

IMS 망에서 세션 모니터링을 위한 비주얼 프로토콜 분석기

정회원 정인환*

A Visual Protocol Analyzer for Session Monitoring over IMS Network

In Hwan Jung* *Regular Member*

요 약

본 논문에서는 IMS 망에서 단말기간 세션 연결 정보를 그래프로 표현해주는 비주얼 프로토콜 분석기(VIPA: Visual IMS Protocol Analyzer)를 설계하고 구현한다. 구현된 프로토콜 분석기는 IMS 망에서 세션 설정을 위해 사용되는 SIP 프로토콜 정보를 패킷 단위로 수집하고 분석하여 세션 상태를 그래프로 표현함으로써 프로토콜 분석의 편의성을 제공한다. 기존의 프로토콜 분석기가 단말기 측면의 부분적인 세션 연결 상태만 보여주는 한계가 있었으나 구현된 프로토콜 분석기는 세션에 참여하는 모든 사용자 기기들 간의 세션 연결 상태를 확인할 수 있다.

Key Words : IMS, IP Multimedia Subsystem, pSIP, Protocol Analyzer

ABSTRACT

In this paper, we design and implement a visual IMS protocol analyzer, VIPA(Visual IMS Protocol Analyzer), providing a graphical view of session connection information between user equipments on IMS network. The VIPA captures SIP packets, which are used to setup sessions on IMS networks, and shows graphical view of session information to provide easy way to analyze IMS protocols. The existing protocol analyzer has limitation to show only terminal side protocol analysis information. Whereas the VIPA can provide not only terminal side but also server side analysis result so that the connection status between all the session participating user agents can be monitored.

1. 서 론

IMS(IP Multimedia Subsystem)는 IP(Internet Protocol) 기반으로 VoIP, 비디오, PoC(Port of Call), 프레즌스(Presence) 서비스, 컨퍼런스(Conference), 응급전화 서비스, 인스턴트 메시지(Instant Message) 등의 다양한 패킷 기반 서비스를 플랫폼에 상관없이 제공할 수 있는 망 구조로 최초 3GPP(3rd Generation

Partnership Project)^[1]에서 3세대 이동 통신망으로 제안된 후, 무선 통신망 뿐만 아니라 유/무선, 인터넷, TV 등의 멀티미디어 서비스를 통합하는 형태로 발전되었다^[2].

호 설정을 기반으로 하는 IMS 망에서 세션 제어 프로토콜로 사용되는 SIP(Session Initiation Protocol)^[4]는 IETF(Internet Engineering Task Force)^[3]에서 발표한 프로토콜로써, 단말 또는 사용자들 간 기존의

※ 본 연구는 한성대학교 교내 연구비 지원 과제임.

* 한성대학교 컴퓨터공학과(ihjung@hansung.ac.kr)

논문번호 : KICS2010-10-470, 접수일자 : 2010년 10월 4일, 최종논문접수일자 : 2011년 3월 2일

VoIP 서비스뿐만 아니라 다양한 멀티미디어 서비스를 위하여 사용자간의 세션 설정, 변경 및 종료룰 수행하는 응용계층의 프로토콜이다⁴⁾. 기존의 호 설정 프로토콜들이 아닌 SIP가 차세대 통신망을 위한 IMS의 기본 제어 프로토콜로 채택된 이유로는 SIP의 간결성과 확장성에 있다. SIP는 WWW(World Wide Web)에서 일반적으로 사용되는 HTTP(Hyper Text Transfer Protocol)처럼 텍스트 기반으로 되어있어 프로토콜에 대한 이해가 쉽고, 텍스트를 이용할 수 있는 모든 기기와 호 설정이 가능하다는 점에서 IP를 기반으로 플랫폼에 상관없이 다양한 기기와 멀티미디어 서비스를 통합하고자 하는 IMS의 미래와 부합한다.

이러한 IMS 망의 특징을 기반으로 전 세계 통신회사들은 차세대 통신망을 구축중이거나, 준비하고 있다. 이에 IMS와 관련한 많은 제품들이 연구되고, IMS와 관련한 사용자가 늘어남에 따라 IMS 망을 이해하고 시험, 관리를 위하여 패킷 분석을 통한 네트워크 모니터링 방법의 필요성이 대두된다.

일반적으로 네트워크를 모니터링하기 위한 방법으로는 능동적 방법과 수동적 방법으로 나눌 수 있다. 능동적 방법의 경우 임의의 시험 트래픽을 생성하여 네트워크에 부가하기 때문에 사용자가 원하는 형태의 종단 간 지연, 지터 및 손실률 등을 정확하게 파악할 수 있지만, 실제 사용자 트래픽이 아니기에 실제 트래픽을 반영하는 데에는 한계점을 지니고 있다. 이에 반하여 수동적 방법의 경우 사용자가 원하는 임의의 환경을 만들어 시험하지는 않지만, 실제 사용자의 트래픽을 기준으로 패킷을 분석하여 측정을 하기 때문에 능동적 방법의 단점을 보완하면서 실사용 환경에 맞는 트래픽을 모니터링 하여 실제 사용 환경에서의 성능, 문제점, 상태 등을 확인 할 수 있어 많이 이용되고 있다⁵⁾.

본 논문에서는 IMS 망에서 사용자가 망의 상태를 쉽게 이해하고, 관리 할 수 있는 수동적 모니터링 방법으로, IMS 망에서 호 설정 프로토콜로 이용되는 SIP 패킷들을 분석하여 그래프를 이용한 결과를 보여주는 비주얼 프로토콜 분석기를 설계 및 구현하였다.

본 논문의 구성은 다음과 같다. 2장 관련 연구에서는 IMS와 SIP의 구조와 기존 프로토콜 분석기와 관련된 연구들에 대하여 기술한다. 3장에서는 IMS망에서 세션 모니터링을 위한 비주얼 프로토콜 분석기를 설계하고 구현 내용의 세부사항에 대하여 기술한다. 4장은 구현한 프로토콜 분석기를 통하여 IMS 망에서의 SIP 패킷들을 분석하고 결과를 기술한다. 끝으로 5장에서는 결론 및 향후 연구에 대하여 기술한다.

II. 관련연구

2.1 IMS 통신망의 구조

IMS 통신망은 3GPP에 의해 최초 3세대 이동통신으로 제안되었으며, 현재 IP기반으로 기존의 유선, 무선, 인터넷 등을 통합하는 멀티미디어 서비스를 제공하기 위한 차세대 통신망으로 표준화를 진행하고 있다¹⁾.

그림 1은 IMS망 내 시스템 계층도이다. IMS는 어플리케이션 계층, 컨트롤 계층, 트랜스포트 계층으로 구성된다. 첫째, 어플리케이션 계층은 SIP(Session Initiation Protocol)기반의 오디오, 비디오, 채팅등과 같은 어플리케이션을 제공 해주는 여러 AS(Application Server)들로 구성되어 있다. 둘째, 컨트롤 계층은 SIP서비스를 제어하는 콜 세션 컨트롤 기능(CSCF : Call Session Control Function)과 사용자 정보 저장을 하는 HSS(Home Subscriber Server)등의 기능을 수행한다. 마지막으로 트랜스포트 계층에서는 IMS 네트워크를 다른 IMS 네트워크, 또는 PSTN 등의 기존 네트워크와 연결해주는 게이트웨이의 기능을 수행한다.

그림 2는 IMS망의 구성을 주요 기능을 이용하여 나타내고 있다. 이중 콜 세션 컨트롤 기능을 담당하는 CSCF는 수행하는 각 기능에 따라 P-CSCF (Proxy-CSCF), I-CSCF (Interrogating-CSCF), S-CSCF (Serving-CSCF)로 나뉘어져 있다. 각 CSCF들은 수행하는 기능에 따라 개념적으로 나뉘어져 있을 뿐, 그 기능들을 실제 각각의 서버를 통하여 제공해야하는 것은 아니다. HSS는 이동통신의 HLR(Home Location Register)에서 발전한 것으로 가입자 정보를 관리하는 중앙화된 데이터베이스이다. HSS는 사용자 등록/변경 관리, 인증, 권한 부여, 로케이션 정보, 세션 라우팅, 요금 부여 등의 기능을 수행한다.

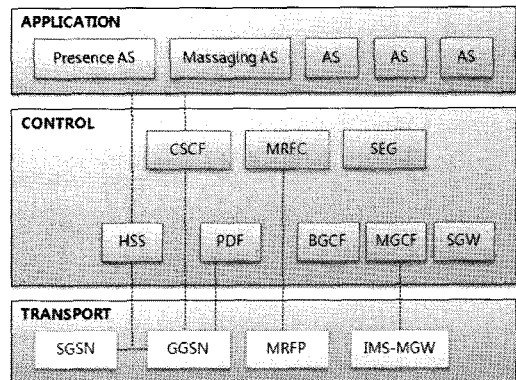


그림 1. IMS망 내 시스템 계층도

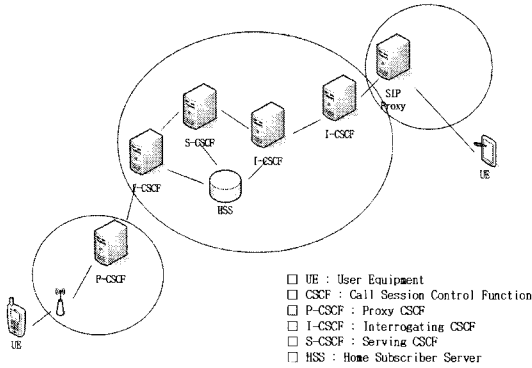


그림 2. IMS 망 구성도

2.2 SIP(Session Initiation Protocol)

SIP는 IETF에서 제안된 프로토콜로 클라이언트/서버 구조에 기반을 두고 사용이 간단한 텍스트 기반으로 사용자간의 상호 통신 세션을 초기화 하는 용도로 설계되었다⁴⁾.

그림 3은 SIP 메시지의 구조를 보여주고 있다. SIP 메시지는 시작 줄, 헤더, 메시지 본문, 그리고 헤더와 본문 사이의 CRLF로 구성되어 있다. SIP는 시작 줄과 헤더의 값만 기술하고, 메시지 본문은 세션 정보를 기술하는데 사용되는 프로토콜인 SDP¹⁵⁾가 이용된다. SIP 메시지 시작 줄의 내용에 따라 클라이언트에서 서버로 보내는 요청(Request) 메시지와 서버에서 클라이언트로 보내는 응답(Response) 메시지로 구분된다.

그림 4는 UA(User Agent)간의 SIP를 이용한 기본적인 호 설정 흐름을 보여주고 있다. Proxy Server는 각 UA들 간의 중계자 역할을 하게 된다. 호 설정은 UA1이 Proxy Server를 통하여 INVITE 요청 메시지

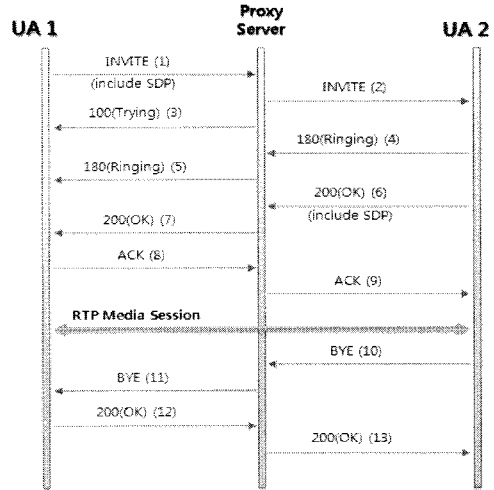


그림 4. SIP 호출 흐름

를 보내면서 시작된다. Proxy Server는 UA1에게는 INVITE 요청을 시도하고 있다는 100(Trying) 메시지를 보내고, UA2에게는 UA1으로부터 받은 INVITE 메시지를 전달한다. INVITE 메시지를 받은 UA2는 벨 소리를 울리며 180(Ringing) 응답을 보내고, 사용자의 응답에 따라 200(OK) 응답 메시지를 UA1로 전송하면서 호 설정은 이루어진다. 호 설정이 이루어진 후에 전송되는 미디어 데이터의 경우에는 Proxy Server를 이용하지 않고, UA1과 UA2가 직접 RTP Real-time Transport Protocol을 이용하여 통신 후, 종료를 원할 때 BYE 요청 메시지를 보냄으로써 세션이 종료를 하게 된다.

2.3 IMS에서의 SIP

IMS 통신망에서 호 설정에 사용되는 SIP는 기본적인 SIP 형식을 따르면서 IMS 통신망을 위해 확장시킨 Private Header를 이용한다. 표 1은 IMS를 위하여 확장된 대표적인 Private Header를 보여준다^{6,9)}.

그림 5는 IMS 망에서 SIP를 이용하여 사용자를 등록 하는 REGISTER 요청 메시지를 보내는 흐름을 보여주고 있다. REGISTER 메시지는 UE1(User Equipment)에서 P-CSCF1으로 보내어진 후, P-CSCF에서 Home Network에 있는 I-CSCF1으로 전달된다. 그리고는, 메시지를 보낸 UE의 인증을 위해 I-CSCF1은 REGISTER 메시지를 S-CSCF로 보내게 된다. 그림 6은 UE1에서 P-CSCF1으로 전달되는 REGISTER 메시지의 예이다.

그림 6의 UE1에서 P-CSCF1으로 보내는 REGISTER 메시지에서는 일반적인 SIP 기본 REGISTER 메시지

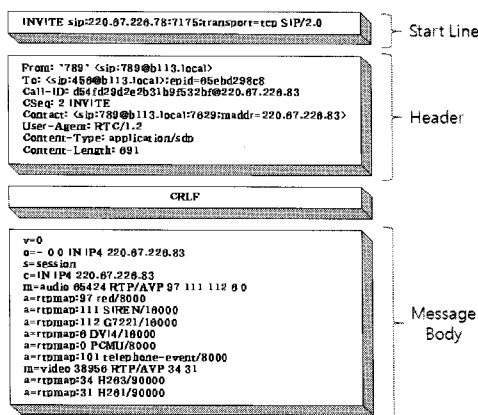


그림 3. SIP 메시지의 구조

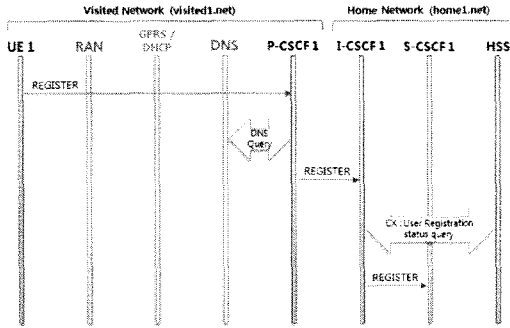


그림 5. IMS에서의 REGISTER 메시지 흐름

```
REGISTER sip:registrar.home1.net SIP/2.0
Via: SIP/2.0/UDP [5555:aaa:bbb:ccc:ddd]:comp=sigcomp;branch=z9hG4bKnashds7
Max-Forwards: 70
P-Access-Network-Info: 3GPP-UTRAN-TDD;utran-cell-id-3gpp=234151D0FCE11
From: <sip:user1_public1@home1.net>;tag=4fa3
To: <sip:user1_public1@home1.net>
Contact: <sip:[5555:aaa:bbb:ccc:ddd]:comp=sigcomp;expires=600000
Call-ID: apb03a09d9djkjgik49111
Authorization: Digest username="user1_private@home1.net", realm="registrar.home1.net",
nonce="", uri="sip:registrar.home1.net", response=""
Security-Client: ipsec-3gpp;alg= hmac-sha-1-96;spi-c=23456789;spi-s=12345678;
port-c=2468;port-s=1357
Require: sec-agree
Proxy-Require: sec-agree
CSeq: 1 REGISTER
Supported: path
Content-Length: 0
```

그림 6. From UE1 to P-CSCF1

표 1. 대표적인 Private Header (RFC3455)

Private Header	설 명
P-Associated-URI	저장된 Associated-URI 정보
P-Called-Party-ID	Call Filtering 등의 서비스를 위한 정보
P-Visited-Network-ID	Visited Network의 식별자
P-Access-Network-Info	사용자의 접속 망 정보
P-Charging-Function-Address	요금 부과 방법과 관련한 주소 정보
P-Charging-Vector	요금 부과를 위한 icid-value 정보

와 다른 Private Header인 P-Access-Network-Info가 존재한다. IMS의 경우 기존의 인터넷망을 이용한 VoIP처럼 하나의 망을 이용하는 것이 아닌, 유/무선 전화망, 인터넷 망 등을 모두 통합하는 형태이기 때문에 사용자(UE1)가 접속하고 있는 Visited Network의 종류 등의 망 정보를 알려줄 필요성이 있다. 그렇기에 P-Access-Network-Info 헤더를 통하여 사용자의 접속 망 정보를 REGISTER 메시지에 포함시켜 전달한다.

그림 7은 P-CSCF1에서 I-CSCF1로 보내어 지는 메시지를 보여준다. 그 중 P-Visited-Network-ID 헤더는 사용자가 속해있는 Visited Network의 식별자로, P-CSCF1에서 Home Network의 I-CSCF1로 메시지가

```
REGISTER sip:registrar.home1.net SIP/2.0
Via: SIP/2.0/UDP pccsf1.visited1.net;branch=z9hG4bK240f34.1
SIP/2.0/UDP [5555:aaa:bbb:ccc:ddd]:comp=sigcomp;branch=z9hG4bKnashds7
Max-Forwards: 69
P-Access-Network-Info: 3GPP-UTRAN-TDD;utran-cell-id-3gpp=234151D0FCE11
Path: < sip:term@pccsf1.visited1.net />
Require: path
P-Visited-Network-ID: "Visited Network Number 1"
P-Charging-Vector: icid-value="AyretyU0dm-602f75AFrbHlso=023551024
From: <sip:user1_public1@home1.net>;tag=4fa3
To: <sip:user1_public1@home1.net>
Contact: <sip:[5555:aaa:bbb:ccc:ddd]:comp=sigcomp;expires=600000
Call-ID: apb03a09d9djkjgik49111
Authorization: Digest username="user1_private@home1.net", realm="registrar.home1.net",
nonce="", uri="sip:registrar.home1.net", response="", integrity-protected="no"
CSeq: 1 REGISTER
Supported: path
Content-Length: 0
```

그림 7. From P-CSCF1 to I-CSCF1

전달되면서 같이 전달된다. 또한 P-Charging-Vector 헤더는 요금 부과를 위한 Icid-value(IMS Charging Identity - value) 정보를 포함한다.

그림 8은 UE와 Visited Network에 대한 정보와 함께 사용자 인증을 위하여 I-CSCF1에서 S-CSCF1로 전달되는 메시지를 보여준다.

REGISTER 요청 메시지와 같이 기존 VoIP용으로 설계되었던 SIP가 유/무선, 인터넷 등을 모두 통합하는 통신망인 IMS에 시그널링 프로토콜로 적용되면서 SIP는 텍스트 기반이라는 장점을 이용하여 3GPP의 IMS 통신망에서 필요로 하는 정보를 포함하도록 확장 되어 IMS 통신망에서 이용된다.

```
REGISTER sip:registrar.home1.net SIP/2.0
Via: SIP/2.0/UDP icscf1.phome1.net;branch=z9hG4bK351g45.1
SIP/2.0/UDP pccsf1.visited1.net;branch=z9hG4bK240f34.1
SIP/2.0/UDP [5555:aaa:bbb:ccc:ddd]:comp=sigcomp;branch=z9hG4bKnashds7
Max-Forwards: 68
P-Access-Network-Info: 3GPP-UTRAN-TDD;utran-cell-id-3gpp=234151D0FCE11
Path: < sip:term@pccsf1.visited1.net />
Require: path
P-Visited-Network-ID: "Visited Network Number 1"
P-Charging-Vector: icid-value="AyretyU0dm-602f75AFrbHlso=023551024
From: <sip:user1_public1@home1.net>;tag=4fa3
To: <sip:user1_public1@home1.net>
Contact: <sip:[5555:aaa:bbb:ccc:ddd]:comp=sigcomp;expires=600000
Call-ID: apb03a09d9djkjgik49111
Authorization: Digest username="user1_private@home1.net", realm="registrar.home1.net",
nonce="", uri="sip:registrar.home1.net", response="", integrity-protected="no"
CSeq: 1 REGISTER
Supported: path
Content-Length: 0
```

그림 8. From I-CSCF1 to S-CSCF1

2.4 기존의 프로토콜 분석기

기존의 프로토콜 분석기 중 SIP 패킷 분석을 지원 하는 대표적인 프로토콜 분석기로 공개 소프트웨어인 Wireshark^[10]가 있다. Wireshark는 작은 용량과 사용이 편리한 분석 기능을 갖는 범용 프로토콜 분석기이며 기업체에서 만든 상용 프로토콜 분석기인 Solomon^[18]등과 같은 분석기는 SIP 분석 기능이 부족하므로 Wireshark를 기준으로 기존 프로토콜 분석기의 기능과 부족한 점을 분석한다. 그림 9는 Wireshark의 실행 화면을 보여준다.

SIP 패킷 분석에 관한 Wireshark의 대표적인 기능

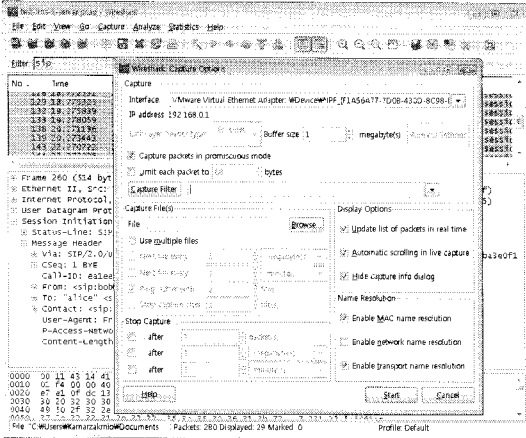


그림 9. Wireshark 실행 화면

은 다음과 같다.

(1) 실시간 패킷 수집 기능. 일반적인 프로토콜 분석기와 마찬가지로 실시간으로 SIP 패킷을 수집하는 기능을 가지고 있다. Ethernet Card를 통과하는 SIP 패킷을 수집하고, SIP 패킷 수집이 정지된 후에 일괄적으로 화면에 표시한다.

(2) SIP 분석 기능. 수집된 SIP, SDP, RTP 패킷들의 헤더를 분석하여 각 필드별로 설명하는 기능을 가지고 있다. 대표적인 필드로는 Request Method, State Code 등이 있다.

(3) SIP 패킷을 위한 추가의 필터 지정 기능. SIP 패킷을 위한 필터링 기능을 추가하여, 효과적인 필터링 기능을 강화하였다.

(4) 통계 기능. SIP 패킷에 대한 상태 코드를 통계 내는 기능이다. 세션에 사용된 패킷의 수와 상태 코드를 나열하는 기능을 가지고 있다. 이는 맺어진 세션의 수를 표시함으로써 수집된 패킷에 대해서 몇 개의 세션이 맺어졌는지 파악 할 수 있게 한다. 또한, RTP 패킷들만 분류하여 보여주는 기능을 포함하고 있다.

(5) 그래프 기능. SIP 패킷에 의해 세션이 맺어지게 되면 RTP 패킷에 의해서 음성 혹은 영상 데이터가 전송된다. 이렇게 전송되는 RTP 패킷들의 트랙픽을 세션이 맺어진 시점부터 종료시점까지 그래프로 보여준다. 그림 10은 Wireshark에서 SIP 세션을 그래프로 보여주는 화면이다.

Wireshark의 경우 SIP 패킷에 대하여 다양한 분석 기능과 통계 기능을 지원하지만, 일반적인 사용자 UA(User Agent) 기준으로 하였기 때문에 서버 입장에서 필요로 하는 정보인 각 UA들의 대한 상태 정보는 지원하지 않는다. 또한 일반적인 VoIP용도의 SIP

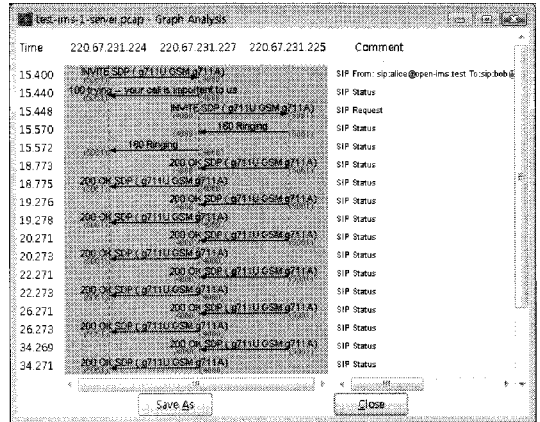


그림 10. SIP 세션 그래프 화면

를 기준으로 하였기 때문에 IMS에서 이용되는 Private Header들에 대해서도, 일반적인 Header들과 같이 동일하게 보여준다.

표 2는 Wireshark와 제안된 프로토콜 분석기를 비교하였다. 본 논문에서 구현한 프로토콜 분석기 VIPA (Visual IMS Protocol Analyzer)는 Wireshark와 같은 범용 프로토콜 분석기는 아니지만 IMS 망에서 Session 연결 정보를 서버 측면에서 분석하여 보여줌으로써 Session에 참가한 모든 UA의 상태를 추적하고 분석할 수 있는 기능을 장점을 가진다.

표 2. 프로토콜 분석기 비교

기능/분석기	Wireshark	VIPA
범용 프로토콜 분석	O	X
실시간 패킷 수집	O	O
SIP/SDP/RTP 분석	O	O
SIP 필터 지정	O	O
통계	O	O
그래프	O	O
UA 기준 분석	O	O
서버 기준 분석	X	O
Session 참여 모든 UA 도식화	X	O

III. 비주얼 프로토콜 분석기의 설계 및 구현

본 장에서는 관련 연구를 통하여 알아보았던 기존의 프로토콜 분석기의 분석을 통하여 IMS 기반의 프로토콜 분석기에서 필요로 하는 사항에 대해 정리하고, 그것을 기반으로 IMS 기반의 비주얼 프로토콜 분

석기를 설계 및 구현한다.

3.1 비주얼 프로토콜 분석기의 요구 사항

2장의 관련 연구를 통하여 분석 하였던 것을 토대로 IMS 기반의 비주얼 프로토콜 분석기가 지원해야 할 주요 요구 사항을 정리해 보면 3가지로 요약할 수 있다.

첫째, 기본적인 SIP 프로토콜 분석기로서의 기능에 충실해야 한다. 지정된 Ethernet Card를 통하는 모든 SIP 패킷들을 빠짐없이 수집하고, 실시간으로 분석된 결과를 사용자에게 보여주어야 한다.

둘째, IMS기반에서 필요로 하는 분석 기능을 지원해야 한다. IMS 망에서 이용되는 SIP의 Private Header들을 사용자가 한눈에 알아 볼 수 있도록 일반 Header들과는 다르게 표현해 주어야 한다. 또한, 일반적인 프로토콜 분석기와는 다르게 패킷 분석을 하는데 있어 P-CSCF를 패킷 수집 지점으로 하고 있기 때문에 지정된 P-CSCF를 통하여 REGISTER 과정을 거친 UE(User Equipment)들의 상태 정보를 사용자에게 분석하여 보여줄 수 있어야 한다. 마지막으로, 분석된 각 세션 별, UE 별로 이용된 패킷들을 분류하여 보여줄 수 있어야 한다.

셋째, 사용자가 쉽게 이해 할 수 있도록 모든 분석 결과를 그래프와 같은 비주얼한 형식으로 보여주는 기능을 지원해야 한다. 분석된 각 세션의 흐름만 그래프로 보여주는 것이 아닌, 일반 패킷과 분석된 UE의 상태 역시 그래프와 같은 형식으로 사용자가 쉽게 한눈에 이해할 수 있도록 비주얼한 분석 결과를 보여주어야 한다.

3.2 비주얼 프로토콜 분석기 설계

3.2.1 IMS 기반 환경

비주얼 프로토콜 분석기를 설계 및 구현하는데 기준으로 한 IMS망 환경은 오픈소스로 공개 되어있는 Open IMS Core^[11]를 이용하였다. 그림 11은 Open IMS Core의 구성도이다.

Open IMS Core는 가입자 정보를 관리하는 HSS, 그리고 콜 세션을 컨트롤 하는 I-CSCF, S-CSCF, P-CSCF의 기능을 지원한다. Linux를 기반으로 소스 파일, 또는 설치가 되어있는 VMWare 이미지 형식으로 공개되어 있다. 본 논문의 구현을 위해서 HSS와 CSCFs가 설치되어있는 VMWare 이미지를 이용하여 환경을 구성하였다. 표 3은 IMS 환경이 설치된 컴퓨터의 사양과 IMS 이미지에 할당된 사양을 나타낸다.

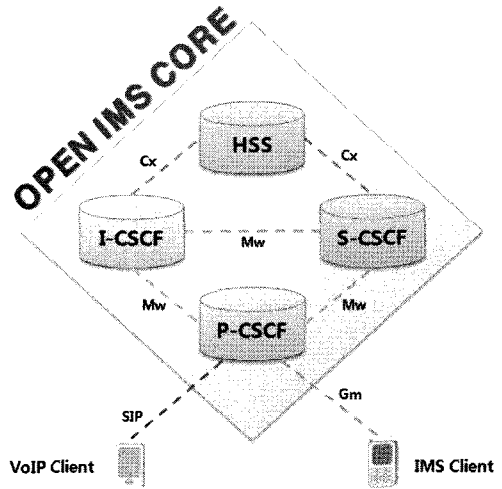


그림 11. Open IMS Core의 구성

표 3. 실험에 사용된 시스템 사양

구성	사양
VMWare가 설치된 컴퓨터	Intel Core2 6400@2.13GHz 2048MB RAM 250G HDD
Open IMS Core 이미지	1 Processor 512MB RAM 10G HDD

그림 12에서 보는 바와 같이 비주얼 프로토콜 분석기로 SIP 패킷을 분석하는 위치는 UE가 IMS망에 연결되는 관문 격인 P-CSCF에서의 패킷 분석을 기반으로 하고 있다. P-CSCF에서의 패킷 분석을 위해서는 Line상에 더미 허브나 탭 장비 등을 이용하여 패킷을 받아 분석할 수 있다. 하지만, 본 논문에서 이용한 IMS 망의 기능들이 모두 VMWare를 이용하여 서비스를 제공하기 때문에, 다른 장비를 이용하지 않고,

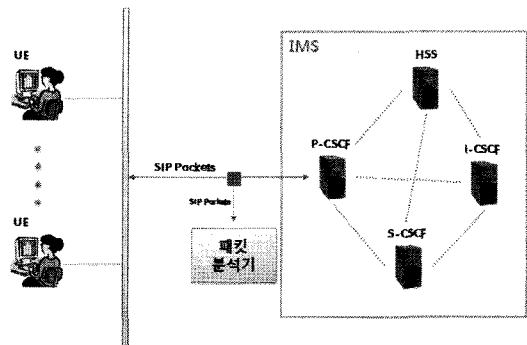


그림 12. 비주얼 프로토콜 분석기의 패킷 수집 위치

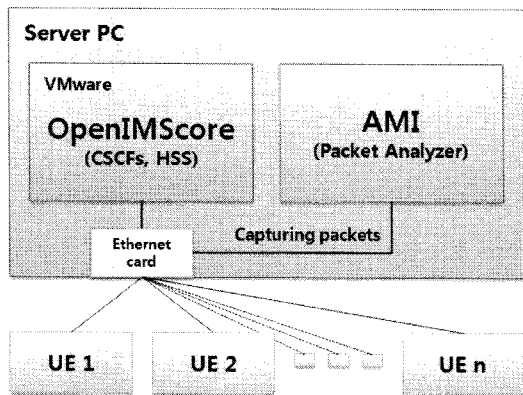


그림 13. 분석기의 패킷 수집 구조

VMWare와 윈도우 상에서의 Ethernet Card를 공유함으로써 P-CSCF로 전송되는 패킷들을 윈도우 상에서 수집한다. 그림 13은 Ethernet Card를 공유하여 패킷을 수집하는 구조를 나타낸다.

3.2.2 분석기 설계

비주얼 프로토콜 분석기의 주요 기능으로는 위의 요구사항에서와 같이 기본적인 SIP 패킷 수집 및 분석 기능, Session과 UE 별 분석기능, 그리고 마지막으로 분석된 결과를 비주얼한 화면으로 보여줄 수 있는 그래프 기능이 있다. 그림 14는 본 논문에서 설계한 프로토콜 분석기의 간단한 구조이다.

그림 14에서처럼 비주얼 프로토콜 분석기의 구조는 크게 4가지로 구분된다. 패킷을 수집하는 Packet Receiver, 수집된 패킷 또는 분석된 data를 저장할 저장 공간, 그리고 패킷 분석을 통하여 세션, UE등의 정보를 얻어내는 Analyzer, 마지막으로 분석된 데이터를 통하여 사용자에게 그래프를 그려주는 Painter 부분이다.

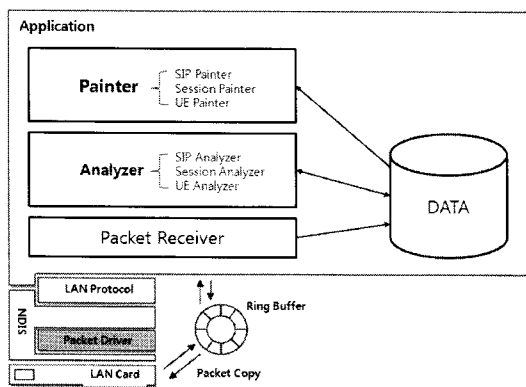


그림 14. 프로토콜 분석기 S/W 구조

Packet Receiver를 통하여 수집된 Packet을 pcap.Packet 클래스 형태로 저장하고, 저장된 패킷은 SIP Analyzer, Session Analyzer, UE Analyzer의 순서대로 Analyzer들을 거치면서 각각의 분석을 통하여 분석된 결과를 따로 저장한 후, SIP 패킷이 아닌 패킷은 버려지게 된다. 이후 사용자가 UI를 통하여 테이블에 나열되는 SIP 패킷, Session, 또는 UE 정보를 선택하게 되면, Painter를 통하여 각 결과에 맞는 그래프를 그려줌으로써 사용자에게 비주얼한 결과물을 보여주게 된다.

3.3 비주얼 프로토콜 분석기 구현

3.3.1 UI 화면

비주얼 프로토콜 분석기의 기본적인 UI 화면은 그림 15와 같다. 그림과 같이 UI는 크게 4부분으로 나뉜다. 1번 툴바에는 Ethernet 장치의 선택을 위한 환경 설정 버튼, 수집 및 분석 시작 버튼과 중지 버튼 등이 있으며, 2번 Progress Bar는 패킷 수집과 패킷 분석 스레드가 각각 실행되기 때문에 현재까지 분석된 패킷의 비율을 보여준다. 3번은 데이터 테이블 창으로 비주얼 프로토콜 분석기의 3가지 분석 기능인 패킷, 세션, UE 별로 분석된 결과를 테이블 형식으로 나타낸다. 4번은 3번의 데이터 테이블을 통해 선택된 데이터에 대한 자세한 사항을 보여준다. Contents 창을 통해서 선택된 SIP 패킷의 헤더들을 보여주고, Packets 창을 통해서 3번에서 세션이나 UE 데이터가 선택되었을 때 선택된 세션 및 UE와 관련한 패킷들을 테이블의 형태로 나열해 준다. 마지막 창인 5번의 패널을 통하여서는 선택된 패킷, 세션, 또는 UE 정

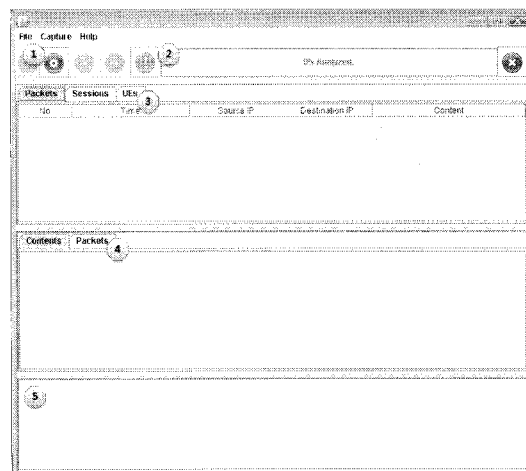


그림 15. 비주얼 프로토콜 분석기의 기본 UI

표 4. JPCAP 라이브러리 지원 클래스

클래스	설 명
ARPPacket	ARP 패킷 정보
DatalinkPacket	Datalink 레이어 정보
EthernetPacket	Ethernet 패킷 정보
ICMPPacket	ICMP 패킷 정보
IPPacket	IP 패킷 정보
IPv6Option	IPv6 관련 옵션 헤더 정보
JpcapCaptor	패킷 수집 or 파일로부터 패킷 읽음
JpcapSender	패킷 보냄
JpcapWriter	패킷을 파일로 저장
NetworkInterface	해당 기기의 네트워크 장치
NetworkInterface Address	네트워크 장치의 주소
Packet	수집된 패킷의 기본 형태
PacketReceiver	수집된 패킷을 실제로 읽어옴
TCPPacket	TCP 패킷 정보
UDPPacket	UDP 패킷 정보

보를 그래프의 형식으로 사용자에게 보여주어 비주얼한 결과물을 출력한다.

3.3.2 기본적인 패킷 수집 기능

비주얼 프로토콜 분석기는 기본적으로 SIP 패킷을 수집하고 분석하기 위한 기능을 지원한다. 모든 패킷을 분석하는 일반 프로토콜 분석기와는 다르게 SIP 패킷의 수집 및 분석만을 위한 기능만으로 구성되어 있다. 비주얼 프로토콜 분석기는 패킷 수집을 위하여 JAVA를 기반으로 한 RAW Packet 수집 공개 라이브러리인 jpcap^[12]을 이용한다. jpcap은 리눅스의 패킷 수집 라이브러리인 libcap^[13]을 기반으로 한 라이브러리로 표 3.2와 같은 클래스를 지원한다.

그림 16은 패킷을 수집하는 Packet Captor 클래스 관련 다이어그램이다. 메인 컨트롤러인 AMIController 클래스의 호출에 의해 AMIInformation 클래스에서 패킷을 수집할 Ethernet 장치의 정보를 받아, 수집 스레드를 시작하고, 멈추는 명령을 수행한다. AMI PacketCaptor 클래스의 startCaptureThread() 메소드 내에서 jpcap.PacketReceiver 인터페이스를 이용함으로써 해당 장치를 통하는 모든 패킷을 수집하여 jpcap.Packet 형태 그대로 AMIDataInformation 데이터 저장 클래스에 패킷을 저장하고, 패킷 분석 스레드가 실행되도록 관련 Flag의 값을 true로 변경한다. 기본적인 패킷 수집하는 기능의 경우 jpcap 라이브러

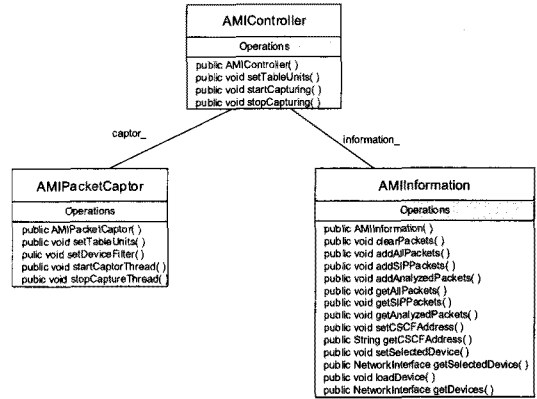


그림 16. 클래스 다이어그램

에서 패킷 수집에 관련한 클래스를 지원하기 때문에 간단한 인터페이스의 사용을 통해 패킷 수집을 수행할 수 있다.

3.3.3 패킷의 분석 기능

수집된 패킷을 분석하기 위하여 세 개의 Analyzer 클래스를 이용한다. 그림 17은 패킷을 분석하는 흐름을 보여준다.

PacketReceiver를 통하여 저장된 패킷이 들어오면 Analyzer 스레드가 시작되면서 패킷 분석을 시작한다. 먼저 패킷의 data부분을 분석하여 SIP 패킷인지 확인하고, SIP 패킷이 아니라면 패킷을 버린다. SIP 패킷

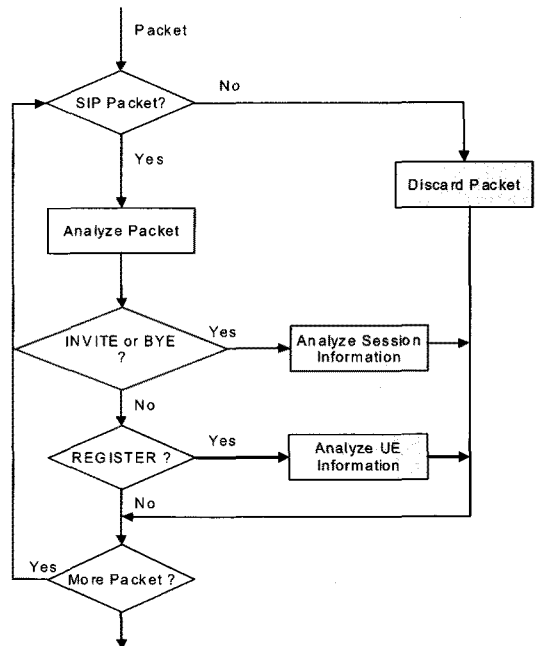


그림 17. 패킷 분석 흐름

이러면 패킷의 헤더를 분석한 후, 분석된 데이터를 토대로 Session Analyzer, UE Analyzer를 통하여 더 많은 분석하게 된다.

그림 18은 Session 분석을 하는 흐름을 보여준다. Session Analyzer에서는 우선 SIP 패킷이 200 OK 응답 메시지인지를 판단하게 된다. 200 OK 응답 메시지라면, 세션의 설정, 종료와 관련된 INVITE 또는 BYE 요청 메시지에 대한 응답인지를 판단한다. 만약 INVITE에 대한 응답 메시지라면, 저장된 세션 정보와 비교한 후 새로운 사용자간의 메시지인지 여부를 확인하여 세션 정보를 추가하게 된다. 세션 정보를 추가 할 때, INVITE 요청 메시지가 시작된 패킷의 고유순번을 같이 저장하여 각 세션에 이용된 패킷을 분류할 때 이용한다.

또는, SIP 패킷이 BYE 메시지에 대한 200 OK 응답이라면, 현재 저장된 세션 정보에서 세션 사용자 정보를 검사하여 현재 상태를 Disconnect 상태로 바꾸게 된다. 200 OK 응답 메시지의 경우뿐만 아니라, 한 사용자가 BYE 메시지를 보냈음에도 상대방의 응답이 없는 경우가 있기 때문에 BYE 요청 메시지가 5번 이상 누적 될 경우 세션 정보의 상태를 Disconnect 상태로 바꾸게 된다.

그림 19는 UE에 대한 정보를 분석하는 흐름을 나타낸다. UE Analyzer에서는 현재 SIP 패킷이 200 OK 응답 메시지의 여부를 확인한다. 200 OK 응답 메시지일 경우 CSeq 헤더 정보를 확인하여 REGISTER 요청 메시지의 응답인지를 확인하여 REGISTER 메시지의 응답일 경우 Contact 헤더의

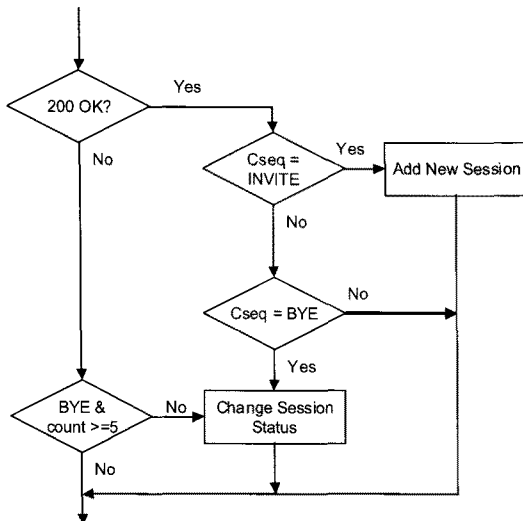


그림 18. Session 분석 흐름

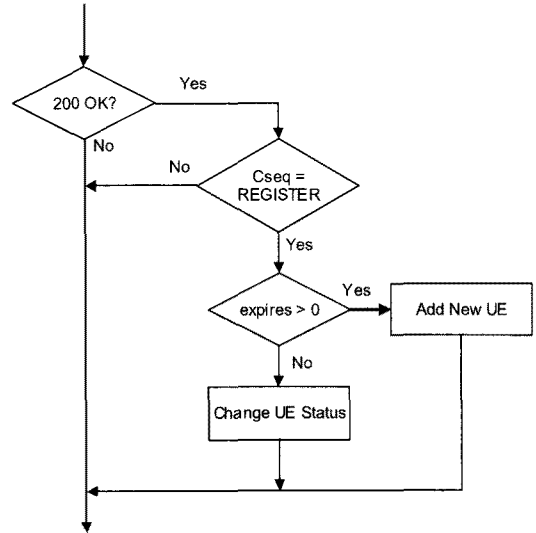


그림 19. UE 분석 흐름

expire값을 확인한다. expire값이 0보다 클 경우 UE가 P-CSCF에 최초 등록을 하거나, 또는 expire 시간을 연장하기 위한 요청이기 때문에 기존의 이미 동일한 사용자가 UE 테이블에 등록되어 있다면 사용자의 expire 시간을 새로 세팅하고, 동일한 사용자가 없다면, 새로운 UE로 등록을 하게 된다. 반대로 expire 값이 0일 경우엔 등록된 사용자가 접속을 끝내려는 메시지가기 때문에 UE 테이블을 확인하여 동일한 사용자의 상태를 Unregistered상태로 바꾸게 된다. 또한 위의 흐름에서 뿐만 아니라 일정 시간 마다 저장된 UE정보 내의 expire값을 확인하여 기한이 지난 UE의 경우는 상태를 Unregistered로 변경한다.

3.3.4 그래프 기능

UI는 기본 창의 폼을 위한 MainFrame 클래스와 각각의 테이블들을 관리하는 PacketTableModel, SessionTableModel, UETableModel, SubPacketTableModel과 각 테이블의 status 셀을 관리하는 PacketCellRenderer, SessionCellRenderer, UECellRenderer, 그리고 디바이스의 설정 창인 DeviceFrame, SIP 헤더 정보를 보여주는 ContentsPanel, 마지막으로 분석 결과를 그래프로 그려주는 VisualPanel, VisualData 클래스로 구성되어 있다.

그 중 VisualPanel을 통하여 실제 결과 값을 그래프 형태로 사용자에게 보여주게 된다. Packet, Session, UE 정보를 통하여 그래프에 사용되는 각각의 유닛과 연결선을 VisualData 클래스의 형태로 저장하고 VisualPanel에서 VisualData 클래스들을 실제 그리게

된다.

그림 20은 Packet을 분석한 그래프 화면이다. 수집된 패킷 리스트가 1번 테이블에 요청과 응답 메시지로 구분되어 표시된다. 테이블에서 선택된 패킷에 관한 헤더 정보가 2번 패널을 통하여 보이고, 3번과 같이 IMS에서 사용되는 Private Header의 경우 쉬운 구별을 위하여 색을 다르게 하여 표시된다. 4번의 패널에는 선택된 패킷 정보가 그래프로 그려진다. 위 그래프의 경우 caller인 cindy가 callee인 mike에게 보내는 메시지이다. 현재 선택된 패킷의 경우 cindy에게서 P-CSCF로 보내어지는 패킷이며, INVITE 요청 메시지 패킷이라는 것을 보여준다.

그림 21은 분석된 세션 정보를 그래프로 보여주는 화면이다. 1번 세션 테이블에서 세션을 선택하면, 선택된 세션에서 교환된 메시지 리스트가 2번의 서브

테이블을 통하여 나열된다. 그리고 선택된 세션의 설정이 시작된 시간부터 교환된 패킷의 흐름을 3번 패널을 통해 그래프로 보여준다. 시간 순에 따라 모든 패킷의 흐름 간단한 정보와 함께 보여준다. 또한 2번의 서브 테이블을 통해 선택된 패킷이 그래프에서 어느 곳에 위치하는 지를 4번에서처럼 색을 다르게 하여 표현함으로써 사용자의 이해를 돕는다.

그림 22는 UE의 상태를 분석한 결과를 그래프로 보여주는 화면이다. 1번의 UE 테이블을 통하여 선택된 UE와 관련한 모든 패킷들을 2번의 서브 테이블에 나열하여 보여준다. 그리고 3번의 패널을 통하여 선택된 UE에 대한 상태를 그래프로 보여준다. 위의 그림의 경우 선택된 cindy라는 사용자가 등록되어 있는 동안 mike와 tom이라는 사용자와 세션이 설정되었던 것을 보여준다. 또한 각 사용자와의 현재 상태를 connected와 disconnected로 나누어 다르게 보여줌으로써 세션의 현재 상태를 쉽게 확인할 수 있도록 하였다.

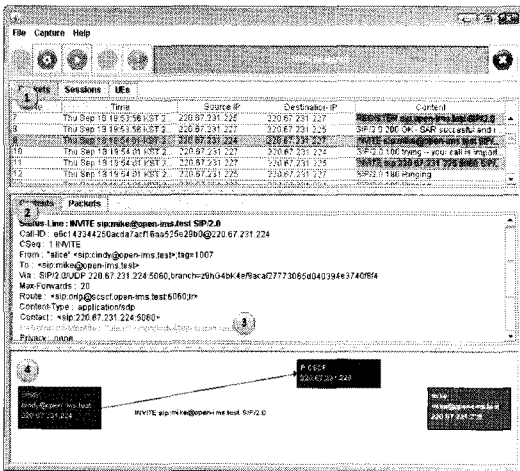


그림 20. Packet 분석 그래프

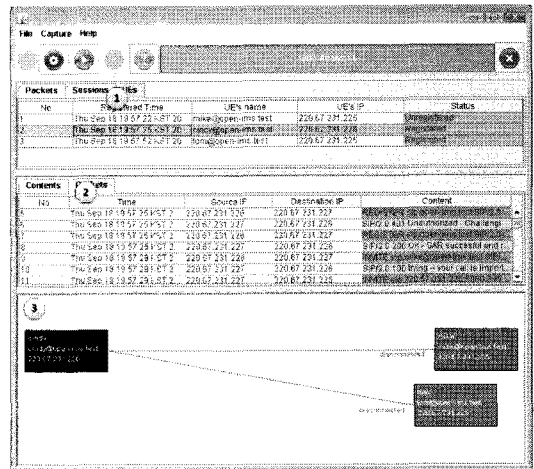


그림 22. UE 분석 그래프

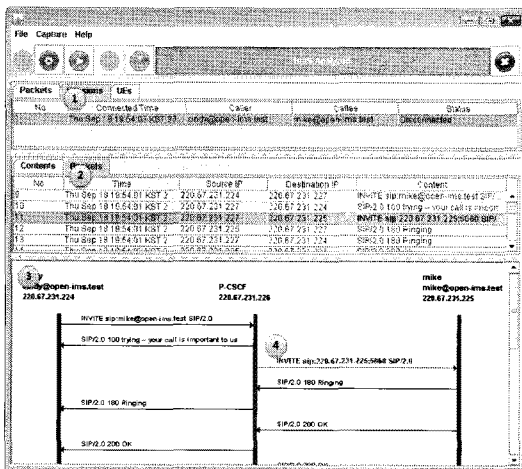


그림 21. Session 분석 그래프

IV. 실험

이 장에서는 앞 장에 기술했던 비주얼 프로토콜 분석기의 요구사항을 기준으로 실험한다. 8개의 UE들을 이용하여 등록/세션 연결 시나리오를 구성하고, 의도한 대로 세션과 UE 정보를 분석하고, 그 결과를 그래프로 보여주는가를 확인한다.

4.1 실험 방법

비주얼 프로토콜 분석기의 분석 및 그래프 기능을 실험하기 위하여 그림 23과 같은 접속 환경을 만들고

분석 결과를 확인한다. 8개의 UE를 등록시키고, 각 UE간의 세션 연결을 하여 분석된 결과와 비교를 한다. 그림에서의 실선은 현재 유지되고 있는 세션 또는 등록 상태를 뜻하고, 점선의 경우 현재 연결되어 있지 않은 상태를 나타낸다. 표 5는 그림 23의 각 UE별 상황을 나타낸다.

실험을 위한 UE로는 Fraunhofer FOKUS^[16]의 Open IMS Client Lite 버전 1.2를 이용한다. 그림 24는 Open IMS Client Lite의 실행화면이다. Open IMS Client Lite는 기본적인 VoIP기능을 가지고 있는 공개용 IMS Client로 Open IMS Core와의 호환성을 가지고 있다.

표 5. UE별 등록 및 세션 연결 상태

User Equipment	등록 상태	세션 연결 상태	
		대상	상태
Andy	등록	Andrew	종료
		Sally	연결
Andrew	종료	Andy	종료
Sally	등록	Andy	연결
		Tom	종료
		Mike	종료
Craig	종료		
Tom	등록	Sally	종료
		Mike	연결
Mike	등록	Sally	종료
		Tom	연결
Michelle	종료	Mike	종료
		Bill	종료
Bill	종료	Michelle	종료

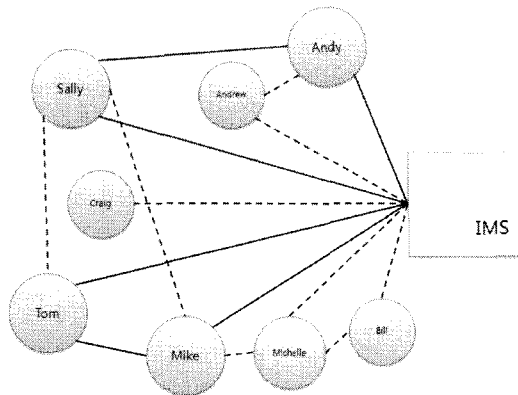


그림 23. 그래프 기능 실험을 위한 UE간의 상태

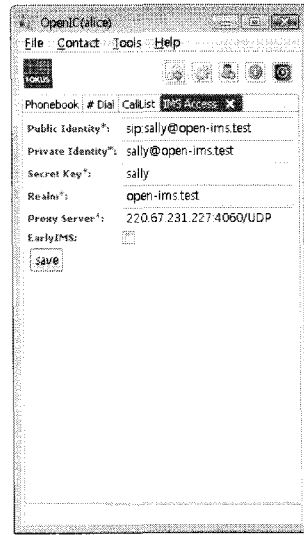


그림 24. Open IMS Client Lite

4.2 실험 결과 및 분석

실험 결과 확인은 분석된 정보가 방법에서 제시한 환경과 동일한가를 확인하고, 분석된 정보를 바탕으로 알기 쉽게 그래프를 그려주는가를 요점으로 하여 확인하였다.

그림 25는 패킷 분석 결과이다. 수집된 모든 패킷의 종류를 요청과 응답 메시지 별로 구분하여 보여주었다. 그래프의 경우 REGISTER 메시지는 최종 목적지가 다른 사용자가 아닌 P-CSCF이기 때문에 메시지를 보낸 사용자의 정보, P-CSCF, 그리고 메시지의 종류를 나타내었다.

그림 26는 시나리오에서의 세션 연결 부분을 나타낸 그림이고 그림 27은 세션 분석 결과이다. 그림 28

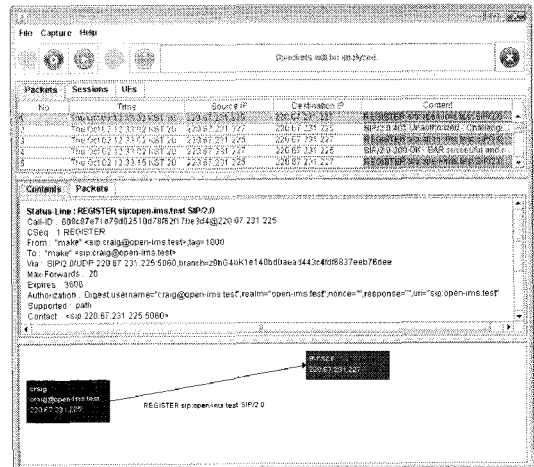


그림 25. 패킷 분석 실험 결과

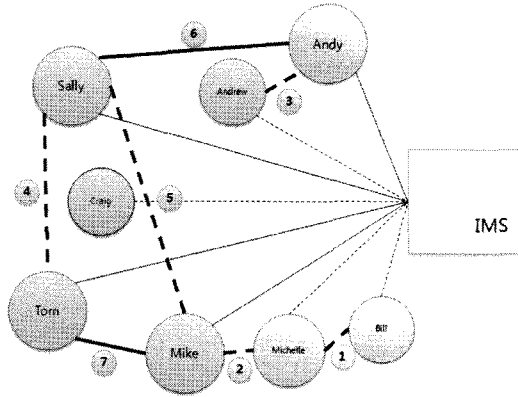


그림 26. 시나리오에서의 세션 연결

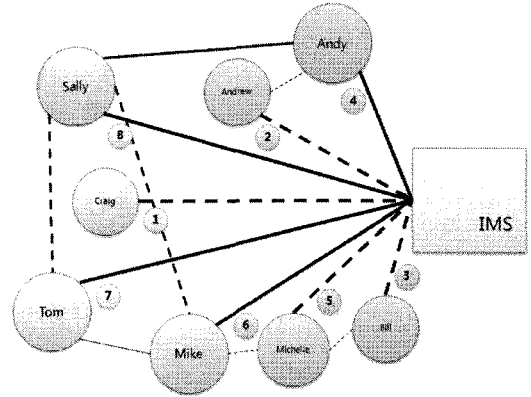


그림 28. 시나리오에서의 UE 등록

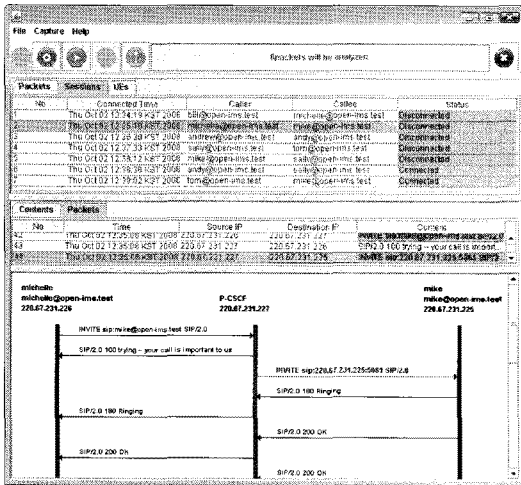


그림 27. Session 분석 실험 결과

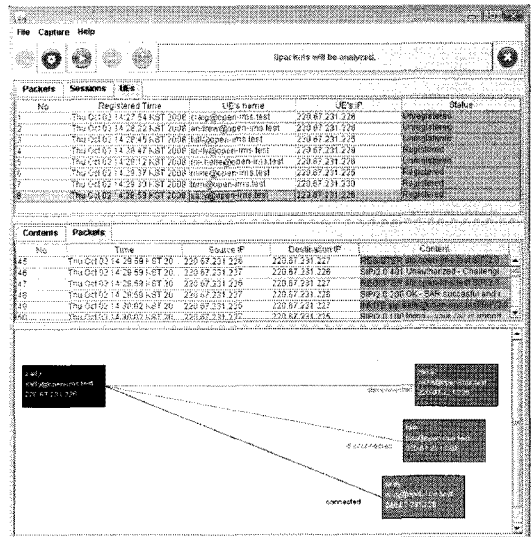


그림 29. UE 분석 실험 결과

과 같이 시나리오에서 설정하였던 환경대로 총 7개의 세션이 분석되었고 그 중 6번, 7번 두개의 세션을 연결 상태의 세션으로 분석하여 실험 시나리오와 같은 결과를 보여주었다. 또한 그래프에 있어서도 선택된 2번 세션에 대한 패킷들을 분석하여 세션의 흐름을 결과로 보여주었고, 서브 패킷 테이블을 통해 선택된 패킷에 대하여서도 그래프에 색을 다르게 표현하는 결과를 보여주었다.

그림 29는 그림 28 환경을 토대로 한 UE 분석 결과이다. 그림 28의 시나리오에서 설정한 것과 같이 총 8개의 UE가 등록되었고, 그 중 Andy, Mike, Tom, Sally만 현재 등록되어 있는 상태라는 결과를 보여주었다. 그리고 8번 UE인 Sally를 선택했을 때 Sally와 세션을 설정하였던 Mike, Tom, Andy를 그래프로 보여주고, 실험 설정과 같이 Andy에 대해서만 connect 상태로 구분하는 결과를 보여주었다.

V. 결론 및 향후 연구

본 논문에서는 IMS를 기반으로 한 비주얼 프로토콜 분석기를 설계 및 구현하였다. 비주얼 프로토콜 분석기는 차세대 통신망의 기준으로써 많은 사람들의 관심을 받고 있는 IMS를 기반으로 네트워크 모니터링을 수행하는데 있어 도움을 제공한다. 사용자에게 비주얼한 결과를 보여주어 편의성을 제공하고 기존의 프로토콜 분석기의 패킷 수집 기준 위치와는 다르게 서버 위치에서의 패킷 수집을 지원하여 UE(User Equipment) 상태를 분석하는 비주얼 프로토콜 분석기를 구현하였다.

본 논문의 비주얼 프로토콜 분석기는 실험을 토대로 SIP패킷 분석을 통한 패킷, 세션, UE 분석 기능을

확인하여, IMS망을 사용하는 사용자들에게 현 네트워크 상태를 모니터링하고 쉽게 분석할 수 있도록 하였다.

본 논문에 이은 향후 연구로는 네트워크 모니터링 도구로서의 성능 평가, 문제점 분석 등의 기능 추가하고, 기본 UE로 이용했던 VoIP 프로그램뿐만 아니라 Application Server, Presence Service 등의 기능을 갖는 여러 UE들을 기반으로 호환성을 높이는 것이다.

참고 문헌

- [1] 이영석, 고석갑, 김영환, “IMS 기반 NGN 기술동향”, *전자공학회지*, 제 33권 8호, pp. 2~4, 2006.
- [2] 3GPP, <http://www.3gpp.org>
- [3] IETF, <http://www.ietf.org>
- [4] Rosenberg J., Schulzrinne H., “SIP: Session Initiation Protocol”, RFC3261, 2002
- [5] 정재훈, 이승윤, 김용진, 「인터넷 트래픽 측정 방법 및 시스템」, *전자통신동향분석*, 16권, 5호, 2001
- [6] Garcia-Martin M., Henrikson E., Mills D., “Private Header (P-Header) Extensions to SIP for the 3GPP”, RFC3455, 2003
- [7] Camarillo G., Blanco B., “The SIP P-Profile-Key Private Header (P-Header)”, RFC5002, 2007
- [8] Camarillo G., Blanco G., “The SIP P-User-Database Private Header (P-Header0)”, RFC 4457, 2006
- [9] Garcia-Martin M., “Input 3GPP Release 5 Requirements on SIP”, RFC4083, 2005
- [10] Wireshark Protocol Analyzer, <http://www.wireshark.org>
- [11] Open IMS Core, <http://www.openimscore.org>
- [12] Jpcap, <http://netresearch.ics.uci.edu/kfujii/jpcap/doc/index.html>
- [13] libpcap, V. Jacobson, C. Leres and S. McCanne, Lawrence, Berkeley Laboratory, Berkeley, 1994. <http://www.tcpdum.org/>
- [14] Wedlund E., Schulzrinne H., “Mobility Support using SIP”, VonEurope Spring 2000, 2000
- [15] Handley M., Jacobson V., Perkins C., “SDP: Session Description Protocol”, RFC4566, 2006
- [16] Ejzak R., “Private Header (P-Header) Extension to SIP for Authorization of Early Media”, RFC5009, 2007

[17] FOKUS, <http://www.fokus.fraunhofer.de>

[18] SOLOMON 프로토콜분석기, <http://www.smon.co.kr/>

정인환 (Inhwan Jung)

정회원



1984년 2월 한양대학교 원자력 공학과 학사

2000년 2월 한국과학기술원 정보및통신공학과 박사

1985년~1998년 삼성전자 시스템사업부 수석 연구원

2007년 University of Colorado 방문 교수

2001년~현재 한성대학교 컴퓨터공학과 부교수

<관심 분야> 멀티미디어 통신, 분산 처리