목 차

그림 목차

표 목차

요 약

인터넷이 확대되고 인터넷 텔레포니 기술(VoIP : Voice over IP)이 시장성 있는 기술로서 각광을 받고 있으며 멀티미디어 기술과 결합하여 기존 망 또는 차세대 이동 통신망에서 핵심적인 서비스로 발전할 것이다. 인터넷 텔레포니 서비스는 음성 및 화상과 같은 대용량의 멀티미디어 데이터를 전송하기 때문에 망에 많은 트래픽이 발생되고 발생된 트래픽 정도에 따라서 멀티미디어 서비스의 품질이 결정된다. 이러한 인터넷 텔레포니 서비스의 세션 제어 프로토콜로 ***SIP(Session Initiation Protocol)*** 가 제안되었으며 향후 SIP는 세션 제어의 표준 프로토콜로 정착될 것으로 보인다. 이러한 배경하에 VoIP 서비스의 측정과 검증을 위한 트래픽 분석 도구가 필요하다. 하지만 아직까지 SIP 기반 멀티미디어 서비스의 트래픽 분석을 위한 도구는 찾아보기 어렵다.

본 논문은 SIP 기반의 인터넷 텔레포니 시스템 환경에서 인터넷 텔레포니 서비스에 의해 발생되는 멀티미디어 통신 트래픽을 측정하고 분석하는 망 진단 도구를 제안한다. 제안된 도구는 저수준 패킷 수집을 통해 SIP 기반 멀티미디어 통신 프로토콜 및 트래픽의 측정 및 분석한다. 분석된 정보를 이용하여 서비스 종류별 트래픽 통계 정보와 그래프로 표시되는 실시간 트래픽 모니터링 정보를 제공함으로써 망의 상태를 점검하고 효과적으로 보완할 수 있는 방법을 제시한다. 본 논문에서 구현된 트래픽 분석기는 SIP 기반 멀티미디어 통신 시스템의 개발 및 진단 도구로 활용될 수 있다.

제 1장 서론

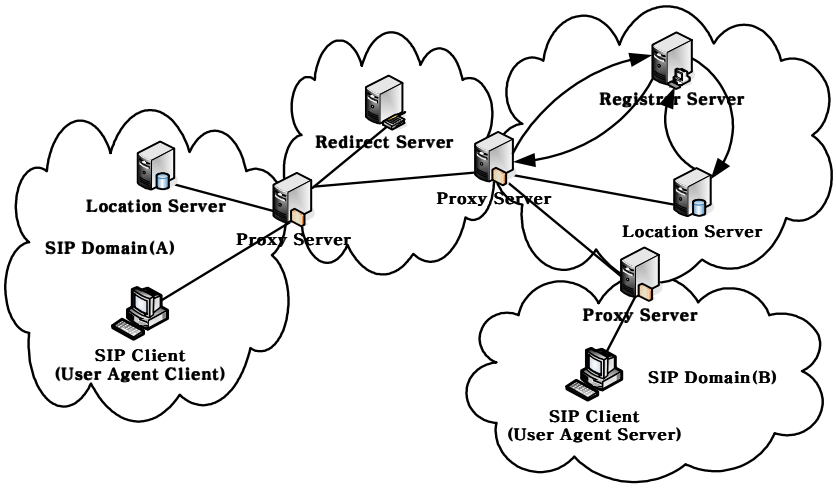
1.1 연구배경

인터넷 텔레포니 기술(Voice over Internet Protocol)이란 지금까지 공중전화망을 통해 이루어졌던 음성 서비스를 Internet Protocol을 이용해 여러 가지 다양한 서비스를 제공하는 기술을 말한다[1]. 이렇게 IP 망을 이용함으로서 기존의 전화망에서 하지 못했던 많은 서비스들이 이루어지고 있으며, 대표적인 응용 프로그램들은 Internet Call Center(Web Call Center), Instance Message, CTI(Computer Telephony Integration), UMS(Unified Messaging System) 등을 들 수 있으며, 이에 대한 연구가 활발히 이루어지고 있다[2][3]. 이러한 인터넷 텔레포니 기술은 저렴한 가격에 장거리 전화를 이용할 수 있다는 장점과 함께 기존의 인터넷에서 사용 가능한 다양한 멀티미디어 서비스를 쉽게 수용할 수 있다는 장점을 가지고 있다. 최근에는 일반 사용자뿐만 아니라 기업에서도 인터넷 텔레포니 기술을 적극적으로 활용하고 있다.

인터넷 텔레포니를 위한 표준 호 설정 프로토콜은 ITU-T(International Telecommunication Union-Telecommunication)의 H.323[4]과 IETF(Internet Engineering Task Force)의 *SIP(Session Initiation Protocol)[5]*를 들 수 있다. 현재 대부분의 응용들이 H.323을 기반으로 되어 있으나, 프로토콜의 복잡성으로 인해 구현이 어렵고, 대역폭이 증가하는 단점을 가지고 있다. 이에 비하여 SIP는 HTTP(HyperText Transfer Protocol)와 유사한 텍스트 기반의 메시지를 사용함으로서 이를 구성하는 헤더의 확장이 용이하고, 간단하게 세션을 설정하고 수정 및 종료할 수 있기 때문에 인터넷 응용 분야로의 적용이 가능한 장점을 가지고 있다[6]. 이런 장점으로 인하여 다수의 표준화 단체들이 차세대 호 설정 프로토콜로서 SIP를 채택함에 따라 SIP를 기반으로 한 많은 응용 연구가 진행되고 있다. 즉, 초고속 무선 네트워크를 통해 멀티미디어를 제작, 전달 및 재생을 논의하는 표준화 단체인 3GPP(3rd Generation Partnership Project)[7]와 동기식 IMT—2000 이동통신 규격을 논의하는 표준화 단체로, 최근 멀티미디어 서비스를 위한 규격들을 제정하고 있는 3GPP2(3rd Generation Partnership Project 2)[8]의 경우, SIP를 멀티미디어 서비스를 위한 표준 프로토콜로 채택하면서, 멀티미디어 서비스(Multimedia Service), 팩스(Fax), 홈 네트워킹(Home Networking), 인스턴스 메세징(Instant Messaging), 망 관리(Network Management), 웹(World Wide Web) 및 SMTP(Simple Mail Transport Protocol) 등

의 응용 서비스들이 SIP를 기반으로 통합되는 구조를 보이고 있다[9][10].

SIP를 이용한 인터넷 텔레포니 시스템은 [그림 1.1]과 같은 구조를 가진다. UAC(User Agnet Clinet)와 UAS(User Agent Server)는 SIP 메시지를 생성하고 메시지를 처리할 수 있는 사용자 측면의 응용 프로그램이다. 프록시 서버(Proxy Server)와 재지정 서버(Redirect Server)는 SIP 메시지를 처리하기 위한 핵심요소이다. 프록시 서버는 수신한 메시지를 직접 다음 노드로 전달하고 재지정 서버는 수신한 메시지가 전달되어야 할 새로운 노드의 주소를 알려주게 된다. 위치 서버(Location Server)는 한 도메인 내에 속하는 사용자의 위치를 기록한다. 등록 서버(Registrar Server)는 사용자에 대한 정보를 등록 및 관리한다.



[그림 1.1] SIP 기반의 VoIP의 구조

1.2 연구동기

일반적으로 SIP 기반 응용 프로그램은 상대편을 세션에 참석시키기 위하여 호출하는 형태로 전개되는데 간략히 설명하면 다음과 같다.

송신자는 호출시 자신에 대한 정보 즉, 자신의 SIP 주소와 호에 대한 정보를 전달한다. 이때 멀티미디어 서비스 통신을 위하여 세션에 표현되어야 할 세션 정보들은 ***SDP(Session Description Protocol)[12]***를 이용하여 기술한다. 세션 정보에는 호출자가 통신에서 사용할 주소, 포트번호, 미디어 종류 등이 포함된다. 이후에 세션 협

약 과정이 발생한다. 즉, 호출을 받은 수신자는 송신자의 세션 정보를 검토하여 수용 및 거부를 표시하여 송신자에게 전달한다. 호 설정 후 *RTP(Realtime Transport Protocol)[13]* 세션이 열리고, 이 세션을 통하여 오디오와 비디오 정보는 UDP 상으로 전송된다.

SIP를 이용한 인터넷 텔레포니 서비스는 오디오, 비디오, 화이트 보드 등과 같은 대용량의 멀티미디어 데이터를 전송하기 때문에 망에 많은 트래픽이 발생된다. 또한 인터넷 텔레포니 서비스를 사용하는 사용자들은 각자의 편의와 상황에 여러 종류의 단말들(데스크탑이나 노트북 또는 PDA와 같은 휴대 단말)들과 엑세스 망을 사용할 수 있다. 이러한 경우 각 단말이 서로 다른 통신 능력과 미디어 코딩방식을 가지고 있기 때문에 SIP 도메인 내에서 발생되는 트래픽 정도에 따라서 멀티미디어 서비스 품질이 결정된다. 따라서 망의 운영 상태를 파악하고 검사하는데 있어서 SIP 기반의 응용 서비스들을 구분하여 분석할 수 있는 망 진단 도구의 필요성을 절실히 느끼고 있다.

일반적으로 망 진단 도구로서 가장 많이 활용되는 것이 트래픽 분석기[11]이다. 트래픽 분석기는 이더넷(Ethernet)의 특성인 동보 기능(broadcasting)을 이용하여 LAN에 흘러 다니는 모든 패킷들을 실시간으로 수집하여, 패킷의 세부 내용을 보여주는 프로토콜 분석(protocol analyze) 기능과 망 사용 통계를 보여주는 트래픽 측정 기능을 가지고 있다. 그러나 아직까지 SIP 기반 응용 프로그램에서 발생되는 트래픽을 세부적으로 분석할 수 있는 트래픽 분석기는 찾아보기 힘들다. 따라서 SIP를 기반으로 하는 응용 프로그램에서 발생되는 멀티미디어 통신 트래픽을 분석할 수 있는 도구를 설계하고 구현하는 일이 중요하다.

일반적인 망 진단 도구들은 패킷의 헤더 정보, 현재의 통신 속도와 프로토콜 별 사용량을 보여주는 구조로 되어있다. 이런 진단 도구들은 SIP 기반 인터넷 텔레포니 서비스를 이용하는 사용자가 어느 정도인지 즉, 세션의 수가 몇 개이고 그 세션들에서 발생되는 멀티미디어 통신 트래픽이 어느 정도인지 분석할 수 없고, 또한 발생되는 트래픽이 어떤 종류의 미디어 타입인지와 같은 세부적인 정보를 알 수가 없다.

그러므로 SIP 기반 인터넷 텔레포니 시스템 환경에서 멀티미디어 통신 트래픽을 분석하기 위해서는 트래픽 분석기가 다음과 같은 기능을 가져야 한다.

- SIP 프로토콜 분석(요청 또는 응답 메시지, 세션 설정/변경/종료 등) 기능
- SDP 프로토콜 분석(사용할 주소, 포트번호, 미디어 종료 등) 기능
- RTP 프로토콜 분석(미디어 타입, 전송된 데이터 량 등) 기능
- 세션 관리(세션의 수, 세션에서 발생되는 트래픽의 량 등) 기능
- 미디어 별 통계 표시 기능

본 논문에서는 위와 같은 기능을 갖는 새로운 트래픽 분석기 STAT(Sip based Traffic Analysis Tool)를 설계하고 구현한다. STAT는 일반 PC에 기반을 둔 소프트웨어이므로 확장성과 범용성을 갖는다.

본 논문의 구성은 다음과 같다. 2장 관련연구에서는 인터넷 멀티미디어 통신 프로토콜과 일반적인 트래픽 분석기에 대해 알아보고, 3장에서는 STAT 설계 및 구현에 대하여 설명한다. 4장에서는 실험 및 성능 평가에 대하여 설명한다. 마지막으로 5장에서는 결론 및 향후 연구에 대하여 기술한다.

제 2장 관련연구

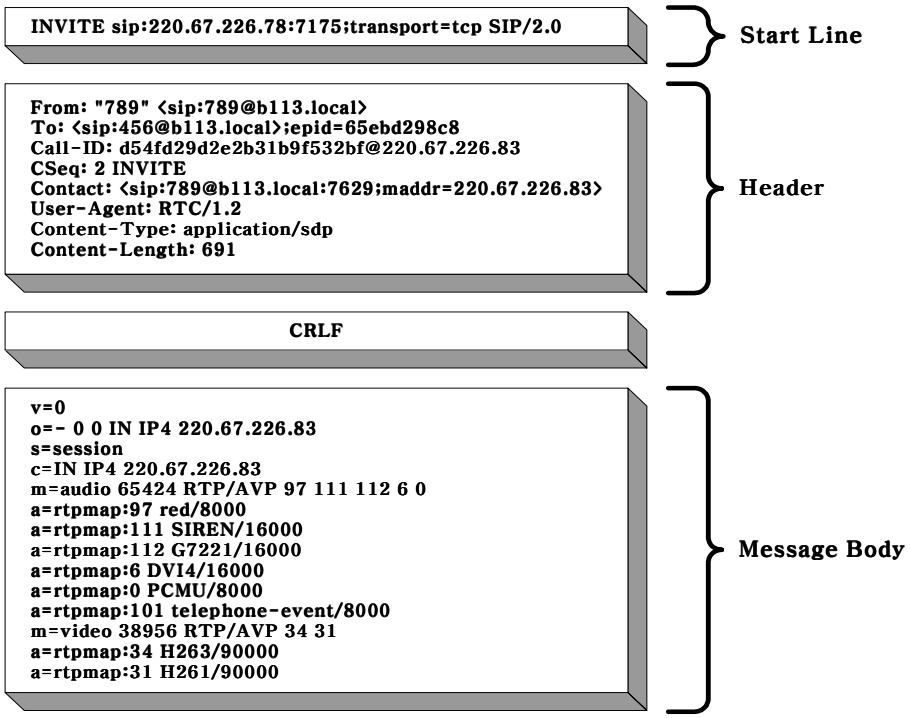
본 장에서는 우선 대표적인 인터넷 멀티미디어 통신 프로토콜인 SIP 프로토콜, SDP 프로토콜 및 RTP 프로토콜에 대해 살펴보고 일반적인 트래픽 분석기의 구조에 대하여 설명한다.

2.1 SIP(Session Initiation Protocol)

SIP는 ITU-T(International Telecommunication Union-Telecommunication)의 H.323에 대응되는 프로토콜로서 IETF(Internet Engineering Task Force) MMUSIC(Multiparty Multimedia Session Control) WG(Working Group)에서 개발되어 오다가 IETF SIP WG를 신설하여 작업을 진행하고 있다. SIP는 클라이언트들이 호출을 시작하면 서버가 그 호출에 응답을 하는 클라이언트/서버 구조에 기반을 두고 있다. 이 프로토콜은 사용이 간단한 텍스트 기반의 인터넷 프로토콜로서 HTTP(Hyper Text Transfer Protocol), SMTP(Simple Mail Transfer Protocol)와 유사한 E-mail 주소나 WWW(World Wide Web) 주소의 형태를 가지고 있으며, 사용자 간의 상호 통신 세션을 초기화하는 용도로 설계되었다. 즉, 하나 이상의 참여자로 구성되는 세션을 초기화하고, 변경 및 종료하기 위해 사용하는 용도로 설계되었다. 세션은 오디오, 비디오, 화이트 보드 등과 같은 하나 또는 그 이상의 미디어 타입으로 이루어진 멀티미디어 데이터 전송 등을 포함한다.

2.1.1 SIP 특성

SIP 메시지는 시작 줄, 헤더, 헤더의 마지막을 가리키는 CRLF와 메시지 본문으로 구성되어 있다. SIP는 시작 줄과 헤더의 값만을 정의하고, 메시지 본문의 값은 세션 정보를 기술하는데 사용하는 프로토콜인 SDP가 정의한다. [그림 2.1]은 SIP 메시지의 구조를 보여주고 있다.



[그림 2.1] SIP 메시지의 구조

SIP 메시지의 시작 줄에 따라 클라이언트에서 서버로 보내는 요청(Request) 메시지와 서버에서 클라이언트로 보내는 응답(Response) 메시지로 구분된다. [표 2.1]에 설명되어 있는 바와 같이 시작 줄 구문은 SIP 메시지가 요청인지 응답인지에 따라 달라진다.

시작 줄	구 문
요청 메시지	신호 명령, 요청 URI, SIP 버전
응답 메시지	SIP 버전, 상태 코드, 응답 구

[표 2.1] SIP 메시지 시작 줄 구문

가. SIP 요청 메시지

SIP 요청 메시지의 시작 줄의 첫 번째 항목은 신호 명령[15]이다. SIP의 주요 신호 명령과 기능을 [표 2.2]에 나열하였다.

SIP 신호 명령	설 명
INVITE	세션 설정
ACK	INVITE의 마지막 응답에 대한 승인
BYE	세션 종료
CANCEL	대기중인 세션 취소
REGISTER	사용자의 URL 등록
SUBSCRIBE	이벤트 통지를 요청
UNSUBSCRIBE	이벤트 통지를 취소
NOTIFY	구독된 이벤트 통지서를 전송

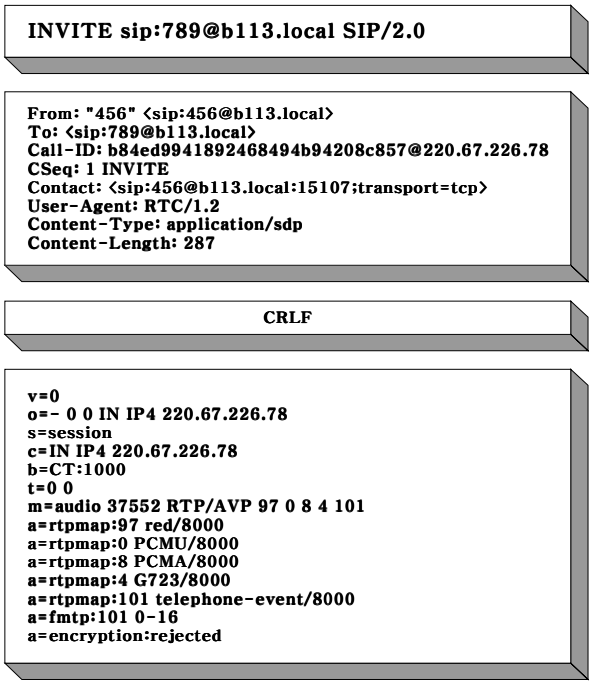
[표 2.2] SIP 신호 명령과 기능

SIP 요청 메시지의 두 번째 항목은 수신자 URL을 포함하는 요청 URI(Uniform Resource Identifier)이다. [표 2.3]은 지원되는 SIP URI 중 일부를 나열한 것이다.

SIP URI 형식	설 명
sip:user@b113.local	◦ 기본 SIP URL
sip:user@b113.local;transport=TCP	◦ TCP의 전송 프로토콜을 포함하는 기본 SIP URL ◦ 전송 프로토콜이 지정되지 않은 경우 기본값은 UDP
sip:user@220.67.226.78	◦ IP 주소를 포함하는 SIP URL
sip:+1-425-707-9796@b113.local;user=phone	◦ 국제 전화 번호를 포함하는 SIP URL
sip:b113@b113.local;maddr=225.0.2.1;ttl=64	◦ 이전에 지정된 호스트 이름을 무시하는 멀티캐스트 주소를 포함하는 SIP URL ◦ TTL 값이 64로 설정 ◦ 멀티캐스트 주소와 UDP를 전송 프로토콜로 사용할 경우 TTL을 설정

[표 2.3] SIP 요청 URI

SIP 요청 메시지의 마지막 항목은 SIP 버전이며 현재는 2.0이다. [그림 2.2]는 SIP 요청 메시지의 예이고, SIP 요청 메시지의 시작 줄 부분이 크게 표시되어 있다.



[그림 2.2] SIP 요청 메시지

나. SIP 응답 메시지

SIP 응답 메시지의 첫 번째 항목은 SIP 요청 메시지의 마지막 항목과 같은 SIP 버전이고, SIP 응답 메시지의 두 번째 항목은 숫자로 된 상태 코드이다. SIP 상태 코드는 [표 2.4]와 같이 정보, 성공, 재지정, 클라이언트 오류, 서버 오류 및 전역 실패의 6개의 범주로 되어있다.

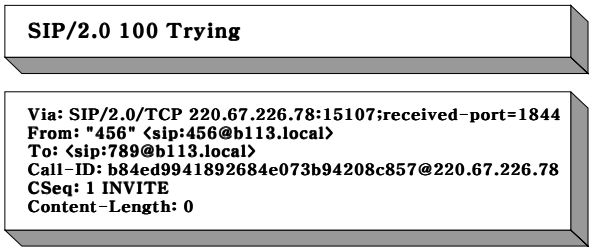
SIP 상태 코드	응답 범주	설 명
1XX	정보(Informational)	호의 진행 상태 표시
2XX	성공(Success)	요청이 성공적으로 처리
3XX	재지정(Redirection)	요청에 대한 위치 정보를 표시
4XX	클라이언트 오류(Client Error)	클라이언트 에러를 표시
5XX	서버 오류(Server Error)	서버 에러를 표시
6XX	전역 실패(Global Failure)	요청에 대한 지원이 불가능한 경우

[표 2.4] SIP 응답 코드

SIP 응답 메시지의 마지막 항목은 응답 구이다. [표 2.5]에는 SIP 버전 2.0에 정의된 SIP 상태 코드와 해당 범주 및 응답 구를 나열하였다. [그림 2.3]은 SIP 응답 메시지의 예이고, SIP 응답 메시지의 시작 줄 부분이 크게 표시되어 있다.

SIP 상태 코드	응답 범주	응답 구
100	정보	시도 중
180	정보	신호가 울림
200	성공	확인
300	재지정	다중 선택
301	재지정	영구적으로 이동됨
302	재지정	임시로 이동됨
303	재지정	기타 참조
305	재지정	프록시 사용
380	재지정	대체 서비스
400	클라이언트 오류	잘못된 요청
401	클라이언트 오류	권한이 없음
402	클라이언트 오류	지불 필요
403	클라이언트 오류	사용할 수 없음
404	클라이언트 오류	없음
405	클라이언트 오류	메서드 허용 안 함
406	클라이언트 오류	받아들일 수 없음
407	클라이언트 오류	프록시 인증 필요
408	클라이언트 오류	요청 시간 초과
409	클라이언트 오류	충돌
410	클라이언트 오류	없음
411	클라이언트 오류	길이 필요
413	클라이언트 오류	요청 엔터티가 너무 큼
414	클라이언트 오류	요청 URI이 너무 김
415	클라이언트 오류	지원되지 않는 미디어 유형
420	클라이언트 오류	잘못된 확장
500	서버 오류	서버 내부 오류
501	서버 오류	구현되지 않음
502	서버 오류	잘못된 게이트웨이
503	서버 오류	서비스 사용할 수 없음
504	서버 오류	게이트웨이 시간 초과
505	서버 오류	지원되지 않는 SIP 버전
600	전역 실패	모든 곳에서 사용 중
603	전역 실패	적용 안 함
604	전역 실패	어느 곳에도 없음
606	전역 실패	받아들일 수 없음

[표 2.5] SIP 응답 코드 및 응답 구



[그림 2.3] SIP 응답 메시지

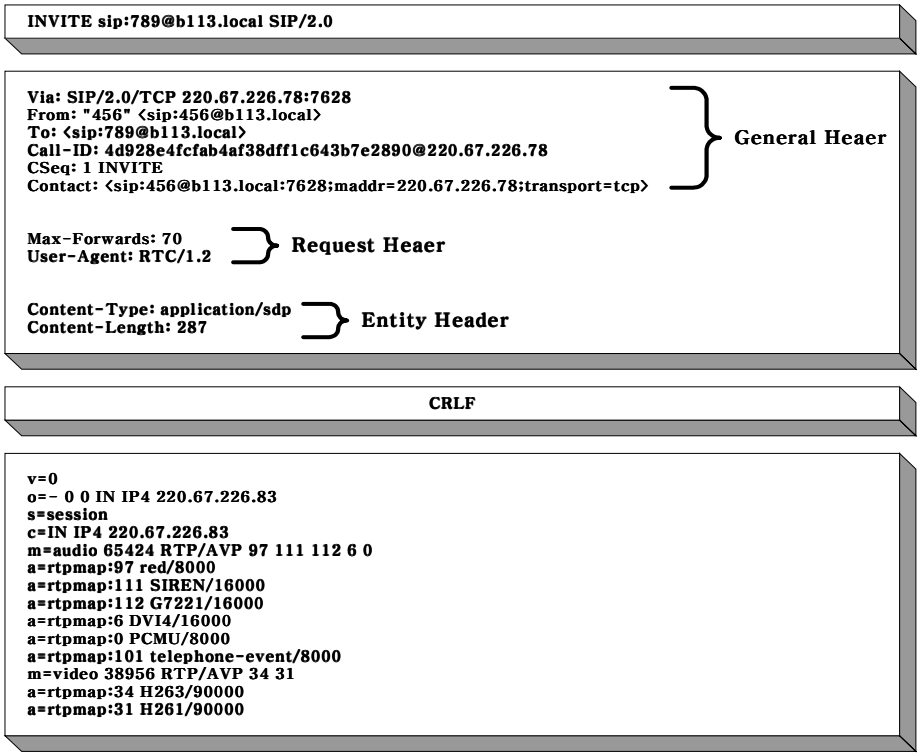
다. SIP 헤더

SIP 메시지의 시작 줄 다음에는 하나 이상의 헤더가 온다. 포함된 헤더는 메시지가 응답인지, 요청인지에 따라 달라진다. SIP 헤더는 [표 2.6]과 같이 일반, 요청, 응답 및 본체의 네 가지 범주로 구분되고 일반 범주의 헤더는 요청 및 응답 메시지 모두에 사용할 수 있다.

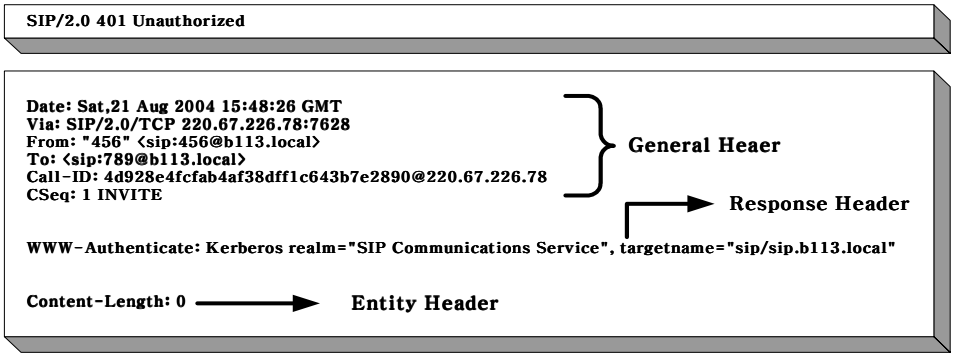
일 반	요 청	응 답	본 체
Accept	Authorization	Allow	Content-encoding
Accept-encoding	Hide	Proxy-authenticate	Content-length
Accept-language	Max-forwards	Retry-after	Content-type
Call-ID	Organization	Server	
Contact	Priority	Unsupported	
Cseq	Proxy-authorization	Warning	
Date	Proxy-require	WWW-authenticate	
Encryption	Route		
Expires	Require		
From	Response-key		
Record-route	Subject		
Time stamp	User-agent		
To			
Via			

[표 2.6] SIP 헤더 종류

SIP 헤더의 구조는 하나 이상의 일반 헤더, SIP 메시지 종류에 따라 요청 헤더 또는 응답 헤더와 본체 헤더로 구성된다. [그림 2.4]와 [그림 2.5]는 SIP 메시지의 종류에 따른 SIP 헤더의 구조의 예를 보여주고 있다.



[그림 2.4] SIP 요청 메시지의 헤더



[그림 2.5] SIP 응답 메시지의 헤더

2.1.2 SIP 구성 요소

SIP UA(User Agent)는 SIP 메시지 전송을 요구하고 전송된 SIP 메시지를 수신하는 단말 시스템으로, UAC(User Agent Client)와 UAS(User Agent Server)로 나뉜다. 일반적으로 UAC는 요청 메시지를 만들어서 전송하는 단말을 말하며, UAS는 요청 메시지를 받으면 클라이언트에게 알리거나 내부적 판단에 의하여 대응되는 응답 메시지를 전송하는 단말을 말한다.

SIP 서버들은 프록시 서버(Proxy Server), 재지정 서버(Redirect Server), 등록 서버(Registrar Server), 위치 서버(Location Server) 등이 있다. 프록시 서버는 UA로부터의 요청 메시지를 실제 메시지가 전달되어야 할 곳으로 전달(forwarding)하는 역할을 한다. 재지정 서버는 수신한 메시지가 전달되어야 할 새로운 위치를 응답 메시지를 통해 알려주어 사용자가 직접 전송하도록 하며, 등록 서버는 UA를 등록할 수 있게 하고 이를 관리하며 인증 과정을 통해 정당한 사용자인지를 구별하는 역할도 한다. 또한 위치 서버는 한 도메인 내에 속한 사용자의 현재의 위치를 기록하기 위한 서버이다. 이러한 위치 서버를 이용할 경우 현재 사용자가 접속한 시스템의 위치를 파악할 수 있기 때문에 사용자의 이동성(User Mobility)[14]을 지원해 줄 수 있다.

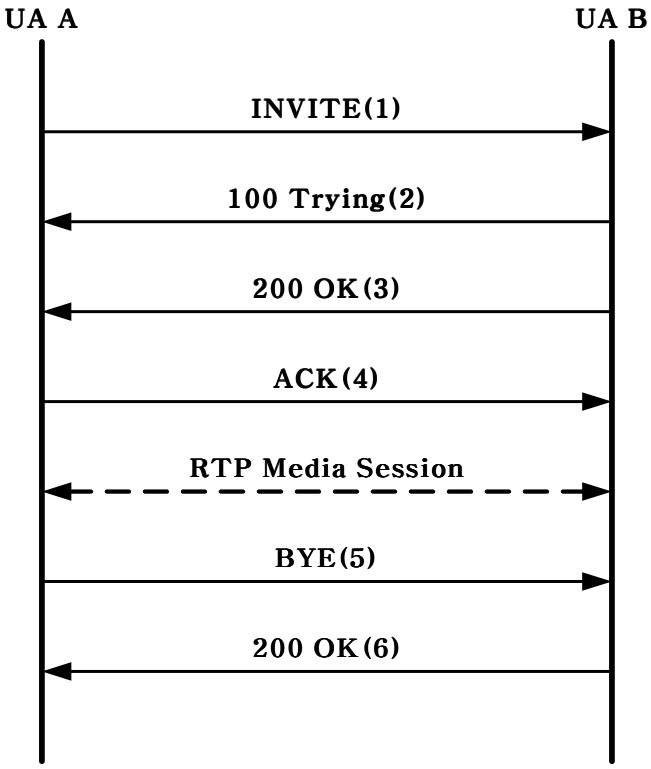
2.1.3 SIP 세션의 호출 흐름

SIP 세션의 호출 흐름은 SIP 세션이 SIP UA간에 직접 설정되었는지 아니면 SIP 서버(프록시, 등록 또는 재지정)가 SIP UA 사이에 있는지 여부에 따라 달라진다. 각각의 SIP 세션의 호출 흐름을 설명하면 다음과 같다.

가. 두 UA간의 일반적인 호출 흐름

[그림 2.6]은 두 UA간의 일반적인 호출 흐름을 보여 주며 각 단계는 괄호로 되어있다. 우선 UA A는 호출을 시작하기 위해 INVITE 요청을 보낸다. 그러면 UA B는 호출 요청이 처리 중임을 나타내는 시도 중 응답 코드(100)로 응답한다. UA B가 호출을 수락했음을 나타내는 확인 응답 코드(200)로 응답하면 UA A는 UA B의 최종 응답 코드를 UA A가 수신했음을 나타내는 승인(ACK) 요청으로 UA B에 응답한다. 실시간 데이터가 오디오 및 비디오 코덱을 통하여 압축된 정보를 UA A와 B

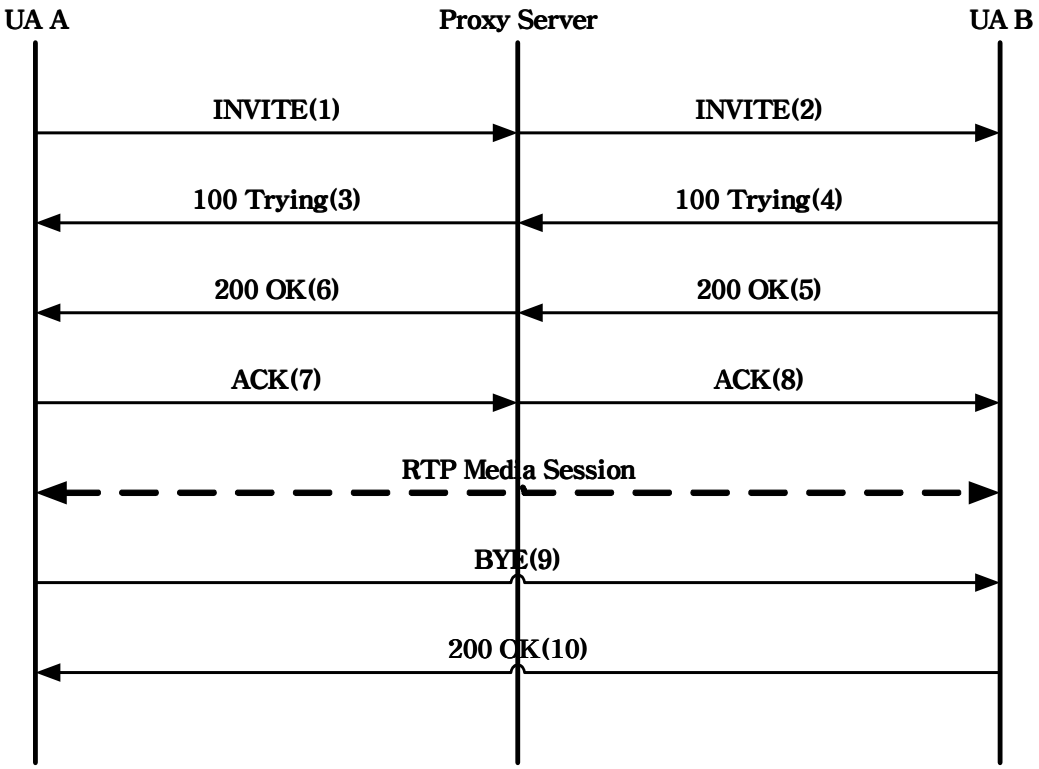
사이에 전송되고 나면, UA A 또는 B는 UA가 세션을 종료하고자 함을 나타내는 BYE 요청을 보낼 수 있다. 그러면 UA B는 요청 성공을 나타내는 확인 응답 코드(200)를 UA A에게 보낸다.



[그림 2.6] 두 UA간의 호출 흐름

나. 두 UA 경로 사이에 프록시 서버가 있을 경우

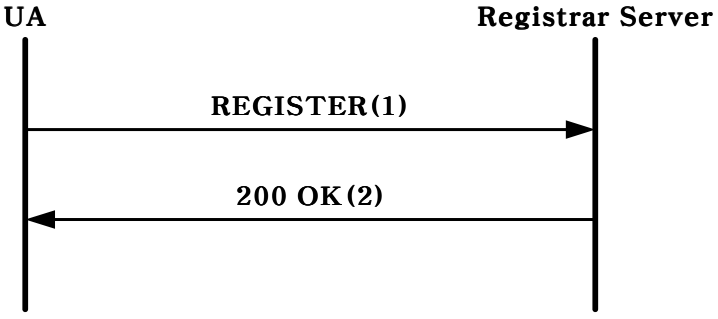
[그림 2.7]은 두 UA 경로 사이에 프록시 서버가 있을 경우의 일반적인 호출 흐름을 보여 준다. 프록시 서버는 기본적으로 UAS 및 UAC의 기능을 모두 수행하는 통신 중간 지점으로 사용된다. UAS로 작동할 경우 프록시는 SIP 요청을 수신하여 대상 UA로 전달하고, UAC로 작동할 경우 SIP 응답을 수신하여 대상 UA로 전달한다.



[그림 2.7] 두 UA 경로 사이에 프록시 서버가 있을 경우

다. UA와 등록 서버간의 일반적인 호출 흐름

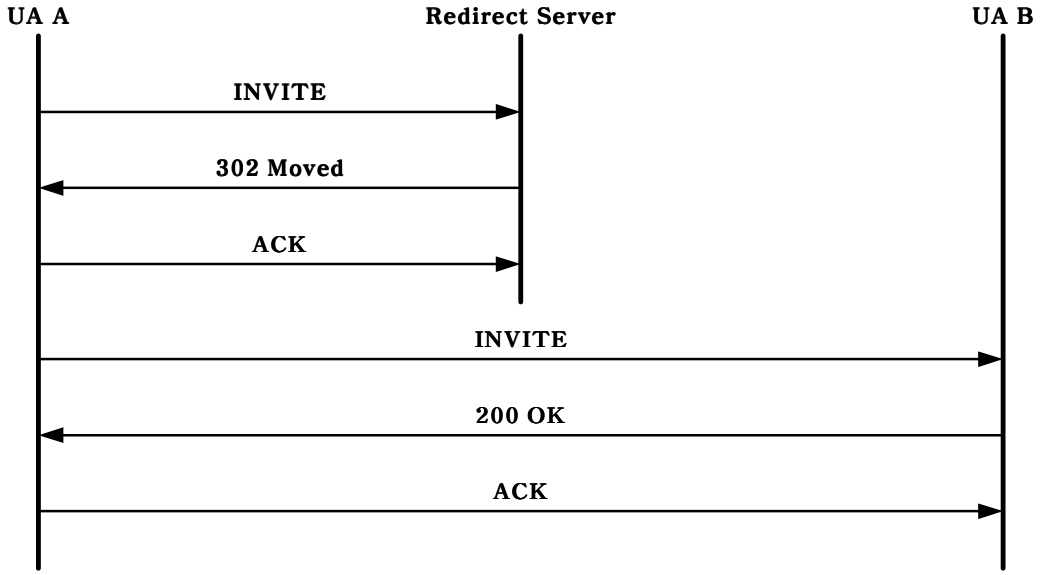
[그림 2.8]은 UA와 등록 서버간의 일반적인 호출 흐름을 보여준다. 등록 서버는 UA가 도달할 수 있는 주소를 나타내는 UA의 REGISTER 요청을 수락한다. 등록 서버는 일반적으로 프록시 또는 재지정 서버와 함께 위치한다.



[그림 2.8] UA와 등록 서버간의 호출 흐름

라. 재지정 서버가 두 UA 사이에 있을 경우의 일반적인 호출 흐름

[그림 2.9]는 재지정 서버가 두 UA 사이에 있을 경우의 일반적인 호출 흐름을 보여준다. UA A가 호출을 시작하기 위해 INVITE 요청을 보내면, 재지정 서버는 UA B가 일시적으로 이동했음을 나타내는 이동됨 응답 코드(302)로 응답한다. 이어서 UA A는 재지정 서버의 응답 코드를 UA A가 수신했음을 나타내는 ACK 요청으로 응답한다. 그러면 UA A는 UA B의 새로 얻은 주소로 또 다른 INVITE 요청을 바로 보낸다.



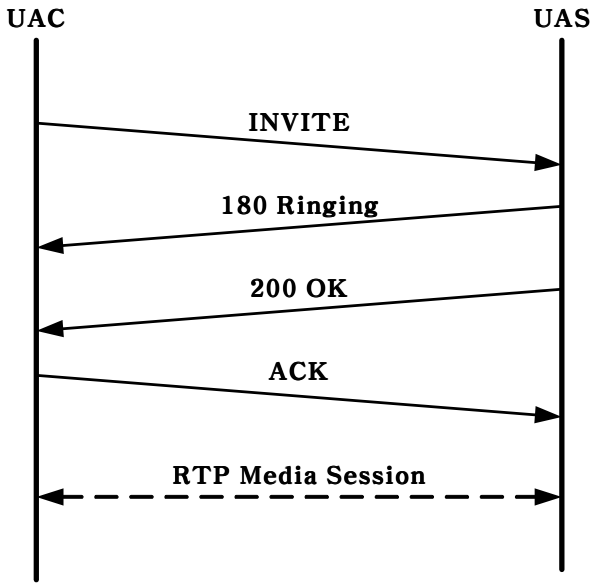
[그림 2.9] 재지정 서버가 두 UA 사이에 있을 경우의 호출 흐름

2.1.4 SIP UA의 동작원리

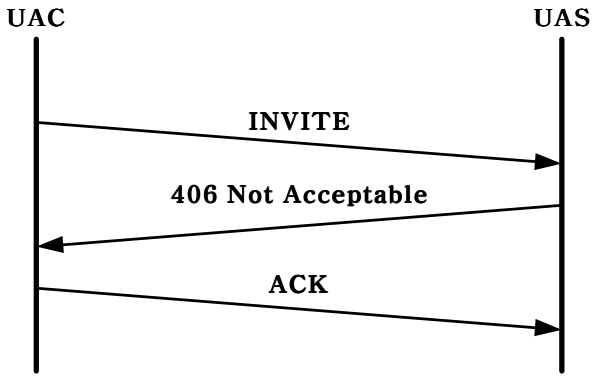
가. 초기 세션 설정

클라이언트가 통신을 하기 위해서는 INVITE 메시지를 전송하여 세션 설정을 요청을 해야한다. 이때 INVITE 메시지의 메시지 본문에 세션 정보가 기술된다. INVITE 메시지를 받은 착신자는 세션 정보를 검토하여 각 미디어 스트림을 수용할

지 거부할지 판단한다. 수용된 미디어 스트림의 정보는 그대로 응답 메시지에 복사한다. 거부된 스트림은 포트번호를 0으로 지정한다. 수정된 세션 정보를 응답 메시지 200 OK에 실어서 송신자에게 전송하여 세션 참여를 알린다. 송신자가 ACK 메시지를 전송하면 오디오 코덱과 비디오 코덱이 결정되고 세션이 열린다. 세션이 열린 후에 오디오 및 비디오 코덱을 통하여 압축된 정보를 세션을 통하여 전송한다. [그림 2.10]은 수용된 미디어 스트림의 초기 세션 설정 과정을 보여주고 있고, [그림 2.11]은 거부된 미디어 스트림의 초기 세션 설정 과정을 보여주고 있다.



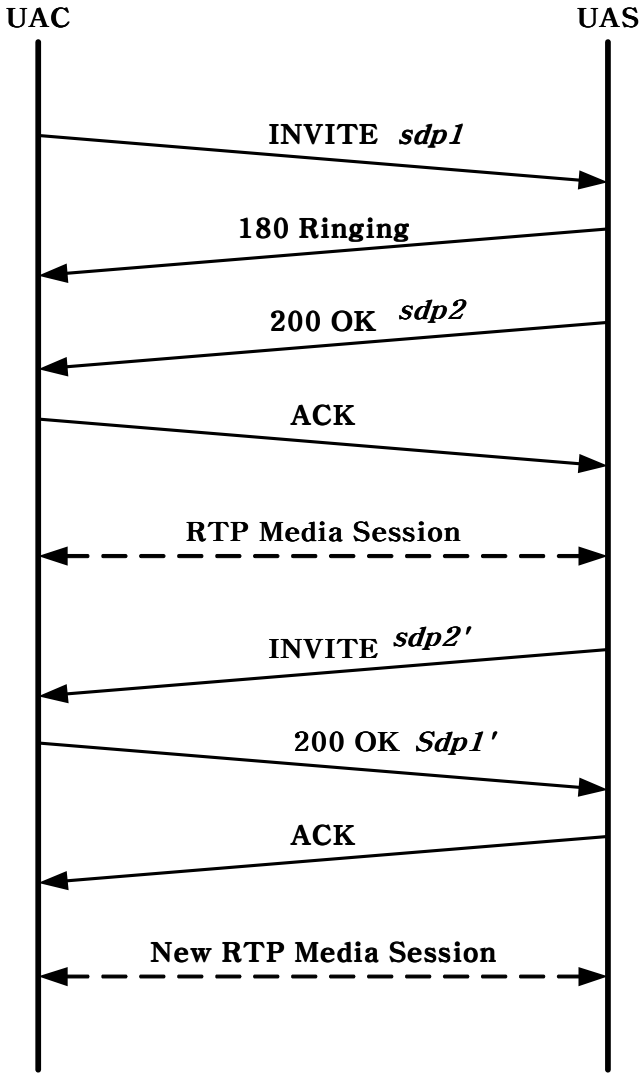
[그림 2.10] SIP 초기 세션 설정 - 미디어 수용



[그림 2.11] SIP 초기 세션 설정 - 미디어 거부

나. 세션 변경

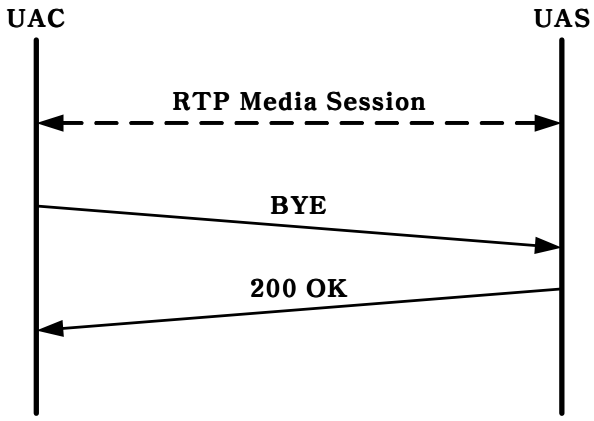
세션이 설정된 이후에 세션은 변경될 수 있다. 세션 변경의 종류로는 미디어 스트림의 추가 및 삭제, 코덱 변경, 주소 정보 변경 등이 있다. INVITE 메시지를 전송하여 세션을 변경한다. 이 INVITE가 거부되어도 기존 세션에는 전혀 영향을 주지 않는다. [그림 2.12]는 세션의 변경 과정을 나타낸다.



[그림 2.12] SIP 세션 변경

다. 세션 종료

설정된 세션을 종료하기 위해서는 [그림 2.13]과 같이 BYE 메시지를 이용하여 세션의 종료를 알린다. BYE 메시지를 받은 착신자는 자신의 세션을 종료하고, 200 OK 응답 메시지를 송신자에게 전송한다. 200 OK 응답 메시지를 받은 송신자는 자신의 세션을 종료한다.



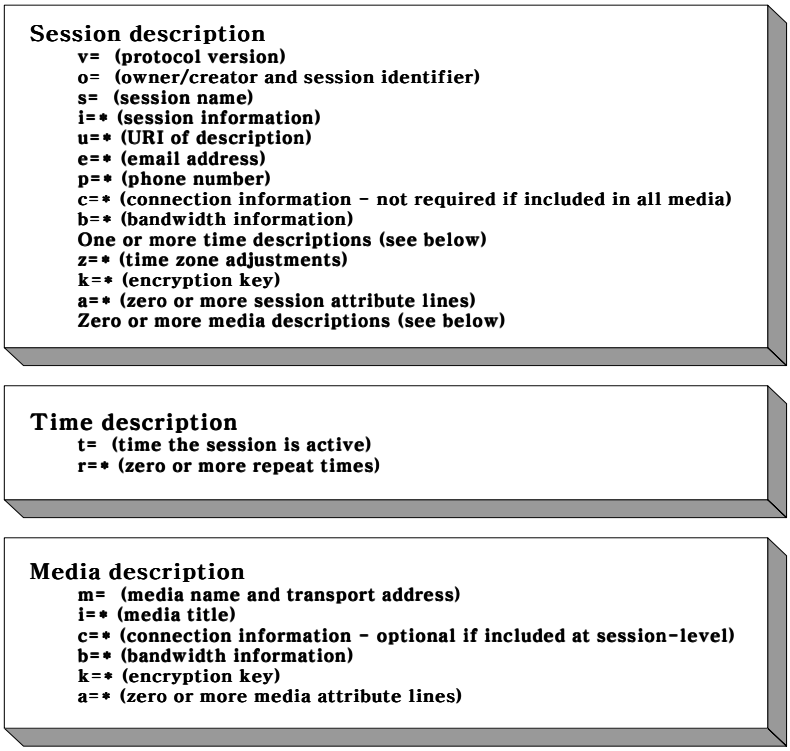
[그림 2.13] SIP 세션 종료

위와 같이 SIP 특성, SIP 구성요소, SIP 세션의 호출 흐름과 SIP UA의 동작원리에 대해 살펴보았다. SIP는 클라이언트/서버 구조에 기반을 두고 있으면 SIP 메시지는 요청 메시지와 응답 메시지로 구분된다. 각각의 SIP 메시지가 어떠한 구조를 가지고 있으며 SIP 메시지의 종류에 따라서 헤더의 종류가 결정되고 세션 정보가 기술되는 메시지 본문이 어떤 경우에 존재하는지 알 수 있었다. 또한 SIP 세션 설정 과정 중에 전달되는 SIP 메시지의 종류들을 확인하였다.

본 논문에서는 SIP 메시지들을 수집하고 분석하여, 세션에 대한 설정이나 변경 또는 세션 종료에 관련된 메시지인 경우에는 해당하는 SIP 메시지를 구조화하고, 구조화된 SIP 메시지를 통하여 세션을 관리하고 트래픽을 분석한다. 구조화된 SIP 메시지에는 SIP 요청 메시지인 경우에는 신호 명령이 저장되고 SIP 응답 메시지인 경우에는 상태 코드가 저장되며 호를 식별하는데 필요한 SIP 일반 헤더의 Call-ID, From, To, Cseq와 SIP 본체 헤더의 Content-length, Content-type 등이 저장된다.

2.2 SDP(Session Description Protocol)

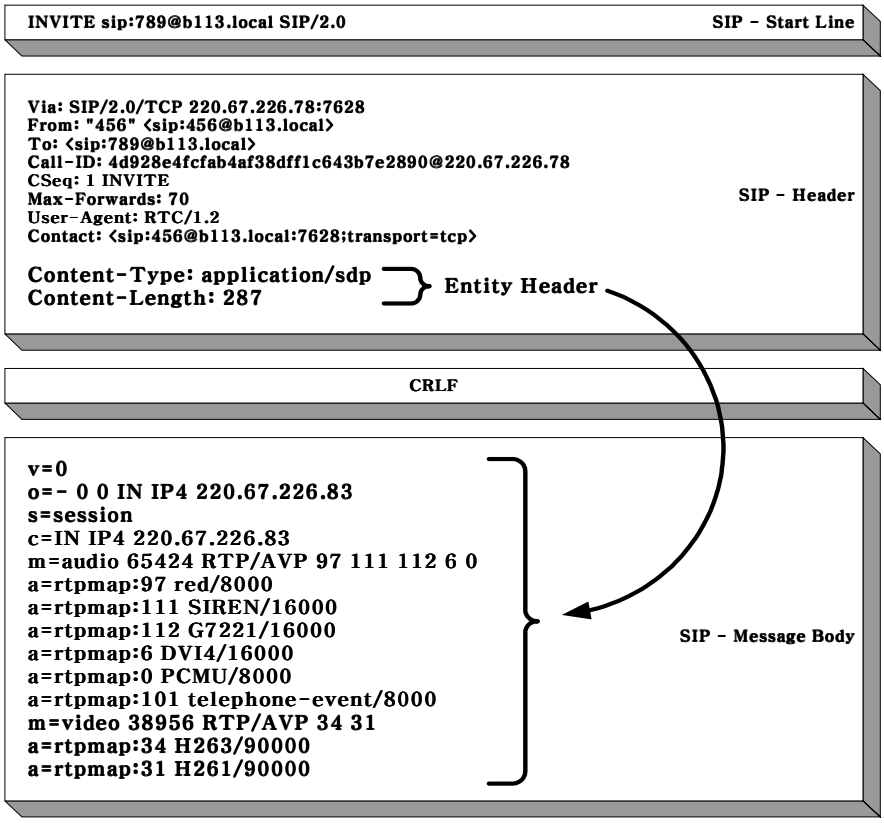
SDP는 멀티미디어 세션을 기술하기 위해 IETF에서 제안되었고, 멀티미디어 회의를 알리고 설명하기 위한 표준이다. SIP 메시지의 메시지 본문에는 SDP에 의해 정의된 바와 같이 값이 기술되어 있다. SDP 세션의 기술(Description)은 필드 이름과 속성 이름으로 이루어진 세션 기술(Session Description), 시간 기술(Time Description)과 미디어 기술(Media Description)의 3개 부분으로 기술된다. 세션의 기술은 하나의 세션 기술(Session Description), 0개 이상의 시간 기술(Time Description) 및 0개 이상의 미디어 기술(Media Description)의 세 부분으로 구성되며, [그림 2.14]는 SDP 프로토콜의 구조를 보여주고 있다.



[그림 2.14] SDP 프로토콜 구조

세션 기술(Session Description)은 전체 회의나 모든 미디어 스트림에 적용되는 전역 특성을 포함하고, 시간 기술(Time Description)은 회의 시작, 중지 및 반복 시간

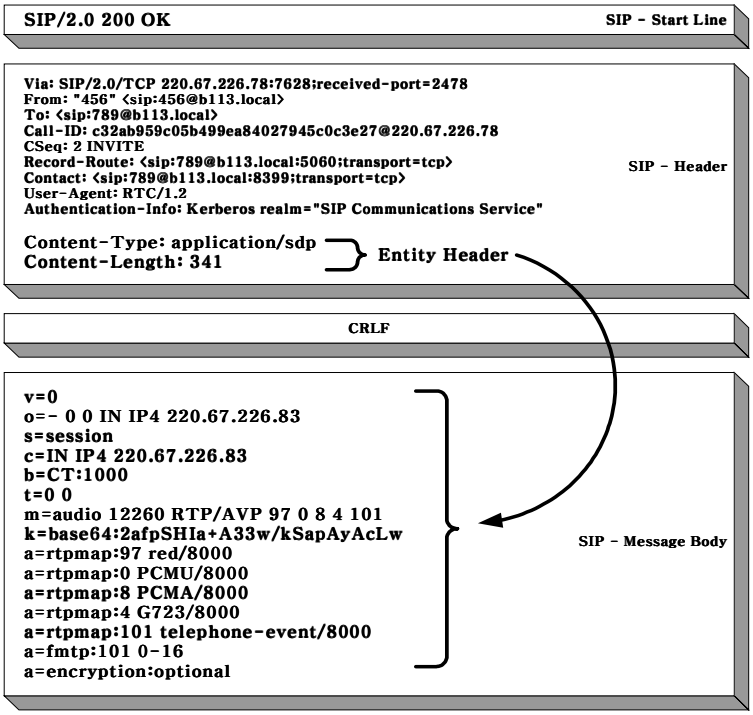
정보를 포함하며, 미디어 기술(Media Description)은 특정 미디어 스트림에 대한 세부 정보를 포함한다. [그림 2.15]는 SIP 메시지와 SDP 메시지와의 관계를 보여주고 있다.



[그림 2.15] SIP 메시지에 기술된 SDP 메시지

SIP는 SIP 메시지의 시작 줄과 헤더의 내용을 정의하고 메시지 본문은 SDP에 의해 기술된다. SIP 메시지에 세션 정보가 기술되었을 경우에는 [그림 2.15]에서 보듯이 SIP 본체 헤더가 존재된다. 기술된 SIP 본체 헤더는 본체-종류(Content-type) 헤더의 값으로 application/sdp를 갖고 본체-길이(Content-length) 헤더의 값으로 SIP 메시지 본문에 기술된 내용의 크기를 값으로 갖는다.

SIP 메시지에서 SDP가 기술되는 경우는 크게 두 가지로 볼 수 있다. 하나는 SIP INVITE 요청 메시지와 다른 하나는 SIP INVITE 요청 메시지에 대한 응답으로 상태 코드가 200인 SIP 응답 메시지이다. [그림 2.16]은 SIP 응답 메시지에 기술된 SDP 메시지이고, SDP의 주요 필드의 속성 내용을 [표 2.7]에 나열하였다.



[그림 2.16] SIP 응답 메시지에 기술된 SDP 메시지

필드	속성 내용	설 명
c	IN IP4 220.67.226.83	접속 정보(Connection Information)
	<network type>	Internet을 의미 - IN
	<address type>	Internet Protocol 버전 4 - IPv4
	<connection address>	사용할 주소 - 220.67.226.83
m	video 38956 RTP/AVP 34 31	미디어 안내(Media Announcements)
	<media>	미디어 종류 - video
	<port>	포트 번호 - 38956
	<transport>	사용할 전송계층 프로토콜 - RTP/AVP RTP/AVP : Audio/Video Profile를 사용하는 RTP
	<fmt list>	미디어 형식 - 34 41 RTP/AVP에 정의되어 있는 미디어 페이로드 타입
a	rtpmap:34 H263/90000	속성(Attribute)
	rtpmap:<payload type>	페이로드 종류 - 34 RTP/AVP의 페이로드 종류
	<encoding name>	코덱 이름 - H.263
	/<clock rate>	표본화 속도 - 90000

[표 2.7] SDP의 주요 필드의 속성 내용

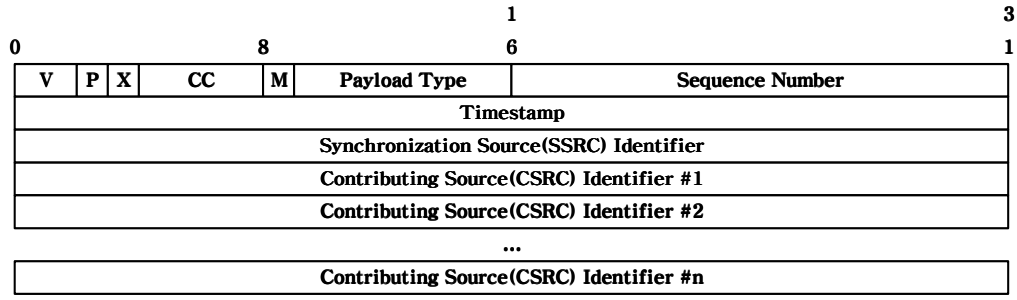
본 논문에서는 SIP 메시지의 메시지 본문에 기술된 SDP 메시지를 구조화하고, 구조화된 SDP 메시지를 통하여 발생된 트래픽의 미디어 종류와 미디어 별 통계를 측정한다. 구조화된 SDP 메시지에는 세션 기술(Session Description)의 접속 정보(connection information)와 미디어 기술(Session Description)의 미디어 이름 및 전송 주소와 미디어 특성 등이 저장된다.

2.3 RTP(Realtime Transport Protocol)

RTP는 IETF에서 표준화한 실시간 전송 프로토콜로써, 패킷 기반 네트워크를 통한 실시간 통신 요구를 충족하도록 설계되었다. RTP는 실시간 응용 프로그램을 위한 실시간 데이터의 종단간 네트워크 전송 기능을 제공하고, 오디오 및 비디오 등의 실시간 데이터는 부호화 및 압축을 통해 패킷 기반 네트워크를 통한 전송에 맞게 최적화 된 후에 RTP 내에 내장(encapsulation)된다. RTP는 패킷 전송의 보증이나 정해진 시간 내에 정확하게 전송시키기 위한 기법을 제공하는 것은 아니지만 RTP에는 실시간 세션에 관한 정보가 포함되어 있으므로 응용 프로그램은 순간 흐트러짐(jitter), 비순차 패킷 및 패킷 손실을 쉽게 조정할 수 있다.

일반적으로 RTP는 UDP와 함께 기본 전송 계층으로 사용되고, IP와 함께 기본 네트워크 계층으로 사용된다. RTP는 특정 미디어 스트림의 송신자와 수신자간에 협상된 동적 UDP 포트를 사용한다. 그러나 RTP는 기본 전송 및 네트워크 계층과 무관하며, 전송 및 네트워크 프로토콜로서 UDP 및 IP와 함께 사용할 필요가 없다.

RTP 패킷은 [그림 2.17]과 같이 12 바이트 헤더와 페이로드(Payload)로 구성된다. RTP 해더는 오디오와 비디오 데이터를 동기화하고 화면에 출력하며, 패킷 손실과 비순차 패킷을 검사하는데 필요한 시간 정보를 제공한다. [표 2.8]은 RTP 패킷의 헤더 내용이다.



[그림 2.17] RTP 패킷의 헤더

헤 더	내 용
Version	◦RTP 프로토콜의 버전을 표시
Padding	◦암호화를 위해 padding된 octet이 있을 경우에 설정 ◦가장 마지막 바이트는 padding 수를 표시
Extention	◦고정 RTP 헤더에 추가된 확장 헤더가 존재
CSRC Count	◦고정 RTP 헤더 뒤에 오는 CSRC 식별자의 수
Maker	◦RTP 프로필에 따라 마커 비트의 정의 및 사용이 결정 -비디오 페이로드 : 프레임의 끝을 표시 -오디오 페이로드 : 셀 발생(Talk Spurt)의 시작을 표시
Palyload Type	◦페이로드의 종류를 식별(JPEG 비디오 또는 GSM 오디오 등)
Sequence Number	◦손실한 패킷 발견에 사용 ◦시간표(Timestamp)를 가진 패킷들의 순서를 정함 ◦전송된 각 RTP 패킷에 대해 1씩 증가
Timestamp	◦데이터가 패킷에 저장되는 순간을 기록 ◦값은 Payload에 따라 다름
SSRC Identifier	◦RTP 스트림의 소스와 동기를 맞추기 위한 식별자 ◦SSRC는 임의로 선택
CSRC Identifiers	◦RTP 세션에 기여한 여러 스트림의 소스를 표시 ◦CSRC Count에서 CSRC의 수를 표시하며 16개까지 펴시가능

[표 2.8] RTP 패킷의 헤더 내용

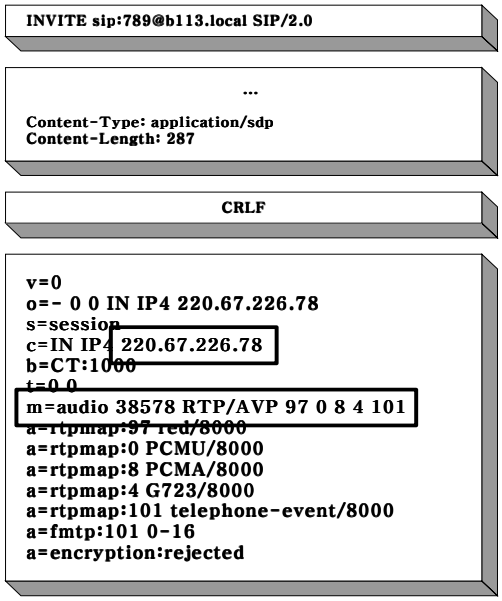
페이로드 형식(Payload Format)은 오디오 또는 비디오 부호화와 같은 특별한 페이로드의 내용을 RTP로 전송하는 방법을 정의한다. 현재 RTP를 위해 정의된 페이로드 종류[16]가 [표 2.9]에 나열되어 있다. RTP/AVP(Audio/Video Profile)는 응용 프로그램에서 사용될 페이로드 형식의 집합을 위한 페이로드 숫자를 할당한다.

페이로드 종류	코덱 이름	오디오/비디오 (A/V)	표본화 속도 (Hz)	채널 수
0	PCMU	A	8000	1
1	1016	A	8000	1
2	G.721	A	8000	1
3	GSM	A	8000	1
5	DVI4	A	8000	1
6	DVI4	A	16000	1
7	LPC	A	8000	1
8	PCMA	A	8000	1
9	G.722	A	8000	1
10	L16	A	44100	2
11	L16	A	44100	1
12	QCELP	A	8000	1
13	CN	A	8000	1
14	MPA	A	90000	
15	G.728	A	8000	1
16	DVI4	A	11025	1
17	DVI4	A	22050	1
18	G.729	A	8000	1
25	CelB	V	90000	
26	JPEG	V	90000	
28	nv	V	90000	
31	H.261	V	90000	
32	MPV	V	90000	
33	MP2T	AV	90000	
72-76	Reserved	N/A	N/A	N/A
96-127	Dynamic	?		

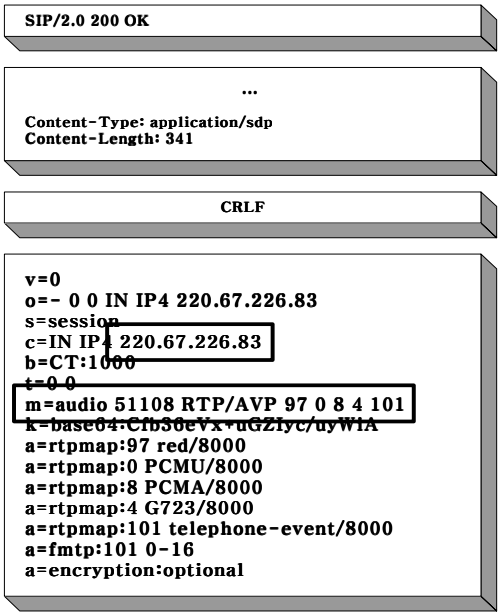
[표 2.9] 페이로드 종류

SIP는 SIP 메시지의 메시지 본문에 기술된 SDP 메시지의 접속 정보, 미디어 종류, 전송 주소(포트 번호), 미디어 특성 정보 등을 참조하여 세션에 참가하는 클라이언트에게 RTP를 이용하여 실시간 데이터를 전송한다. 실시간 데이터의 전달 과정을 상세히 설명하면 다음과 같다.

[그림 2.18]과 [그림 2.19]는 SIP 요청 메시지와 SIP 응답 메시지의 메시지 본문에 기술된 SDP 메시지를 보여주고 있고, [표 2.10]은 송신자의 SDP 메시지와 수신자의 SDP 메시지의 주요 필드를 세부적으로 나열한 것이다.



[그림 2.18] 송신자의 SDP 메세지



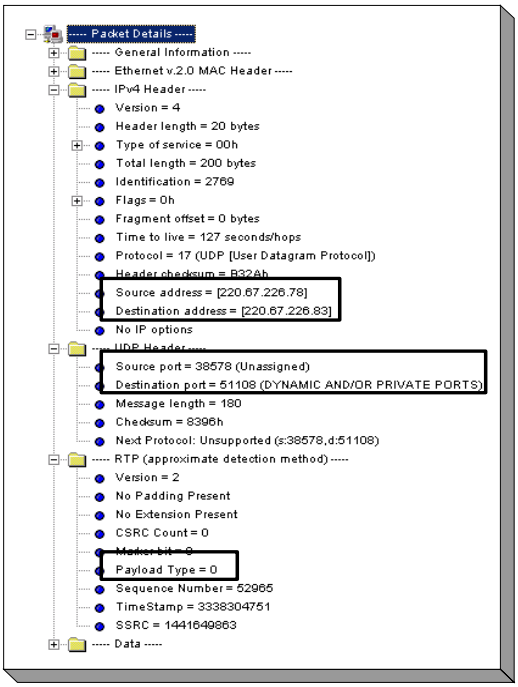
[그림 2.19] 수신자의 SDP 메세지

	속 성	값
송신자의 SDP		
	접속 정보	220.67.226.78
	미디어 종류	오디오
	전송 주소	38578
	전송 계층 프로토로	RTP/AVP (Audio/Video Profile를 사용하는 RTP)
	미디어 페이로드 종류	97 0 8 4 101
수신자의 SDP		
	접속 정보	220.67.226.83
	미디어 종류	오디오
	전송 주소	51108
	전송 계층 프로토로	RTP/AVP (Audio/Video Profile를 사용하는 RTP)
	미디어 페이로드 종류	97 0 8 4 101

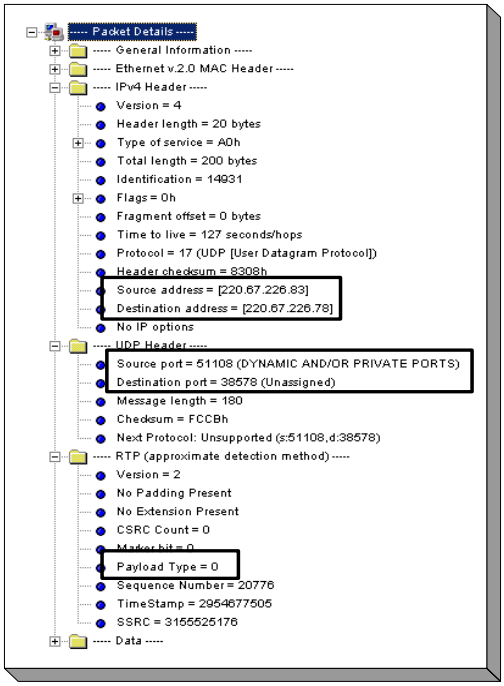
[표 2.10] 송신자와 수신자의 주요 필드의 상세 정보

[표 2.10]의 필드의 상세 정보를 통하여 송신자가 수신자에게 또는 수신자가 송신자에게 실시간 데이터를 전송할 때 다음과 같은 정보를 이용한다. 이와 같은 사실은 일반적인 프로토콜 분석기를 사용하여 패킷 헤더를 분석한 결과 확인할 수 있었다. [그림 2.20]과 [그림 2.21]은 송신자에서 발생된 RTP 패킷과 수신자에서 발생된 RTP 패킷을 분석한 결과이다.

- 소스 및 목적지 주소 : 220.67.226.78 ⇒ 220.67.226.83,
  220.67.226.83 ⇒ 220.67.226.78
- 소스 및 목적지 포트 : 38578 ⇒ 51108, 51108 ⇒ 38578
- 전송 계층 프로토콜 : RTP/AVP(Audio/Video Profile를 사용하는 RTP)
- 미디어 페이로드 타입 : 97, 0, 8, 4, 101



[그림 2.20] 송신자의 RTP 패킷 분석

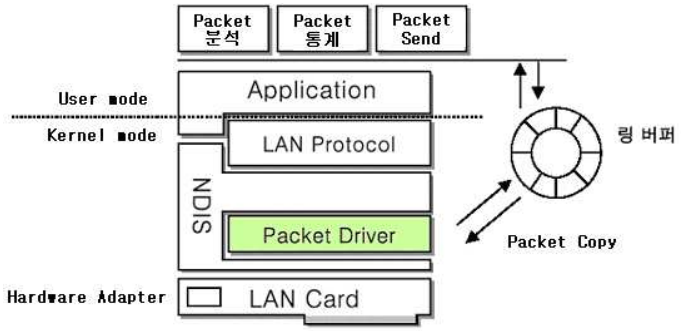


[그림 2.21] 수신자의 RTP 패킷 분석

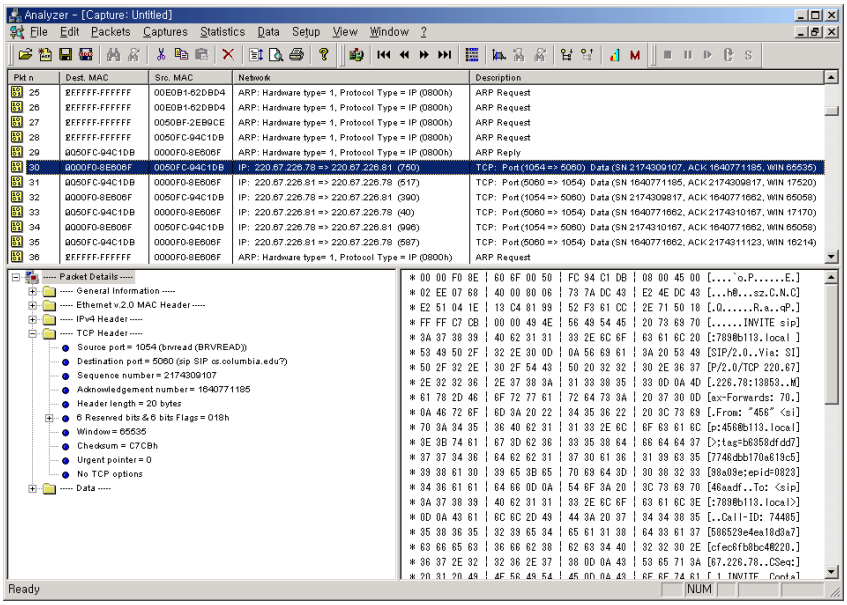
본 논문에서는 SIP의 실시간 데이터 전송에 사용되는 RTP 패킷을 분석하여, 구조화된 SDP 메시지와 연계하여 실시간 데이터의 미디어 종류와 미디어 데이터의 량을 유추하고 분석하여 ***미디어 별 통계를 측정***할 수 있도록 한다.

2.4 트래픽 분석기

일반적인 트래픽 분석기는 이더넷(Ethernet)의 특성인 동보 기능(broadcasting)을 이용하여 LAN에 흘러 다니는 모든 패킷들을 실시간으로 수집하는 기능과 수집된 패킷의 세부 내용을 보여주는 프로토콜 분석(protocol analyze) 기능과 망 사용 통계를 보여주는 트래픽 측정 기능 등을 가지고 있다. [그림 2.22]는 일반적인 트래픽 분석기의 구조이고 [그림 2.23]은 프로그램 실행 모습 예이다.



[그림 2.22] 일반적인 트래픽 분석기의 구조



[그림 2.23] 일반적인 트래픽 분석기 실행 예

구조와 기능에 대한 설명은 다음과 같다.

가. 실시간 패킷 수집 기능

이더넷은 이론적으로는 LAN 카드에서 LAN에 흘러 다니는 모든 패킷을 볼 수 있다. 그러나 일반적인 TCP/IP 및 이더넷 응용 프로그램들이 수행되는 환경에서는 LAN 카드가 모든 패킷을 읽지만 해당 LAN 카드의 MAC(Medium Access Control) 주소에 맞는 패킷만 선택하여 상위 계층인 응용 프로그램으로 전달하고 다른 주소로 보내지는 패킷들은 버리게 된다. 하지만 망 진단 도구는 모든 패킷들을 볼 수 있어야 한다. 따라서 LAN 카드에서 모든 패킷을 수집하여 응용 계층으로 전달해 주는 기능이 필요한데 이러한 기능을 담당하는 부분이 패킷 드라이버(Packet Driver)[17][18]이다.

패킷 드라이버에서 수집된 패킷들은 링 버퍼(ring buffer) 또는 원형 큐(circular queue)를 통해서 응용 프로그램으로 전달되게 된다. 원형 큐를 사용하는 이유는 응용 프로그램에서는 패킷 드라이버에서 수집한 패킷을 순서대로 분석해야 하며 패킷 드라이버는 실시간으로 패킷을 수집하여 상위 계층인 응용 프로그램으로 순서에 맞게 전달해야 하기 때문이다.

실시간 패킷 수집은 주로 패킷 드라이버가 담당하지만 패킷 드라이버는 수집된 패킷을 원형 큐를 사용하여 응용 프로그램으로 전달하게 되므로 응용프로그램에서는 전달된 패킷들을 분석을 위해 메모리에 복사하여 보관한다.

나. 프로토콜 분석 기능

프로토콜 분석은 저장된 패킷의 헤더를 분석하여 각각의 헤더 내용을 프로토콜의 필드별로 설명하는 기능이다. [그림 2.23]에서와 같이 패킷 목록과 헤더 분석 부분, 그리고 패킷의 Dump 내용이 한 화면에 나온다

다. 조건적 패킷 수집을 위한 필터(filter) 지정 기능

필터링은 트래픽 분석기가 특정 조건에 맞는 패킷들만 수집하여 분석이 가능

하도록 사용자에게 선택권을 주는 기능이다. 트래픽 분석기에서는 사용자에게 패킷 필터를 지정하도록 메뉴를 제공하며 지정된 필터를 패킷 드라이버에 설정함으로써 패킷 드라이버는 조건에 맞는 패킷만을 수집하여 응용 프로그램에게 전달한다.

라. 트래픽 통계(속도, 패킷 수 등) 표시 기능

트래픽 통계 표시 기능은 프로토콜 분석 기능과 더불어 트래픽 분석기의 가장 핵심적인 기능이다. 트래픽 통계 표시는 필터링 조건에 맞는 패킷들의 사용 통계를 다음과 같은 세부 조건 및 기준으로 표시하는 것을 말한다.

- 패킷 길이 또는 트래픽(Bits, Bytes, 패킷 수) 기준
- 레이어(Layer) 별 필터링 조건(Mac 레이어, 응용 레이어 등)
- 통계를 표시하는 방법 설정(수평 또는 수직 그래프, 원 그래프 등)
- 통계를 표시하는 시간 간격 설정

위와 같이 일반적 트래픽 분석기의 구조에 대해 알아보았다. 트래픽 분석기가 SIP 기반 응용 프로그램에서 발생되는 트래픽을 분석하기 위해서는 SIP 프로토콜과 SIP 프로토콜과 관련된 인터넷 멀티미디어 프로토콜을 분석하는 기능을 가져야한다. 또한 트래픽 분석기는 SIP 프로토콜 연동의 기본 단위인 트랜젝션(Transaction)을 관리할 수 있는 구조가 필요하다.

다음 장에서는 본 논문에서 제안하는 새로운 트래픽 분석기에 대하여 설명한다.

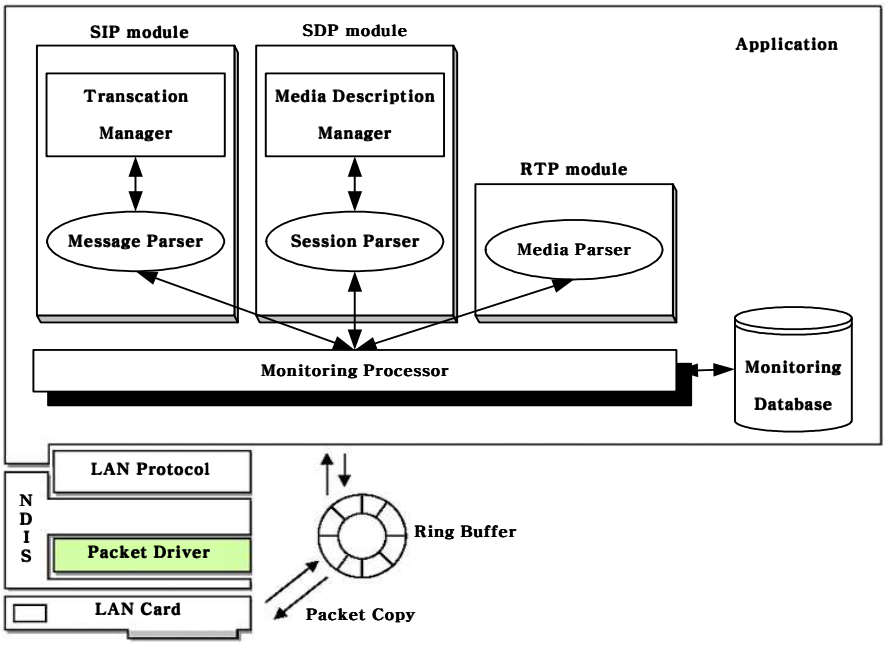
제 3장 STAT 설계 및 구현

본 장에서는 2장에서 소개된 SIP 기반 응용 프로그램에서 발생되는 트래픽을 분석하기 위해서 일반적인 트래픽 분석기가 가져야할 구조와 기능을 반영하여 STAT 설계 및 구현에 대해 기술한다.

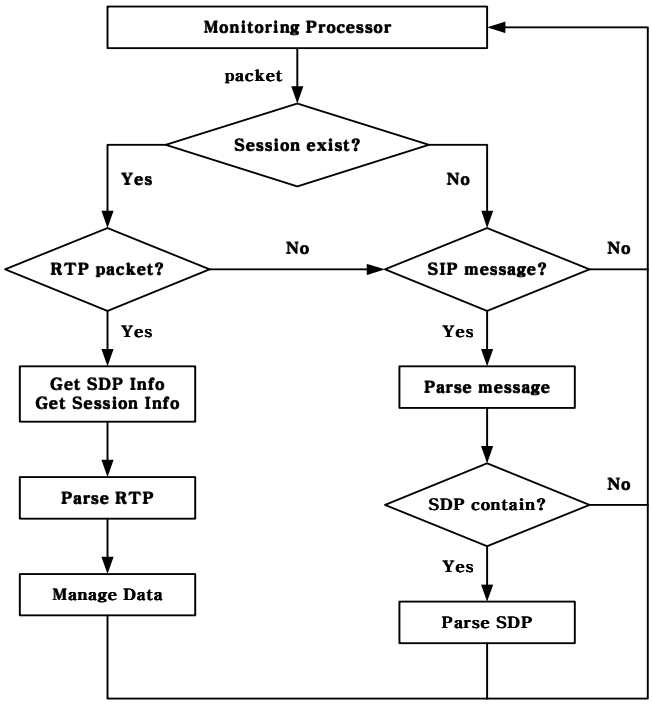
본 논문에서 제안된 트래픽 분석 도구는 Windows를 운영체제로 사용하는 일반 PC에서 수행되는 응용 프로그램으로 패킷 드라이버에서 정의된 API(Application Programming Interface)와 Windows의 사용자 GUI(Graphical User Interface) 함수를 이용하여 개발되었다.

3.1 STAT 구조

본 논문에서 제안하는 새로운 트래픽 분석 도구는 [그림 3.1]과 같이 패킷 드라이버와 응용 프로그램으로 구성되어 있으며 본 논문에서 구현한 부분은 응용 프로그램 부분이다. 응용 프로그램은 모니터링 처리기(Monitoring Processor)에서 SIP 모듈, SDP 모듈과 RTP 모듈을 호출하여 패킷 드라이버를 통하여 실시간으로 수집되는 패킷을 분석하며, 분석된 데이터는 모니터링 저장소(Monitoring Database)에 저장된다. [그림 3.2]는 제안된 새로운 트래픽 분석 도구의 흐름도이다.



[그림 3.1] STAT 구조



[그림 3.2] STAT 흐름도

STAT는 다음과 같이 크게 4가지 부분으로 구성되어 있고 설명은 다음과 같다.

가. 모니터링 처리기

모니터링 처리기는 패킷 드라이버로부터 전달된 패킷의 종류를 구분하고 SIP, SDP, RTP 모듈로 전달하여 각 모듈이 분석할 수 있도록 하고, 각 모듈에 의해 분석된 데이터는 모니터링 저장소(Monitoring Database)에 저장하여 트래픽을 분석하는데 사용한다. 모니터링 저장소에 저장되는 데이터들은 SIP 세션 색인 번호, 실시간 데이터 전송에 사용되는 호출자 및 호출 받은 사용자의 미디어 종류 및 미디어 데이터 량 등이 있다. 또한 모니터링 처리기는 SIP 메시지 또는 SIP 메시지 본문에 기술되는 세션 정보의 크기로 인하여 패킷이 분할되는 현상을 관리하고 처리한다.

나. SIP 모듈

SIP 모듈은 메시지 분석기(Message Parser)와 트랜젝션 관리자(Transaction Manager)로 구성되어 있다. 메시지 분석기는 모니터링 처리기에서 전달된 SIP 요청 메시지 및 SIP 응답 메시지의 헤더 및 헤더 내용을 분석하고 주요 헤더들을 구조화하여 저장한다. SIP 메시지에서 구조화되는 헤더들은 SIP 수신 명령(Method) 또는 SIP 상태 코드(Status Code)와 트랜젝션을 구분하는데 사용되는 호 식별자(Call-ID), 호출자 주소, 호출 받은 사용자 주소, 순서 번호 등이 있다. 트랜젝션 관리자는 메시지 분석기에 의해 저장된 구조화된 SIP 메시지를 이용하여 세션 상태를 확인하고 세션 설정에 대한 트랙젝션이 완료되었을 경우에 모니터링 처리기에게 해당 세션에 대한 트래픽을 분석하도록 알려주며 세션 별로 구조화된 SIP 메시지를 저장한다. 트랙젠션 관리자가 추가적으로 세션 별로 관리하는 데이터들은 호출자의 세션 정보, 호출 받은 사용자의 세션 정보, 세션 설정 및 종료 상태를 나타내는 세션 상태 정보 등이 있다.

다. SDP 모듈

SDP 모듈은 세션 분석기(Session Parser)와 미디어 기술 관리자(Media Description Manager)로 구성되어 있다. 세션 분석기는 SIP 모듈의 메시지 분석기에서 전달된 SIP 본체(Entity) 헤더의 값을 이용하여 SDP 메시지의 필드 및 필드 내용

을 분석하고 주요 필드들을 구조화하여 저장한다. SDP 메시지에서 구조화되는 필드들은 접속 정보, 미디어 안내 정보(미디어 이름, 포트 번호, 사용할 전송계층 프로토콜, 미디어 형식), 미디어 속성(페이로드 종류, 코덱 이름, 표본화 속도) 등이 있다. 미디어 기술 관리자는 세션 분석기에 의해 저장된 세션 정보를 색인 목록을 만들어 관리하고 세션 정보를 SIP 모듈의 트랜젝션 관리자에게 전달하여 트랜젝션 관리자가 호출자 및 호출 받은 사용자의 세션 정보를 관리할 수 있도록 한다.

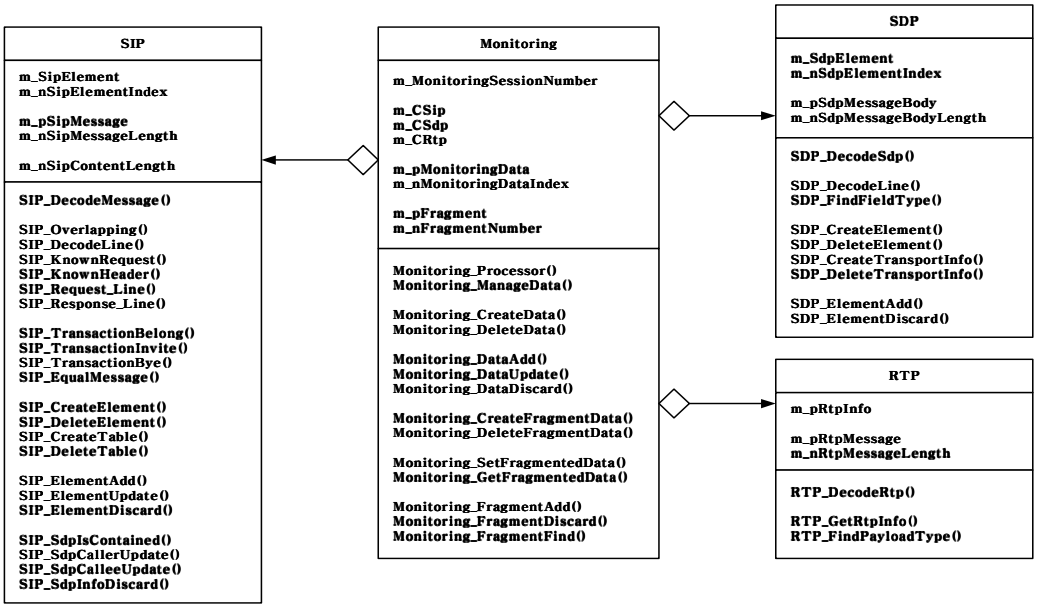
라. RTP 모듈

RTP 모듈은 모니터링 처리기에 의해 전달된 RTP 패킷의 헤더를 분석하고 저장한다. RTP 모듈의 미디어 파서(Media Parser)는 저장된 RTP 패킷의 페이로드의 종류를 식별하고 모니터링 처리기에게 페이로드의 종류와 미디어 데이터 량을 전달하여 트래픽을 분석할 수 있도록 한다.

3.2 STAT 클래스 설계

본 논문에서 구현한 STAT는 객체지향 개념을 근간으로 하는 컴포넌트 기반의 소프트웨어 개발 방법을 적용하여 설계하였다. STAT는 [그림 3.3]과 같이 4개의 클래스로 구성되어 있으며 모니터링 클래스는 SIP, SDP, RTP 클래스의 객체를 가지고 있다. 이와 같은 클래스들간의 관계는 특정 클래스에서 모든 데이터를 저장하고 처리하는 것이 아니라 각각의 클래스에서 데이터를 나누어 관리하기 위한 것이다. 다음은 각각의 클래스에서 처리하는 데이터의 종류를 설명한 것이다.

모니터링 클래스에서는 미디어 별 통계에 관련된 데이터를 저장하고 설정된 세션과 세션 설정과 관련된 SIP 메시지들은 SIP 클래스에서 관리하고 저장한다. SDP 클래스에서는 세션 정보만을 저장하고 미디어 데이터 처리는 RTP 클래스에서 담당한다. [표 3.1]은 각 클래스의 중요 함수를 기능별로 나열한 것이다.

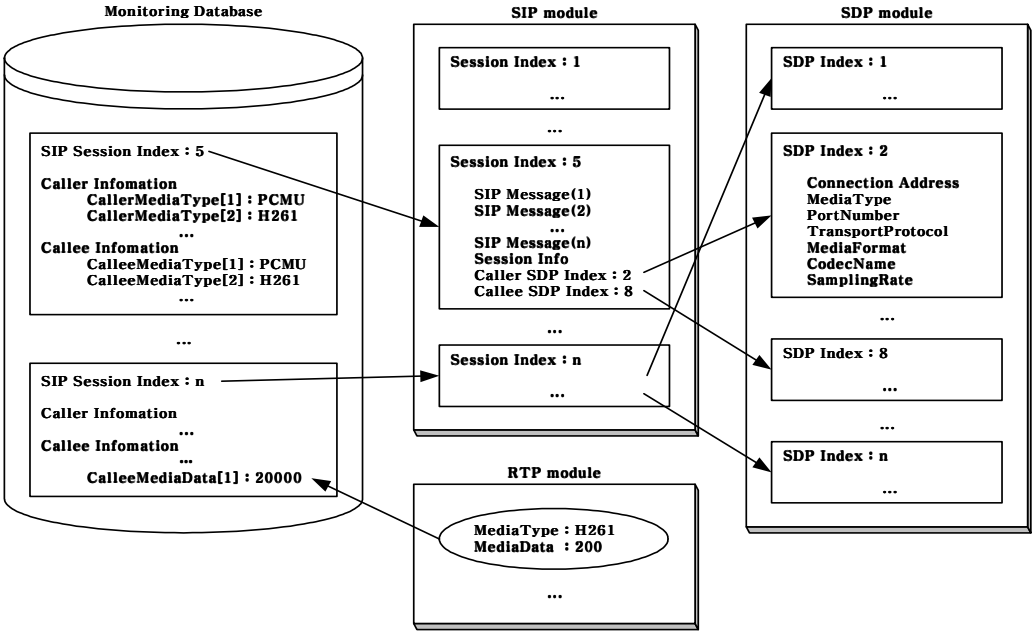


[그림 3.3] STAT 클래스 다이어그램

기 능	관련 함수
분석 기능	Monitoring_Processor()
	SIP_DecodeMessage()
	SDP_DecodeSdp()
	RTP_DecodeRtp()
데이터 관리 기능	Monitoring_ManageData() Monitoring_FragmentFind()
	Monitoring_DataAdd() Monitoring_DataUpdate() Monitoring_DataDiscard()
	Monitoring_FragmentAdd() Monitoring_FragmentDiscard()
	SIP_ElementAdd() SIP_ElementUpdate() SIP_ElementDiscard()
	SDP_ElementAdd() SDP_ElementDiscard()
세션 처리 기능	SIP_TransactionBelong() SIP_TransactionInvite() SIP_TransactionBye()
	SIP_GetSessionIndex() SIP_GetSessionNumber()

[표 3.1] STAT 클래스들의 기능별 중요 함수

STAT에서는 각 객체에 저장되어 있는 데이터에 접근하기 위해서 색인 방식을 이용하였으며 [그림 3.4]는 STAT의 자료구조를 보여주고 있다.



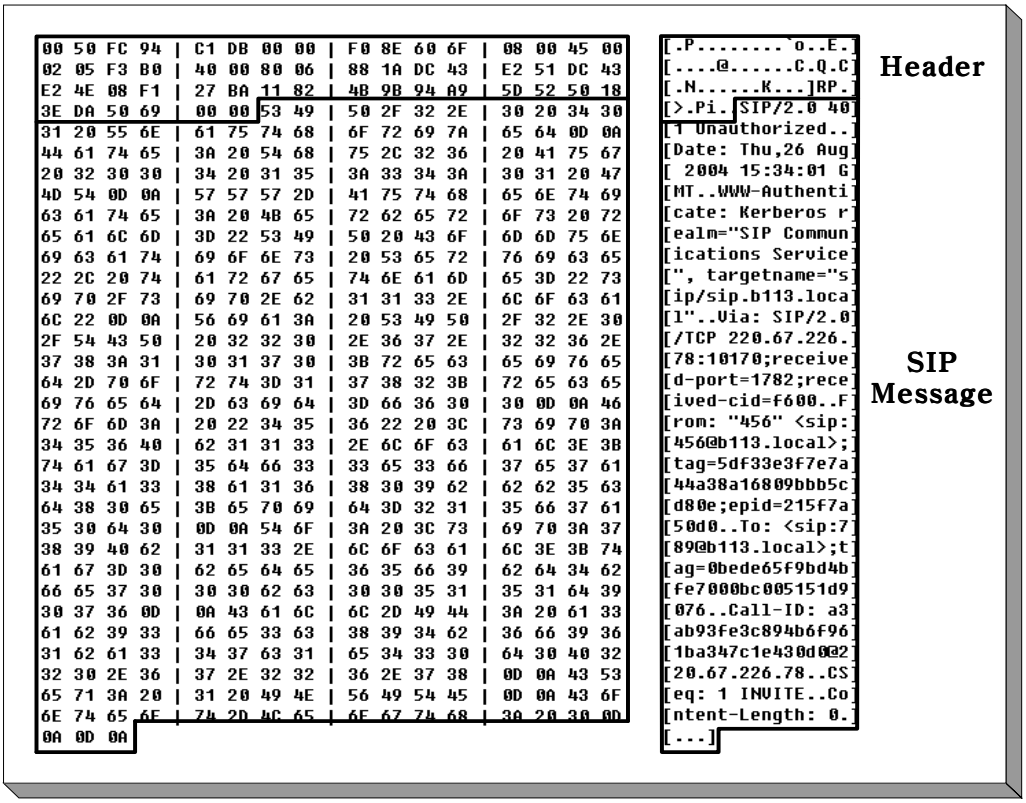
[그림 3.4] STAT 자료구조

[그림 3.4]에서 보듯이 모니터링 저장소는 SIP 세션 색인 번호와 미디어 별 통계 데이터를 가지고 있고, SIP 모듈은 세션을 구성하는 SIP 메시지들, 세션 상태 정보와 호출자 및 호출 받은 사용자의 SDP 세션 정보 색인 번호를 가지고 있으며, SDP 모듈은 각 세션 정보만을 가지고 있다. RTP 모듈은 실시간 데이터의 미디어 종류와 미디어 데이터 량 데이터를 모니터링 저장소에 전달한다.

3.3 모듈별 상세 설계 및 구현

3.3.1 SIP 모듈

SIP 패킷은 이더넷 헤더, IP 헤더, TCP 헤더 또는 UDP 헤더와 SIP 메시지로 구성되어 있다. SIP 패킷의 SIP 메시지는 이더넷 패킷의 응용 데이터 부분에 실려온다. [그림 3.5]는 SIP 응답 메시지가 포함된 이더넷 패킷의 예이다.



[그림 3.5] SIP 응답 메시지가 포함된 이더넷 패킷의 예

[그림 3.5]와 같은 이더넷 패킷이 SIP 패킷인지 판단하기 위해서는 [그림 3.5]의 헤더(Header)를 이용하여 포트 정보를 분석하는 과정이 필요하다. [그림 3.5]의 이더넷 패킷은 근원지 포트 번호로 2289(0x08F1)와 목적지 포트 번호로 10170(0x27BA)을 사용하여 전송된 SIP 패킷이다. 하지만 SIP 패킷은 포트 번호 5060(0x13C4)을 사용하여 전송된다. [그림 3.5]의 SIP 패킷에 포트 번호 5060이 사용되지 않았던 이유는 SIP 서버가 전송한 패킷이기 때문이다. 따라서 SIP 패킷을 판단하는데 있어 포트 정보 외에 SIP 메시지의 시작 줄 부분을 분석하는 과정이 필요하다. SIP 메시지의 시작 줄 부분 분석을 통하여 SIP 패킷인지 판단 할 수 있을 뿐만 아니라 SIP 패킷의 메시지 종류인 SIP 요청 메시지와 응답 메시지를 구분할 수 있다.

본 논문에서 구현된 SIP 모듈은 [그림 3.6]의 패킷 정보 구조체를 사용하여 이더넷 패킷의 중요 헤더 정보와 패킷 정보를 분석하여 저장하며 [그림 3.7]의

SIP_INFO 구조체를 사용하여 SIP 메시지의 시작 줄 부분을 분석하여 저장한다. SIP_INFO 구조체는 SIP 메시지의 시작 줄에 대응되는 부분으로써 SIP 메시지가 요청 메시지이면 SIP_INFO 구조체에는 신호 명령 정보가 저장되고 응답 메시지이면 응답 코드 정보가 저장된다.

```
////////////////////////////////////
// struct Packet Information

typedef struct _PACKET_INFO
{
    BOOL        bSip;
    BOOL        bRtp;
    UINT        nDataLen;
    UINT        nSrcAddr;
    USHORT      nSrcPort;
    UINT        nDstAddr;
    USHORT      nDstPort;
} PACKET_INFO, *PACKET_INFO_POINTER;
```

[그림 3.6] 패킷 정보 구조체

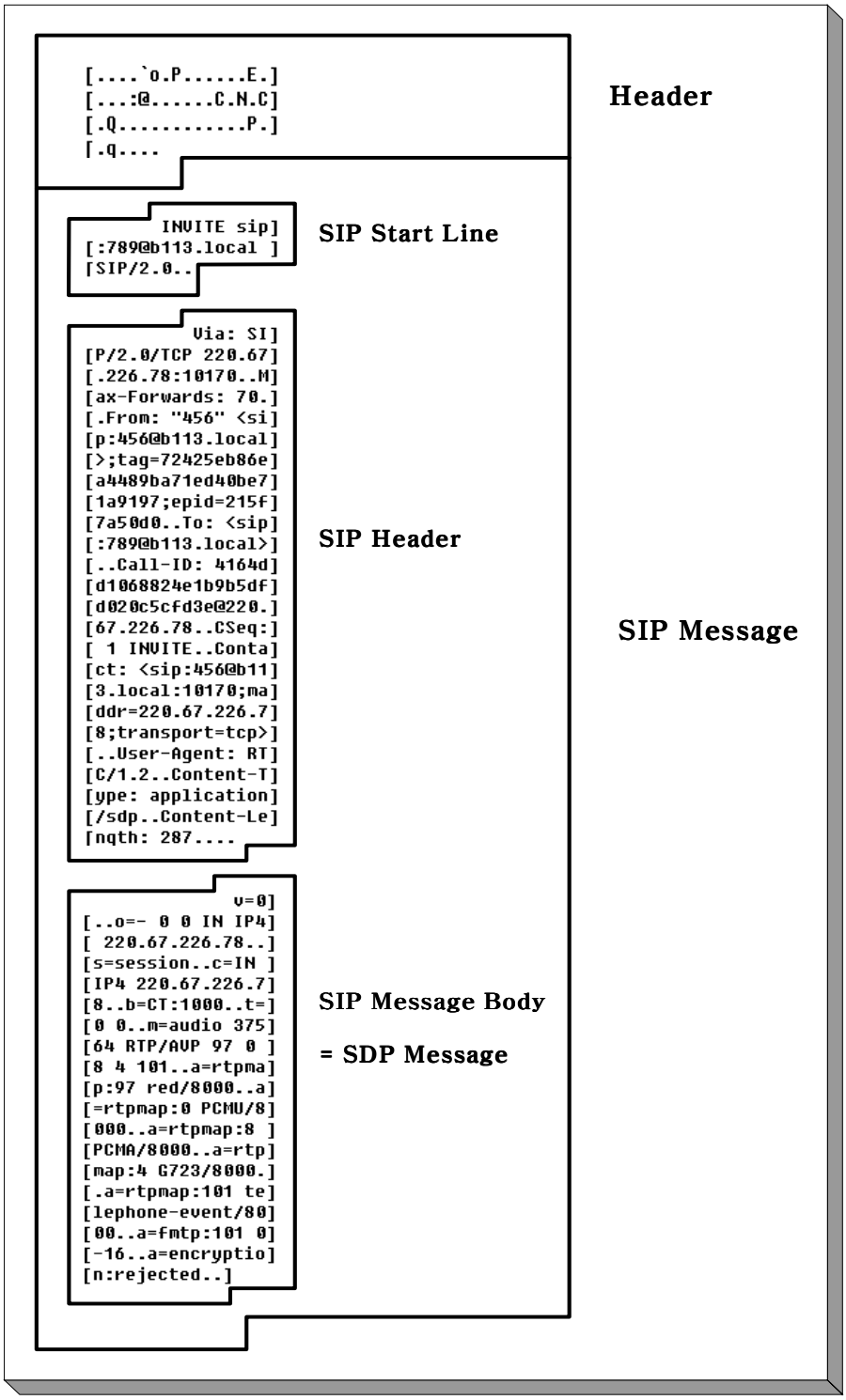
```
////////////////////////////////////////////////
// SIP & Call Information

typedef struct _SIP_INFO
{
    PCHAR   pRequest_Method;
    int     nResponse_Code;
} SIP_INFO, *SIP_INFO_POINTER;

typedef enum _TRANSACTION
{
    ETC,
    REQUEST,
    PROVISIONAL_RESPONSE,
    FINAL_RESPONSE
} TRANSACTION, *TRANSACTION_POINTER;

typedef struct _SIP_CALL_INFO
{
    char            Call_ID[SIP_CALL_ID_SIZE];
    char            From[SIP_URI_MAX_SIZE];
    char            To[SIP_URI_MAX_SIZE];
    UINT            nCSequenceNumber;
    TRANSACTION     Transaction;
} SIP_CALL_INFO, *SIP_CALL_INFO_POINTER;
```

[그림 3.7] SIP 메시지를 저장하기 위한 구조체

SIP 메시지의 시작 줄 다음에는 하나 이상의 헤더가 온다. SIP 헤더의 구조는 헤더 이름, ':' 문자, 공백 문자, 헤더 내용, 헤더의 끝을 나타내는 CRLF로 구성된다. 구현된 SIP 모듈은 SIP 헤더 구조를 참조하여 각각의 헤더와 헤더 내용을 분석하고 [그림 3.7]의 SIP_CALL_INFO 구조체를 사용하여 정보를 저장한다. SIP_CALL_INFO 구조체는 SIP 메시지의 헤더에 대응되는 부분으로써 SIP_CALL_INFO 구조체에는 SIP 트랜젝션을 구분하는데 필요한 정보를 저장한다. [그림 3.8]은 SIP 모듈의 핵심 함수인 메시지 분석기의 SIP 메시지를 분석 과정을 기술한 것이고, [그림 3.9]는 SIP 요청 메시지가 포함된 이더넷 패킷을 세부적으로 분할한 것으로 SIP 헤더의 구조를 확인할 수 있다.

```
SIP 메세지를 저장하기 위하여 SIP_INFO 구조체와 SIP_CALL_INFO 구조체를 생성한다.

SIP 메세지의 첫번째 줄(SIP 시작 줄)을 읽어 SIP 메세지의 종류를 구분한다.

switch( SIP 메세지의 종류 )
{
    SIP 메세지의 종류에 따라 각 신호 명령 또는 응답 코드를 저장한다.

    case SIP_TYPE_REQUEST :     SIP_INFO에 SIP 신호 명령을 저장한다.
    case SIP_TYPE_RESPONSE :    SIP_INFO에 SIP 응답 코드를 저장한다.
    case SIP_ETC :              에러
}

SIP 메세지에서 한줄씩 읽어와 SIP Header를 분석한다.

while( SIP 메세지 길이 > 분석된 길이 )
{
    SIP 메세지의 한줄을 읽는다.

    읽은 줄에서 ':' 문자를 찾고 SIP 헤더인지 확인하고 Index로 변환한다.

    switch( SIP 헤더 Index )
    {
        각 SIP 헤더 종류에 따라 각 SIP 헤더 내용을 저장한다.

        case SIP_HEADER_From :      SIP_CALL_INFO에 From 헤더 내용을 저장한다.
        case SIP_HEADER_To :        SIP_CALL_INFO에 To 헤더 내용을 저장한다.
        case SIP_HEADER_Call_ID :   SIP_CALL_INFO에 Call-ID 헤더 내용을 저장한다.
        case SIP_HEADER_CSeq :      SIP_CALL_INFO에 CSeqeuence 헤더 내용을 저장한다.
        ...
    }
}

분석된 SIP 메세지를 저장한다.
```

[그림 3.8] SIP 메시지 분석 과정



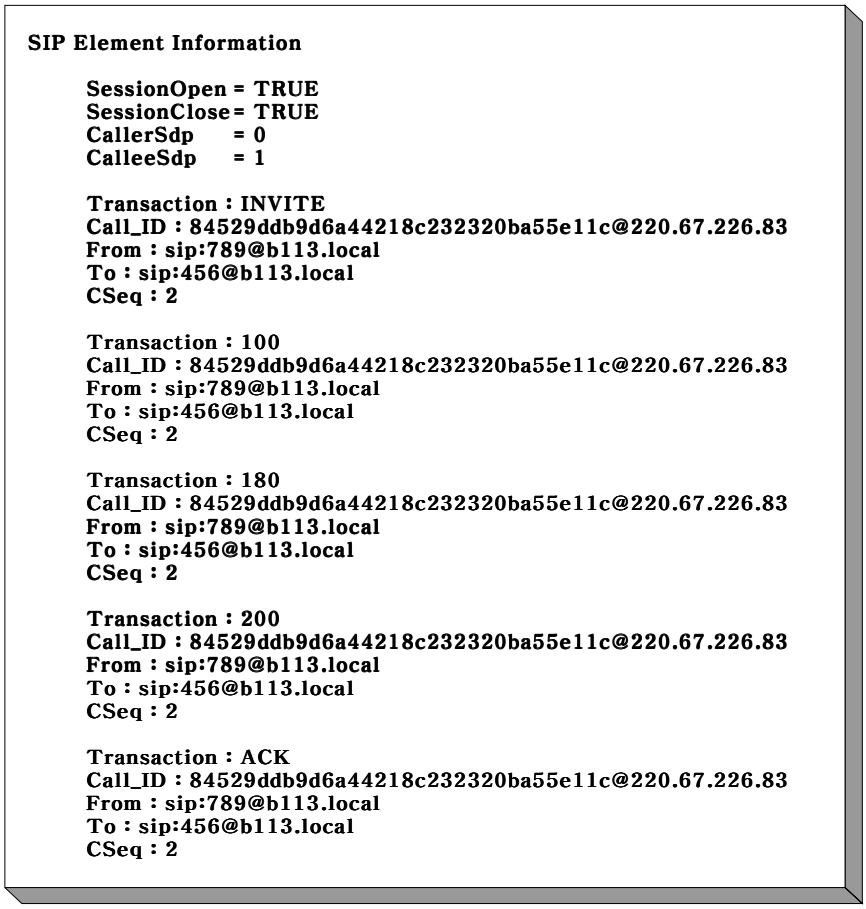
[그림 3.9] SIP 요청 메시지가 포함된 이더넷 패킷의 구조

[그림 3.9]의 SIP 요청 메시지는 SIP 헤더 다음에 SIP 메시지 본체가 포함되어 있다. SIP 메시지 본체가 존재하는 경우는 크게 두 가지로 나눌 수 있다. 하나는 SIP 요청 메시지 중에 상대편을 세션에 참석시키기 위하여 호출하는 INVITE 메시지와 다른 하나는 이에 대한 응답으로 SIP 응답 메시지 중에 상태 코드가 200인 응답 메시지인 경우이다. 구현된 SIP 모듈의 메시지 분석기는 SIP 메시지 분석 중 SIP 본체 헤더, Content-Type과 Content-Length를 발견하면 헤더의 내용을 저장하는 구조를 가진다. SDP 모듈은 저장된 SIP 본체 헤더 정보를 이용하여 세션 정보를 분석한다. [표 3.2]는 SIP 메시지 분석기가 이용하는 중요 함수들이다.

함 수	설 명
SIP_DecodeLine()	◦ SIP 메시지의 시작 줄을 분석하여 SIP 요청 메시지인지 SIP 응답 메시지인지 알아낸다.
SIP_Request_Line() SIP_Response_Line()	◦ SIP 메시지의 신호 명령과 상태 코드를 가져온다.
SIP_KnownHeader()	◦ SIP 메시지의 헤더 내용을 분석하기 위하여 헤더를 식별한다.
SIP_TransactionBelong() SIP_TransactionInvite() SIP_TransactionBye()	◦ SIP 메시지가 특정 트랜젝션에 속하는지 판단한다. ◦ 트랜젝션이 세션 설정 또는 세션 종료인지 판단한다.
SIP_CreateElement() SIP_DeleteElement()	◦ 세션을 관리하기 위한 구조를 생성하고 삭제한다.
SIP_CreateTable() SIP_DeleteTable()	◦ SIP 메시지들을 저장하기 위한 구조를 생성하고 삭제한다.
SIP_ElementAdd() SIP_ElementUpdate() SIP_ElementDiscard()	◦ 세션을 추가, 변경 및 삭제를 수행한다.
SIP_SdpIsContained() SIP_SdpCallerUpdate() SIP_SdpCalleeUpdate() SIP_SdpInfoDiscard()	◦ SIP 메시지에 세션 정보가 포함되어 있을 경우에 SIP 메시지 내에 포함된 세션 정보의 크기를 반환한다. ◦ 해당 세션에 대한 호출자의 세션 정보를 변경한다. ◦ 해당 세션에 대한 호출 받은 사용자의 세션 정보를 변경한다. ◦ 해당 세션에서 호출자 및 호출 받은 사용자의 세션 정보를 삭제한다.

[표 3.2] SIP 모듈의 중요 함수

[그림 3.10]은 SIP 모듈에 의해 세션별로 관리되는 정보를 보여주고 있으며, 세션별 정보는 세션을 구성하는 SIP 메시지들, 세션 설정 정보와 호출자 및 호출 받은 사용자의 SDP로 구성되어 있다.



[그림 3.10] SIP 세션별 정보

3.3.2 SDP 모듈

SDP 모듈은 SDP 메시지의 필드 및 필드 내용을 분석하고 주요 필드들을 구조화하여 저장한다. SIP 메시지의 메시지 본체에는 SDP에 의해 세션에 대한 정보가 기술된다. [그림 3.9]에서 보듯이 SDP 메시지의 구조는 필드 이름, '=' 문자와 속성 이름으로 구성된다. 본 논문에서 구현된 SDP 모듈은 SDP의 이런 특성을 이용하여 필드와 속성을 분석하고 [그림 3.11] TRANSPORT_INFO 구조체를 사용하여 저장한다. TRANSPORT_INFO 구조체에는 접속 정보, 미디어 안내 정보(미디어 이름, 포트 번호, 사용할 전송계층 프로토콜, 미디어 형식)와 미디어 속성(페이로드 종류, 코덱 이름, 표본화 속도)을 저장한다.

```
////////////////////////////////////////////////////////////////////////////////////////////
// SDP Transport Information and Table

typedef struct _TRANSPORT_INFO
{
    PCHAR   pConnectionAddress;
    PCHAR   pMediaName[SDP_TRANSPORT_INFO_CHANNEL_SIZE];
    USHORT  pMediaPortNumber[SDP_TRANSPORT_INFO_CHANNEL_SIZE];
    USHORT  pMediaNumberOfPorts[SDP_TRANSPORT_INFO_CHANNEL_SIZE];
    PCHAR   pMediaProtocol[SDP_TRANSPORT_INFO_CHANNEL_SIZE];
    SHORT   pMediaTypes[SDP_TRANSPORT_INFO_CHANNEL_SIZE][SDP_TRANSPORT_INFO_MEDIA_TYPES_SIZE];
    PCHAR   pMediaTypesValue[SDP_TRANSPORT_INFO_CHANNEL_SIZE][SDP_TRANSPORT_INFO_MEDIA_TYPES_SIZE];
    UINT    nMediaTypesNumber[SDP_TRANSPORT_INFO_CHANNEL_SIZE];
    UINT    nMediaNumber;
} TRANSPORT_INFO, *TRANSPORT_INFO_POINTER;

typedef struct _SDP_ELEMENT
{
    TRANSPORT_INFO_POINTER      pTI;
} SDP_ELEMENT, *SDP_ELEMENT_POINTER;
```

[그림 3.11] 세션 정보 저장에 사용되는 구조체

SDP 메시지를 구성하는 필드들은 필드 종류에 따라 속성이 기술되는 방식이 다르기 때문에 각 필드를 분석할 수 있는 함수를 구현하였다. [표 3.3]은 SDP 모듈에서 사용하는 함수들이다. [그림 3.12]는 SDP 메시지의 필드 종류 정보를 이용하여 각 필드를 분석하는 함수를 호출하는 SDP_DecodeLine() 함수를 보여주고 있다.

함 수	설 명
SDP_DecodeSdp()	◦ SDP 메시지의 필드 및 필드 내용을 분석한다.
SDP_FindFieldType()	◦ SDP 메시지의 각 필드의 종류를 알려준다.
SDP_DecodeLine()	◦ SDP 메시지의 필드 종류 정보를 이용하여 해당하는 각 필드를 분석할 수 있는 함수를 호출한다.
SDP_DecodeLine_O() SDP_DecodeLine_C() SDP_DecodeLine_B() SDP_DecodeLine_T() SDP_DecodeLine_R() SDP_DecodeLine_Z() SDP_DecodeLine_K() SDP_DecodeLine_SA() SDP_DecodeLine_M() SDP_DecodeLine_MA()	◦ SDP 메시지의 각 필드를 분석한다.
SDP_CreateElement() SDP_DeleteElement()	◦ 세션 정보를 저장하기 위한 구조를 생성하고 삭제한다.
SDP_ElementAdd() SDP_ElementDiscard()	◦ 세션 정보를 추가하고 삭제한다.

[표 3.3] SDP 모듈의 중요 함수

```
BOOL SDP_DecodeLine(..., TRANSPORT_INFO_POINTER pTi, SDP_FIELD_TYPE SFT)
{
    if( SFT == OWNER_CREATOR_AND_SESSION_IDENTIFIER )   return SDP_DecodeLine_O(..., pTi);
    if( SFT == CONNECTION_INFORMATION )                 return SDP_DecodeLine_C(..., pTi);
    if( SFT == BANDWIDTH_INFORMATION )                  return SDP_DecodeLine_B(..., pTi);
    if( SFT == TIME_THE_SESSION_IS_ACTIVE )             return SDP_DecodeLine_T(..., pTi);
    if( SFT == REPEAT_TIMES )                           return SDP_DecodeLine_R(..., pTi);
    if( SFT == TIME_ZONE_ADJUSTMENTS )                  return SDP_DecodeLine_Z(..., pTi);
    if( SFT == ENCRYPTION_KEY )                         return SDP_DecodeLine_K(..., pTi);
    if( SFT == SESSION_ATTRIBUTE_LINES )                return SDP_DecodeLine_SA(..., pTi);
    if( SFT == MEDIA_NAME_AND_TRANSPORT_ADDRESS )       return SDP_DecodeLine_M(..., pTi);
    if( SFT == MEDIA_ATTRIBUTE_LINES )                  return SDP_DecodeLine_MA(..., pTi);

    return TRUE;
}
```

[그림 3.12] SDP 필드 분석 함수

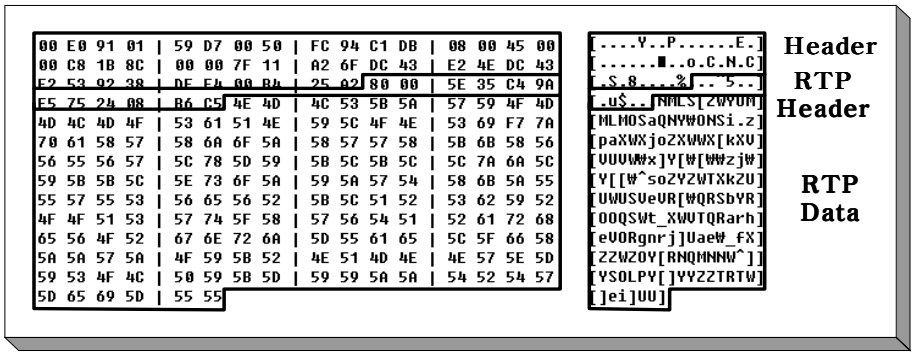
[그림 3.13]은 세션 설정 과정에 호출자와 호출 받은 사용자간에 전달된 세션 설정 정보를 SDP 모듈이 분석하여 저장한 정보들이다.

```
SDP Information
        220.67.226.83
        audio 5558/0 RTP/AVP 97 111 112 6 0 8 4 5 3 101
                red/8000
                SIREN/16000
                G7221/16000
                DVI4/16000
                PCMU/8000
                PCMA/8000
                G723/8000
                DVI4/8000
                GSM/8000
                telephone-event/8000
        video 19540/0 RTP/AVP 34 31
                H263/90000
                H261/90000

SDP Information
        220.67.226.78
        audio 37554/0 RTP/AVP 97 0 8 4 101
                red/8000
                PCMU/8000
                PCMA/8000
                G723/8000
                telephone-event/8000
        video 0/0 RTP/AVP 34
                (null)
```

[그림 3.13] 세션 설정 과정에 포함된 세션 정보

3.3.3 RTP 모듈

RTP 모듈은 RTP 패킷의 헤더를 분석하고 저장한다. RTP 패킷은 이더넷 헤더, UDP 헤더, RTP 헤더와 RTP 데이터로 구성되어 있다. [그림 3.14]는 SIP 패킷을 이용하여 세션이 설정된 후에 호출자와 호출 받은 사용자간에 미디어 데이터 전달에 사용되는 RTP 패킷의 예이다.



[그림 3.14] 미디어 데이터 전달에 사용되는 RTP 패킷의 예

RTP 모듈은 RTP 헤더를 분석하고 [그림 3.15]의 RTP_INFO 구조체를 사용하여 저장한다. RTP_INFO 구조체는 RTP 패킷의 고정 헤더의 정보와 RTP 데이터의 크기 정보를 가지고 있으며 nPayloadType 값을 이용하여 미디어의 종류를 판단한다.

```
////////////////////////////////
// RTP Header

typedef struct _RTP_INFO
{
    BOOL    bPaddingFlag;
    BOOL    bExtensionFlag;
    USHORT  nNumberOfMixedField;
    BOOL    bMarkerBit;
    USHORT  nPayloadType;

    USHORT  nSequenceNumber;
    UINT    nTimestamp;
    UINT    nSSRC;

    UINT    nDataLength;
} RTP_INFO, *RTP_INFO_POINTER;
```

[그림 3.15] RTP 정보 구조체

[그림 3.16]은 RTP 모듈에 의해 분석된 RTP 패킷의 헤더 정보와 미디어 데이터의 크기를 보여주고 있다.

```
RTP Information
      bPaddingFlag              = 0
      bExtensionFlag            = 0
      nNumberOfMixedField       = 0
      bMarkerBit                = 1
      nPayloadType              = 0
      nSequenceNumber           = 31813
      nTimestamp                = 3302184443
      nSSRC                     = 1027423292
      nDataLength               = 160

RTP Information
      bPaddingFlag              = 0
      bExtensionFlag            = 0
      nNumberOfMixedField       = 0
      bMarkerBit                = 0
      nPayloadType              = 0
      nSequenceNumber           = 31814
      nTimestamp                = 3302184603
      nSSRC                     = 1061043516
      nDataLength               = 160

RTP Information
      bPaddingFlag              = 0
      bExtensionFlag            = 0
      nNumberOfMixedField       = 0
      bMarkerBit                = 0
      nPayloadType              = 0
      nSequenceNumber           = 31815
      nTimestamp                = 3302184763
      nSSRC                     = 993671739
      nDataLength               = 160
```

[그림 3.16] 분석된 RTP 패킷 정보

3.3.4 모니터링 처리기

모니터링 처리기는 패킷 드라이버로부터 패킷이 전달되면 동작을 시작하고 PACKET_INFO 구조체에 저장된 정보를 이용하여 SIP, RTP 모듈을 호출하며 각 모듈에 의해 분석된 데이터는 [그림 3.15]의 모니터링 저장소인 MONITORING_DATA 구조체, [그림 3.18]의 분할된 SIP 메시지를 저장하는 FRAGMENT와 FRAGMENT_DATA 구조체를 사용하여 저장한다.

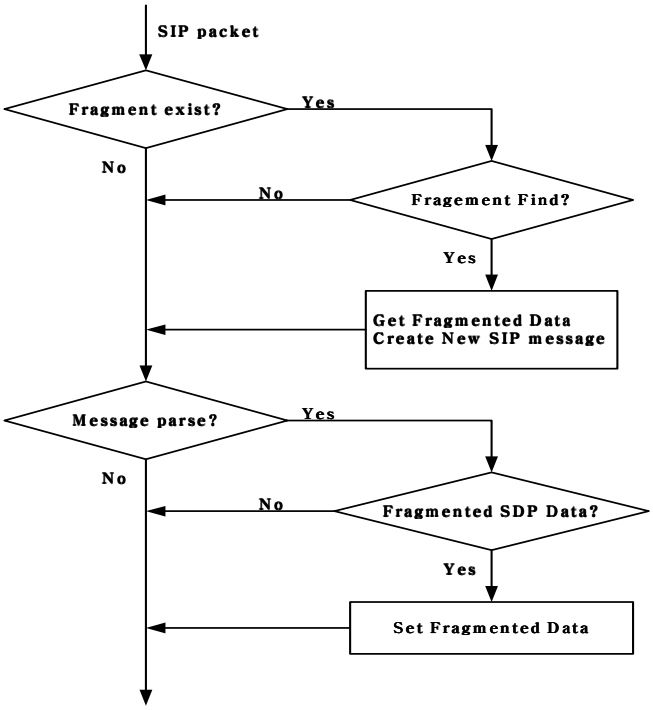
```
////////////////////////////////////////////////////////////////////////
// struct Monitoring Data

typedef struct _MONITORING_DATA
{
    int         nSipElementIndex;
    SHORT       nCallerPayloadType[SDP_TRANSPORT_INFO_CHANNEL_SIZE];
    SHORT       nCalleePayloadType[SDP_TRANSPORT_INFO_CHANNEL_SIZE];
    UINT        nCallerTotalData[SDP_TRANSPORT_INFO_CHANNEL_SIZE];
    UINT        nCalleeTotalData[SDP_TRANSPORT_INFO_CHANNEL_SIZE];
    UINT        nCallerDataNum[SDP_TRANSPORT_INFO_CHANNEL_SIZE];
    UINT        nCalleeDataNum[SDP_TRANSPORT_INFO_CHANNEL_SIZE];
    UINT        nCallerMediaNumber;
    UINT        nCalleeMediaNumber;
} MONITORING_DATA, *MONITORING_DATA_POINTER;
```

[그림 3.17] 모니터링 저장소의 구조체

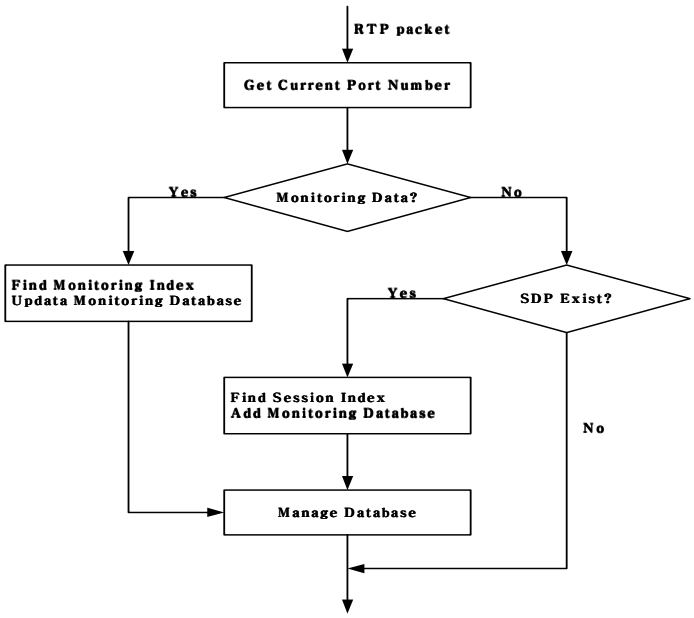
```
////////////////////////////////////////////////////////
// structs Fragmented Data

typedef struct _FRAGMENT_DATA
{
    UINT                    nBeNotEnoughLen;
    PACKET_INFO_POINTER     pPI;
    PUCHAR                  pSM;
} FRAGMENT_DATA, *FRAGMENT_DATA_POINTER;

typedef struct _FRAGMENT
{
    UINT                        nFragmentedDataIndex;
    FRAGMENT_DATA_POINTER       pFD;
} FRAGMENT, *FRAGMENT_POINTER;
```

[그림 3.18] 분할된 SIP 메시지 저장을 위한 구조체

구현된 모니터링 처리기의 동작 흐름은 패킷 드라이버로부터 전달된 패킷이 SIP 패킷인 경우와 RTP 패킷인 경우의 두 가지로 나누어진다. 전달된 패킷이 SIP 패킷인 경우에 모니터링 처리기의 동작은 [그림 3.19]와 같이 분할된 SIP 메시지를 저장하거나 분할되어 저장된 SIP 메시지를 찾아서 결합시켜 SIP 메시지를 분석할 수 있도록 하며, 전달된 패킷이 RTP 패킷인 경우에는 [그림 3.20]에서 보듯이 모니터링 저장소에 관리되고 있는 미디어 데이터인지를 판단하여 미디어 데이터를 갱신하거나 모니터링 저장소에 새로 추가한다.



[그림 3.19] 모니터링 처리기의 동작 흐름 - SIP 패킷



[그림 3.20] 모니터링 처리기의 동작 흐름 - RTP 패킷

모니터링 처리기는 [표 3.4]와 같이 모니터링 처리기 모듈 내에 구현된 함수들을 사용하고 [그림 3.21]과 같이 모니터링 저장소는 미디어 트래픽 정보를 관리한다.

함 수	설 명
Monitoring_ManageData()	◦ 세션 별 미디어 통계 정보를 관리한다.
Monitoring_CreateData() Monitoring_DeleteData()	◦ 세션 별 미디어 통계 정보를 저장할 수 있는 구조를 생성하고 삭제한다.
Monitoring_DataAdd() Monitoring_DataUpdate() Monitoring_DataDiscard()	◦ 해당 세션에 대한 미디어 통계 정보를 추가, 변경 및 삭제한다.
Monitoring_CreateFragmentData() Monitoring_DeleteFragmentData()	◦ 분할된 SIP 메시지들을 저장하고 처리할 수 있는 구조를 생성하고 삭제한다.
Monitoring_SetFragmentedData() Monitoring_GetFragmentedData()	◦ 분할된 SIP 메시지들과 분할된 정보를 저장하고 분할된 정보를 이용하여 결합시켜 전달한다.
Monitoring_FragmentAdd() Monitoring_FragmentDiscard() Monitoring_FragmentFind()	◦ 분할된 SIP 메시지들을 저장 및 삭제한다. ◦ 분할되어 저장된 SIP 메시지를 찾아준다.

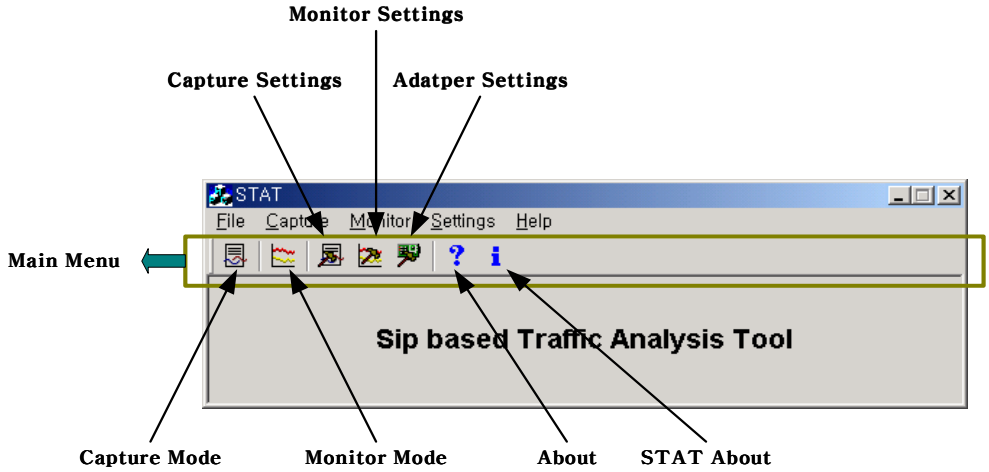
[표 3.4] 모니터링 처리기 모듈의 중요 함수

```
MD Information
      m_nMonitoringDataIndex  = 5
      nSipElementIndex        = 1
      nCallerPayloadType      = 34(58779)(209) 111(3400)(85)
      nMediaNumber            = 2
      nCalleePayloadType      = 111(3600)(90)
      nMediaNumber            = 1

MD Information
      m_nMonitoringDataIndex  = 7
      nSipElementIndex        = 3
      nCallerPayloadType      = 0(14560)(91)
      nMediaNumber            = 1
      nCalleePayloadType      = 0(13600)(85)
      nMediaNumber            = 1

MD Information
      m_nMonitoringDataIndex  = 8
      nSipElementIndex        = 4
      nCallerPayloadType      = 0(38720)(242)
      nMediaNumber            = 1
      nCalleePayloadType      = 0(17760)(111)
      nMediaNumber            = 1
```

[그림 3.21] 분석된 미디어 트래픽 정보

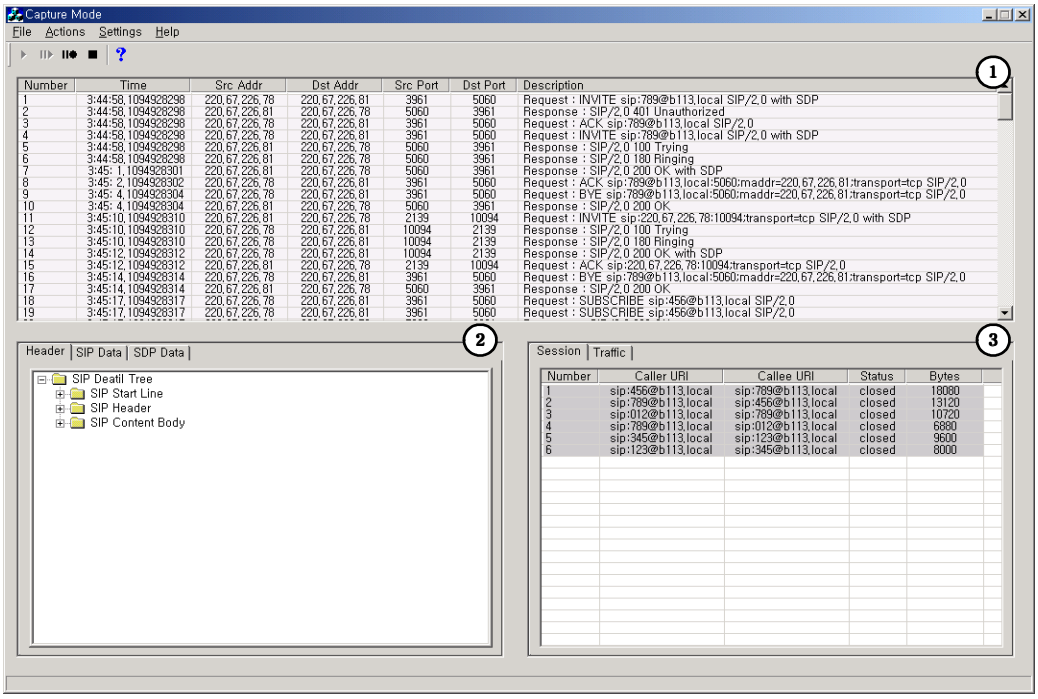
3.4 STAT 화면 설계 및 구현

STAT는 [그림 3.22]와 같은 인터페이스를 가지고 있으며 “Capture Mode”와 “Monitor Mode”로 나누어 동작할 수 있는 구조를 가지고 있고 또한 동시에 다수의 모드를 실행할 수 있다.



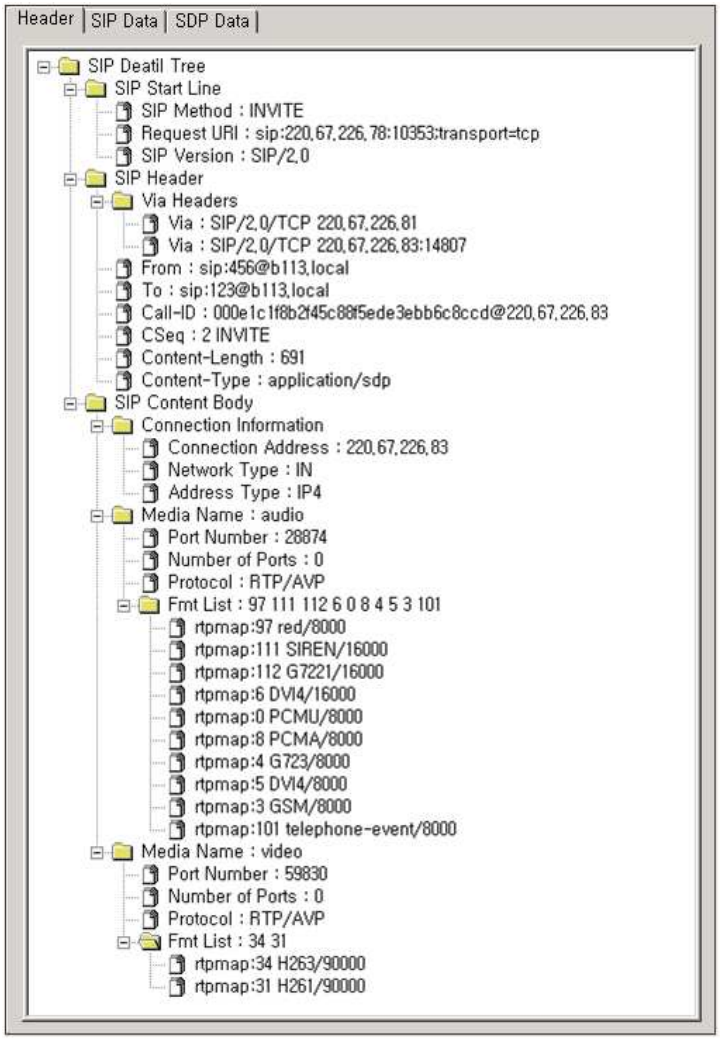
[그림 3.22] STAT 메인 화면

“Capture Mode”는 실시간으로 들어오는 SIP 패킷을 수집하여 패킷 목록, 패킷 세부 내용과 트래픽 정보를 보여주는 기능으로써 수집된 SIP 패킷들의 목록을 보여주는 리스트 컨트롤(①), 세부 내용을 보여는 탭 컨트롤(②)과 트래픽 정보를 보여주는 탭 컨트롤(③)로 구성되어 있고, “Monitor Mode”는 SIP 세션이 설정된 이후에 발생되는 미디어 트래픽 통계를 표시하는 기능으로써 선 그래프, 원 그래프와 막대 그래프로 구성되어 있다. [그림 3.23]은 “Capture Mode”로 실행되는 STAT의 실행 예이다.

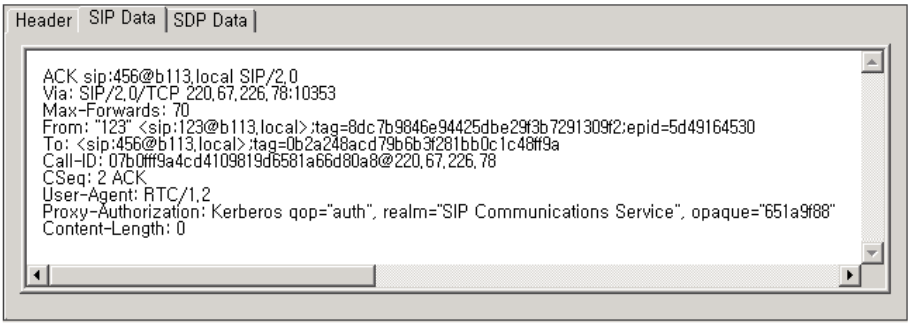


[그림 3.23] STAT Capture Mode 화면

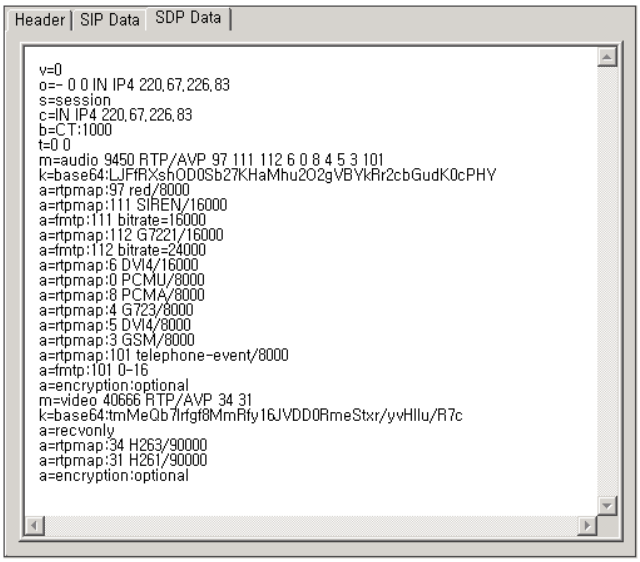
“Capture Mode”의 세부 내용을 보여는 탭 컨트롤(②)은 분석된 패킷을 헤더별로 보여주는 “Header”와 패킷 내용을 보여주는 “SIP Data”와 “SDP Data”로 구분된다. [그림 3.24], [그림 3.25]와 [그림 3.26]은 “Capture Mode”에 의해 수집된 패킷의 세부 내용을 보여주고 있다.



[그림 3.24] STAT Capture Mode - Header 화면

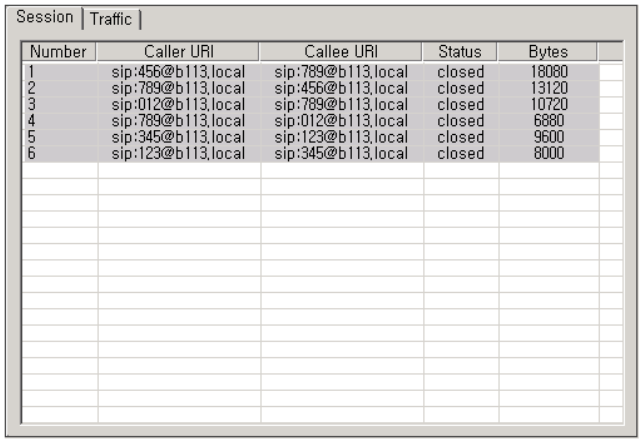


[그림 3.25] STAT Capture Mode - SIP Data 화면

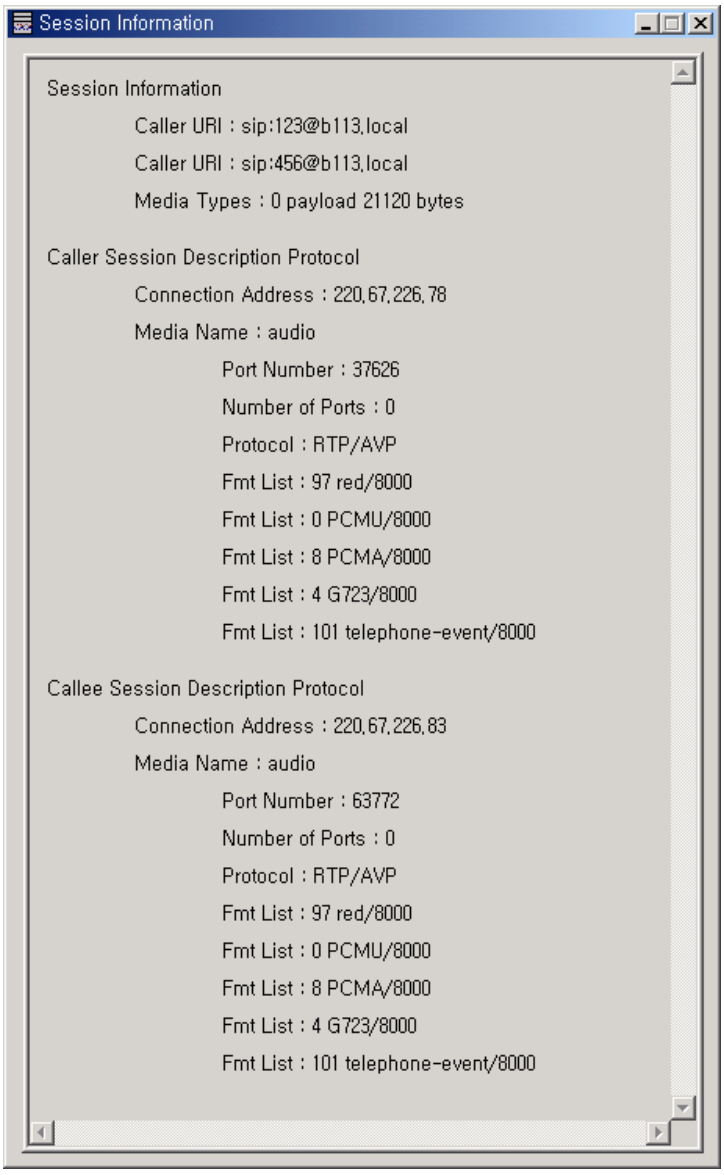


[그림 3.26] STAT Capture Mode - SDP Data 화면

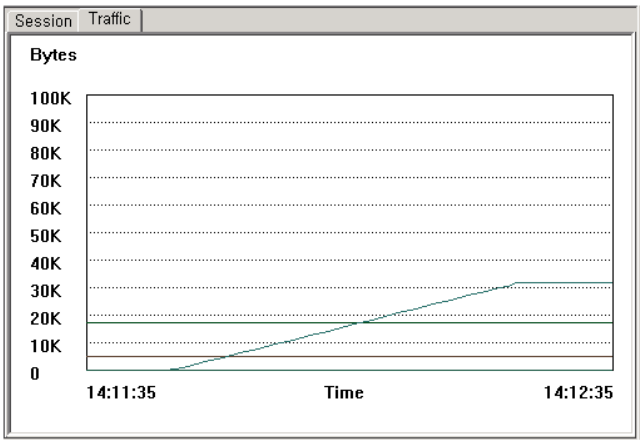
"Capture Mode"의 트래픽 정보를 보여주는 탭 컨트롤(③)은 설정된 SIP 세션 정보를 보여주는 "Session"과 SIP 세션별 발생된 트래픽 량을 보여주는 "Traffic"으로 구분된다. [그림 3.27]과 [그림 3.28]은 "Capture Mode"에 의해 수집된 SIP 세션과 SIP 세션 정보를 보여주고 있으며 [그림 3.29]는 SIP 세션별 트래픽 정보를 보여주고 있다.



[그림 3.27] STAT Capture Mode - Session 화면

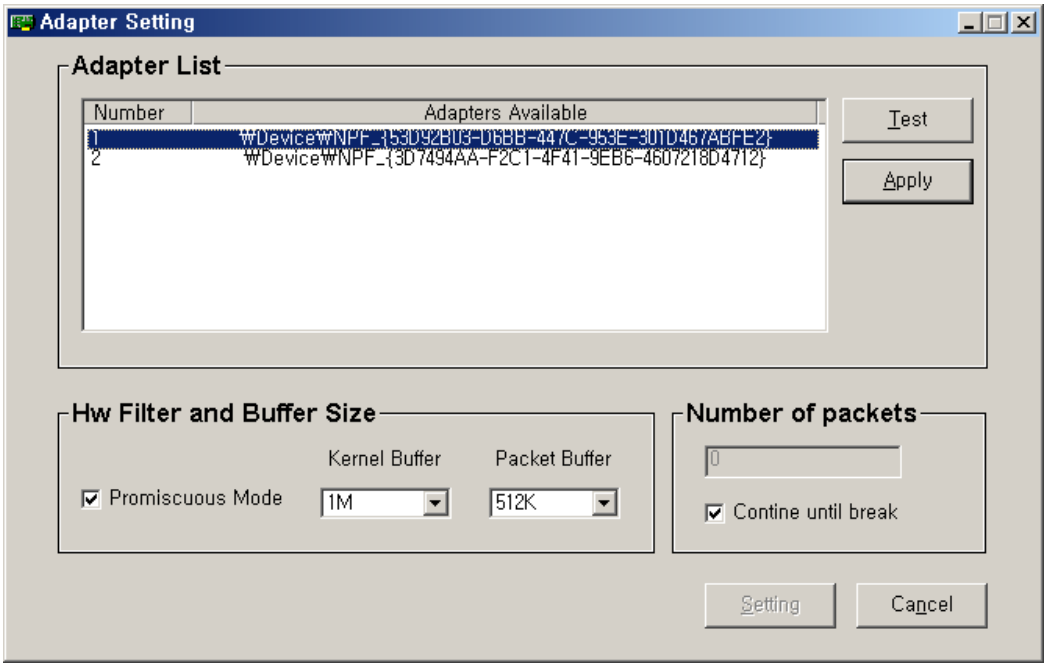


[그림 3.28] STAT Capture Mode - Session 정보 화면



[그림 3.29] STAT Capture Mode - Traffic 화면

"Adapter Setting"은 [그림 3.30]과 같으며 그림 패킷 수집에 사용될 네트워크 카드를 설정하는 기능으로써 네트워크 카드 선택, 패킷 수집 방법, 버퍼 설정 및 패킷 개수 설정 항목으로 구성되어 있다.



[그림 3.30] STAT - Adapter 설정 화면

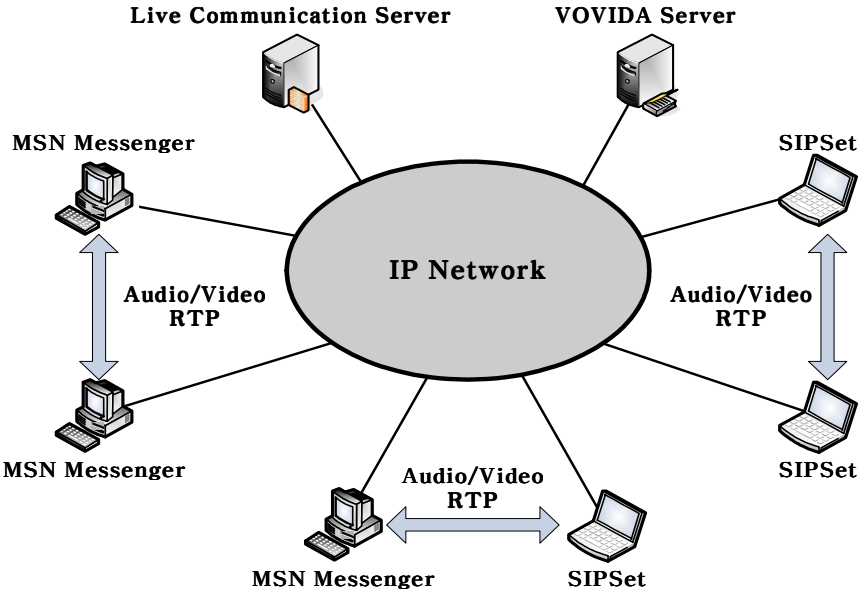
제 4장 실험 및 성능 평가

4.1 실험 방법

본 논문에서 구현된 STAT의 실험 환경과 실험 방법은 다음과 같으며, 이를 통하여 성능을 평가한다.

가. 실험 환경

실험 환경은 [그림 4.1]과 같으며 상세히 설명하면 다음과 같다.



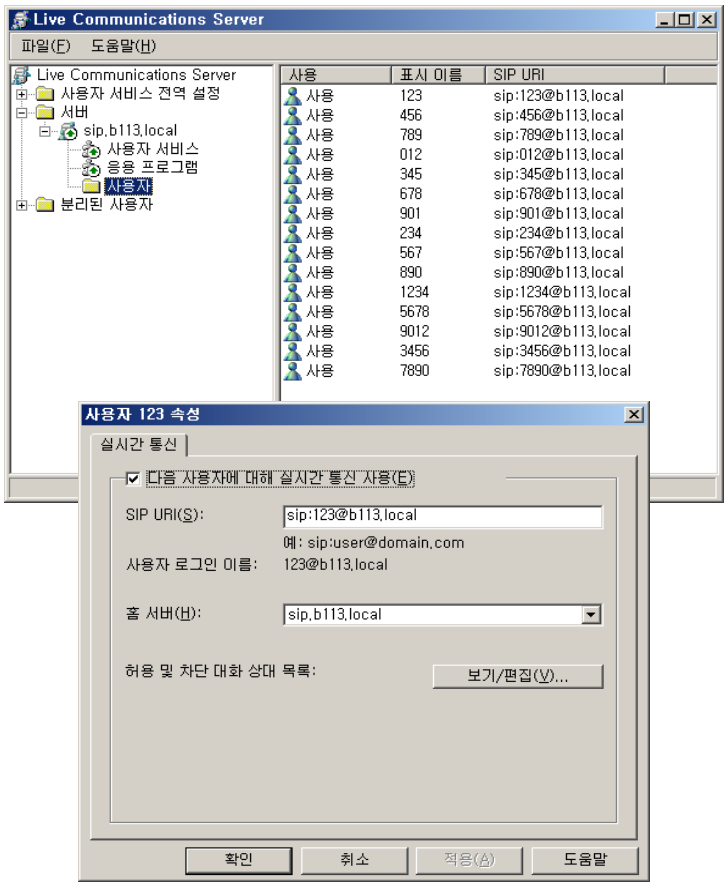
[그림 4.1] STAT 실험 환경

SIP 서버는 Microsoft Windows Server 2003 운영 체제에서 동작하고 SIP 기반 인스턴트 메시징 서버인 실시간 통신 서버(Live Communications Server 2003)[19]와 Linux 기반인 VOVIDA Open Communication Application Library(VOCAL)-1.5.0[20]을 사용하였다. [표 4.1]은 SIP 서버의 시스템 사양을 나열한 것이다.

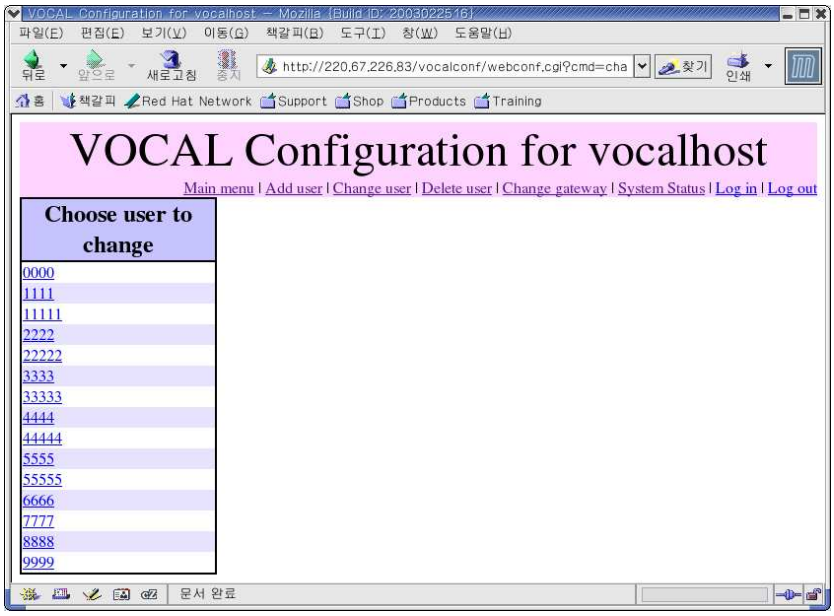
구 분	실시간 통신 서버	VOVIDA 서버
OS	Windows Server 2003	Redhat 9.0
CPU	Pentium 4 - 2.8GHz	Pentium M - 1.4GHz
RAM	512MB	256MB
NIC	Intel(R) PRO/100 Mbps VE	3Com 3C90x 10/100 Mbps PCI

[표 4.1] SIP 서버의 시스템 사양

SIP 서버에 30개의 사용자 계정을 나누어 생성하여 15개의 SIP 세션이 수립될 수 있도록 설정하였다. [그림 4.2]와 [그림 4.3]은 실시간 통신 서버와 VOVIDA 서버에 생성된 계정을 보여주고 있다.



[그림 4.2] SIP 서버 - 실시간 통신 서버



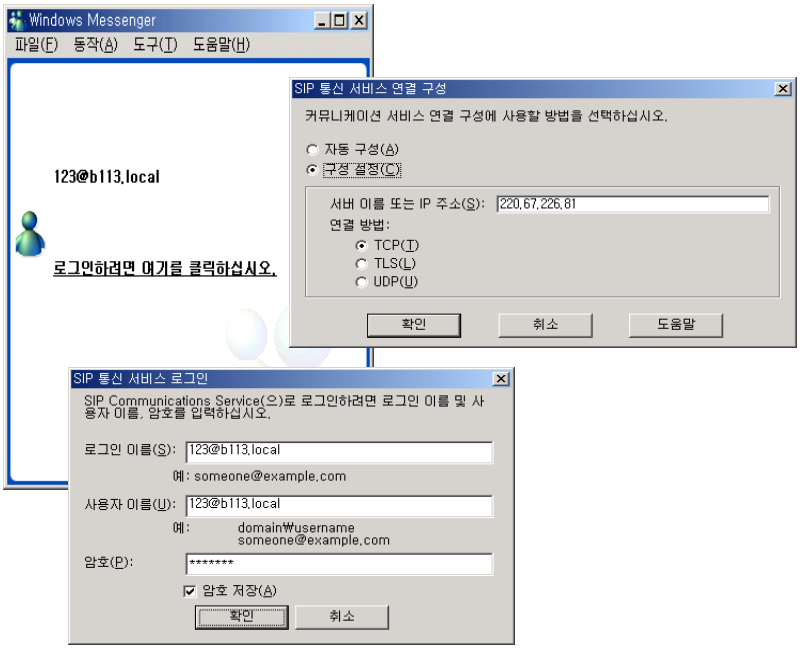
[그림 4.3] SIP 서버 - VOVIDA 서버

SIP User Agent는 SIP 프로토콜을 지원하고 Microsoft Windows XP 또는 Windows 2000 운영 체제에서 동작하는 Microsoft 실시간 통신 서버용 Windows Messenger 5.0[21]과 VOVIDA SIPSet-1.5.0[22]를 사용하였다. [표 4.2]는 SIP User Agent가 사용될 시스템의 사양을 나열한 것이다.

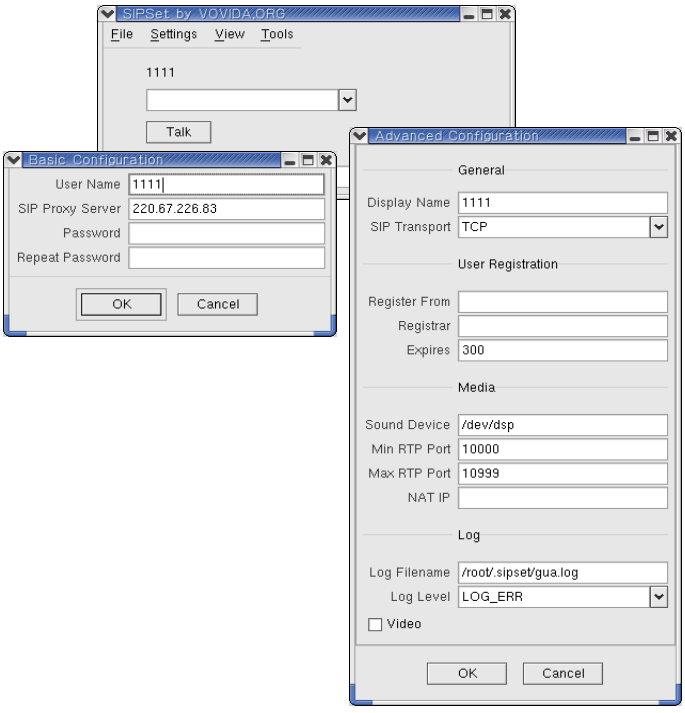
구 분	SIP User Agent	
OS	Windows 2000, XP	Redhat 9.0
CPU	Pentium 4 - 2GHz	Pentium M - 1.4GHz
RAM	512MB	256MB
NIC	Intel(R) PRO/100 Mbps VE	3Com 3C90x 10/100 Mbps PCI

[표 4.2] SIP User Agent의 시스템 사양

SIP User Agent는 음성 채팅 기능 또는 PC 카메라를 이용한 화상 채팅 기능을 갖도록 설정하였다. [그림 4.4]와 [그림 4.5]는 SIP User Agent에 서버 설정 및 로그인 화면을 보여주고 있다.



[그림 4.4] SIP User Agent - MSN Messenger 설정



[그림 4.5] SIP User Agent - SIPSet 설정

나. 실험 방법

실시간 통신 서버와 VOVIDA 서버에 생성된 계정을 이용하는 MSN Messenger와 SIPSet를 사용하여 4가지 실험과 트래픽 발생기(Traffic Generator)[23]를 이용하여 1가지 실험을 하였다.

실험 1 - SIP User Agent가 해당 SIP 서버에 로그인, SIP 세션 요청 및 로그아웃 과정에서 발생되는 SIP 패킷을 수집하여 정상적으로 분석할 수 있는지 SIP User Agent의 수를 늘려가며 실험을 하였다.

실험 2 - SIP 세션 수립 과정에서 발생되는 세션 정보(SDP 패킷)를 분석하는 실험과 SIP 세션이 수립되었을 경우에 SIP 세션의 호출자 정보, 호출 받은 사용자 정보 및 사용되는 미디어 정보 등의 SIP 세션별 세션 정보를 관리하는 능력을 실험하였다.

실험 3 - 미디어 트래픽을 분석하는 실험을 하였다. SIP 세션 수립 후에 RTP 패킷을 이용하여 발생되는 멀티미디어 통신 트래픽을 분석하여 SIP 세션별 발생된 트래픽 통계와 미디어 종류별 트래픽 통계를 잘 처리할 수 있는지 실험하였다.

실험 4 - 멀티미디어 통신 트래픽을 그래프 인터페이스의 성능 실험이다. SIP 세션별 실시간으로 발생되는 트래픽 정보를 이용하여 그래프가 정상적으로 표현되는지 실험하였다.

실험 3과 4에서 사용된 SIP 세션의 생성 시나리오는 [표 4.3]과 같으며 호출자와 호출 받을 사용자가 사용할 계정, SIP 서버, SIP User Agent와 트래픽 발생을 위해 사용될 미디어 종류가 나열되어 있다. 생성된 SIP 세션의 수는 15개로 실시간 통신 서버와 계정을 이용하여 5개의 SIP 세션이 생성되었고 VOVIDA 서버와 계정을 이용하여 10개의 SIP 세션이 생성되었다. VOVIDA 서버를 이용하여 생성된 10개 SIP 세션 중에 5개의 세션은 MSN Messenger와 SIPSet를 함께 사용하였다. SIPSet은 다른 SIP 서버와 호환성을 갖지 않고 VOVIDA 서버와만 동작되도록 설계되어 실시간 통신 서버를 이용한 실험은 하지 못하였다.

Caller			Callee			Media Type
ID	Server	SIP UA	ID	Server	SIP UA	
123	Live Comm. Server	Messenger	456	Live Comm. Server	Messenger	음성
789			012			
345			678			
901			234			
567			890			화상
0000	VOVIDA Server	SIPSet	1111	VOVIDA Server	SIPSet	음성
2222			3333			
4444			5555			
6666			7777			
8888			9999			화상
11111	VOVIDA Server	Messenger	22222	VOVIDA Server	SIPSet	음성
33333		Messenger	44444		SIPSet	
55555		SIPSet	66666		Messenger	
77777		SIPSet	88888		Messenger	화상
99999		Messenger	00000		SIPSet	

[표 4.3] SIP 세션의 생성 시나리오

실험 5 - STAT의 부하 테스트이다. 트래픽 발생기를 이용하여 전송속도를 늘려가면서 미디어 트래픽(RTP 패킷)을 발생시켜 트래픽 분석 성능을 측정하는 부하 테스트를 하였다.

4.2 실험 결과 및 분석

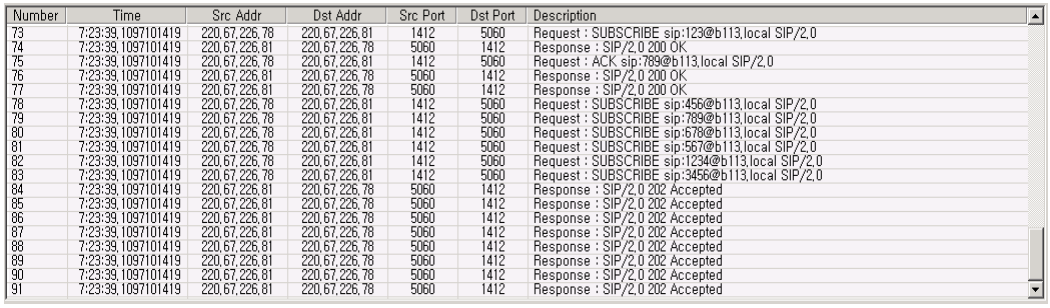
4.2.1 실험 1 - SIP 패킷 수집 및 분석 실험

SIP 서버에 생성된 계정을 이용하는 SIP User Agent가 해당 서버에 로그인, SIP 세션 요청 및 로그아웃 과정에서 발생되는 SIP 패킷을 수집하여 정상적으로 분석할 수 있는지 SIP User Agent의 수를 늘려가며 실험하였다. [표 4.4]는 하나의 SIP User Agent가 해당 서버에 로그인, SIP 세션 요청 및 로그아웃의 각 과정에서 발생된 SIP 패킷들을 수집한 결과를 나열한 것이고 등록된 사용자 수는 14명이다.

SIP 서버	SIP User Agent	로그인/SIP 세션 요청/로그아웃 각 과정에서 발생된 SIP 패킷 수(총 SIP 패킷 수)	
		등록된 사용자 유 ④	등록된 사용자 무 ⑤
실시간 통신 서버	MSN Messenger ①	65/6/20(91)	19/6/10(35)
	SIPSet	·	
VOVIDA 서버	MSN Messenger ②	27/3/10(40)	3/3/6(12)
	SIPSet ③	3/3/6(12)	

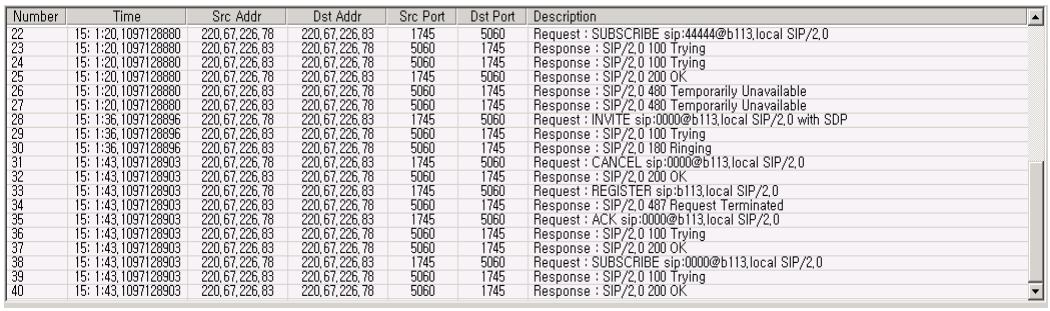
[표 4.4] 하나의 SIP User Agent에서 발생된 SIP 패킷 수

[표 4.4]의 SIP 패킷 수집 실험을 통하여 등록된 사용자가 있을 경우와 없을 경우에 동일한 SIP 서버를 사용하여 발생된 SIP 패킷 수의 비율은 결과 ①에서는 로그인시 약 3.4배와 로그아웃시 약 2배, 결과 ②에서는 약 9배와 약 1.6배로 등록된 사용자가 있을 경우에 보다 많은 패킷이 발생되었고, 서로 다른 SIP 서버를 사용하여 발생된 SIP 패킷 수의 비율은 결과 ④에서는 로그인시 약 2.4배와 로그아웃시 약 2배, 결과 ⑤에서는 약 6.3배와 약 1.6배로 결과 ①, ②와 같이 등록된 사용자가 있을 경우가 그렇지 않은 경우보다 많은 SIP 패킷이 발생되었다는 것을 알 수 있다. 이와 같은 결과는 사용되는 SIP 서버와 SIP User Agent에서 제공하는 기능의 차이로 인하여 발생한 것이다. 즉 다른 사용자의 로그인 상태를 확인할 수 있는 온라인 상태 인식(Presence awareness) 서비스를 제공하는 실시간 통신 서버와 같은 SIP 서버와 사용자 등록 기능을 가지고 있는 MSN Messenger와 같은 SIP User Agent를 사용하여 서버에 로그인 및 로그아웃을 할 경우에 발생되는 SIP 패킷의 수는 등록된 사용자의 수에 비례하여 증가한다. 증가하는 SIP 패킷의 종류로는 자신의 갱신된 상태를 가입한 당사자에게 전달하는 “NOTIFY” SIP 요청 메시지와 발신자가 수신자의 상태 정보에 대한 갱신을 요청하는 “SUBSCRIBE” SIP 요청 메시지가 있었다. [그림 4.6], [그림 4.7]과 [그림 4.8]은 SIP 패킷 수집 실험에서 발생된 SIP 패킷이 모두 분석되어 처리된 결과를 보여주고 있다.



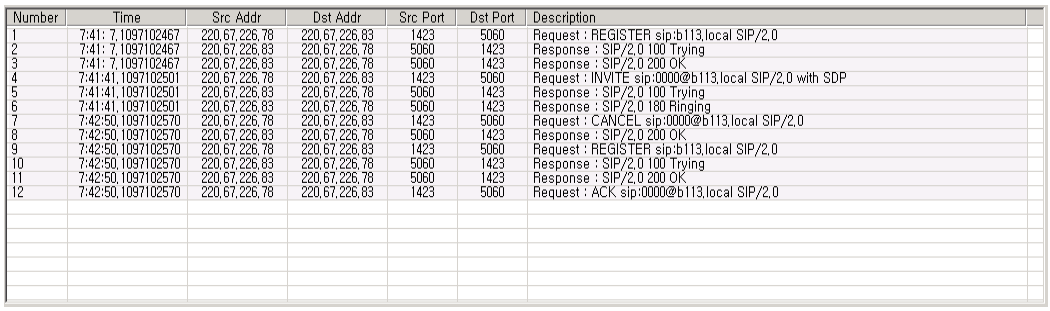
Number	Time	Src Addr	Dst Addr	Src Port	Dst Port	Description
73	7:23:39.1097101419	220.67.226.78	220.67.226.81	1412	5060	Request : SUBSCRIBE sip:123@b113.local SIP/2.0
74	7:23:39.1097101419	220.67.226.81	220.67.226.78	5060	1412	Response : SIP/2.0 200 OK
75	7:23:39.1097101419	220.67.226.78	220.67.226.81	1412	5060	Request : ACK sip:789@b113.local SIP/2.0
76	7:23:39.1097101419	220.67.226.81	220.67.226.78	5060	1412	Response : SIP/2.0 200 OK
77	7:23:39.1097101419	220.67.226.81	220.67.226.78	5060	1412	Response : SIP/2.0 200 OK
78	7:23:39.1097101419	220.67.226.78	220.67.226.81	1412	5060	Request : SUBSCRIBE sip:456@b113.local SIP/2.0
79	7:23:39.1097101419	220.67.226.78	220.67.226.81	1412	5060	Request : SUBSCRIBE sip:789@b113.local SIP/2.0
80	7:23:39.1097101419	220.67.226.78	220.67.226.81	1412	5060	Request : SUBSCRIBE sip:678@b113.local SIP/2.0
81	7:23:39.1097101419	220.67.226.78	220.67.226.81	1412	5060	Request : SUBSCRIBE sip:567@b113.local SIP/2.0
82	7:23:39.1097101419	220.67.226.78	220.67.226.81	1412	5060	Request : SUBSCRIBE sip:1234@b113.local SIP/2.0
83	7:23:39.1097101419	220.67.226.78	220.67.226.81	1412	5060	Request : SUBSCRIBE sip:3456@b113.local SIP/2.0
84	7:23:39.1097101419	220.67.226.81	220.67.226.78	5060	1412	Response : SIP/2.0 202 Accepted
85	7:23:39.1097101419	220.67.226.81	220.67.226.78	5060	1412	Response : SIP/2.0 202 Accepted
86	7:23:39.1097101419	220.67.226.81	220.67.226.78	5060	1412	Response : SIP/2.0 202 Accepted
87	7:23:39.1097101419	220.67.226.81	220.67.226.78	5060	1412	Response : SIP/2.0 202 Accepted
88	7:23:39.1097101419	220.67.226.81	220.67.226.78	5060	1412	Response : SIP/2.0 202 Accepted
89	7:23:39.1097101419	220.67.226.81	220.67.226.78	5060	1412	Response : SIP/2.0 202 Accepted
90	7:23:39.1097101419	220.67.226.81	220.67.226.78	5060	1412	Response : SIP/2.0 202 Accepted
91	7:23:39.1097101419	220.67.226.81	220.67.226.78	5060	1412	Response : SIP/2.0 202 Accepted

[그림 4.6] SIP User Agent에서 발생된 SIP 패킷 - ①



Number	Time	Src Addr	Dst Addr	Src Port	Dst Port	Description
22	15: 1:20.1097128880	220.67.226.78	220.67.226.83	1745	5060	Request : SUBSCRIBE sip:44444@b113.local SIP/2.0
23	15: 1:20.1097128880	220.67.226.83	220.67.226.78	5060	1745	Response : SIP/2.0 100 Trying
24	15: 1:20.1097128880	220.67.226.83	220.67.226.78	5060	1745	Response : SIP/2.0 100 Trying
25	15: 1:20.1097128880	220.67.226.78	220.67.226.83	1745	5060	Response : SIP/2.0 200 OK
26	15: 1:20.1097128880	220.67.226.83	220.67.226.78	5060	1745	Response : SIP/2.0 480 Temporarily Unavailable
27	15: 1:20.1097128880	220.67.226.83	220.67.226.78	5060	1745	Response : SIP/2.0 480 Temporarily Unavailable
28	15: 1:36.1097128896	220.67.226.78	220.67.226.83	1745	5060	Request : INVITE sip:0000@b113.local SIP/2.0 with SDP
29	15: 1:36.1097128896	220.67.226.83	220.67.226.78	5060	1745	Response : SIP/2.0 100 Trying
30	15: 1:36.1097128896	220.67.226.83	220.67.226.78	5060	1745	Response : SIP/2.0 180 Ringing
31	15: 1:43.1097128903	220.67.226.78	220.67.226.83	1745	5060	Request : CANCEL sip:0000@b113.local SIP/2.0
32	15: 1:43.1097128903	220.67.226.83	220.67.226.78	5060	1745	Response : SIP/2.0 200 OK
33	15: 1:43.1097128903	220.67.226.78	220.67.226.83	1745	5060	Request : REGISTER sip:b113.local SIP/2.0
34	15: 1:43.1097128903	220.67.226.83	220.67.226.78	5060	1745	Response : SIP/2.0 487 Request Terminated
35	15: 1:43.1097128903	220.67.226.78	220.67.226.83	1745	5060	Request : ACK sip:0000@b113.local SIP/2.0
36	15: 1:43.1097128903	220.67.226.83	220.67.226.78	5060	1745	Response : SIP/2.0 100 Trying
37	15: 1:43.1097128903	220.67.226.83	220.67.226.78	5060	1745	Response : SIP/2.0 200 OK
38	15: 1:43.1097128903	220.67.226.78	220.67.226.83	1745	5060	Request : SUBSCRIBE sip:0000@b113.local SIP/2.0
39	15: 1:43.1097128903	220.67.226.83	220.67.226.78	5060	1745	Response : SIP/2.0 100 Trying
40	15: 1:43.1097128903	220.67.226.83	220.67.226.78	5060	1745	Response : SIP/2.0 200 OK

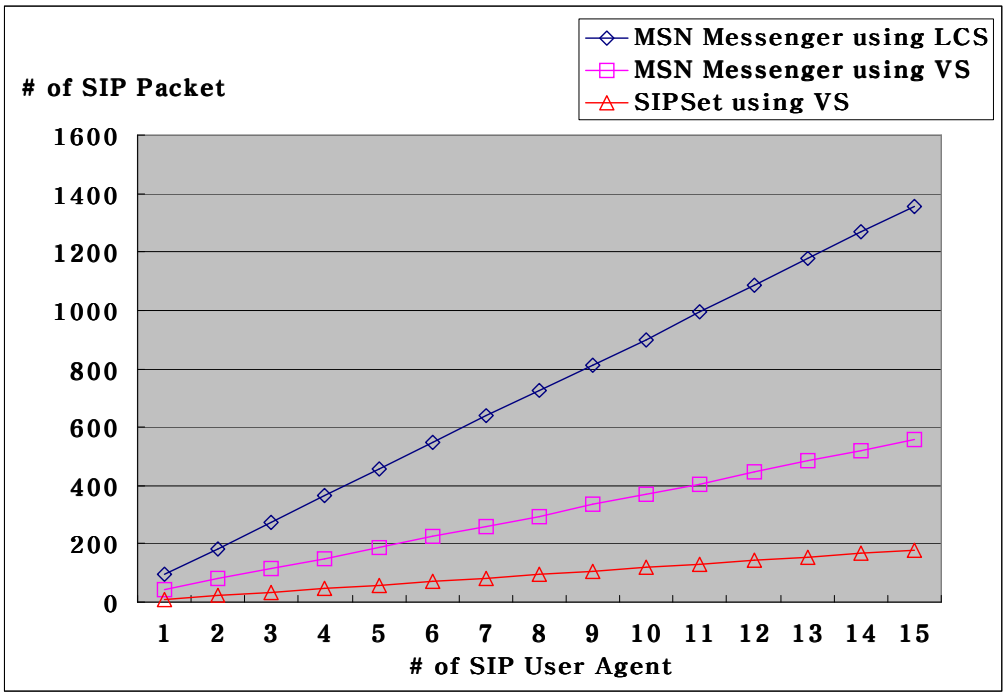
[그림 4.7] SIP User Agent에서 발생된 SIP 패킷 - ②



Number	Time	Src Addr	Dst Addr	Src Port	Dst Port	Description
1	7:41: 7.1097102467	220.67.226.78	220.67.226.83	1423	5060	Request : REGISTER sip:b113.local SIP/2.0
2	7:41: 7.1097102467	220.67.226.83	220.67.226.78	5060	1423	Response : SIP/2.0 100 Trying
3	7:41: 7.1097102467	220.67.226.83	220.67.226.78	5060	1423	Response : SIP/2.0 200 OK
4	7:41:41.1097102501	220.67.226.78	220.67.226.83	1423	5060	Request : INVITE sip:0000@b113.local SIP/2.0 with SDP
5	7:41:41.1097102501	220.67.226.83	220.67.226.78	5060	1423	Response : SIP/2.0 100 Trying
6	7:41:41.1097102501	220.67.226.83	220.67.226.78	5060	1423	Response : SIP/2.0 180 Ringing
7	7:42:50.1097102570	220.67.226.78	220.67.226.83	1423	5060	Request : CANCEL sip:0000@b113.local SIP/2.0
8	7:42:50.1097102570	220.67.226.83	220.67.226.78	5060	1423	Response : SIP/2.0 200 OK
9	7:42:50.1097102570	220.67.226.78	220.67.226.83	1423	5060	Request : REGISTER sip:b113.local SIP/2.0
10	7:42:50.1097102570	220.67.226.83	220.67.226.78	5060	1423	Response : SIP/2.0 100 Trying
11	7:42:50.1097102570	220.67.226.83	220.67.226.78	5060	1423	Response : SIP/2.0 200 OK
12	7:42:50.1097102570	220.67.226.78	220.67.226.83	1423	5060	Request : ACK sip:0000@b113.local SIP/2.0

[그림 4.8] SIP User Agent에서 발생된 SIP 패킷 - ③

[그림 4.9]는 온라인 상태 인식 기능의 유무에 따라 발생되는 SIP 패킷 량을 SIP User Agent 수를 늘려가며 측정한 결과이다.



[그림 4.9] 온라인 상태 인식 기능 유무에 따라 발생된 패킷 량

[그림 4.9]에서 보듯이 SIP 서버와 SIP User Agent의 온라인 상태 인식 기능의 유무에 따라 약 7배 이상의 SIP 패킷이 추가 발생되었다는 것을 알 수 있었다. [그림 4.10]은 [그림 4.9]에서 실시간 통신 서버를 이용하고 15개의 MSN Messenger를 사용하여 발생된 SIP 패킷을 수집하여 분석한 결과를 보여주고 있다.

Number	Time	Src Addr	Dst Addr	Src Port	Dst Port	Description
1348	17:18:11.1097137091	220.67.226.78	220.67.226.81	12282	4971	Response : SIP/2.0 200 OK
1349	17:18:14.1097137094	220.67.226.78	220.67.226.81	1914	5060	Request : SUBSCRIBE sip:123@b113.local SIP/2.0
1350	17:18:14.1097137094	220.67.226.78	220.67.226.81	1914	5060	Request : SUBSCRIBE sip:123@b113.local SIP/2.0
1351	17:18:14.1097137094	220.67.226.81	220.67.226.78	5060	1914	Response : SIP/2.0 200 OK
1352	17:18:14.1097137094	220.67.226.81	220.67.226.78	5060	1914	Response : SIP/2.0 200 OK
1353	17:18:14.1097137094	220.67.226.78	220.67.226.81	1914	5060	Request : SUBSCRIBE sip:456@b113.local SIP/2.0
1354	17:18:14.1097137094	220.67.226.78	220.67.226.81	1914	5060	Request : SUBSCRIBE sip:789@b113.local SIP/2.0
1355	17:18:14.1097137094	220.67.226.78	220.67.226.81	1914	5060	Request : SUBSCRIBE sip:678@b113.local SIP/2.0
1356	17:18:14.1097137094	220.67.226.78	220.67.226.81	1914	5060	Request : SUBSCRIBE sip:234@b113.local SIP/2.0
1357	17:18:14.1097137094	220.67.226.78	220.67.226.81	1914	5060	Request : SUBSCRIBE sip:1234@b113.local SIP/2.0
1358	17:18:14.1097137094	220.67.226.78	220.67.226.81	1914	5060	Request : SUBSCRIBE sip:9012@b113.local SIP/2.0
1359	17:18:14.1097137094	220.67.226.81	220.67.226.78	5060	1914	Response : SIP/2.0 202 Accepted
1360	17:18:14.1097137094	220.67.226.81	220.67.226.78	5060	1914	Response : SIP/2.0 202 Accepted
1361	17:18:14.1097137094	220.67.226.81	220.67.226.78	5060	1914	Response : SIP/2.0 202 Accepted
1362	17:18:14.1097137094	220.67.226.81	220.67.226.78	5060	1914	Response : SIP/2.0 202 Accepted
1363	17:18:14.1097137094	220.67.226.81	220.67.226.78	5060	1914	Response : SIP/2.0 202 Accepted
1364	17:18:14.1097137094	220.67.226.81	220.67.226.78	5060	1914	Response : SIP/2.0 202 Accepted
1365	17:18:14.1097137094	220.67.226.81	220.67.226.78	5060	1914	Response : SIP/2.0 202 Accepted
1366	17:18:14.1097137094	220.67.226.81	220.67.226.78	5060	1914	Response : SIP/2.0 202 Accepted

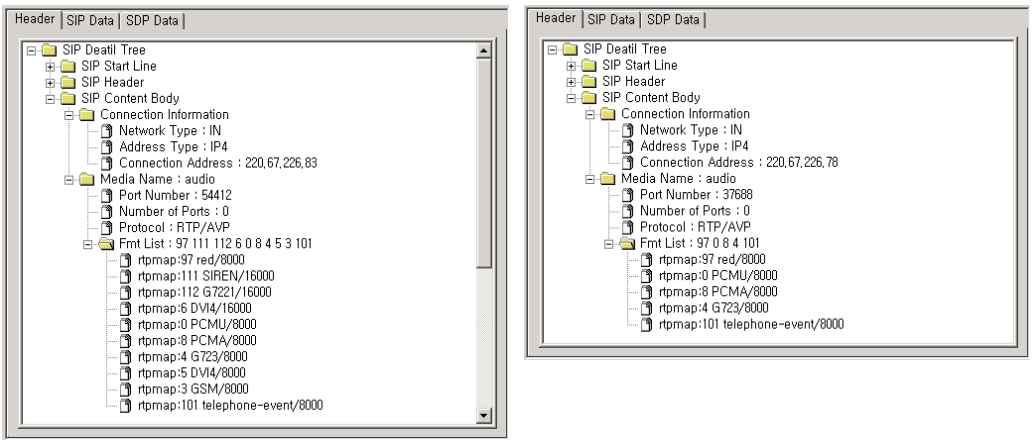
[그림 4.10] 15개의 SIP User Agent에서 발생된 SIP 패킷 량

4.2.2 실험 2 - 세션 정보(SDP 패킷) 분석 및 SIP 세션 정보 관리 실험

SIP 세션 수립 과정에서 발생되는 세션 정보를 분석하는 실험과 이를 통하여 얻어진 호출자와 호출 받은 사용자의 미디어 정보를 이용하여 SIP 세션별 세션 정보를 관리하는 능력을 실험하였다.

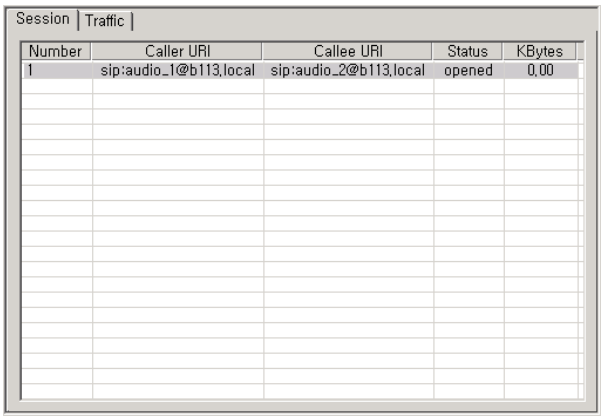
가. 음성 대 음성 채팅

[그림 4.11]은 호출자와 호출 받은 사용자 모두 음성 채팅 기능만을 가지는 SIP User Agent를 사용하여 SIP 세션 수립시 발생된 음성 채팅 요청(좌)과 응답(우)에 포함된 세션 정보를 분석한 결과이다.

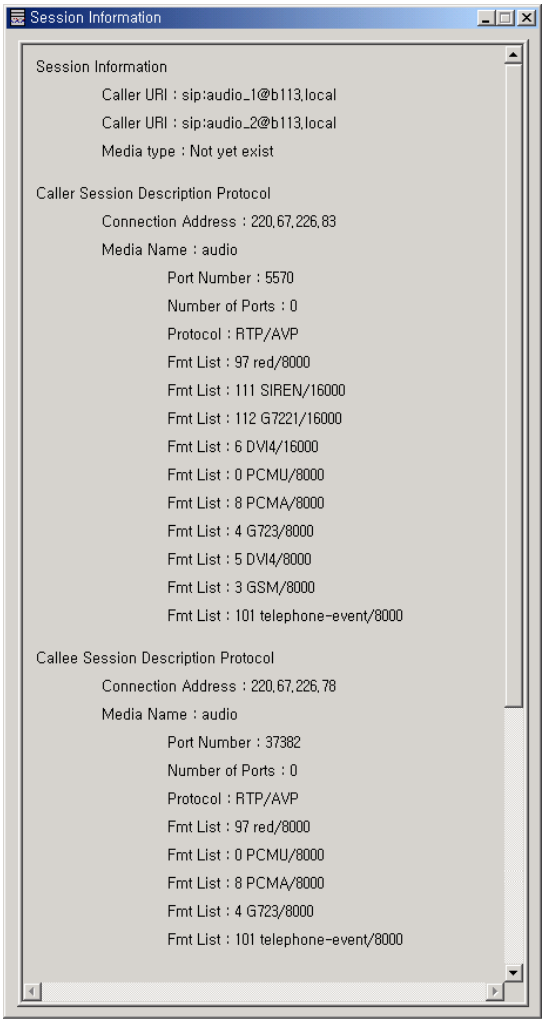


[그림 4.11] 세션 정보 분석 결과 - 음성 대 음성

[그림 4.11]에서 보듯이 호출자는 음성을 사용하고 음성 전달에 사용할 포트 번호는 54412번이며 사용 가능한 오디오 코덱(Fmt List)을 10개 가지고 있다는 정보를 전달하였고, 호출자의 음성 채팅 요청에 대한 응답으로 호출 받은 사용자는 미디어 스트림을 수용하였고 포트 번호 37688번을 사용하여 음성을 전달할 것이며 호출자가 제안한 10개의 오디오 코덱중 5개의 오디오 코덱이 지원 가능하다는 정보를 미디어 협상을 통하여 전달하였다는 것을 분석된 결과에서 알 수 있었다. [그림 4.12]와 [그림 4.13]은 음성 채팅 기능을 가지고 있는 SIP User Agent들간에 수립된 SIP 세션과 세션 정보를 보여주고 있다.



[그림 4.12] 수립된 SIP 세션 - 음성 대 음성



[그림 4.13] SIP 세션 정보 - 음성 대 음성

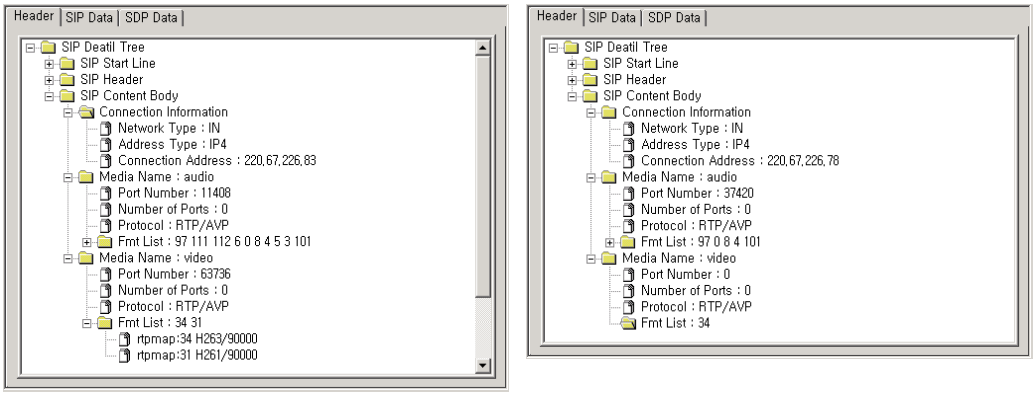
나. 화상 대 음성 채팅

[그림 4.14]는 호출자는 화상 채팅 기능을 갖고 호출 받은 사용자는 음성 채팅 기능을 갖는 SIP User Agent를 사용하여 SIP 세션 수립시 발생된 SIP 패킷을 수집한 화면이다.

Number	Time	Src Addr	Dst Addr	Src Port	Dst Port	Description
1	4:18:57,1097263137	220.67.226.81	220.67.226.78	4150	10682	Request : INVITE sip:220.67.226.78:10682;transport=tcp SIP/2.0 with SDP
2	4:18:57,1097263137	220.67.226.78	220.67.226.81	10682	4150	Response : SIP/2.0 100 Trying
3	4:18:57,1097263137	220.67.226.78	220.67.226.81	10682	4150	Response : SIP/2.0 180 Ringing
4	4:18:59,1097263139	220.67.226.78	220.67.226.81	10682	4150	Response : SIP/2.0 200 OK with SDP
5	4:19: 0,1097263140	220.67.226.81	220.67.226.78	4150	10682	Request : ACK sip:220.67.226.78:10682;transport=tcp SIP/2.0
6	4:19: 0,1097263140	220.67.226.81	220.67.226.78	4150	10682	Request : INVITE sip:220.67.226.78:10682;transport=tcp SIP/2.0 with SDP
7	4:19: 0,1097263140	220.67.226.78	220.67.226.81	10682	4150	Response : SIP/2.0 100 Trying
8	4:19: 0,1097263140	220.67.226.78	220.67.226.81	10682	4150	Response : SIP/2.0 200 OK with SDP
9	4:19: 0,1097263140	220.67.226.81	220.67.226.78	4150	10682	Request : ACK sip:220.67.226.78:10682;transport=tcp SIP/2.0

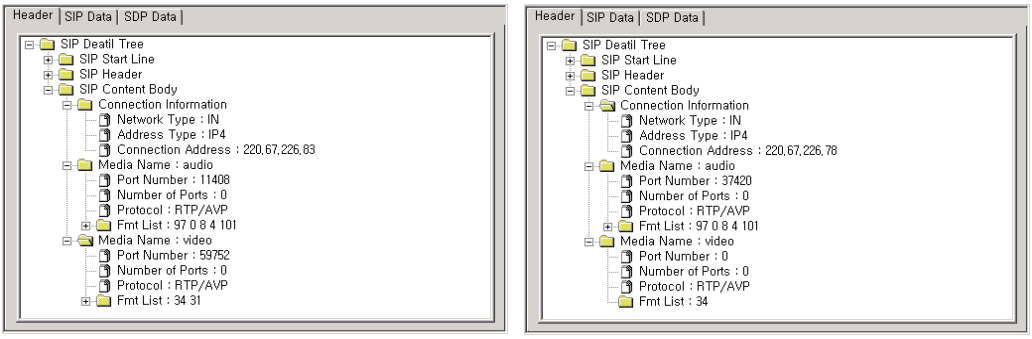
[그림 4.14] 수집된 SIP 패킷 - 화상 대 음성

[그림 4.14]의 수집된 SIP 패킷을 보면 SIP 세션 수립을 요청하는 INVITE 메시지와 SIP 세션 수립에 응답하는 200 OK 메시지가 2개씩 수집되었다는 것을 알 수 있다. 즉 하나의 SIP 세션이 INVITE/200/ACK 메시지를 사용하여 수립된 후에 다른 INVITE/200/ACK 메시지에 의해서 수립된 SIP 세션이 변경된 것이다. [그림 4.15]는 초기 SIP 세션 수립시 발생된 세션 정보를 분석한 결과이고, [그림 4.16]은 SIP 세션 변경시 발생된 세션 정보를 분석한 결과이다.



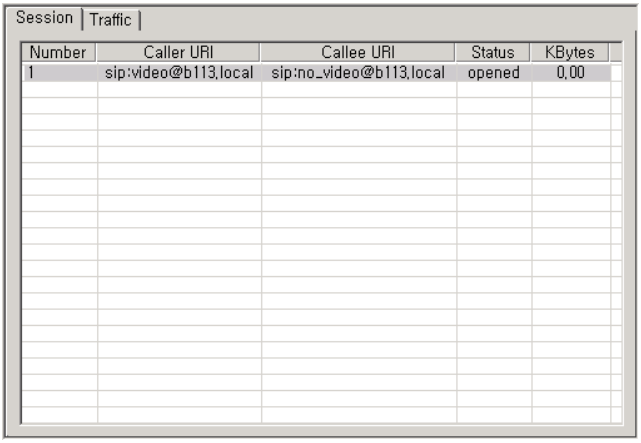
[그림 4.15] 세션 정보 분석 결과(초기 수립시) - 화상 대 음성

[그림 4.15]에서 미디어 협상 과정을 살펴보면 음성의 경우 호출자가 제안한 10개의 오디오 코덱중 호출 받은 사용자는 5개의 코덱이 지원이 가능하다고 알려주었고, 화상의 경우 호출자는 2개의 비디오 코덱을 제안하였지만 호출 받은 사용자는 사용할 포트 번호가 0번이므로 비디오 스트림을 거부하였다는 것을 알 수 있다.

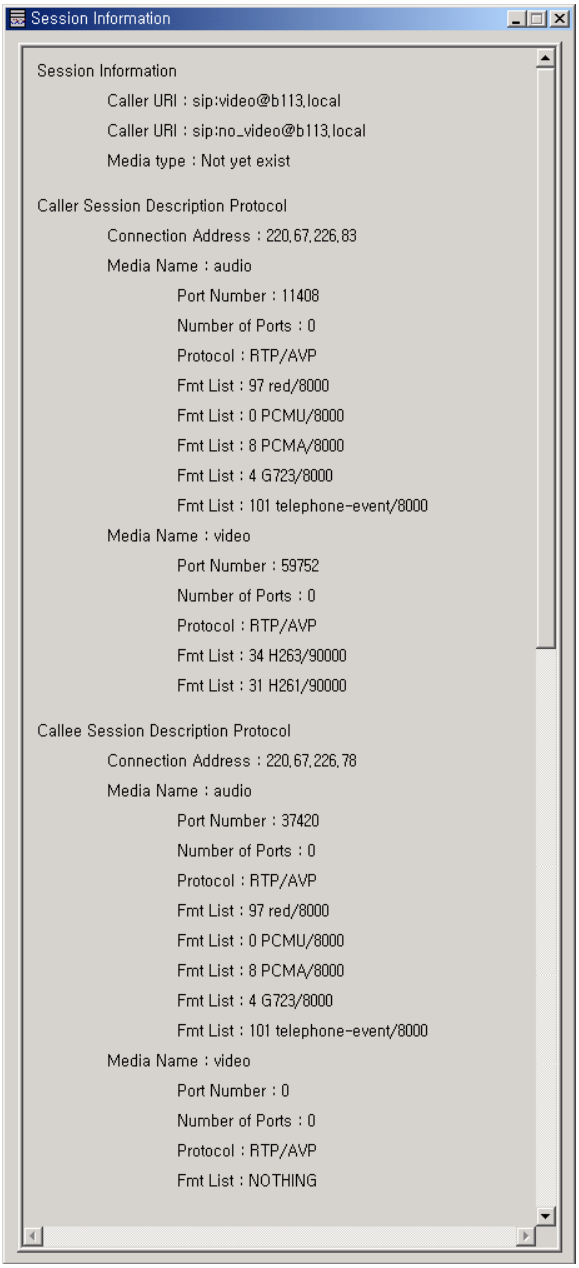


[그림 4.16] 세션 정보 분석 결과(변경시) - 화상 대 음성

[그림 4.16]의 SIP 세션 변경 과정에서 음성의 경우 초기 세션 수립 과정에서 협상한 코덱을 그대로 사용하였고, 화상의 경우 호출자는 다시 화상 채팅을 제안하였지만 거부되었다는 것을 세션 정보 분석 결과를 통하여 판단할 수 있었다. [그림 4.17]과 [그림 4.18]은 화상 채팅 기능을 갖는 SIP User Agent와 음성 채팅 기능을 갖는 SIP User Agent간에 SIP 세션 변경 과정을 통하여 수립된 SIP 세션과 SIP 세션 변경 과정에서 다시 전달된 호출자와 호출 받은 사용자의 세션 정보를 보여주고 있다.



[그림 4.17] 수립된 SIP 세션 - 화상 대 음성

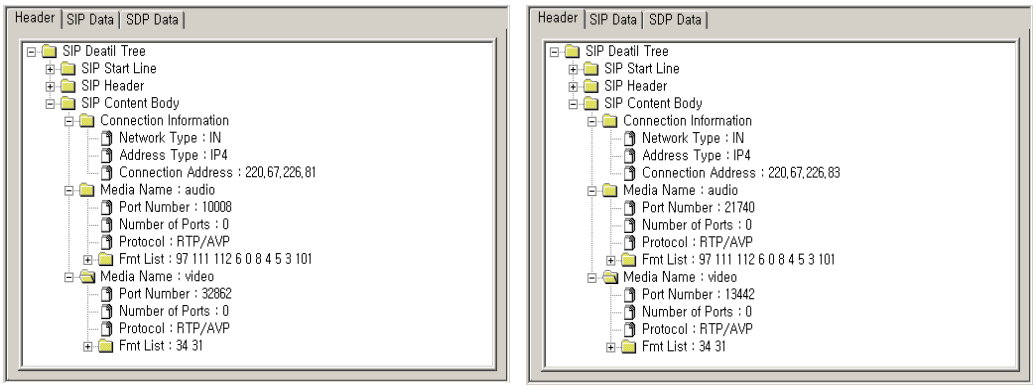


[그림 4.18] SIP 세션 정보 - 화상 대 음성

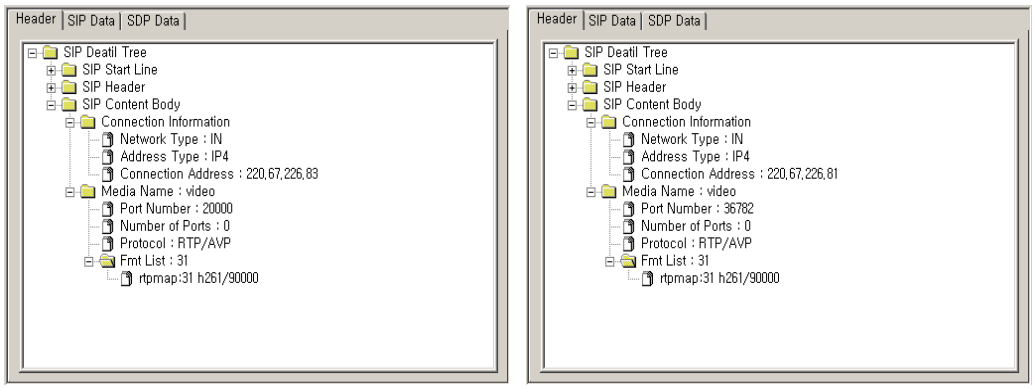
[그림 4.17]과 [그림 4.18]의 결과를 통하여 SIP 세션이 변경되어도 수립된 SIP 세션 정보가 호출자와 호출 받은 사용자의 새로운 세션 정보를 이용하여 수정되어 정상적으로 관리되고 있다는 것을 알 수 있었다.

다. 화상 대 화상 채팅

호출자와 호출 받은 사용자 모두 화상 채팅 기능을 갖는 SIP User Agent를 사용하여 SIP 세션 수립시 발생된 세션 정보를 분석한 결과는 [그림 4.19], [그림 4.20]과 같다.



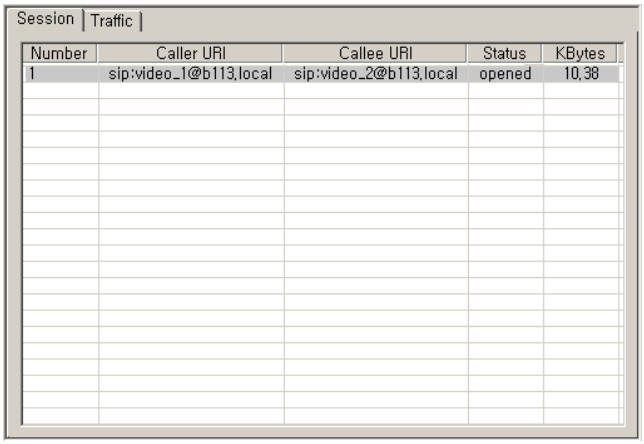
[그림 4.19] 세션 정보 분석 결과 1 - 화상 대 화상



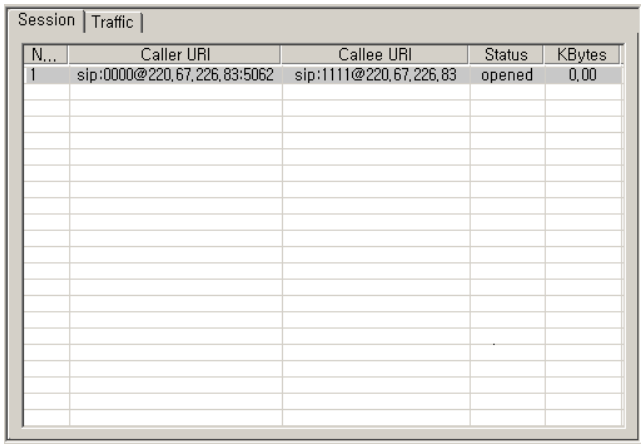
[그림 4.20] 세션 정보 분석 결과 2 - 화상 대 화상

[그림 4.19]는 SIP User Agent로 MSN Messenger를 사용한 경우로 호출자가 화상 채팅 요청시 비디오 코덱 뿐만 아니라 오디오 코덱도 세션 정보에 기술되어 전달되었다는 것을 보여주고 있다. [그림 4.20]은 SIPSet을 사용한 경우로 화상 채팅 기능을 갖도록 설정한 후 채팅 요청시 호출자의 비디오 코덱 정보만 세션 정보에 기술되어 전달되었다는 것을 분석된 결과를 통하여 알 수 있었다. [그림 4.21], [그림

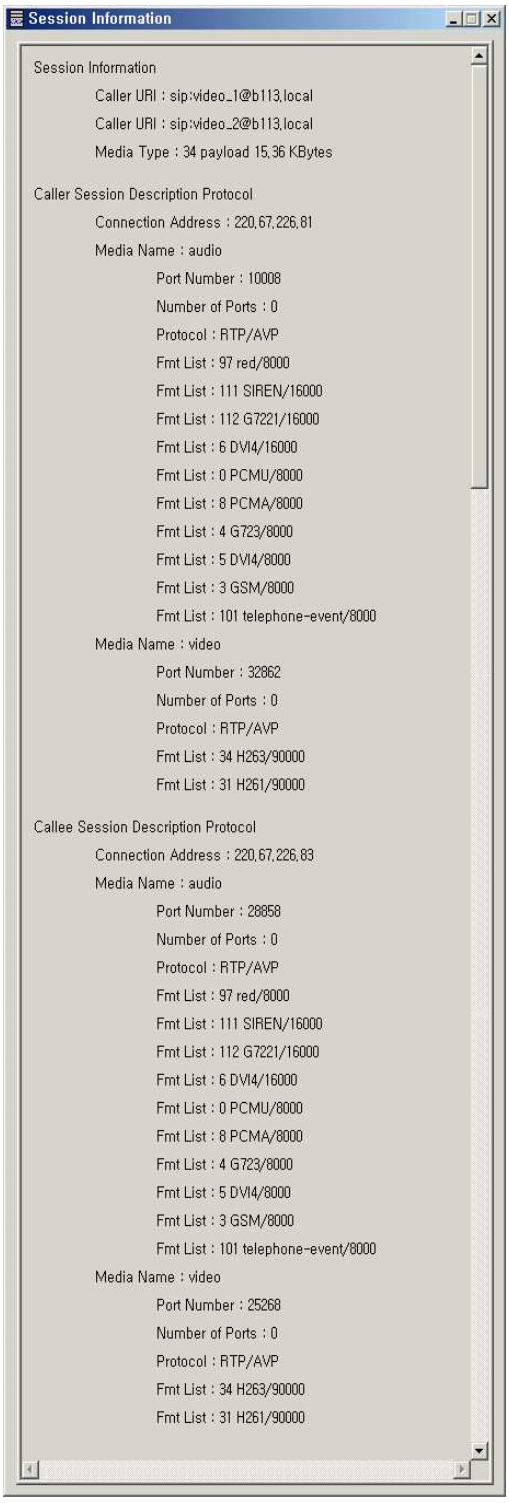
4.22], [그림 4.23]과 [그림 4.24]는 화상 채팅 기능을 갖는 SIP User Agent들의 종류에 상관없이 세션 정보를 분석하고 SIP 세션을 정상적으로 처리한 결과를 보여주고 있다.



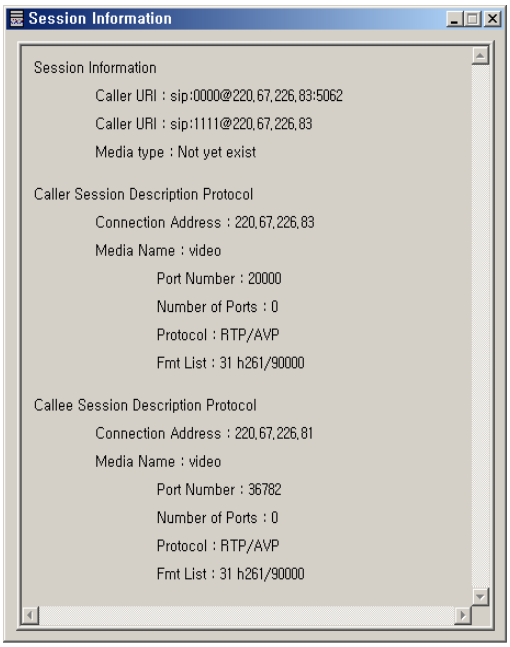
[그림 4.21] 수립된 SIP 세션 1 - 화상 대 화상



[그림 4.22] 수립된 SIP 세션 2 - 화상 대 화상



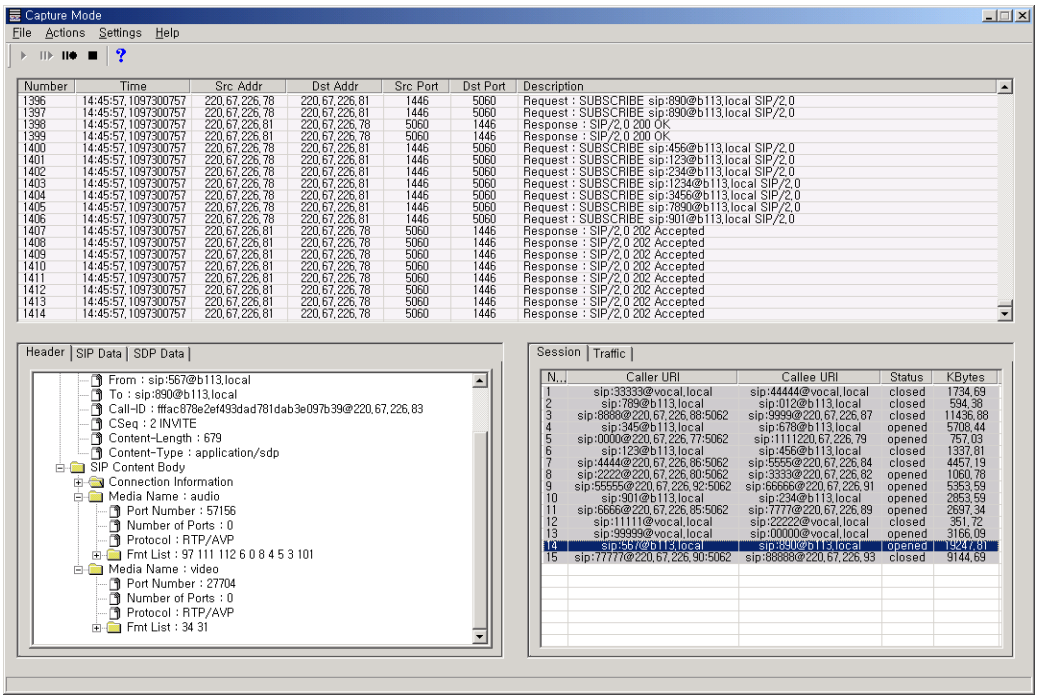
[그림 4.23] SIP 세션 정보 1 - 화상 대 화상



[그림 4.24] SIP 세션 정보 2 - 화상 대 화상

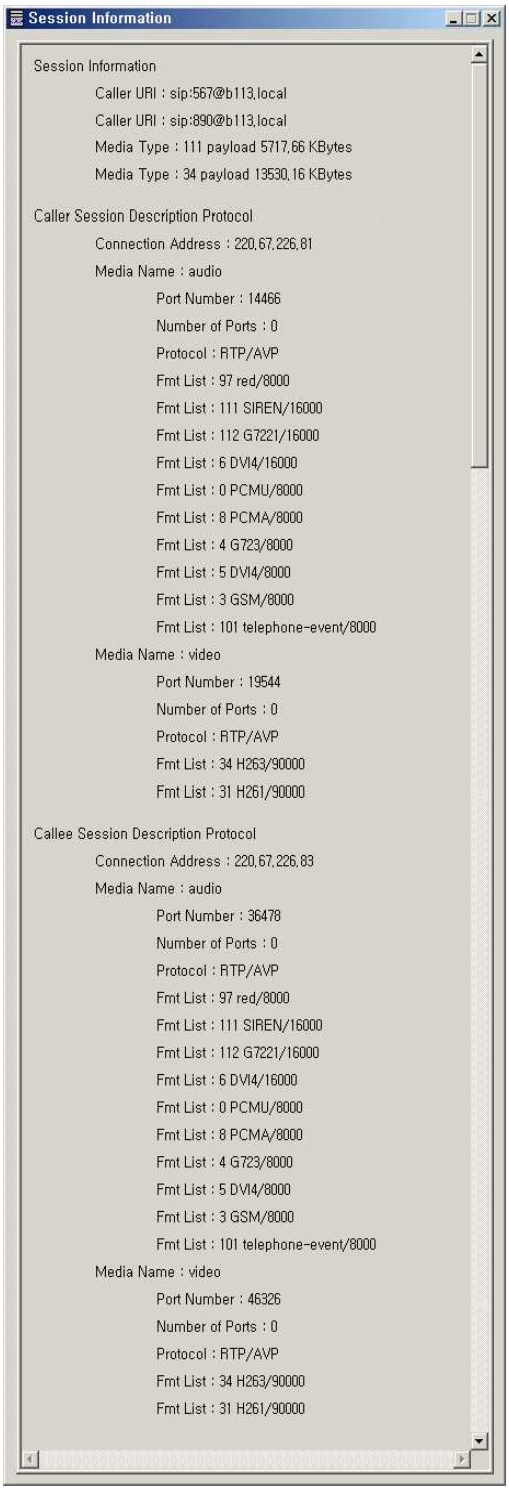
4.2.3 실험 3 - 미디어 트래픽(RTP 패킷) 분석 실험

[표 3.4]의 SIP 세션 생성 시나리오를 이용하여 15개의 SIP 세션을 수립한 후에 RTP 패킷을 이용하여 발생되는 멀티미디어 통신 트래픽을 분석하여 SIP 세션별 발생된 트래픽 통계와 미디어 종류별 트래픽 통계를 처리하는 능력을 실험하였고 결과는 [그림 4.25]와 같다.

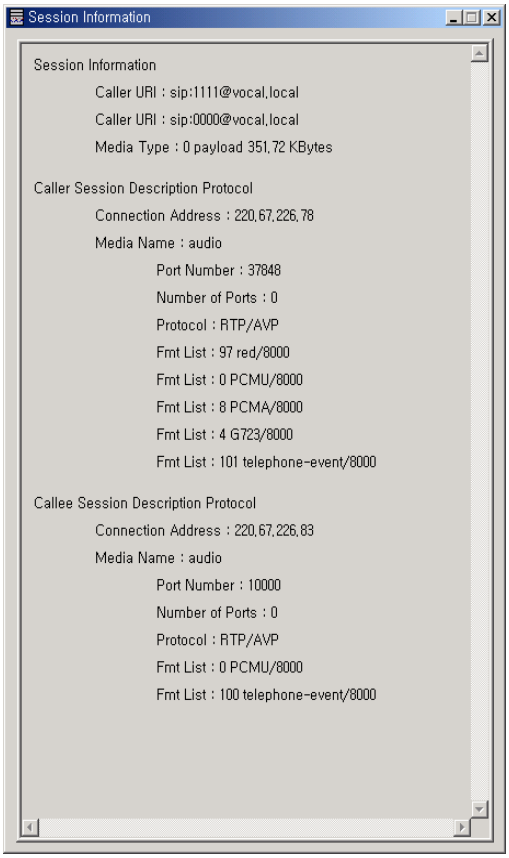


[그림 4.25] 미디어 트래픽 분석 결과

[그림 4.25]에서 보듯이 미디어 트래픽 분석 실험에서 15개의 SIP 세션 수립을 정상적으로 인식하였고 수립된 세션중 14번째 세션에서 가장 많은 트래픽이 발생되었고 12번째 세션에서 가장 적은 트래픽이 발생되었다. 12번째와 14번째 SIP 세션의 미디어 종류별 트래픽 통계 결과는 [그림 4.26], [그림 4.27]과 같다.



[그림 4.26] 미디어 종류별 트래픽 통계 1



[그림 4.27] 미디어 종류별 트래픽 통계 2

[그림 4.26]의 SIP 세션의 세부 정보를 통하여 화상 채팅이 이루어졌으며 H.263 비디오 코덱을 이용한 화상 트래픽이 SIP 세션에서 발생된 총 트래픽의 약 70% 차지하였다는 것을 알 수 있었다. [그림 4.27]의 경우 PCMU 오디오 코덱을 사용한 음성 채팅이 이루어졌으며 SIP 세션에서 발생된 총 트래픽이 음성 트래픽임을 알 수 있었다.

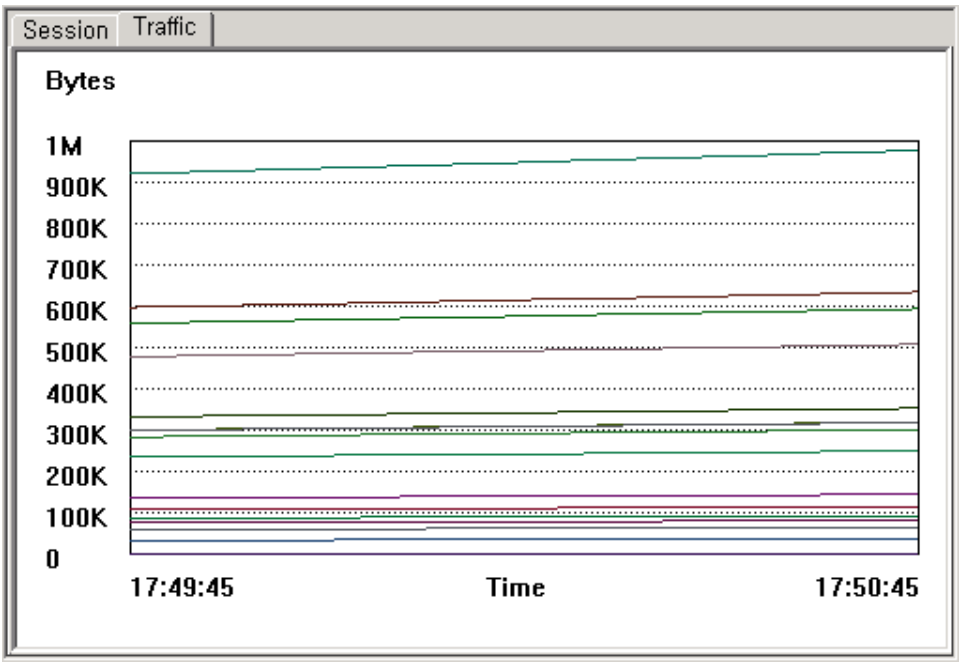
4.2.4 실험 4 - 그래프 인터페이스 실험

실험 3과 같이 SIP 세션 생성 시나리오를 이용하여 15개의 SIP 세션을 수립한 후에 SIP 세션별 실시간으로 발생되는 트래픽 정보를 이용하여 그래프 인터페이스 기능이 정상적으로 동작하는지 실험하였다. [그림 4.28]은 SIP 세션 생성 시나리오에 의해 수립된 SIP 세션과 세션의 상태 정보를 보여주고 있으며, [그림 4.29]는 수립된 SIP 세션의 트래픽 량을 선 그래프를 이용하여 SIP 세션별 트래픽 정보를 표현하고 있다.

Session | Traffic

N...	Caller URI	Callee URI	Status	KBytes
1	sip:123@b113.local	sip:456@b113.local	opened	145.86
2	sip:11111@vocal.local	sip:22222@vocal.local	opened	12.06
3	sip:8888@220.67.226.88:5062	sip:9999@220.67.226.87	opened	986.10
4	sip:4444@220.67.226.86:5062	sip:5555@220.67.226.84	opened	300.12
5	sip:567@b113.local	sip:890@b113.local	opened	638.62
6	sip:6666@220.67.226.85:5062	sip:7777@220.67.226.89	opened	97.81
7	sip:901@b113.local	sip:234@b113.local	opened	48.95
8	sip:2222@220.67.226.80:5062	sip:3333@220.67.226.82	opened	75.22
9	sip:33333@vocal.local	sip:44444@vocal.local	opened	96.38
10	sip:789@b113.local	sip:012@b113.local	opened	254.75
11	sip:77777@220.67.226.90:5062	sip:88888@220.67.226.93	opened	500.02
12	sip:55555@220.67.226.92:5062	sip:66666@220.67.226.91	opened	322.68
13	sip:99999@vocal.local	sip:00000@vocal.local	opened	597.16
14	sip:0000@220.67.226.77:5062	sip:1111220.67.226.79	opened	108.89
15	sip:345@b113.local	sip:678@b113.local	opened	350.66

[그림 4.28] SIP 세션 생성 시나리오에 의해 생성된 세션



[그림 4.29] 그래프 인터페이스 기능의 실험 결과

[그림 4.28]과 [그림 4.29]를 통하여 SIP 세션이 수립되었을 경우 SIP 세션에서 발생되는 트래픽 정보를 그래프 인터페이스가 실시간으로 화면에 표시하는 것을 확인할 수 있었다.

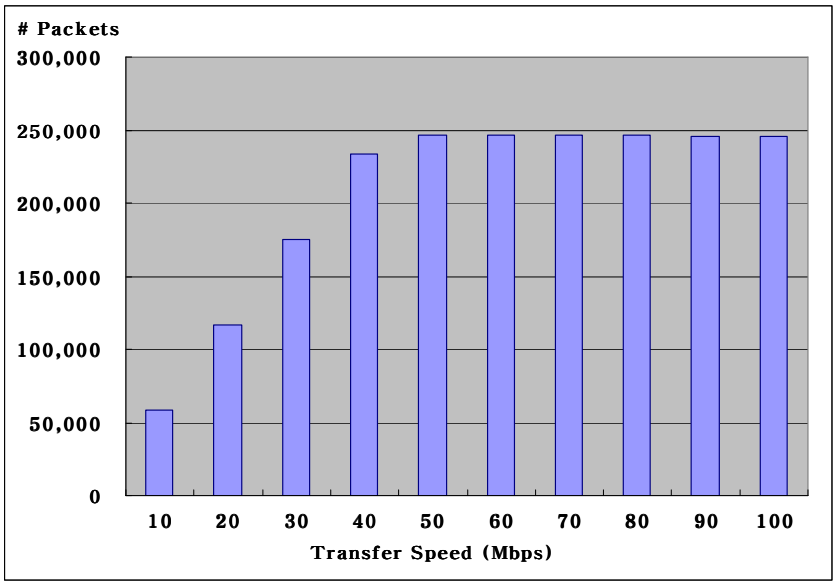
4.2.5 실험 5 - STAT의 부하 테스트

트래픽 발생기를 이용하여 미디어 트래픽에 해당하는 RTP 패킷을 생성하고 전송속도를 10Mbps부터 100Mbps까지 늘려가면서 트래픽을 발생시켜 미디어 트래픽 분석 성능을 측정하였다.

구 분	내 용
OS	Windows 2000 Server
CPU	Pentium 4 - 1.8GHz
RAM	768MB
NIC	Realtek RTL8139(A) PCI Fast Ethernet Adapter

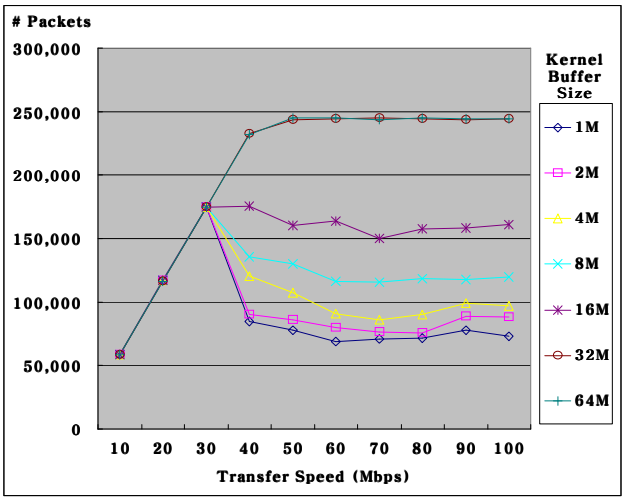
[표 4.5] 부하 테스트시 사용된 시스템의 사양

멀티미디어 트래픽 중 페이로드 타입이 0인 PCMU 코덱을 사용할 경우에 160byte의 음성 데이터가 RTP 패킷에 내장된다. [그림 4.30]은 이와 동일한 음성 트래픽을 트래픽 발생기를 이용하여 10초 동안 지연시간 없이 발생시킨 결과로 트래픽을 전송속도별로 보여주고 있다.



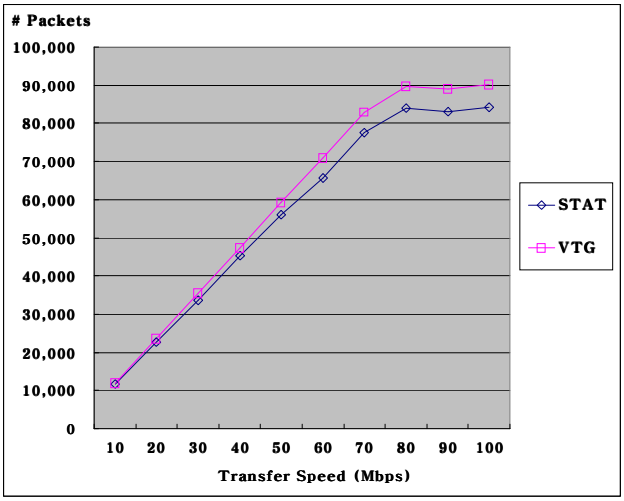
[그림 4.30] 트래픽 발생기에 의해 발생된 음성 트래픽 량

[그림 4.31]은 트래픽 발생기에 의해 발생된 음성 트래픽을 STAT의 커널 버퍼 크기 및 패킷 버퍼 크기 옵션을 다르게 하여 분석한 결과를 보여주고 있다. [그림 4.31]에서 보듯이 커널 버퍼 크기가 16Mbyte 미만인 경우에는 최대 70% 이상 손실이 발생하였다는 것을 알 수 있다. 반면에 커널 버퍼 크기를 32Mbyte 이상으로 설정하였을 경우에는 99% 이상의 미디어 트래픽 분석 성능을 보였다.



[그림 4.31] 부하 테스트 결과 - 음성 트래픽

멀티미디어 트래픽 중 화상 트래픽은 음성 트래픽과 데이터 크기면에서 차이가 있다. 즉 화상 데이터는 프레임 단위로 전송되는데 움직임이 없을 경우에는 하나의 RTP 패킷에 내장되고 움직임이 많을 경우에는 여러 개의 패킷에 내장되어 전송된다. 실험에서는 화상 데이터 크기를 1Kbyte로 고정시키고 트래픽 발생기를 이용하여 트래픽을 발생시켰다. [그림 4.32]는 STAT의 커널 버퍼 크기를 64Mbyte로 설정하여 분석 성능을 측정한 결과로 전송속도가 100Mbps인 경우 약 7% 미만의 패킷 손실이 발생하였다.



[그림 4.32] 부하 테스트 결과 - 화상 트래픽

제 5장 결론 및 향후 연구

인터넷 텔레포니 기술은 기존 공중전화망 중심의 음성 및 데이터망 서비스에서 패킷망 중심의 음성 및 데이터망 서비스로의 진화적인 측면에서 매우 중요한 기술로써 인식되고 있지만 인터넷 텔레포니 기술의 역사는 그리 길지 않고 인터넷 기술의 급속한 발전에 따라 아직도 표준으로 정리되지 않은 분야들이 매우 많다. 기본적인 세션 제어에 대한 기술만이 어느 정도 완성단계에 들어가 있으며 관리를 위한 요소와 각 시스템간의 상호 운용성 등에 대해서는 계속적으로 표준화가 진행되고 있다.

본 논문에서는 이러한 상황을 고려하여 SIP를 이용한 인터넷 텔레포니 시스템 환경에서 발생되는 멀티미디어 통신 트래픽을 분석하여 망을 진단할 수 있는 트래픽 분석기 STAT(SIP based Traffic Analysis Tool)를 설계하고 구현하였다.

4장 실험 및 성능 평가에서는 실제 프로젝트에서 많이 사용되는 SIP 서버인 실시간 통신 서버와 VOVIDA 서버, SIP User Agent인 Microsoft Messenger와 SIPSet를 이용하여 실험과 성능을 평가하였다. 실험 결과에서 보듯이 SIP를 기반으로 하는 SIP User Agent에서 발생된 SIP 패킷이 정상적으로 분석되었고 분석된 결과를 통하여 수립된 SIP 세션이 인식되고 처리되었다. 또한 수립된 SIP 세션에서 발생된 멀티미디어 통신 트래픽을 분석하여 미디어 종류와 미디어 종류별 트래픽 정보를 알아낼 수 있다는 것을 보여주었다.

실험 1에서는 사용되는 SIP 서버와 SIP User Agent의 종류에 상관없이 발생되는 SIP 패킷을 모두 수집하여 분석하는 것을 보았다. 또한 SIP 서버와 SIP User Agent의 기능에 따라 로그인, SIP 세션 요청 및 로그아웃의 각 과정에서 발생되는 SIP 패킷 량에 많은 차이가 있음을 발견하였다. 즉 온라인 상태 인식 기능의 유무에 따라 많은 수의 “NOTIFY”와 “SUBSCRIBE” SIP 패킷이 추가적으로 발생되었다. 멀티미디어 통신 트래픽은 대부분이 미디어 트래픽이 차지한다고 볼 수 있지만 SIP 서버에 로그인, SIP 세션 요청 및 로그아웃 과정에서 추가적으로 발생되는 트래픽을 무시할 수 없음을 확인하였다.

실험 2에서는 SIP 세션 수립 과정에서 발생되고 SDP에 의해 기술되는 세션 정보

를 정상적으로 분석하여 트리 컨트롤을 이용한 사용자 인터페이스에 나열되는 것을 볼 수 있었고, SIP 세션이 수립될 경우 SIP 세션의 상태 정보와 호출자와 호출 받은 사용자의 세션 정보가 지속적으로 유지되고 관리되는 것을 호출자는 화상 채팅 기능을 갖고 호출 받은 사용자는 음성 채팅 기능을 갖는 경우의 SIP 세션 수립 실험에서 발생된 SIP 세션 변경 과정을 통하여 확인할 수 있었다.

실험 3에서는 호출자와 호출 받은 사용자간의 실시간 미디어 데이터 전달에 사용되는 RTP 패킷을 분석하여 SIP 세션별 발생된 트래픽 정보를 얻을 수 있었고 SIP 세션별 발생된 트래픽을 음성 트래픽과 화상 트래픽으로 나누어 세부적으로 표시하여 미디어 종류별 트래픽 통계 정보를 확인하였다.

실험 4에서는 미디어 통계를 표시하는 그래프 인터페이스의 성능을 평가하였다. SIP 세션 생성 시나리오를 이용하여 15개의 SIP 세션을 생성한 후에 SIP 세션별로 발생되는 트래픽 정보를 이용하여 선 그래프, 원 그래프와 막대 그래프가 실시간으로 화면에 트래픽 량을 표시하는 것을 알 수 있었다.

실험 5에서는 트래픽 발생기를 이용하여 실제 환경에서 발생되는 트래픽과 유사한 트래픽을 발생시켜 부하 테스트를 수행한 결과 실험 환경에서 STAT가 정상적으로 동작하기 위해서 요구되는 커널 메모리 크기가 32Mbyte 이상이 되어야 한다는 것을 알 수 있었다.

실험 및 성능 평가를 통하여 하나 이상의 참여자로 구성되는 세션을 초기화하고, 변경 및 종료하기 위해 사용하는 SIP 패킷, SIP 패킷의 메시지 본문에 멀티미디어 세션을 기술하기 위하여 사용되는 SDP와 실시간 미디어 데이터를 세션 참여자간 전송에 사용되는 RTP 패킷을 모두 수집하고 분석하여 멀티미디어 통신 트래픽을 측정하고 세부적으로 분석할 수 있다는 것을 보였다.

본 논문에 이은 향후 연구로는 다양한 조건으로 트래픽을 분석할 수 있도록 기능 보완과 사용자 인터페이스의 개선이다. 그리고 현재 많이 사용되고 있는 이동성이 좋고 휴대하기 편리한 WinCE 기반 PDA용 트래픽 분석 도구를 개발하는 것이다.

참고문헌

[1] http://www.voip-forum.or.kr, VoIP Forum Homepage.

[2] 황세진, 박성순, "차세대 VoIP 통신개요 및 기술동향", 한국멀티미디어학회지, Vol.7, No.5, pp.139-147, 2003.

[3] 이강석, 염창선, 황기현, "CTI/VoIP 기반 인터넷 콜시스템의 설계에 관한 연구", IE Interface, Vol.15, No.4, 2002.

[4] ITU-T H.323, Packet-based Multimedia Communications Systems, September 1999.

[5] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg, SIP: Session Initiation Protocol, RFC 2543, March 1999.

[6] IETF SIP Working Group, http://www.ietf.org/html.charters/sip-charter.html.

[7] http://www.3gpp.org, 3rd Generation Partnership Project Homepage.

[8] http://www.3gpp2.org, 3rd Generation Partnership Project 2 Homepage.

[9] 3rd Generation Partnership Project, "IP Multimedia(IM) Subsystem-Stage 2", 3GPP TS23.228 V5.1.0, 2001.

[10] International Softswitch Consortium, "ISC Reference Architecture V 1.2", 2002.

[11] F. Risso and L. Degioanni, Analyzer: a public domain protocol analyzer, http://analyzer.polito.it.

[12] M. Handley, V. Jacobson, SDP: Session Description Protocol, RFC 2327, April 1998.

[13] H. Schulzrinne, S. Casner, R. Frederick and Y. Jacobson, RTP: A Transport Protocol for Real-Time Applications, RFC 1889, January 1996.

[14] E. Wedlund, H. Schulzrinne, "Mobility Support using SIP", VON Europe Spring 2000, June 2000.

[15] IETF, “SIP: Session Initiation Protocol”, RFC 3261, June 2002.

[16] http://www.iana.org/assignments/rtp-parameters, RTP Parameters.

[17] http://winpcap.polito.it/docs/default.htm, WinPcap Homepage.

[18] F. Risso and L. Degioanni, “An Architecture for High Performance Network Analysis”, Proceedings of the Sixth IEEE Symposium on Computers and Communications, pp.686-693, 2001.

[19] http://www.microsoft.com/office/livecomm/prodinfo/default.mspx, Microsoft Live Communication Server 2003 Homepage.

[20] http://www.vovida.org, VOVIDA Homepage.

[21] http://www.microsoft.com/windows/messenger, Microsoft Messenger Homepage.

[22] http://www.vovida.org/applications/downloads/sipset, SIPSet Homepage.

[23] 정인환, 김진환, 비쥬얼 이더넷 트래픽 발생기의 설계 및 구현, 제18회 정보처리학회 추계학술대회, Vol.9, No.5, 2002.

ABSTRACT

Design and Implementation of Traffic Analyzer for SIP Based Multimedia Communication System

Lee, Jae-Jong
Major in Computer Engineering
Dept. of Computer Engineering
Graduate School
Hansung University

As Internet service is extended, the Internet Telephony technology(VoIP : Voice over IP) is getting more attention as a major communication service combined with multimedia capability. The quality of this service depends on the available traffic rates because it transfers large amount of audio and video data. *SIP(Session Initiation Protocol)* was proposed for session control of this VoIP service and is expected to be a standard session control protocol. In this circumstance, a traffic analyzer for measuring and testing VoIP service is required. However, a tool for traffic analysis of SIP based multimedia service is difficult to find up to now.

In this thesis, we propose a tool that measures and analyzes multimedia communication traffic generated by VoIP service based on SIP protocol. This tool, by capturing low level Internet packets, measures and analyzes SIP based multimedia communication protocol and traffic. By utilizing the statistics of each service and showing a realtime monitoring information graphically, this tool provides a useful information for testing and managing SIP based network. The traffic analyzer, which is called *STAT*(SIP based Traffic Analysis Tool) in this thesis, will be a helpful tool for the development and management of SIP based multimedia communication system.