1에서 DTLS/UDP 연결을 통해 프록시 (노드 3) 으로 전송되고, 프록시와 프록시 사이에서는 한 개의 연결에 다수의 SIP 메세지가 다중화되어 (multiplexing) 전송된다고 가정하고 있다. 따라서 두 프록시 간(노드 3 4)의 홉에서는 네트워크 혼잡이 발생하여 전송 프로토콜과 보안 프로토콜에 따라 큰 성능 차를 보일 수 있다. 그러므로 본 시뮬레이션에서는 프록시 사이에 다양한 전송 프로토콜과 보안 프로토콜을 적용하여 종단간 세션생성 지연의 변화를 살펴본다. 프록시 사이에서 TCP와 SCTP 같은 연결 지향의 전송 프로토콜은 하나의 지속성 연결을 (persistent connection) 통해 모든 SIP 메시지가 다중화되어 전송된다고 본 시뮬레이션에서 가정한다. 프록시 사이에서 UDP 같은 비연결 지향의 전송 프로토콜을 사용할 경우에는 프록시 사이에 연결이 존재하지 않고 노드에서 발생되는 SIP 메세지들은 하나의 UDP 패킷에 실려 다른 프록시로 전송이 된다고 가정한다.

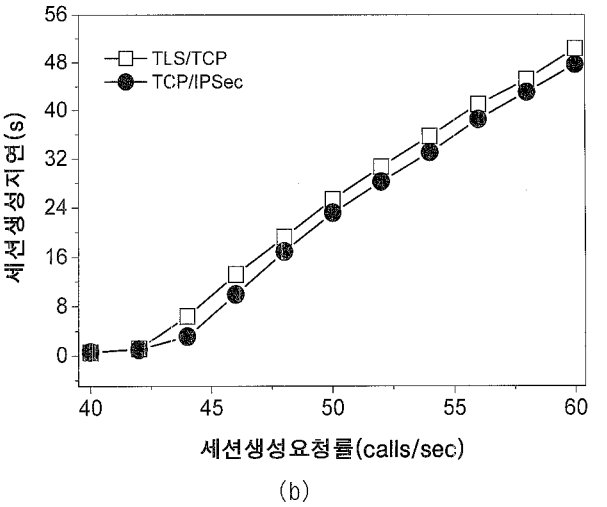
5. 시뮬레이션 결과

보안 프로토콜의 적용이 SIP 세션생성 지연에 주는 영향을 분석하기 위해 4장에서 시나리오의 시뮬레이션을 수행하였다.

시간당 발생하는 세션 생성 요청 수를 증가시키며 시뮬레이션을 반복적으로 수행하였다. 시뮬레이션 시간은 400초이며 그 시간 동안 생성된 세션들의 평균 세션생성 지연을 계산하였다. 이 시뮬레이션의 목적은 각각의 보안 메커니즘의 적



(a)



(b)

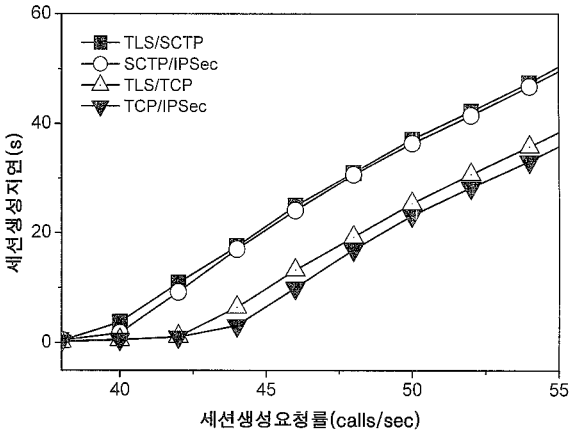
(그림 4) TCP에서 보안을 적용했을 때의 세션생성 지연. (a) 낮은 세션생성요청률에서의 세션생성 지연과 (b) 높은 세션생성요청률에서의 세션생성 지연

용과 전송 프로토콜들의 조합이 세션 생성 지연에 미치는 영향을 확인하기 위함이다. (그림 4)는 프록시 사이에 TLS/TCP와 TCP/IPSec을 적용하였을 때의 세션생성 지연을 보여준다. (그림 4.a)와 같이 세션생성요청률이 초당 30 세션보다 작을 때는 TLS/TCP와 TCP/IPSec 사이의 차이는 거의 보이지 않는다. 현재 시나리오 상에서 TLS와 IPSec의 차이는 암호화와 메시지 인증의 적용으로 인해 증가하는 패킷의 크기뿐이며 SHA 1을 메시지 인증 알고리즘으로 사용할 경우 TLS는 패킷을 25 바이트 증가시키며 ESP를 사용하는 IPSec은 36 바이트[1] 증가시킨다. 본 논문에서 사용한 시뮬레이션에서 SIP 메시지 크기의 평균은 458 바이트이다[21]. 그러므로 TLS와 IPSec의 적용에 의한 패킷 크기의 오버헤드는 각각 5.5%와 7.9%이며 네트워크 혼잡이 발생하지 않을 경우 지연 시간에 큰 영향을 주지 않는다. 또한 보안 프로토콜을 적용했을 경우가 그렇지 않았을 때에 비해 2배 이상 세션생성 지연을 발생시키는 것을 볼 수 있다. 이 오버헤드의 주 요인은 SIP 세션 생성마다 사용자 에이전트들((그림 3의 노드 1과 5)과 프록시((그림 3의 노드 3과 4) 사이에서 수행되는 각각 두 번의 DTLS 핸드쉐이크이다.

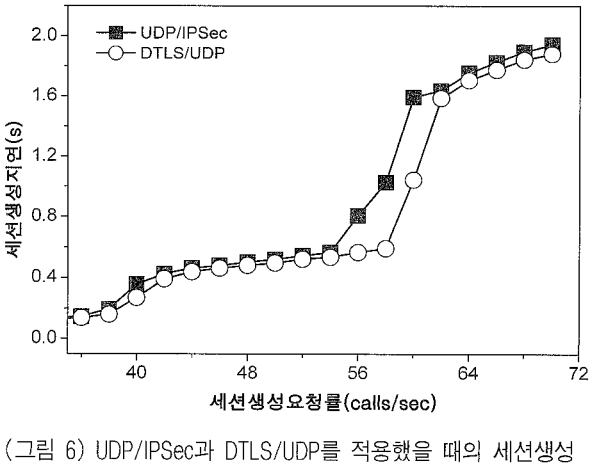
(그림 4.b)에서 볼 수 있듯이 세션 요청 수가 증가하면서 세션생성 지연은 빠르게 증가하며 IPSec과 TLS 사이의 성능 차이가 나타나기 시작한다. 패킷 증가의 오버헤드가 더 큰 IPSec이 TLS 보다 더 낮은 지연을 발생시키는 이유는 두 프로토콜이 동작하는 계층이 다르기 때문이다. 네트워크 계층에서 동작하는 IPSec의 경우 하나의 IP 패킷에 포함되는 SIP 메시지의 수에 상관없이 하나의 AH 헤더 혹은 ESP 헤더가 추가된다. 반면에 전송 계층과 응용계층 사이에서 동작하는 TLS는 두 개 이상의 SIP 메시지가 하나의 IP 패킷에 포함되면 SIP 메시지 각각에 메시지 인증 코드와 TLS 헤더가 붙은 후에 IP 패킷이 된다. 예를 들어 하나의 IP 패킷에 두 개의 SIP 메시지가 포함될 경우 증가하는 패킷 크기는 36 바이트이다. 그러나 TLS는 SIP 메시지 각각에 25 바이트의 오버헤드를 패킷에 추가시키므로 결과적으로 50 바이트의 오버헤드가 발생한다. 그러므로 세션 요청 수가 증가하고 네트워크 혼잡이 발생하면서 하나의 패킷에 포함되는 SIP 메시지가 많아지게 되면 이는 TLS의 오버헤드가 IPSec보다 커지게 된다.

(그림 5)는 (그림 4.b)의 TLS/TCP ,TCP/IPSec과 TLS/SCTP, IPSec/SCTP의 경우를 비교하여 보여준다. TLS/SCTP와 SCTP/IPSec은 초당 세션 생성 수 40 calls/sec가 될 때까지는 TLS/TCP, IPSec/TCP와 거의 차이를 보이지 않지만 초당 세션 생성 수가 그 이상 많아져 네트워크 혼잡이 발생하면서 차이가 생기기 시작한다. 세션생성 요청율이 46 calls/sec 일 때의 세션생성 지연은 TLS/TCP는 TLS/SCTP의 85.9%, 그리고 TCP/IPSec은 SCTP/IPSec의 83.4%로 측정되었다. 이는 SCTP가 TCP보다 헤더 크기가 더 크기 때문이다. TCP 헤더의 경우 옵션이 없을 경우 20 바이트 이다. SCTP는 12 바이트의 공통 헤더(common header)를 가지며 데이터 청크(data chunk) 마다 16 바이트의 청크 헤더(chunk header)를 가지므로 최소한 28 바이트의 헤더를 포함하게 된다.

(그림 6)은 프록시 사이에 UDP/IPSec과 DTLS/UDP를 적용했을 경우의 세션생성 지연을 보여준다. 결과로 UDP를 사용하였을 때 (그림 5)의 TCP에 비해 IPSec은 평균 3.1%, TLS는 평균 2.9%의 세션생성 지연(세션생성 요청율 64 calls/sec에서)을 가지며, SCTP에 비해서는 IPSec은 평균 2.2%, TLS는 평균 1.6%의 세션생성 지연(세션생성 요청율 64 calls/sec에서)을 유지하였다. 세션생성 지연은 UDP/IPSec의 경우 초당 54 세션 이후에, DTLS/UDP의 경우에는 초당 58 세션 이후에 급격하게 증가하는데 이는 응용 계층에서의 재전송 때문이다. UDP는 TCP와 SCTP와 달리 혼잡 제어를 하지 않는다. 혼잡에 의해 패킷 손실이 발생해도 같은 전송률을 유지하기 때문에 네트워크 혼잡 시에도 세션생성 지연이 크게 증가하지 않는다. 그러나 손실된 패킷을 재전



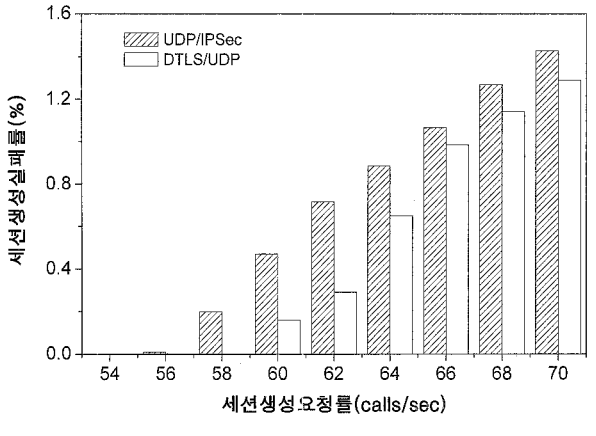
(그림 5) SCTP와 TCP 상에서의 보안을 적용한 세션생성 지연 비교



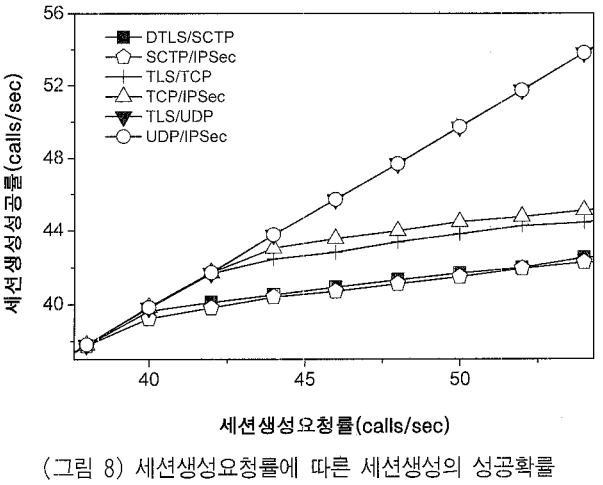
(그림 6) UDP/IPSec과 DTLS/UDP를 적용했을 때의 세션생성 지연

1) IPSec의 ESP transport 모드 기준

송하기 위한 방법이 없기 때문에 UDP에서의 패킷 재전송은 응용 계층에서의 재전송 메커니즘에 의존한다. 또한 TLS/TCP가 TCP/IPSec 보다 낮은 성능을 보였던 것과는 달리 DTLS/UDP는 UDP/IPSec에 비해 높은 성능을 유지하였다. UDP의 경우 하나의 IP 패킷은 하나의 DTLS 레코드만을 포함하므로[2] TLS에서 발생하는 추가적인 오버헤드가 발생하지 않는다. 이 경우 TLS에 비해 큰 IPSec의 패킷 증가 오버헤드가 그대로 적용되어 IPSec을 적용하였을 때 더 빨리 네트워크 혼잡이 발생하게 하는 요인이 된다. 세션생성 지연의 급격한 증가 이후에 증가하는 속도가 초당 62 세션과 64세션 이후에서 둔화하는 것은 재전송 횟수가 증가함에 따라 세션생성의 실패률도 증가하기 때문이다. SIP의 경우 7번의 재전송 후에 재전송을 포기하는데[1]UAC가 재전송을 포기하여 세션을 생성하지 못하는 경우를 실패한 세션생성으로 구분하였다. 세션생성에 실패한 세션은 세션생성 지연의 정의에 따라 평균에 포함시키지 않았다. 세션생성에 실패한 세션은 7 번의 재전송에 의해 잠재적으로 높은 세션생성 지연을 가지고 있으므로 이 세션들이 평균에 포함되지 않는 것은 세션생성 지연의 평균치를 감소시킨다.



(그림 7) 세션생성요청률의 증가에 따른 UDP에서의 세션생성실패



(그림 8) 세션생성요청률에 따른 세션생성의 성공확률

2) DTLS 표준[15]에서는 하나의 UDP 데이터그램에 여러 DTLS 레코드를 포함할 수 있다고 명시하고 있다. 그러나 일반적으로 하나의 UDP 데이터그램은 하나의 상위 계층 메시지만을 포함하므로 시뮬레이션에서는 하나의 UDP 데이터그램이 하나의 DTLS 레코드만을 포함하도록 구현하였다.

(그림 7)은 프록시 사이에 UDP/IPSec과 DTLS/UDP를 각각 적용하였을 때의 세션생성 실패율을 보여준다. TCP와 SCTP는 재전송만 있었을 뿐 요청된 모든 세션이 생성에 성공하였으므로 UDP만을 비교한다. 언급한 바와 같이 SIP는 7번의 재전송 후 세션 생성을 포기하며 세션 생성이 실패하게 된다. 그러므로 네트워크 혼잡의 과중은 많은 재전송으로 인한 세션 생성 실패률의 증가로 이어진다. 보안 프로토콜의 적용으로 인한 오버헤드는 네트워크 대역폭을 더 많이 소비하게 하여 네트워크 혼잡을 더욱 과중 시키며 결과적으로 세션 생성 실패할 확률이 높아지게 된다. (그림 7)에서 보는 것처럼 패킷 크기 증가에 의한 오버헤드가 더 큰 IPSec의 경우가 TLS에 비해 더 높은 세션 생성 실패율을 보인다. UDP가 낮은 세션생성 지연을 유지하게 해주지만 혼잡 제어의 부재로 인한 세션 생성 실패는 QoS에 부정적인 영향을 미칠 수 있다. 그러므로 UDP 상에서 혼잡 제어를 제공하는 DCCP(Datagram Congestion Control Protocol)[21]와 같은 프로토콜의 적용이 필요하다.

(그림 8)은 초당 세션 생성이 성공한 횟수를 보여주고 있다. TCP와 SCTP의 경우 초당 세션 요청의 수가 약 40 세션에 이를 때까지 세션생성 성공률은 선형적으로 증가하며 그 후 세션생성요청률이 초당 40 세션 이상으로 증가하여 네트워크 혼잡이 발생하면서 TCP와 SCTP의 혼잡 제어에 의해 증가하는 속도가 급격하게 감소하게 된다. UDP의 경우 혼잡에 상관 없이 선형적으로 증가하는 것을 볼 수 있다.

6. 결 론

SIP에서 중요한 두 가지 요구사항은 크게 1) 기존 전화망의 신호프로토콜 수준의 성능과 2) 과금과 개인 프라이버시 보호를 위해 정보보안를 제공하는 것이다. SIP에 보안 프로토콜을 적용할 때 발생하는 오버헤드 때문에 두 요구사항들이 상충되는 문제가 발생한다. 본 논문은 보안 프로토콜의 적용이 SIP의 성능 측정의 중요한 기준이 되는 세션생성 지연에 미치는 영향에 대한 연구를 수행하여 SIP에 보안 프로토콜을 적용하기 위한 기준점을 제시하고자 하였다.

본 논문에서는 SIP의 세션생성 지연과 보안 사이의 관계에 대한 연구를 전송 지연에 초점을 맞춰 수행하였다. 우리는 보안 프로토콜의 적용이 SIP에 영향의 미치는 요인을 두 가지(보안 핸드쉐이크, 보안 메시지 처리)로 구분하고 각각에 대한 영향을 분석하였다.

시뮬레이션 결과 보안 메시지 처리에 의한 오버헤드는 일반적인 네트워크 상황에서는 크게 영향을 미치지 않지만 네

트워크 혼잡 시 세션생성 지연을 증가시키는 것을 확인하였다. 또한 여러 SIP 메시지가 하나의 IP 패킷에 전송될 경우 4계층의 상위에서 동작하는 TLS가 3계층의 IPSec에 비해 더 많은 오버헤드를 가지는 것을 발견하였다. 보안 핸드쉐이크의 경우 추가적인 메시지 전송으로 인해 세션생성 지연에 직접적인 영향을 끼친다.

보안 프로토콜들은 다양한 전송 프로토콜 상에서 동작하는 것이 가능하기 때문에 전송 프로토콜과 보안 프로토콜의 조합에 따른 변화 역시 중요한 요소이다. 시뮬레이션 결과 UDP에서 동작하는 UDP/IPSec과 DTLS/UDP가 가장 좋은 성능을 보였다. UDP를 사용할 경우 네트워크 혼잡 상황에서도 전송률을 떨어뜨리지 않아 낮은 세션생성 지연을 유지하는 것을 확인하였다. 시뮬레이션 결과로 UDP를 사용할 경우 네트워크 혼잡 상황(세션생성 요청율 64 calls/sec)에서 TCP에 비해 IPSec은 평균 3.1%, TLS는 평균 2.9%의 세션생성 지연을 가지며, SCTP에 비해서는 IPSec은 평균 2.2%, TLS는 평균 1.6% 낮은 세션생성 지연 유지하였다. 그러나 혼잡 제어의 부재는 재전송 실패로 인한 세션 생성 실패률을 높이는 부작용을 가져온다. SCTP는 멀티 스트림과 멀티 호밍과 같이 신호 프로토콜을 전송하기에 적합한 특성을 가지고 있지만 비교적 큰 오버헤드로 인해 네트워크 혼잡 상황에서는 다른 프로토콜들에 비해 좋지 않은 성능을 보였다. 시뮬레이션 결과 SCTP를 사용할 경우의 세션생성 지연의 평균은 TCP의 1.2배로 나타났다.

본 연구에서는 처리 지연이 고려되지 않았다. 이 후의 연구에서는 보안 프로토콜의 처리 지연을 고려하여 보다 정확한 세션생성 지연을 측정할 것이다. 또한 DCCP를 적용하여 SIP/UDP의 낮은 지연을 유지하며 세션생성 실패를 줄이기 위한 연구를 수행할 예정이다.

참 고 문 헌

[1] P. Metha and S Ubani, "Voice over IP," IEEE Potentials Magazine, Vol.20, Iss.4, Oct., 2001.

[2] U. Varshney et al., "Voice over IP," Communications of the ACM, Vol.45, No.1, Jan., 2002.

[3] Thomas J. Walsh and D. Richard Kuhn, "Challenges in Securing Voice over IP," IEEE Security & Privacy Magazine, Vol.3, iss. 3, May, 2005.

[4] H. Schulzrinne and J. Rosenberg, "The Session Initiation Protocol : Internet Centric Signaling", IEEE Communications Magazine, Vol.38, Iss.4, Oct., 2001.

[5] J. Rosenberg et al., "SIP: Session Initiation Protocol," IETF RFC 3261, June, 2002.

[6] (IMS); Stage 2(Realease 7)," 3GPP specification, June 2006.

[7] H. Fathi et al., "On SIP Session Setup Delay for VoIP services Over Correlated Fading Channels," IEEE Trans. Veh. Technol., Vol.55, No.1, Jan. 2006.

[8] I. D. Curio and M. Lundan, "SIP Call Setup Delay in 3G Networks," in Proc. 7th IEEE ISCC '02, July, 2002.

[9] T. Eyers and H. Schulzrinne, "Predicting internet telephony call setup delay," in Proc. IP Telephony Workshop. Apr., 2000.

[10] G. Camarillo, R. Kantola, and H. Schulzrinne, "Evaluation of transport protocols for the session initiation protocol," IEEE Network, Vol.17, No.5, Oct., 2003.

[11] Swapna S. Gokhale., "Signaling Performance of SIP Based VoIP: A Measuremnet Based Approach," Globecom 05', Vol.2, Nov., 2005.

[12] S. Salsano, L. Veltri, and D.Papalilo, "SIP Security Issues: The SIP Authenication Procedure and its Processing Load," IEEE Networks, Vol.16, No.6, Dec., 2002.

[13] T. Dierks, C. Allen, "The Transport Layer Security (TLS) Protocol Version 1.1," IETF RFC 4346, Apr., 2006.

[14] A. Jungmaier, E. Rescorla, and M. Tuexen, "Transport Layer Security over Stream Control Transmission Protocol," IETF RFC 3436, Dec., 2002.

[15] E. Rescorla, and N. Modadugu, "Datagram Transport Layer Security," IETF RFC 4347, Apr., 2006.

[16] M. Tuexen, C. Hohendorf, and E. Rescorla, "Datagram Transport Layer Security for Stream Control Transmission Protocol," IETF Internet Draft <draft tuexen dtls for sctp 00.txt>, Aug., 2006.

[17] S. Kent, and K. Seo, "Security Architecture for the Internet Protocol," IETF RFC 4301, Dec., 2005.

[18] S. Kent, "IP Authentication Header," IETF RFC 4302. Dec., 2005.

[19] S. Kent, "IP Encapsulating Security Payload (ESP)," IETF RFC 4303. Dec., 2005.

[20] D. Harkins, and D. Carrel, "The Internet Key Exchange (IKE)," IETF RFC 2409. Nov., 1998.

[21] E. Kohler, M. Handley, and S. Floyd, „Datagram Congestion Control Protocol (DCCP)," IETF RFC 4340, Mar., 2006.

차 은 철

e-mail : iris1212@ece.skku.ac.kr

2005년 성균관대학교 정보통신공학부 (학사)

관심분야 : 인터넷 보안, VoIP

최 형 기

e-mail : hkchoi@ece.skku.ac.kr

1992년 성균관대학교 전자공학과 (공학사)

1996년 Polytechnique University 전기전자 (공학석사)

2001년 Georgia Institute of Technology 전기전자 (공학박사)

2001년 2004년 미국 Lancope. Inc. 연구원

2004년 2006년 성균관대학교 정보통신공학부 전임강사

2006년 현재 성균관대학교 정보통신공학부 조교수

관심분야 : 인터넷 보안, 모바일 커뮤니케이션 등