

碩士學位論文

指導教授 鄭仁煥

IMS 망에서 PTT 서비스의 호 설정 및 통화
성능향상을 위한 프로토콜 압축 기법

A Protocol Compression Method for Improving Call Setup and
Call Bandwidth of Push-To-Talk Service over IMS

2006年 12月 日

漢城大學校 一般大學院

컴퓨터 工學科

컴퓨터 工學 專攻

金 秀 熙

碩士學位論文
指導教授 鄭仁煥

IMS 망에서 PTT 서비스의 호 설정 및 통화
성능향상을 위한 프로토콜 압축 기법

A Protocol Compression Method for Improving Call Setup and
Call Bandwidth of Push-To-Talk Service over IMS

위 論文을 컴퓨터工學 碩士學位論文으로 提出함

2006年 12月 日

漢城大學校 一般大學院

컴퓨터 工學科

컴퓨터 工學 專攻

金 秀 熙

金秀熙의 工學 碩士學位論文을 인정함

2006年 12月 日

심사위원장 김 영 응



심사위원 황 기 태

A handwritten signature in black ink, corresponding to the name Hwang Gi-tae.

심사위원 정 인 환



목 차

제 1장 서론	1
1.1 연구배경	1
1.2 연구동기	5
제 2장 관련연구	9
2.1 프로토콜 헤더 압축	9
2.1.1 기존 프로토콜 문제점	9
2.1.2 프로토콜 헤더 압축 기법	10
2.2 호 처리 향상 기법	14
2.2.1 Always On	14
2.2.2 Dormant 페이징과 Multiple Access 채널	17
2.2.3 페이징 부하 개선	18
2.2.4 단말기의 버퍼 추가	21
2.3 통화 성능 향상 기법	23
2.4 호 처리 지연 및 통화 대역폭 부족 요인	24
제 3장 호 설정 및 통화 성능 향상을 위한 프로토콜 압축 기법	27
3.1 A+Sigcomp : 호 설정 향상을 위한 프로토콜 압축 기법	28
3.2 A+ROHC : 통화 성능 향상을 위한 프로토콜 압축 기법	36
제 4장 성능평가	48
4.1 호 설정 향상 성능 평가	48
4.1 통화 성능 향상 성능 평가	50
제 5장 결론 및 향후 연구	54
참고문헌	55
Abstract	59

표 목차

[표 1.1] IMS 구성요소	4
[표 1.2] 각 통신 방식별 비교	7
[표 2.1] Compressing TCP/IP Headers for Low-Speed serial Links 패킷 타입	11
[표 2.2] IP Header Compression 패킷 타입	12
[표 2.3] Compression IP/UDP/RIP Headers for Low-Speed Serial Links 패킷 타입	14
[표 2.4] 단말의 상태구분	15
[표 3.1] Sigcomp Stack 구성요소 기능	31

그림 목차

[그림 1.1] 각 기반 망에 대한 PPT 서비스 비교 분석 도표	2
[그림 1.2] IMS 망 구성도	4
[그림 1.3] PoC 시스템의 망 구성 및 PoC 동작 원리	5
[그림 2.1] RTP 헤더 크기	9
[그림 2.2] Compression_TCP 헤더 포맷	11
[그림 2.3] Always On 상태 천이도	16
[그림 2.4] GAN 단위의 GPM 전송 방법 - TCH 이용 INVITE 전송	19
[그림 2.5] GAN 단위의 GPM 전송 방법 - TCH 이용 INVITE 전송	20
[그림 2.6] 단말기 음성 버퍼 메시지 흐름	22
[그림 2.7] 단말기 내부 동작 흐름도	22
[그림 2.8] VoIP 망 구성 예	25
[그림 2.9] 지터 개념	26
[그림 3.1] SIP 스택	29
[그림 3.2] Sigcomp Interface 모델	30
[그림 3.3] SIP 메시지의 구조	32
[그림 3.4] SIP Directory	34
[그림 3.5] 기존 Sigcomp와 A+Sigcomp Stack	35
[그림 3.6] IP/UDP/RTP 패킷 크기 비교	37
[그림 3.7] Compressor에서 상태 전이	38
[그림 3.8] Decompressor에서 상태 전이	39
[그림 3.9] ROHC 초기화 과정 및 동작절차	41
[그림 3.10] W-LSB 동적 헤더 Field 동작과정	45
[그림 4.1] 패킷 크기 비교	48
[그림 4.2] 패킷 오버 헤드 비교	49
[그림 4.3] 메시지 압축 처리 시간 비교	50
[그림 4.4] SO 상태에서 오버 헤드 비교	51
[그림 4.5] 절약할 수 있는 대역폭 비교	52

약어표

3GPP	3rd Generation Partnership Project
A+ROHC	Advanced ROHC
A+SigComp	Advanced SigComp
AS	Application Server
BS	Base Station
CDR	Charging Data Record
CID	Context IDentifier
CNG	Comfortable Noise Generation
CRC	Cyclic Redundancy Check
CSCF	Call Session Control Function
DTX	Discontinuous Transmission
EV-DO	EVolution Data Only
FC	Full Context
FO	First Order
GAN	Global Action Network
GLMS	Group/List Management Server
GPM	General Page Message
HLR	Home Location Register
HSDPA	High Speed Data Packet Access
HSS	Home Subscriber Server
HTTP	HyperText Transfer Protocol
I-CSCF	Interrogating-CSCF
IETF	Internet Engineering Task Force
IM	Instance Message
IMS	IP Multimedia Subsystem
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
IR	Initialization and Refresh
LD	Load Distributor
MDN	Mobile Directory Number
MRS	Media Relay Server
MS	Mobile Station
MSRP	Message Session Relay Protocol
NACK	Negative ACK
NC	No Context

OAM	Operation, Administration, Maintenance
O-mode	Bidirectional Optimistic mode
PCF	Packet Control Function
P-CSCF	Proxy-CSCF
PDSN	Packet Data Switch Network
PoC	Push-to-talk over Cellular
PTD	Push-To-Data
PTT	Push-To-Talk
R-mode	Bidirectional Reliable mode
ROHC	RObust Header Compression
RTCP	Real Time Control Protocol
RTP	Real-time Transport Protocol
SC	Static Context
S-CSCF	Serving-CSCF
SCT	Session Closure Timer
SDB	Short Data Burst
SDP	Session Description Protocol
SigComp	Signalling Compression
SIP	Session Initiation Protocol
SMTP	Simple Mail Transport Protocol
SO	Second Order
TCH	Traffic CHannel
TDM	Time Division Multiplexing
TRS	Trunked Radio System
UDVM	Universal Decompressor Virtual Machine
U-mode	Unidirectional mode
URL	Public Uniform Resource Location
VAD	Voice Activity Detection
VJHC	Van Jacobson Header Compression
VOX	Voice Operated Transmitting
WCDMA	Wide Code Division Multiple Access
WIBRO	Wireless Broadband Internet
W-LSB	Window-Least Significant Bit

요 약

전 세계 대부분의 이동통신업체들은 무선 패킷 망을 통한 고속의 멀티미디어 데이터 전송이 가능함으로 인하여 유/무선, 음성/영상/데이터, 통신/방송/사내 망들의 통합할 수 있는 관리구조로 3GPP(3rd Generation Partnership Project)에서 IMS(IP Multimedia Subsystem)구조가 제안되었다. 통신사업자들은 IMS망구조를 이용하여 다양한 멀티미디어 서비스 고도화된 신규 서비스를 보다 쉽고, 비용 효율적으로 개발이 가능할 것으로 전망된다. 이러한 신규 서비스 중의 하나가 IMS 망 기반의 PTT(Push-To-Talk) 이다.

PTT란 유, 무선 통합 인터넷 환경으로 진화하고 있는 통신망을 통하여 음성 및 텍스트 기반의 일대일 및 그룹 즉시 통신을 제공하기 위한 서비스이다. 단어의 의미 그대로 스위치를 누르고 말하면서 즉시 의사소통을 할 수 있으므로 일반적인 통화 유형의 대기시간에 비교할 때 상대적으로 빠른 통신 서비스를 제공할 수 있다.

IMS 기반 망에서 PTT 서비스를 하기 위기 위해서는 현재 상용화 되고 있는 Mobile 망, TRS 망과 동일한 콜 셋업을 가져야 하며, 콜 셋업 및 통화시의 주파수 자원 사용이 충분해야 한다. IMS망에서 PTT 서비스를 할 때 호 설정에는 SIP가 사용되며, 호 설정이 완료 된 후 통화 시에는 RTP가 사용된다. 두 프로토콜은 멀티미디어 통신을 위해 고안된 프로토콜이며 SIP 경우 1000byte이상의 큰 메시지를 가지고 있으며, RTP의 경우 음성 데이터에 비해 헤더의 크기가 크다는 것을 알 수 있다. 본 논문에서는 IMS 망 PPT 서비스의 성능 향상을 위해 통화 설정 대기 시간 단축 및 대역폭 확보를 위한 프로토콜 압축 기법인 *A+SigComp*, *A+ROHC* 기법을 제안하고 실험을 통해 성능을 평가하였다.

제 1장 서론

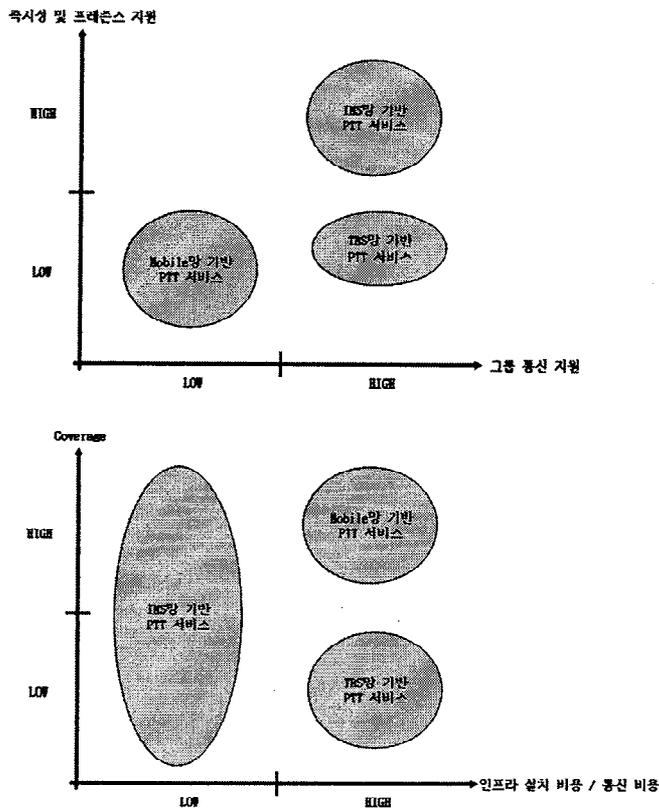
1.1 연구배경

인터넷의 보급과 확산에 따른 인터넷 기술의 발전과 이동통신 사용자들의 고속 데이터 Traffic과 다양한 서비스 기능에 대한 요구가 증가함에 따라 현재 이동통신망은 기존의 TDM(Time Division Multiplexing)방식의 코어 망에서 IP(Internet Protocol) 방식의 코어 망으로 이동하고 있다. 전 세계 대부분의 이동통신업체들은 이동통신망의 발전 및 광대역 코드 분할 다중 접속(WCDMA : Wide Code Division Multiple Access), 초고속 데이터 패킷 접속 (HSDPA : High Speed Data Packet Access), WIBRO(Wireless Broadband Internet) 등의 시스템 도입에 따라 무선 패킷 망을 통한 고속의 멀티미디어 데이터 전송이 가능하다.[1] 이러한 통신서비스의 발달로 인하여 유/무선, 음성/영상/데이터, 통신/방송/사내 망들의 통합할 수 있는 관리구조로 3GPP(3rd Generation Partnership Project)에서 IMS(IP Multimedia Subsystem)구조가 제안되었다.[2] 통신사업자들은 IMS 망구조를 이용하여 다양한 멀티미디어 서비스와 유/무선 환경 통합 및 통신/방송 융합에 대응하는 고도화된 신규 서비스를 보다 쉽고, 비용 효율적으로 개발이 가능할 것으로 전망된다. 이러한 신규 서비스 중의 하나가 IMS 망 기반의 PTT(Push-To-Talk)이다. 이동 전화 시장의 가입자 수가 포화 상태에 이르면서 각 업체들은 가입자 당의 평균수익을 증대 시키려는 방안으로 기본적인 음성서비스 이외에 SMS, 무선인터넷, 포토메시지, Mobile Game등 다양한 멀티미디어 서비스를 제공하고 있으며, 한 걸음 더 나아가 차별적이고 독특한 부가서비스를 개발 중이다. 그 중에서도 음성 및 텍스트 기반의 일대일 및 그룹 즉시 통신이 가능한 PTT(Push-To-Talk)서비스를 개발 중이다.[3]

PTT란 유, 무선 통합 인터넷 환경으로 진화하고 있는 통신망을 통하여 음성 및 텍스트 기반의 일대일 및 그룹 즉시 통신을 제공하기 위한 서비스이다. 단어의

의미 그대로 스위치를 누르고 말하면서 즉시 의사소통을 할 수 있으므로 일반적인 통화 유형의 대기시간에 비교할 때 상대적으로 빠른 통신 서비스를 제공할 수 있다.

PTT의 활용사례를 살펴보면, 운송/창고/유통/제조업체의 현장 내 혹은 사무실과의 즉시 그룹통신, 직원 간 1:1 및 1:N 동시통화로 조직 내 신속한 의사전달 및 결정 지원, 영업, A/S 요원의 신속한 호출, 병원 등 의료기관에서 신속한 호출 수단, 놀이공원/스키장에서 사용 등 다양한 용도로 사용할 수 있는 장점이 있으며, 기존의 휴대폰으로 PTT를 사용할 수 있고 저렴한 요금의 안정적 서비스를 제공한다면, 대부분의 휴대폰 이용자들은 이 서비스를 이용하리라 예상된다.[3]



[그림 1.1] 각 기반 망에 대한 PPT 서비스 비교 분석 도표

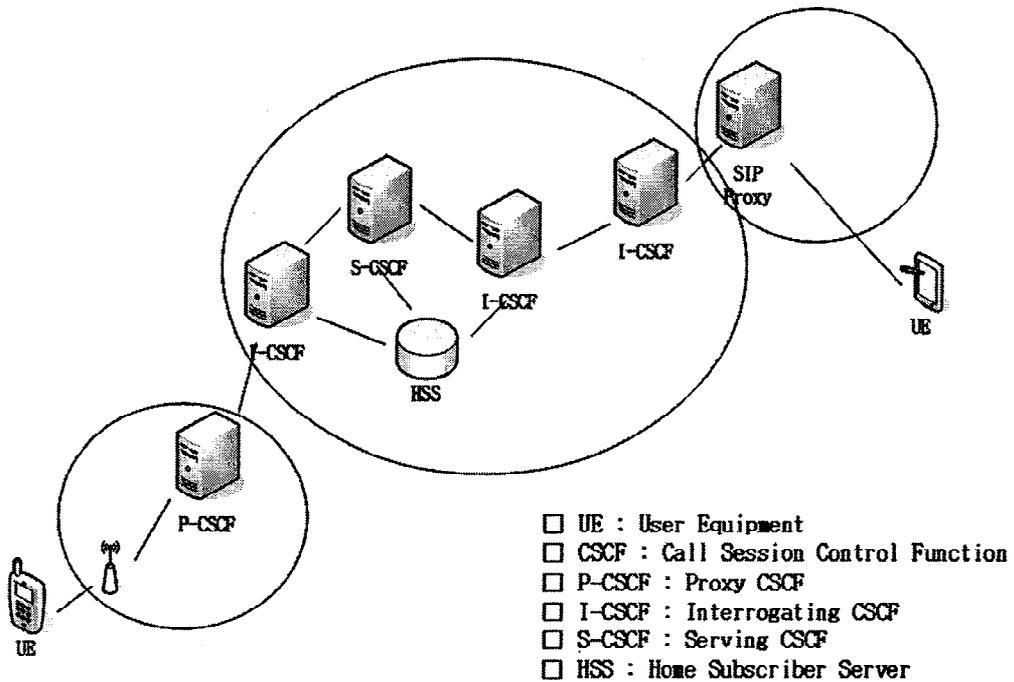
현재 상용화 되고 있는 PTT는 TRS(Trunked Radio System)망과 Mobile망에서 서비스를 하고 있다. Mobile망에서는 아직 활성화 되지 않은 상태에 있으며, TRS망에서는 택시, 버스와 같은 교통통신체계에 적용하여 상용화 되고 있다. [그림 1.1]과 같이 PTT는 즉시성 및 프레즌스 정보를 제공하며 그룹 통신 지원 측면에서 Mobile망에 부족한 면을 볼 수 있다. 즉시성 서비스는 PPT의 핵심 기능으로 통화 버튼을 누른 동시에 말을 할 경우 통화에 해당하는 그룹 전원이 현재 말한 음성을 들을 수 있는 서비스다. 프레즌스 서비스는 가입자의 여러 가지 상태를 나타내는 데 접속 중, 미 접속 중, 회의 중, 통화 중, 수신 거부 등이 포함된다. TRS망에서는 즉시성은 있으나 프레즌스 지원이 안 되며 Coverage가 낮고 인프라 설치에 많은 비용이 소요 된다는 단점을 가지고 있다. 또한 전용 주파수 할당을 요구하여야 하는 단점도 포함하고 있다. 이에 반해 IMS망 기반 PTT 서비스는 기존망과 비교하여 넓은 Coverage와 저렴한 인프라 설치비용, 통신비용을 가지고 있으며 즉시성 및 프레즌스 정보를 제공한다.[4][5]

IMS는 세션 제어 계층과 게이트웨이 계층, 그리고 애플리케이션 계층 등 세 가지 논리적 핵심 구성요소로 정의된다.[6] 첫째, 세션 제어 계층은 SIP(Session Initiation Protocol)서비스를 가능하게 해주는 서버들로 구성돼 있으며, 콜 세션 컨트롤 기능(CSCF : Call Session Control Function)과 HSS(Home Subscriber Server)를 포함한다. CSCF는 SIP기반의 멀티미디어 세션 제어를 위한 기본 기능을 수행하는 인프라 시스템으로서 역할에 따라 P-CSCF(Proxy-CSCF), I-CSCF(Interrogating-CSCF), S-CSCF(Serving-CSCF)로 나뉜다. 이 시스템에서는 가입자 등록, 인증, 과금, 서비스별 트리거링 및 해당 애플리케이션 서버로의 라우팅, 착신자 위치 조회, SIP 메시지 압축 및 해제를 처리한다. HSS는 이동통신의 HLR(Home Location Register)에서 발전한 것으로 가입자 정보를 관리하는 중앙화 된 데이터베이스다. HSS는 사용자 등록/변경 관리, 인증, 권한

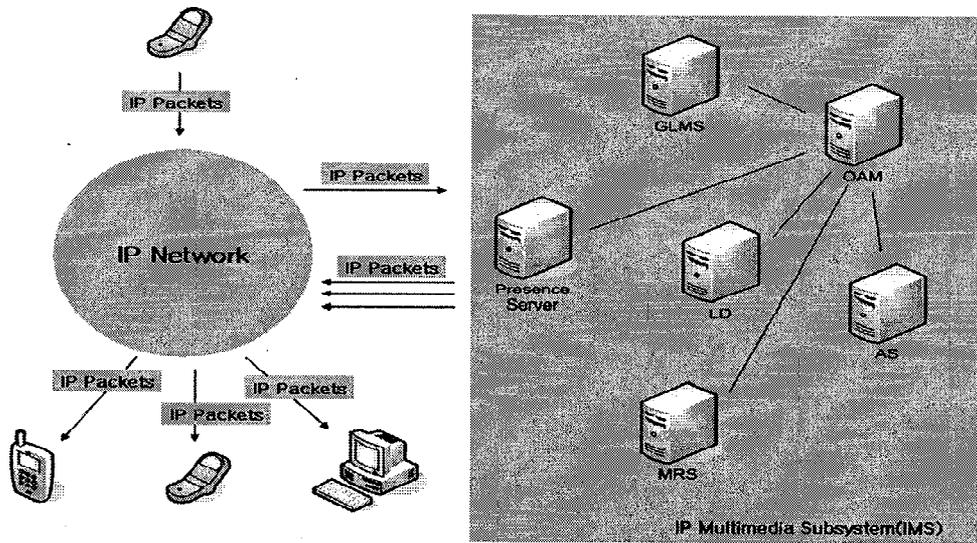
부여, 로케이션, 세션 라우팅, 과금 등의 기능을 수행한다. 둘째, 게이트웨이 계층은 IMS 네트워크를 다른 IMS 네트워크와 PSTN 및 기타 기존 네트워크와 통신사 운영 지원 시스템과 연결해주는 역할을 한다. 셋째, 애플리케이션 계층은 게임, 비디오, 그룹 채팅과 같은 SIP 기반 애플리케이션을 제공하며, 사용자서비스 로직을 포함하고 있다.[7][8] [표 1.1]은 IMS 구성 요소에 관한 설명이며, [그림 1.2]는 IMS 망 구성도이다. [그림 1.3]은 IMS 망에서의 PTT 서비스 모델이다.[9][10][11]

Session Control Layer	- SIP 서비스 지원하며 CSCF와 HSS 기능 포함
Gateway Layer	- IMS 네트워크를 다른 네트워크에 연결
Application Layer	- SIP 기반 어플리케이션 제공

[표 1.1] IMS 구성요소



[그림 1.2] IMS 망 구성도



[그림 1.3] PoC 시스템의 망 구성 및 PoC 동작 원리

1.2 연구 동기

IMS망에서 PTT를 서비스하기 위해서는 IMS망에서 기본으로 제공하는 PoC(Push-to-talk over Cellular)서버를 이용하여 PTT서비스를 제공 할 수 있다. [그림 1.3]은 PoC 시스템의 망 구성 및 PoC 동작 원리에 대한 그림이다. AS(Application Server)는 PTT, PTD(Push-To-Data), IM(Instance Message) 등의 호 처리에 대한 제어를 담당하며 과금을 위한 기초 데이터를 생성한다. PoC 서비스를 제공하는데 있어 가장 중요하고 핵심적인 역할을 수행하므로 보통 PoC서버라고 하면 AS를 의미한다. MRS(Media Relay Server)는 AS와 연동하여 PTT 및 PTD 세션에 대한 생성, 삭제, 변경을 수행한다. 또한 RTCP(Real Time Control Protocol)를 통한 사용자간 발언권 제어, RTP(Real-time Transport Protocol) 미디어 전송 중계, MSRP(Message Session Relay Protocol)를 통한 파일 미디어 전송 중계를 담당한다. LD(Load Distributor)는 SIP(Session Initiation Protocol) 프록시 서버로부터 수신되는 그룹통화 요청, 그룹통화 취소, PTD, IM

요청 등을 AS로 로드를 분산하여 라우팅 하는 역할을 담당한다. GLMS(Group/List Management Server)는 PoC 서비스 가입자의 공용 URL(Public Uniform Resource Location), IMSI(International Mobile Subscriber Identity), MDN(Mobile Directory Number), 제공 가능 서비스 등의 정보와 수신거부 모드 설정, 응답모드, 표시이름 등의 속성을 관리한다. 또한 그룹 목록리스트와 목록에 대한 수신 허용/금지 등을 관리한다. 프레즌스 서버(Presence Server)는 가입자의 프레즌스 정보를 관리하는 서버이다. OAM(Operation, Administration, Maintenance) 서버는 전체 시스템에 대한 시스템 운용 및 관리, 운용자 정합, 구성 관리, 성능관리, 장애관리, 통계 생성 및 출력, NMS(Network Management System)연동, 과금 CDR(Charging Data Record) 수집 및 전송 등의 운용 및 유지 보수 관련 기능의 마스터 기능을 수행한다.[12]

IP 기반인 IMS망에 PTT 서비스하기 위해 고려해야 할 두 가지 사항이 있다. 첫 번째는 현재 상용화 되고 있는 Mobile 망, TRS 망과 동일한 콜 셋업을 제공해야 한다. 현재 국내에서 상용화되고 있는 대표적 PTT 서비스는 주파수 공용 통신 (TRS : Trunked Radio System)방식이다.[13] 2005년 1사분기까지 가입자 30만 이상을 확보하여 상용중인 TRS는 이동통신망이 복합된 형태로 두 개의 망에 접속할 수 있는 겸용 단말기를 사용하여야 한다. PTT 서비스는 Hot-Key 버튼을 이용하여 1:1 또는 1:N 그룹통화가 가능하다. 콜 셋업 시간이 0.5~1초 정도이며 TRS 망을 사용함으로써 이동통신망과의 충돌을 피할 수 있다. 그러나 TRS망을 별도로 운영함으로써 막대한 투자비가 소요된다. IMS망 PTT 서비스는 SIP 프로토콜을 이용하여 통신한다. IMS 망에서 음성을 무선데이터와 같이 무선 패킷 망에 실어 전송하는 방식으로 다수의 그룹 원에게 동시에 음성 패킷을 One-Way로 전송하는데 다른 통신망에 비해 콜 셋업이 느리다는 단점이 있다. 콜 셋업이 느린 단점에 비해 투자비가 저렴하고 PTT 서비스 이외 영상 PTT 서비스, 데이터 전송 서비스, 프레즌스 서비스 등 다양한 서비스를 제공받을 수 있다.[14][15] [표 1.2]는 PTT 구현방식에 대하여 비교하였다.

구분	TRS망	Mobile망	IMS망
무선망	TRS망+이동통신망	이동통신망	IP망
투자	TRS망 투자 및 단말 개발 투자	서버 개발 투자	서버 및 단말개발
투자비용	대	소	소
콜 셋업	0.5 - 1초	이동전화와 동일	5 - 10초
데이터 형태	Circuit 방식	Circuit 방식	패킷 방식

[표 1.2] 각 통신 방식별 비교

두 번째는 IMS망에서의 주파수 자원 사용이 충분해야 한다. 2003년 8월 세계 최초로 PTT 상용화를 한 버라이즌사는 최근 신규 가입자 유치를 잠정적으로 중단하였다. 중단의 가장 큰 원인은 기존 음성통화대비 PTT 주파수 자원의 과다 사용으로 인한 기존 이동전화 서비스에 영향을 미쳤기 때문이다.[15]

IMS 기반 망에서 PTT 서비스를 하기 위기 위해서는 현재 상용화 되고 있는 Mobile 망, TRS 망과 동일한 콜 셋업을 가져야 하며, 콜 셋업 및 통화시의 주파수 자원 사용이 충분해야 한다. 본 논문에서는 콜 셋업 시간의 단축과 통화 성능 향상을 위한 방안으로 프로토콜 압축 방법을 제시하고자 한다.

PTT 서비스는 1:N 의 통화가 순간적으로 이루어 져야 하기 때문에 기존의 1:1 음성 및 영상 서비스 보다 훨씬 더 많은 음성 대역폭을 요구하게 된다. 또한, IMS 망에서 사용하는 SIP 경우 대역폭이 풍부한 네트워크에서의 사용을 전제로 개발되었고, SIP 메시지들은 텍스트 기반이어서 크기가 수백에서 수천 바이트로서 매우 크다. SIP를 대역폭이 풍부하지 못한 3G와 같은 무선 이동 통신 네트워크에서 사용할 경우 유선망에서는 발생하지 않았던 콜 셋업 지연과 대역폭 부족의 문

체들이 나타나고 있다. 콜 셋업 지연과 대역폭 부족의 문제를 해결하기 위하여 SIP, RTP, UDP, IP와 같은 프로토콜 헤더의 오버헤드를 압축하여 대역폭을 절약할 수 있는 헤더 압축 기술이 요구되고 있다. SIP 프로토콜 압축 방법에는 Deflate 메시지 압축 방법을 이용한 SigComp(Signalling Compression)[16] 알고리즘을 이용한다. IP/UDP/RTP 압축방법에는 대표적인으로 Header Removal/Generation, Header Striping/Regeneration, 헤더 압축[17] 등이 있으며, 그 중 헤더 압축 방법이 가장 효율적이다. 헤더 압축 연구는 유선 망 환경에서는 TCP/IP 프로토콜 헤더 압축[18][19], RTP/UDP/IP 헤더 압축[20]이 연구 되었으며, 무선망에서는 ROHC(RObust Header Compression)[21] 헤더 압축 기법이 연구되고 있다. 본 논문에서는 IMS 망 PPT 서비스의 성능 향상을 위해 통화 설정 대기 시간 단축 및 대역폭 확보를 위한 프로토콜 압축 기법을 제안한다.

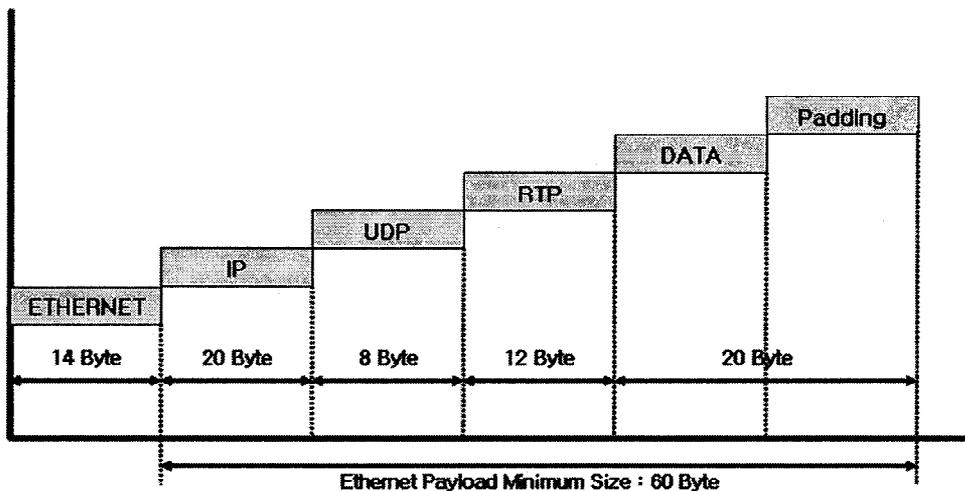
본 논문의 구성은 다음과 같다. 2장 관련연구에서는 프로토콜 압축 기법과 다양한 호 처리 및 통화 성능 향상 기법, 호 처리 지연 및 통화 대역폭 부족 요인에 대해 기술하였다. 3장에서는 IMS망에서 PTT 서비스를 위한 호 설정 및 통화 성능향상을 위한 새로운 프로토콜 압축 기법 방안을 제시하고 4장에서 성능평가를 하였다. 마지막으로 결론 및 향후 과제를 5장에 기술하였다.

제 2장 관련연구

2.1 프로토콜 헤더 압축

2.1.1 기존 프로토콜 문제점

기존의 프로토콜의 헤더 크기는 IP 20바이트, TCP 20바이트, UDP 8바이트, RTP 12바이트이다. [그림 2.1]을 보면 IP/UDP/RTP 헤더의 크기를 제외하면 전송되는 데이터 및 패딩의 크기가 헤더에 비해 작다는 것을 알 수 있다. 이와 같은 헤더들은 고정적인 값을 가지는 필드, 일정한 규칙에 맞게 변하는 필드, 몇몇 필드를 통하여 추론할 수 있는 필드를 포함하고 있다. FTP와 같이 헤더에 비해 데이터가 큰 경우에는 문제가 되지 않지만, 대부분의 실시간 서비스에서와 같이 헤더에 비해 데이터가 작은 서비스의 큰 오버헤드는 네트워크 효율을 저하 시키는 원인이 된다. 이와 같은 데이터에 비하여 헤더가 큰 경우 헤더 압축 기술을 이용하면 많은 오버헤드를 줄일 수 있으며, 네트워크 대역폭 확보에 큰 도움이 된다.



[그림 2.1] RTP 헤더 크기

2.1.2 프로토콜 헤더 압축 기법

헤더 압축 프로토콜은 RFC1144의 Van Jacobson 헤더 압축(VJHC : Van Jacobson Header Compression)을 기본으로 출발하였다.[18] 이 프로토콜은 패킷스트림에 그 개념을 두고 있는데, 동일한 세션 내에 존재하는 헤더 영역의 대부분은 동일한 값을 갖게 된다. 예를 들어, IP나 UDP/TCP/RTP 헤더 내에 존재하는 주소영역과 포트영역은 같은 세션 내에서는 동일하다. 또한, RTP의 Timestamp나 Sequence number같은 영역은 값이 순차적으로 일정한 크기를 가지고 변하며, UDP 길이 영역은 링크계층의 영역과 중복된다. 이러한 패턴분석을 기반으로 IP/UDP/RTP 헤더영역의 중복되는 부분은 제거하고 일정하게 증가하는 영역의 값에 대해서는 변화량만 전송하면 헤더압축이 가능하다는 방법이 현재 RFC 2508로 승인되어 있다.[20]

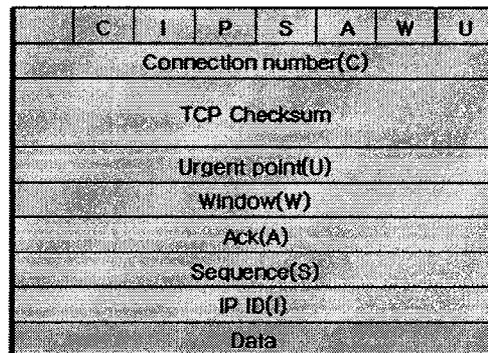
- Compressing TCP/IP Headers for Low-Speed serial Links

헤더 압축의 기본이 되는 Compressing TCP/IP Headers for Low-Speed serial Links 방법은 40 바이트(IP 20바이트 + TCP 20바이트)의 TCP/IP 헤더 중에서 TCP 연결이 성립 후에 수십, 수백 개의 패킷이 교환되는 동안에 많은 헤더 필드들이 일정한 값을 유지하기 때문에 수신되는 헤더 패킷 중 많은 필드의 헤더 정보들이 앞뒤 프레임 사이 중복하게 된다. 이러한 중복된 헤더 필드와 헤더 필드들을 전송할 경우 이전 프레임과 이후 프레임을 비교하여 순차적인 영역의 차이(differential coding)를 보내게 되면 필드 값을 나타내기 위한 바이트를 줄일 수 있는 점 등 두 가지 방법을 이용하여 헤더 압축을 수행한다. 패킷 타입은 Compressed_TCP, Uncompressed_TCP, Non-TCP, TYPE_ERROR 등 4가지 종류의 패킷 타입을 사용한다. [표 2.1]은 각 패킷 타입의 설명을 나타낸다.

패킷 타입	Description
Compressed_TCP	압축된 TCP 헤더
Uncompressed_TCP	압축되지 않은 전체 헤더
Non-TCP	압축 할 수 없는 TCP 프로토콜이 아닌 헤더
TYPE_ERROR	수신측에서 에러 발생 시 사용

[표 2.1] Compressing TCP/IP Headers for Low-Speed serial Links 패킷 타입

[그림 2.2]의 Compression_TCP 헤더 포맷을 살펴보면 처음 8비트는 체인지 마스크로 이전에 전송된 패킷과 현재 보내는 패킷에서 어떤 필드들이 변했는지를 나타낸다. C는 Connection number, I는 Ip id, P는 tcp Push bit, S는 Sequence number, A는 Ack, W는 Window size, U는 Urgent point 필드의 변화를 나타낸다. 이 8비트의 조합으로 256가지의 변화를 나타낸다. 에러 검출은 링크 레벨의 CRC(Cyclic Redundancy Check)를 통해서 검출하며, 싱크 에러는 TCP Checksum을 통해 검출한다.



[그림 2.2] Compression_TCP 헤더 포맷

- IP Header Compression

IP Header Compression은 TCP/UDP 헤더를 4~7 바이트로 줄이는 것이다. 기본적인 개념은 Compressing TCP/IP Headers for Low-Speed serial Links와 같

다. 다른 점은 송/수신 단에서 헤더 정보를 저장하는 context라는 개념을 도입하여, 초기에 모든 필드들의 정보를 갖고 있는 풀 헤더를 전송하고, 수신 단은 풀 헤더를 context에 저장한다. 패킷을 보낼 경우는 Compressing TCP/IP Headers for Low-Speed serial Links와 동일한 방식으로 differential coding을 이용하여 전송하며 수신 단은 받은 패킷을 context를 이용하여 원래의 헤더로 복원하게 된다. 송/수신 단에서 context를 구별하기 위하여 CID(Context Identifier)를 사용하며, Non_TCP 패킷의 압축을 위하여 TCP 프로토콜의 순서 번호와 비슷한 generation을 생성하여 사용한다. 패킷 타입은 [표 2.2]와 같이 5가지 종류를 사용한다.

패킷 타입	Description
FULL_HEADER	압축되지 않은 헤더 패킷으로 초기화 사용
COMPRESSED_NON_TCP	압축된 Non_TCP 패킷으로 differential coding을 이용하지 않는다.
COMPRESSED_TCP	압축된 TCP 패킷으로 differential coding을 이용하여 필드 값을 전달
COMPRESSED_TCP_NODELTA	압축된 TCP 패킷으로 differential coding 대신 받은 값을 그대로 보낸다.
CONTEXT_STATE	Decompressor가 Compressor로 싱크를 잃었을 경우 CID 리스트를 전송하는 경우에 사용한다.

[표 2.2] IP Header Compression 패킷 타입

패킷을 잃어 버렸을 경우 수신 단은 FULL_HEADER를 수신하기 전까지 수신된 패킷을 버리게 되며, FULL_HEADER를 수신하여 context에 저장하고, 헤더 압축을 계속하게 된다. 송신 단은 FULL_HEADER 패킷 분실로부터 빨리 복구하기

위하여 Compression Slow_Start와 수신측의 context손실에서 오는 많은 패킷의 손실을 피하기 위하여 Periodic Header Refreshes를 사용한다. Compression Slow_Start는 송신측에서 풀 헤더 보내는 주기를 지수 적으로 증가시키는 방법이며, Periodic Header Refreshes는 주기적으로 풀 헤더를 보내서 주기적으로 갱신하는 방법이다. 대부분 앞에서 설명한 Compressing TCP/IP Headers for Low-Speed serial Links와 비슷하지만, 필드들의 분류를 더욱 세분화 시켜서 변하지 않는 필드들을 NOCHANGE, differential coding을 할 수 있는 필드들을 DELTA, Random한 값을 가지기 때문에 값을 그대로 전송하는 RANDOM, 다른 필드들의 값을 통해 추론이 가능한 INFERRED로 분류하여 더욱 헤더 압축 효율을 높였다. 그리고 수신 단에서의 에러를 줄이기 위하여 twice 알고리즘을 사용했다.

- Compression IP/UDP/RTP Headers for Low-Speed Serial Links

지금까지는 헤더 압축이 TCP 중심이었다면, 여기에서는 실시간 서비스를 RTP 프로토콜 중심의 헤더 압축이다. IP/UDP/RTP 헤더 압축은 헤더 압축 효율을 높이기 위하여 링크 계층에서 제공되는 IP 전체 길이 필드나 CRC등 중복되는 필드들을 제거하였다. 패킷 타입은 [표 2.3]과 같이 5가지 패킷 타입을 가지고 있다.

패킷 타입	Description
FULL_HEADER	압축되지 않은 IP헤더에 상위 헤더와 데이터를 더한 패킷 타입으로 원형 패킷과 유사하지만 8비트 또는 16비트 CID를 포함한다.
COMPRESSED_NON_TCP	압축된 TCP 프로토콜이 아닌 프로토콜로 generation 필드가 포함
COMPRESSED_UDP	압축된 IP와 UDP헤더에 압축되지 않은 RTP에 데이터를 더한 패킷 타입이며, 이 패킷 타입은 RTP 필드들이 변할 때 사용
COMPRESSED_RTP	RTP/UDP/IP 모두 압축된 패킷 타입
CONTEXT_STATE	송/수신 단에서 Sink를 잃었을 경우에 CID 리스크 전송에 사용

[표 2.3] Compression IP/UDP/RTP Headers for Low-Speed Serial Links 패킷 타입

2.2 호 처리 향상 기법

2.2.1 Always On

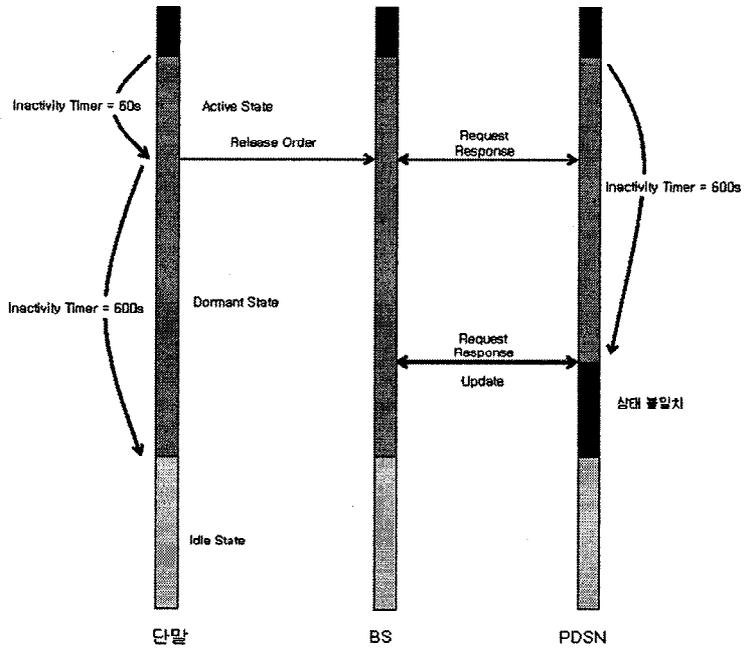
Always On은 단말의 Online 데이터 착신을 위하여 활성화(Active or Dormant) 상태가 영원히 지속되도록 하는 기능이다.[23] 활성화 상태가 영원히 지

속되면 통화 접속 대기 시간을 단축시키는데 효율적이다. 이러한 단말의 상태는 [표 2.4]와 같은 3가지 상태로 구분할 수 있다.

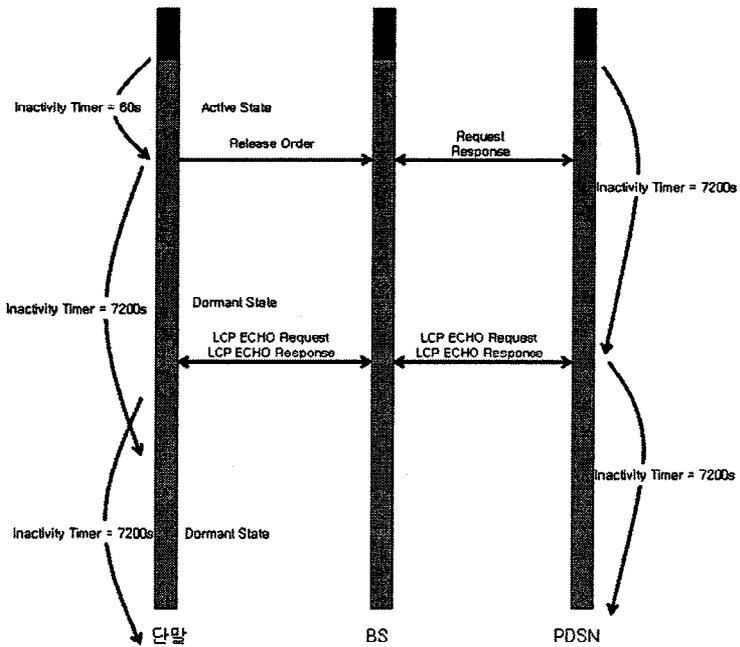
단말상태	IP 주소 할당 여부	트래픽 채널 할당 여부
Idle	×	×
Dormant	○	×
Active	○	○

[표 2.4] 단말의 상태구분

구현방안으로는 두 가지 방법이 제안되었다. 첫 번째 방법은 단말과 패킷 데이터 서비스망(PDSN : Packet Data Switch Network) 사이의 세션 종료 타이머(SCT : Session Closure Timer)를 조정하는 방법이다. 세션 종료 타이머는 단말이 패킷 데이터 서비스망과의 세션을 유지하고 있는 상황에서 무선 환경의 저하 및 단말이 PTT 서비스를 지속 할 수 없을 경우 세션 자원의 효율성을 위해서 일정 시간동안 응답이 없으면 강제로 세션을 종료시키는 역할을 한다. 이때 단말의 Active 상태를 유지하기 위하여 세션 종료 타이머 시점을 무한대로 설정하는 것이다. 두 번째 방법은 세션 종료 타이머는 기존 그대로 유지하며 타이머가 종료되기 전에 keepalive 메시지를 전송하여 세션을 유지하는 방법이 있다.



<1xEV-DO 망의 천이도>



<Always On 상태 천이도>

[그림 2.3] Always On 상태 천이도

[그림 2.3]은 위의 두 가지 방안을 절충하여서 Always On 기능 동작 시 단말과 기지국(BS : Base Station), PDSN의 상태 천이를 나타내었다. 현재 국내의 상용망인 1x EV-DO(EVolution Data Only)에서 사용 중인 세션 종료 타이머는 600초이며 이 시간을 적용 할 시에는 단말과 PDSN의 타이머 불일치로 인하여 상호간의 상태가 불일치 될 수 있다.[24] PTT에서는 이러한 점을 보완하여 세션 종료 타이머를 12배가량 늘린 7200초를 사용하여 첫 번째 제시한 방법 중 하나인 무한대의 효과를 가져 올 수 있다. 또한, 세션 종료 타이머가 종료 될 시에 PDSN에서 SDB(Short Data Burst)[25]기능을 사용한 keepalive(LCP ECHO Request/Response) 메시지를 전송하여 두 번째 제시한 방법인 타이머를 갱신함으로써 세션을 유지 할 수 있다.

2.2.2 Dormant 페이징과 Multiple Access 채널[24]

Dormant 페이징 방법은 단말이 무선 접속 시 액세스 채널을 통하여 통화를 시도 하게 되는데 액세스 채널보다는 트래픽 채널을 사용하는 것으로서 SIP Signaling(INVITE / 200 OK)을 트래픽 채널로 전송하는 방법을 제안한다.

트래픽 채널 사용에 따라 자연히 액세스 채널의 사용은 감소하게 되고 무선 접속 충돌(Access Collision)에 의한 재전송(Re-transmission) 발생은 크게 줄어들 수 있다. 또한 시스템 내부에서 발생하는 PPP Layer의 Encapsulation으로 인한 SDB(Short Data Burst) 사이즈의 증가로 인한 SDB 처리 불가 및 PCF(Packet Control Function)에서 버퍼링되는 메시지, 확인 메시지 지연에 따라 버퍼링되는 메시지의 페이징에 의한 지연현상을 대부분 줄일 수 있게 된다. 이러한 효과로 착신가입자의 입장에서는 세션 참여 시간이 짧아지며 편차도 크게 줄어들게 된다.

Multiple 액세스 채널은 기존에 한 개로 운영되는 액세스 채널수를 여러 개로 늘려주어 다수의 액세스 채널을 사용함으로써 무선 접속 충돌을 방지 할 수 있는

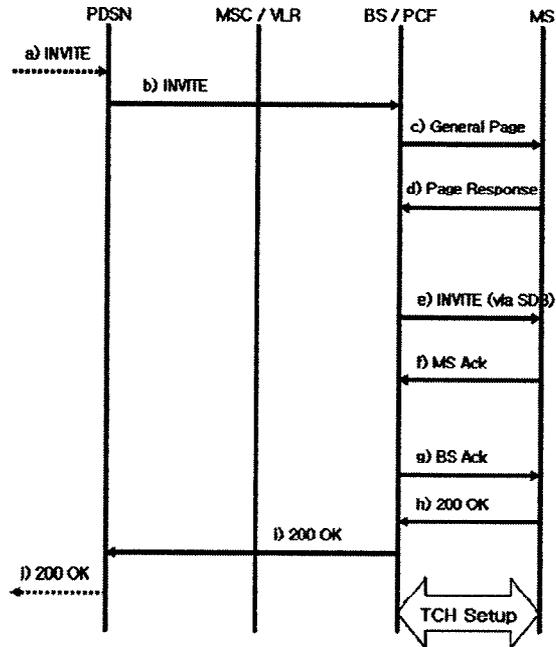
방법이다. Multiple 액세스 채널을 사용한다는 개념 자체는 역방향 링크(Reverse Link)에 간섭(Interference)을 증가시키지는 않는다. 다수의 액세스 채널이 존재하는 경우 단말은 Hashing 함수를 사용하여 특정 액세스 채널을 사용하게 되는데 이 경우 액세스 채널이 수용하게 되는 단말의 수가 분산되는 효과가 발생하므로 역방향 링크 무선 환경에 따른 액세스 충돌에 의해 재전송하는 단말의 액세스 Probe Sequence 횟수가 줄어들어 결과적으로 역방향 링크상의 간섭 발생량이 줄어드는 효과를 가져다준다.

Multiple 액세스 채널을 사용함으로써 기존 서비스에는 미치는 악영향은 없으나 Hardware적으로 한정된 트래픽 채널 자원을 액세스 채널로 용도 변경함에 따라 통화량 증가에 따른 트래픽 채널의 자원 부족 현상이 발생할 수 있다. 따라서 액세스 채널수를 늘리는 것은 해당 기지국의 통화 용량을 검토한 후에 적용을 하는 것이 바람직하다.

2.2.3 페이징 부하 개선[24]

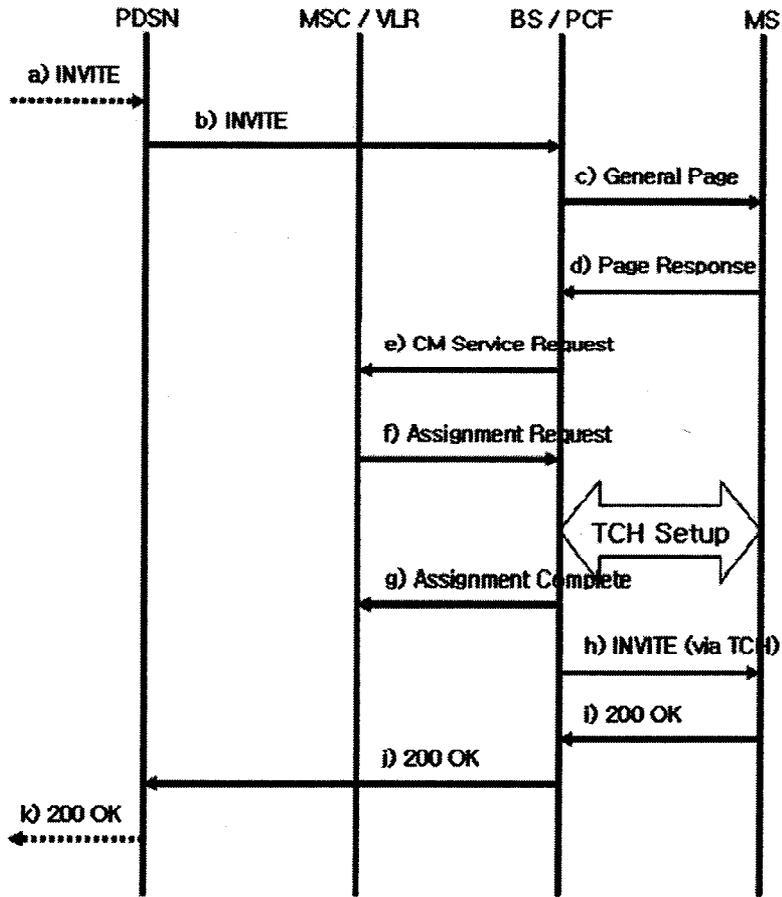
기존 PTT 호 설정 과정에 이용하는 GAN(Global Action Network)단위의 SDB를 전송방법은 호 설정 시간을 최대한 줄이기 위해 고안되었으나 반면에 페이징 부하를 증가시키는 문제가 발생한다. CDMA 2000 1X 망의 경우 GAN 단위의 SDB(INVITE) 페이징이 발생하므로 전체 시스템이 미치는 부하의 영향은 상당하다. 이에 아래와 같은 4가지 방안을 제시하고자 한다.

첫 번째 방안은 GAN 단위의 GPM(General Page Message) 전송 방법으로써 SDB를 이용하여 INVITE 메시지를 전송하는 방법이다. INVITE 메시지를 수신한 착신측은 BS/PCF가 GAN 단위의 GPM을 전송하고 이에 응답하는 특정 Cell/Sector로 INVITE 메시지를 전송하여 Page 채널의 부하를 줄이는 방법이다. [그림 2.4]는 첫 번째 방안에 대하여 표시하였다.



[그림 2.4] GAN 단위의 GPM 전송 방법 - TCH(Traffic CHannel) 이용 INVITE 전송

두 번째 방안은 GAN 단위로 GPM 메시지를 전송하는 방법인데 트래픽 채널을 이용하여 INVITE 메시지를 전송한다. INVITE 메시지를 수신한 착신 측 BS/PCF가 GAN 단위의 GPM을 전송하고 이에 응답하는 특정 Cell/Sector의 MS와 트래픽 채널 설정 후 트래픽 채널을 통해 INVITE 메시지를 전송하여 페이징 채널의 부하를 줄이는 방법이다. [그림 2.5]는 두 번째 방안에 대한 호 처리 절차를 도식화 하였다.



[그림 2.5] GAN 단위의 GPM 전송 방법 - TCH 이용 INVITE 전송

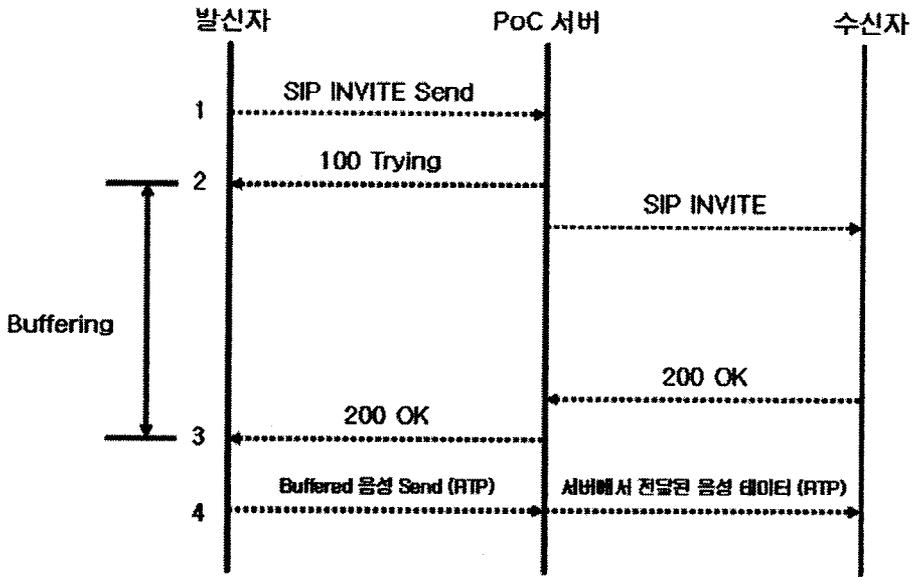
세 번째 방안은 BSC 단위로 GPM 메시지를 전송 방법인데 INVITE 메시지를 수신 한 착신 측 BS/PCF가 이를 MSC/VLR로 전달하고 MSC/VLR의 착신 단말의 위치정보에 따라 특정 BSC 단위의 Paging을 수행 한 후 이에 응답하는 특정 Cell/Sector로 INVITE 메시지를 전송하여 Page 채널의 부하를 줄이는 방법이다.

마지막으로 BSC 단위의 SDB 전송 방법이 있다. 이것은 INVITE 메시지를 수신한 착신 측 BS/PCF가 이를 MSC/VLR로 전달하고 MSC/VLR의 착신 단말의 위치정보에 따라 특정 BSC 단위로 INVITE 메시지를 전송함으로써 페이징 채널의 부하를 줄이는 방법이다.

2.2.4 단말기의 버퍼 추가[26]

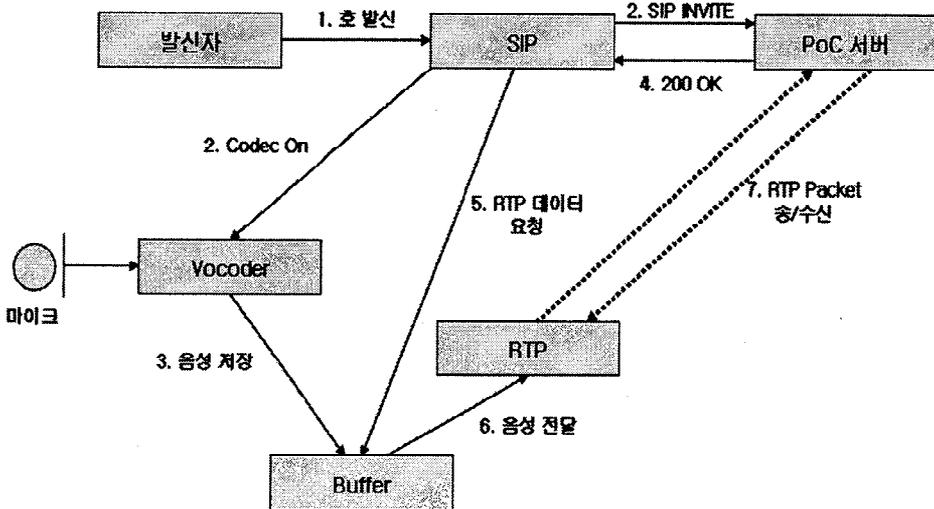
PoC 서버와 단말기 사이의 호 설정 시간 지연의 문제점은 무선 환경이나 호 설정 메시지 문제가 아닌 네트워크 전송 환경에 보다 많이 영향을 받고 있다는 것을 알 수 있다. 따라서 네트워크상에서 데이터 전송 중에 발생하는 호 설정 지연 문제 해결을 단말기 상에 버퍼를 추가하여 호 연결 시간을 단축시키는 방법을 제시한다.

호 설정 초기화 및 종료, Floor Control 절차에서 단말기가 PoC 서버로부터 200 OK를 받은 후 호 설정 되어 Vocoder 측에서 음성 데이터를 PoC 서버를 통해서 상대방에게 전달하게 되는 방법이다. 이러한 기존의 방법을 기지국과 데이터 채널이 연결이 된 후 INVITE 메시지를 보내는 시점에 단말기 내부에서 호 연결 상태로 인식하여 Vocoder 음성 데이터를 임시 버퍼 상에 저장한 후 PoC 서버로부터 200 OK를 받는 순간 RTP 패킷에 단말기 임시 버퍼에 저장된 음성 데이터를 발신하는 방법이다. 이 방법은 이론상으로 단말기 상에서 RTP 한 패킷은 6개의 프레임으로 구성되어 있기 때문에 약 120ms 정도의 음성 데이터 전송 시간을 줄일 수 있다. 그러나 INVITE 메시지를 발신 후 발신 측에서 음성 데이터를 버퍼에 저장하기 때문에 보다 더 많은 시간이 단축되는 것으로 사용자에게 인식된다. [그림 2.6]은 단말기 음성 Buffering 메시지 흐름이다.



[그림 2.6] 단말기 음성 버퍼 메시지 흐름

1번에서 INVITE 메시지를 보낸 시점에서 3번 200 OK를 받은 시점 사이에 Vocoder 단에서 마이크를 통해서 들어온 음성 데이터를 20ms 프레임 내부 음성 버퍼에 저장한다는 원리이다.[27] [그림 2.7]은 단말기 내부 동작 흐름도이다.



[그림 2.7] 단말기 내부 동작 흐름도

단말기 내부 동작은 INVITE 메시지를 발신하는 시점과 동시에 Vocoder에서는 마이크 단에 입력된 음성 데이터를 20ms 프레임 단위로 Encoding 한 후 버퍼로 전달하며 PoC 서버 측에서 200 OK 받은 즉시 이미 단말기 버퍼에 저장된 음성 데이터를 120ms RTP 패킷 데이터에 실어서 PoC 서버로 발신하도록 구현하는 방식이다.[28][29]

2.3 통화 성능 향상 기법

채널의 효율성을 높이기 위하여 입력 음성 신호로부터 음성구간을 구분하고 일정시간이상 묵음이 지속되는 구간에서는 패킷을 전송하지 않고 일정한 시간 간격으로 패킷을 전송하는 방법이 있다. H.323[30] 프로토콜을 이용하여 개발한 OpenPhone[31]의 경우 G.711 코덱을 사용하면 초당 64Kbit를 보내게 된다. 사용자가 말을 하지 않을 경우도 초당 64Kbit를 보내게 되어 불필요한 패킷 낭비 및 통화대역폭의 낭비가 우려된다. 이러한 문제점을 해결하기 위해서 묵음 구간 신호를 효율적으로 코딩하는 기법이 필요하다.

음성 코덱을 사용하는 주된 이유는 음성 신호를 전송 또는 저장하는데 필요한 비트 수를 낮추는데 있으며, 시스템의 특성에 따라 가변 Bit Rate 또는 고정 Bit Rate 음성 코덱을 사용한다. 고정 Bit Rate 코덱에서는 입력 신호의 특성에 상관없이 항상 같은 Bit Rate로 코딩하고, 가변 Bit Rate 코덱에서는 입력 신호의 특성에 따라 다른 Bit Rate로 코딩한다. 멀티미디어 서비스를 위해 채널을 다른 유형의 데이터와 공유할 경우에는 가변 Bit Rate 코덱을 사용하는 것이 효율적이다. 양방향 통화중인 경우에 입력 신호를 분석해 보면 약 60% 이상이 말을 하고 있지 않는 묵음 구간이다. 이 묵음 구간에는 정보가 거의 없으므로 적은 Bit Rate로 표현 가능하며, 이로 인해 가변 전송률 음성 코덱을 사용할 경우에는 거의 음질의 저하 없이 평균 전송률을 낮출 수 있다.

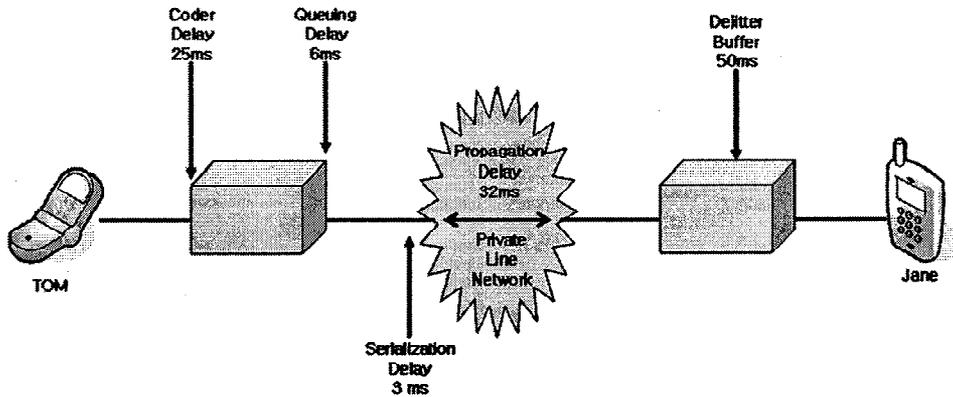
목음 구간 신호를 효율적으로 코딩하기 위해서는 두 가지 알고리즘이 필요하다. 첫 번째는 입력 신호를 음성과 목음 구간으로 분류하는 VAD(Voice Activity Detection) 알고리즘이다. VAD 알고리즘이란 음성과 목음 사이를 구분하는 기능이다. 만일 입력 신호가 목음 구간으로 분류되면 약 1kbit/s의 낮은 Bit Rate로 압축하여 전송하며 DTX(Discontinuous Transmission) 기법을 이용하여 전송을 하지 않는 경우도 있다. DTX 기법이란 이동 전화나 휴대 무선 전화기에 음성 신호가 없을 때 순간적으로 출력을 줄이거나 목음 상태로 만드는 방법이다. 두 사람이 통화할 때 각자의 통화 시간은 각각 전체의 절반 이하에 해당하므로 만약에 송신기에 음성 입력 시간만 연결된다면 송신 시간이 50% 이하가 될 수도 있다. 이는 전지 소모 절약, 송신기 증폭기의 부하 감소, 시분할 다중화(TDM : Time Division Multiplexing)에서 다른 신호와 채널 공유 등의 이점을 살릴 수 있다. DTX 회로는 음성 구동 검파(VAD : Voice Activity Detection)를 이용해 동작하며, 복스(VOX : Voice Operated Transmitting)라고 부른다. 두 번째는 수신 단에서 사람의 귀에 편안한 잡음을 재생하는 CNG(Comfortable Noise Generation) 알고리즘이다. GSM에서는 MS(Mobile Station)에서 전력 소모를 줄이고 전체적인 간섭을 줄이기 위한 목적으로 목음 구간을 검출하여 일정 간격마다 목음 신호에 대한 정보를 전송하기 위해 VAD/DTX/CNG 알고리즘을 사용한다. 또한 평균 Bit Rate를 낮출수록 더 많은 가입자를 수용할 수 있는 CDMA에서는 시스템의 특성상 DTX는 지원하지 않고 목음 구간을 약 1kbit/s로 압축하여 전송한다. 이와 같은 방법은 무선 주파수 자원 절감에도 효과가 있으며 기존 대비 2/3 이상 오버헤더를 줄일 수 있다. [32][33]

2.4 호 처리 지연 및 통화 대역폭 부족 요인

이동간의 타 망과의 서비스가 통합되는 IMS 망에서는 인터넷 망을 이용하여 음성 데이터를 전송하기 때문에 호 설정에 영향을 주는 요인은 크게 네트워크 측

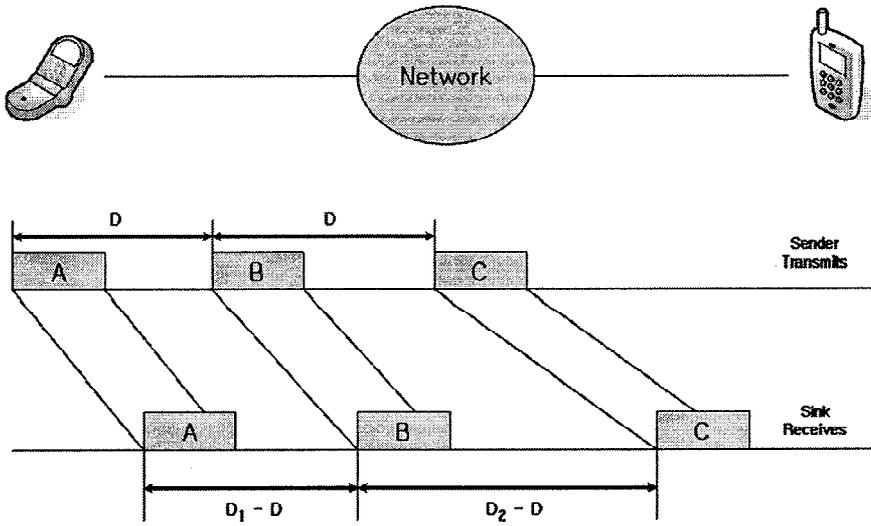
면과 무선 단말기 측면으로 구분된다. 무선 네트워크상에서 음성 품질에 영향을 주는 요인은 지연, 지터, 채널과 같은 문제들이 있다.

지연 Budget이란 네트워크에서 요구되는 품질을 유지하기 위해 허용될 수 있는 지연의 총량을 말한다. ITU-T에서는 VoIP 망에서 “Good Voice Quality” 기준인 150ms 보다도 작아야 한다고 규정하고 있다.[34][35][36] [그림 2.8]은 전체 지연이 $116\text{ms}(=25+6+3+32+50)$ 로 설계된 VoIP 망의 예시를 나타내고 있다.



[그림 2.8] VoIP 망 구성 예

지터란 패킷 도착시간 편차(Variation of Packet Inter-arrival Time)를 말하며, 패킷 수신 예정시간과 실제로 패킷이 도착한 시간 간의 차이를 말한다. 지터 역시 패킷 기반의 네트워크에만 존재하는 문제 중 하나로, 단대단 지연에 영향을 미치는 또 다른 원인이다. 지터가 없는 패킷을 일정한 간격으로 도착시키기 위해서는 지터 버퍼가 필요하며, 지터 버퍼는 패킷 네트워크의 예측할 수 없는 특징을 어느 정도 보완해 주는 역할을 한다. 많은 벤더들은 자신들의 VoIP 제품에 정적 지터 버퍼를 적용하고 있으나, 이 경우 종종 지터 버퍼의 크기가 너무 작거나 반대로 너무 크게 되어 패킷 유실이나 과도한 지연이 발생 할 수 있다. 반대로 일부 벤더의 경우 상황에 따라 버퍼 크기가 변하는 동적 버퍼를 사용한다. [그림 2.9]는 지터의 개념을 설명한 그림이다.



[그림 2.9] 지터 개념

단말기의 RF 성능, 기지국 용량, 전파환경 등과 같은 사항은 H/W적 요소를 제외할 시 PPT 서비스는 호 설정 이후 IP/UDP/RTP로 음성데이터를 전송하게 된다.[37][38][39] 이때 기지국과 데이터 호 연결 시간, 연결 후 RLP/PPP 설정 절차들이 PPT 서비스의 음성 품질에 영향을 주는 요인이다.

단말기의 소프트웨어 측면에서는 호 설정 및 음성 품질 관련해서 두 가지 관련요소가 있다. 기지국에서의 접근과 데이터 세션 설정 시 서비스 연결 후 RLP/PPP 연결 시간이 음성 품질 지연에 영향을 준다. 단말기가 기지국에 트래픽 채널을 획득하기 위해서는 여러 차례 접근 시도하면 그 만큼 연결 시간이 지연된다. 물론 트래픽 채널을 한 번에 확보 할 경우나 패킷 길이가 짧으면 짧을수록 호 설정 시간은 짧아진다. 통화 채널 획득 후 음성 통화를 시도 하게 되는데 동일 서브넷 내에서 다수의 가입자 혹은 동일 세션의 통화가 이루어 질 경우 음성 패킷 특성상 한 수신자에게 초당 20-30개의 패킷을 받게 된다. 이 패킷의 대역폭은 6.3Kbit/s에서 64kbit/s까지 차지하게 된다. 이러한 패킷의 크기를 줄이게 되면 더 많은 트래픽 채널을 다른 사용자게 할당 할 수 있다. 그러므로 두 항목들은 서비

스 사업자 망 환경에서는 최적화 되어야 한다.

제 3장 호 설정 및 통화 성능 향상을 위한 프로토콜 압축 기법

IMS망에서 PPT 서비스를 상용화 할 경우 두 가지 사항을 고려해야 한다. 첫째 현재 상용화 되고 있는 기존 PTT 서비스인 TRS망에서의 동일한 호 설정 시간을 가져야 한다. 둘째 통화 연결이 된 후 통화 품질 또한 기존 서비스와 동일해야 하며 # of User에 대한 충분한 트래픽 채널이 할당 되어야 한다. 기존 연구에서는 호 설정 향상을 위하여 SDB 메시지[25], Always On 기능 활성화[23], 페이지 부하 개선 방안[24], 단말기의 음성 버퍼 추가 방안[26], 채널의 분배 기능[24] 등을 통하여 해결하였다. 통화 성능 향상 부분에서는 음성 코덱을 수정하여 트래픽 채널 할당에 노력을 기울였다. 단말기 자체에 대한 부분 또한 단말기에 음성 버퍼를 추가함으로써 기존 대비 빠른 호 설정을 유도하였다.

본 논문에서는 하드웨어의 문제나 네트워크 트래픽 채널에서의 호 설정 문제를 해결하기 전에 멀티미디어 프로토콜인 SIP에 문제가 있음을 발견하였다. IMS망에서 PTT 서비스를 할 때 호 설정에는 SIP가 사용되며, 호 설정이 완료된 후 통화 시에는 RTP가 사용된다. 두 프로토콜은 멀티미디어 통신을 위해 고안된 프로토콜이며[40] 대역폭이 충분히 확보된 상태에서 텍스트 기반으로 되어 있다. 많은 멀티미디어 통신 프로토콜(SIP, RTP, H.323)은 그 결과 프로토콜 메시지는 충분히 최적화 되어 있지 않는 상태이다. 예를 들어 SIP의 경우를 보면 메시지 길이는 몇 백 바이트에서 2000 바이트 까지 이거나 혹은 그 이상의 길이를 가지고 있다.[40] RTP의 경우는 IP(20)+UDP(8)+RTP(12)+Payload Data(24 : G.723.1)로서 헤더의 길이가 음성 데이터의 길이보다 16바이트 더 길다. 멀티미디어 프로토콜은 대역폭이 풍부하지 못한 2.5G, 3G, IMS와 같은 무선 이동 통신 네트워크에서 사용할 경우 유선망에서 발생하지 않았던 문제들이 나타난다. 무선망에서는 일반 유선망과는 달리 높은 에러 전송률을 나타내는데 헤더 부분의 에러는 치명적인 결과를 낳을 수 있어 효율적인 헤더 처리 기술이 필요하다. 또한 SIP와 같은 큰 메시지로

인하여 무선망에서는 통신하기 위한 멀티미디어 프로토콜로서 최적의 솔루션이 될 수 없다.

본 논문에서는 헤더가 긴 멀티미디어 프로토콜인 SIP를 압축을 통하여 해결하고자 한다. 호 설정 압축 프로토콜은 RFC3320에 명시된 SigComp(Signalling Compression)[16]기법을 향상 시킨 A+SigComp(Advanced Signalling Compression)기법을 이용한다. IP/UDP/RTP의 경우는 RFC3095에 명시된 ROHC(RObust Header Compression)[21]기법을 향상 시킨 A+ROHC(Advanced RObust Header Compression)기법을 이용하여 프로토콜을 압축한다.

3.1 A+SigComp : 호 설정 향상을 위한 프로토콜 압축 기법

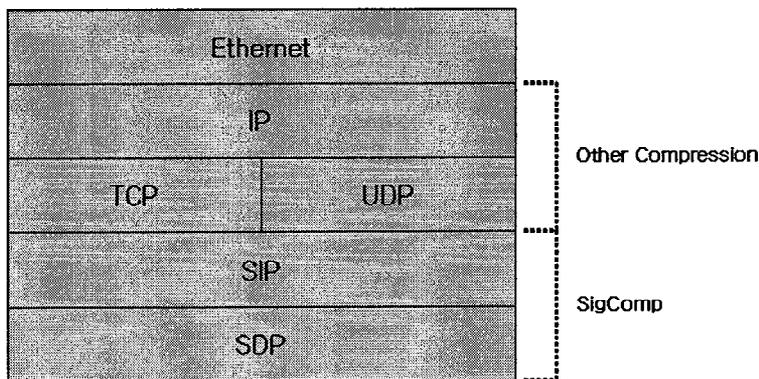
IETF(Internet Engineering Task Force)의 SIP(Session Initiation Protocol)[41]는 HTTP(HyperText Transfer Protocol)와 유사한 텍스트 기반의 메시지를 사용함으로써 이를 구성하는 헤더의 확장이 용이하고 간단하게 세션을 설정하고 수정 및 종료 할 수 있기 때문에 인터넷 응용 분야로의 적용이 가능한 장점을 가지고 있다.[41] 확장이 용이하고 세션 설정이 간단한 장점으로 인하여 다수의 표준화 단체들이 차세대 호 설정 프로토콜로서 SIP를 채택함에 따라 SIP를 기반으로 한 많은 응용 연구가 진행되고 있다.

초고속 무선 네트워크를 통해 멀티미디어를 제작, 전달 및 재생을 논의하는 표준화 단체인 3GPP(3rd Generation Partnership Project)와 동기식 IMT-2000 이동통신 규격을 논의하는 표준화 단체로, 최근 멀티미디어 서비스를 위한 규격들을 제정하고 있는 3GPP2(3rd Generation Partnership Project 2)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) 등의 응용 서비스들이 SIP를 기반으로

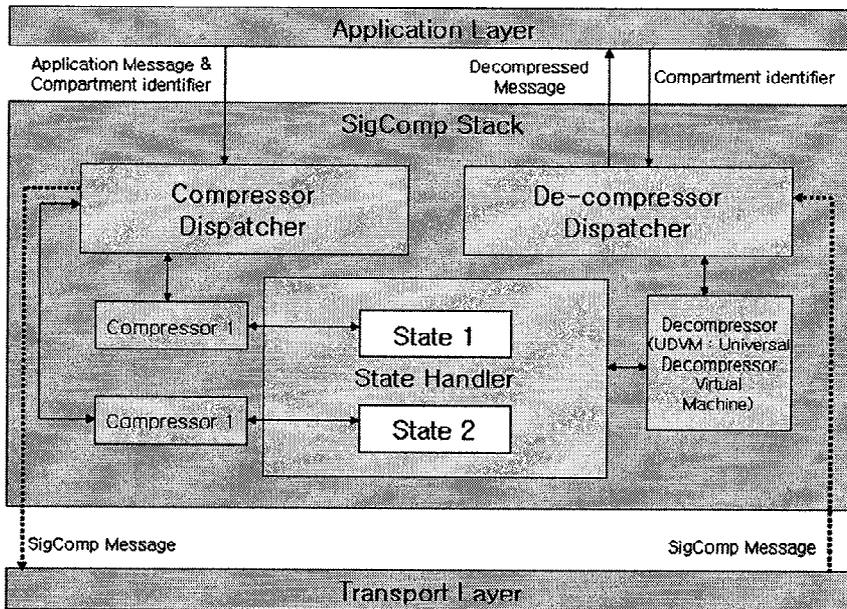
통합되는 구조를 보이고 있다.[41]

SIP는 대역폭이 풍부한 네트워크에서의 사용을 전제로 개발되었고 SIP 메시지는 텍스트 기반이어서 크기가 수백에서 수천 바이트로써 매우 크다. SIP를 대역폭이 풍부하지 못한 3G와 같은 무선 이동 통신 네트워크에서 사용할 경우 유선망에서 발생하지 않았던 문제들이 나타나고 있다. 그 중 가장 큰 문제점은 주파수 사용의 낭비를 들 수 있다. 풍부하지 못한 대역폭에서 텍스트 기반의 큰 메시지는 다른 통신에 지장을 주게 된다. 풍부하지 못한 대역폭의 문제점을 해결하기 위해 RFC3320에 명시된 SigComp(Signalling Compression)을 이용하여 SIP 메시지를 압축하는 방법이 있다.

[그림 3.1]은 SIP 스택을 나타낸다. SIP는 응용 계층에서 SIP, SDP 스택을 포함한 메시지를 TCP/UDP로 전송한다. Sigcomp 압축 알고리즘은 응용 계층(SIP, SDP)을 TCP/UDP에 포장하기 전에 Deflate 압축 알고리즘을 이용하여 압축 한 후 TCP/UDP 계층으로 보내게 되며, IP/TCP/UDP 단에서는 다른 헤더 압축 알고리즘을 이용하여 압축 한 뒤 패킷을 전송하게 된다.



[그림 3.1] SIP 스택



[그림 3.2] Sigcomp Interface 모델

[그림 3.2]는 Sigcomp Interface 모델을 나타낸다. SIP는 응용 계층에서 데이터가 만들어 지고 만들어진 데이터는 전송 계층(TCP or UDP 캡슐화)으로 내려가게 된다. 응용 계층에서 만들어진 SIP 메시지를 전송 계층으로 내려가는 중간 단계에 위치하여 SIP 메시지를 압축 하게 된다. Sigcomp Stack에는 Compressor Dispatcher, De-Compressor Dispatcher, Compressor, State Handler, UDVM(Universal Decompressor Virtual Machine)등으로 구성된다. [표 3.1]은 각 구성요소의 기능을 설명해 놓았다.[16]

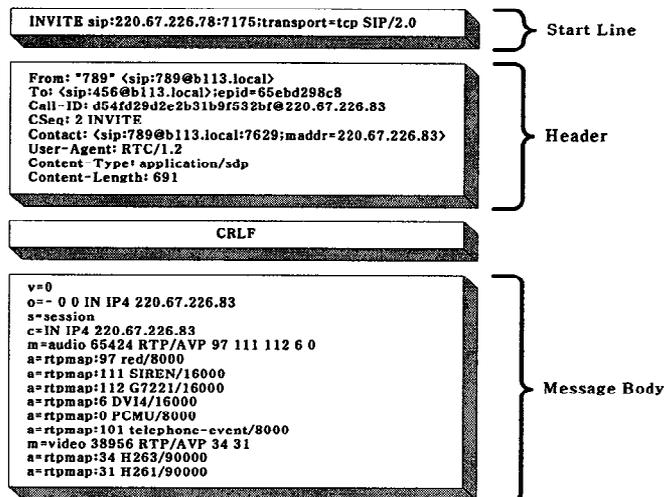
구성요소	기능 설명
Compressor Dispatcher	응용 계층의 인터페이스로써 SIP 메시지와 CID를 받게 된다. 전송 받은 CID를 통하여 해당되는 Compressor로 SIP 메시지를 전송한다. 또한, 압축이 완료된 SIP 메시지를 전송 계층으로 전송한다.
De-Compressor Dispatcher	응용 계층의 인터페이스로써 전송 계층으로부터 Sigcomp 메시지를 받는다. UDVM으로부터 Sigcomp 메시지를 전송하거나 압축이 해제된 SIP 메시지를 응용 계층으로 전송한다. 또한, SIP 메시지 상태를 저장하기를 원하면 응용 계층에서 CID를 할당받은 후 저장하게 된다.
Compressor	한개 이상의 Compressor는 SIP 메시지를 Sigcomp 메시지로 변환한다. 각각의 Compressor는 Application Layer로 내려온 CID를 통하여 해당 Compressor로 이동하게 된다. Compressor는 Compressor dispatcher로부터 받은 SIP 메시지를 받아 압축 한 뒤에 Sigcomp 메시지를 다시 Compressor dispatcher로 돌려준다. 각 Compressor는 데이터를 압축하기 위한 알고리즘을 포함하고 있다.
State Handler	각 상태를 저장하거나 이전의 상태로 돌리는 일을 한다. 각 메시지의 기초 데이터를 업로드 할 필요성을 피하기 위해서 Sigcomp 메시지의 저장된 정보를 가지고 있다. 보완된 상태의 정보라면 응용 계층의 권한에 따라 새로운 상태를 생성해 낸다.
UDVM	Sigcomp 메시지를 압축 해제한다. 압축을 해제 하는 동안 UDVM은 State handler에게 이전에 접근하여 존재하는 상태 인지 새로운 상태를 생성하는지를 상기시켜 준다.

[표 3.1] Sigcomp Stack 구성요소 기능

Sigcomp는 OpenSigComp[42]에서 0.9.3 Version까지 Released 되었으며 압축 알고리즘은 Deflate Algorithm[43]을 사용하였다. Deflate Algorithm은 LZ77 Algorithm를 이용하여 압축하며 중복되는 내용에 대한 포인터를 허프만 부호화를 사용하여 한 번 더 압축한다. Deflate Algorithm은 압축률에 비해 압축과 해제 속도가 빠른 것으로 알려져 있다.[43] SIP는 호 설정을 위한 패킷으로 속도가 빨라야 하므로 압축률 보다는 압축 속도에 비중을 둔 Deflate Algorithm을 이용하였다.

A+Sigcomp 알고리즘은 기존의 Sigcomp의 두 가지 문제점을 해결하기 위해 고안되었다. 첫 번째 문제점은 호 설정 시 SIP 메시지 압축함으로써 생기는 프로세스 처리 시간이다. 두 번째는 기존 Sigcomp Stack에서 ROHC 압축 기법으로 인하여 메시지 처리 속도가 느린 단점이 있다.

Sigcomp Stack은 응용 계층에서 만들어 진 메시지를 전송 계층에 보내기 전에 State에 SIP/SDP 메시지 형태를 저장하게 된다. 저장을 하는 이유는 다음 Compressor와 Decompressor에서 빠른 메시지 복원을 위해서 이다. 따라서 초반에 새로운 메시지 형태를 가진 경우 메시지 형태를 State에 저장해야 하는 과정으로 인하여 처리 속도가 감소된다. 초기 메시지의 State 저장 과정에서 빠른 처리를 위하여 SIP Directory를 추가하여 방법을 해결 하고자 한다.



[그림 3.3] SIP 메시지의 구조

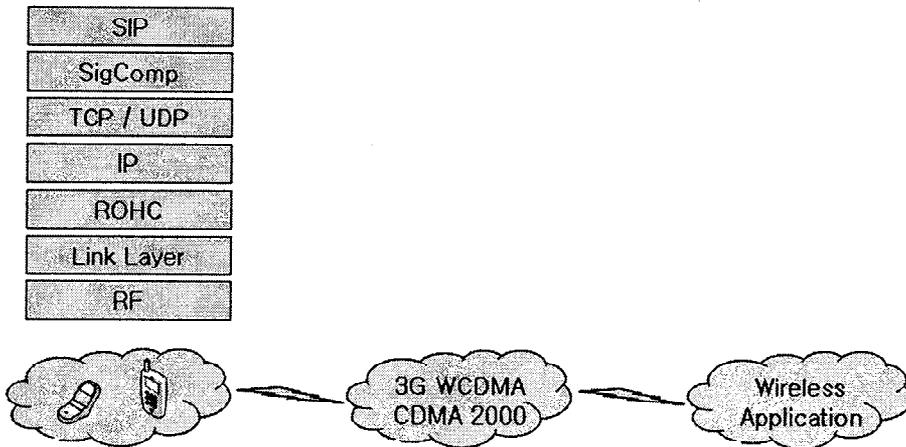
SIP Directory는 RFC 3485[44]에 명시 되어 있으며, 잘 알려진 String을 Static으로 Directory화 한 코드이다. [그림 3.3]은 SIP 메시지 구조를 보여준다. SIP는 Start Line, SIP 헤더, SDP(Session Description Protocol) 헤더로 구성되어 있다. Start Line은 INVITE, 200 OK, TRYING, RINGING등과 SIP 헤더의 From, To, Call-ID등과 SDP 헤더의 session, IP4등은 이미 Static한 String이 존재한다. 그리고 메시지를 송/수신시 자주 사용되는 String이 존재하게 된다. Static한 String과 자주 사용되는 String을 이용하여 SIP Directory를 만들게 된다. 자주 사용되는 String에 대해서는 우선권을 높게 주고 자주 사용되지 않는 String에 대해서는 우선권을 낮게 잡는다. 따라서 메시지 복원 시 SIP Directory를 우선순위로 검색 한 후 offset값과 String 길이를 이용하여 메시지를 받은 후에 복구 String에 포함시켜 빠른 복구를 구현할 수 있다. [그림 3.4]는 SIP Directory[44]의 일부분을 보여준다.

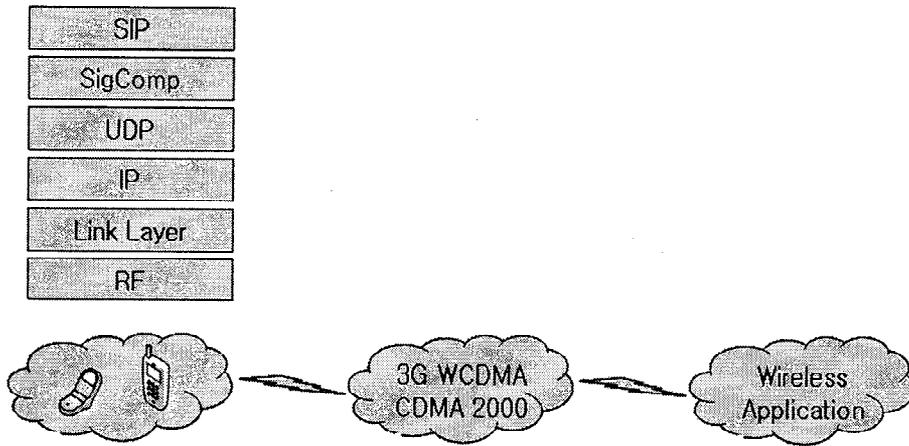
String	Pr	Off	Len
"sip:"	1	0CDD	0004
"sips:"	3	08AC	0005
"tel:"	3	08BD	0004
"SIP/2.0"	1	0CB9	0007
"SIP/2.0/UDP "	1	0CFE	000C
"SIP/2.0/TCP "	2	0CB9	000C
"INVITE"	1	0D4E	0006
"INVITE "	1	0D4E	0007
"ACK"	1	0D4A	0003
"ACK "	1	0D4A	0004
"OPTIONS"	4	0269	0007
"OPTIONS "	4	0269	0008
"BYE"	2	0C8A	0003
"BYE "	2	0C8A	0004
"CANCEL"	4	05E3	0006

"CANCEL "	4	05E3	0007
"REGISTER"	2	0B8F	0008
"REGISTER "	2	0B8F	0009
"INFO"	4	06E9	0004
"INFO "	4	06E9	0005
"SUBSCRIBE"	2	0A6C	0009
"SUBSCRIBE "	2	0A6C	000A
"NOTIFY"	2	0BC6	0006
"NOTIFY "	2	0BC6	0007
"PRACK"	2	0D48	0005
"PRACK "	2	0D48	0006
"UPDATE"	2	0BBF	0006
"UPDATE "	2	0BBF	0007
"REFER"	4	066B	0005

[그림 3.4] SIP Directory

두 번째는 기존 Sigcomp 프로토콜 Stack에서 포함된 ROHC 압축 스택 부분을 제외함으로써 메시지 처리 속도를 빠르게 하는 방법이다 또한, 기존의 SIP 경우는 TCP 통신을 사용하지만 PTT 서비스를 위하여 UDP 스택을 이용하는 방법이다. [그림 3.5]는 기존의 Sigcomp 스택[45]과 확장된 A+SigComp 스택을 보여준다.





[그림 3.5] 기존 Sigcomp와 A+Sigcomp Stack

SIP 메시지 압축 방법은 헤더 압축 방법과는 다른 압축 알고리즘을 이용한 압축 방법이다. IP/UDP/RTP 헤더 압축 방법은 STATIC 필드와 DYNAMIC 필드를 보낸 후 다음 패킷은 이전 패킷과의 차이를 전송하는 방법을 사용하였다. 하지만 SIP 메시지 압축의 경우 String 메시지가기 때문에 String을 압축 할 수 있는 Deflate 알고리즘을 이용하여 메시지를 압축하였다. 따라서 Sigcomp 압축기법과 IP/UDP/RTP 헤더 압축과는 다른 방식을 사용하기 때문에 이중 압축을 이용할 경우 빠른 통화 설정을 위해서는 속도가 느린 단점을 가지고 있다. SIP 메시지의 크기가 크기 때문에 압축 한 결과 50-100 Byte의 크기를 가지고 있기 때문에 IP/UDP 헤더 크기인 28Byte에 비하여 큰 크기를 가지고 있다. IP/UDP 헤더를 압축 할 경우 최대 2Byte까지 압축이 되지만 전체 데이터 크기를 비교 해 볼 때 약 26Byte를 차이를 가지고 있기 때문에 압축된 SIP 메시지 크기에 비해서는 큰 차이를 가지고 있지 않다. 따라서 기존의 IP/UDP 헤더를 ROHC 기법을 이용하지 않고 원본 데이터를 보낼 경우 처리 속도는 더 빨리 질 것이다.

기존의 SIP 경우 TCP 헤더에 메시지를 실어 보내었다. PTT 서비스의 경우는 신뢰성 있는 프로토콜인 TCP를 쓰지 않고 UDP를 사용하여도 IMS망에서 오류

처리를 하기 때문에 UDP 헤더를 이용하여도 상관이 없다. 또한 INVITE 메시지 이후 200 OK 응답 메시지를 받기 때문에 TCP를 쓰지 않아도 오류처리가 가능하다. UDP헤더를 사용함으로써 TCP헤더의 복잡한 헤더 필드를 처리 하지 않아도 되며, 8byte의 헤더 크기를 줄일 수 있다.

3.2 A+ROHC : 통화 성능 향상을 위한 프로토콜 압축 기법

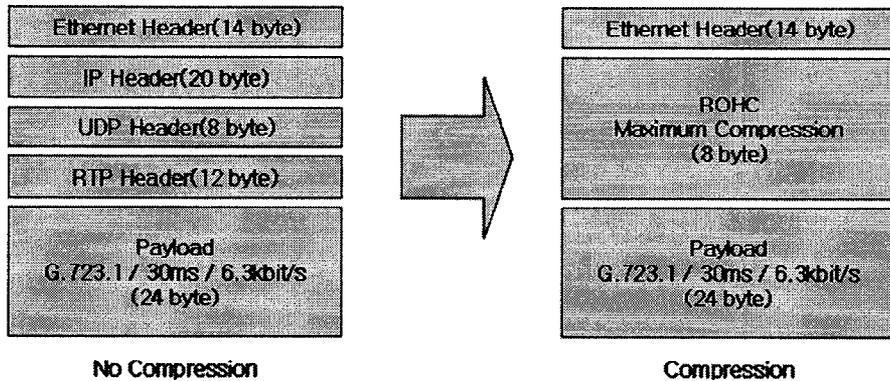
ROHC(RObust Header Compression)는 기존의 헤더 압축 방법들과는 다르게 높은 비트 에러율과 긴 왕복소요시간이 있는 무선망에서 효율적이고 강인한 헤더 압축을 위한 방법이다. 기존의 헤더 압축 방법들은 IP, TCP, UDP 헤더를 STATIC, DYNAMIC, INFERABLE 클래스로 분류하여 압축하는 방식을 사용하였다. 압축한 패킷을 송 / 수신 시 네트워크에 따라 피드백의 유무나 신뢰성, 최적화에 따르지 않고 보내기 때문에 네트워크 대역폭이 풍부한 유선망에 적합한 프로토콜 압축 방법이다. ROHC 경우는 압축 상태와 압축 모드라는 개념을 도입하여 네트워크 상태 및 피드백의 유무에 따라 압축방법이 변하게 된다. 따라서 가장 낮은 압축 상태에서 가장 높은 압축 상태로 변하게 되며 패킷 헤더들의 변화, Decompressor부터의 ACK/NACK, 주기적인 Timeout 등의 요인에 의해 압축 상태가 변하게 된다.

RFC 1889에 의하면 RTP 패킷 크기는 압축되지 않은 상태에서의 헤더크기는 54Byte를 가진다.[27] ROHC 알고리즘을 적용하여 RTP 패킷을 압축 할 경우 최대 22Byte를 가질 수 있다. 아래 수식을 이용할 경우 압축률은 약 60%까지 절약을 할 수 있다. 만일 Payload Data를 더하여 계산할 경우 약 41% 가량의 패킷을 절약 할 수 있다. [그림 3.6]은 압축하지 않은 RTP 패킷 Header와 압축한 RTP 패킷 헤더를 비교해 놓았다.

$$CompressionRate(\%) = \frac{no\ Compression\ Packet\ Size - Compression\ Packet\ Size}{no\ Compression\ Packet\ Size} \times 100$$

$$58.2 \approx \frac{54 - 22}{54} \times 100\%$$

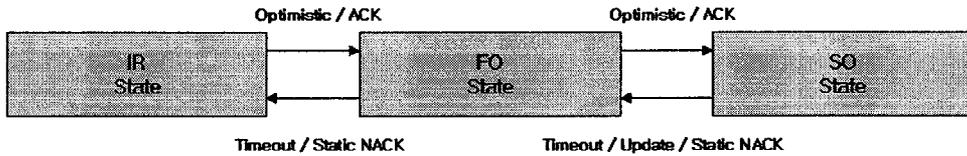
$$41 \approx \frac{(54 + 24) - (22 + 24)}{54 + 24} \times 100\%$$



[그림 3.6] IP/UDP/RTP 패킷 크기 비교

ROHC 알고리즘은 Compressor상에서 IR(Initialization and Refresh), FO(First Order), SO(Second Order)등의 3가지 상태를, Decompressor상에서 NC(No Context), SC(Static Context), FC(Full Context)등의 3가지 상태를 가지고 동작한다.

[그림 3.7]은 Compressor에서의 상태 전이를 나타낸다. Compressor에서는 가장 낮은 압축 상태인 IR에서 시작하여 전체 헤더와 CID를 전송한다. Decompressor에서 ACK를 받을 경우 Compressor는 FO 상태로 전이 한 후에 STATIC Field 및 DYNAMIC Field를 제외한 후 CID와 함께 패킷을 전송한다. Decompressor에서 ACK를 받으면 Compressor에서는 SO 상태로 전이 되며 STATIC Field와 DYNAMIC Field, 추론 가능한 Field를 제외한 CID와 그 외의 Field를 포함하여 전송한다.



[그림 3.7] Compressor에서 상태 전이

• IR State

- 가장 낮은 압축 상태
- Decompressor의 context 초기화 및 실패 후 복구
- Compressor는 완전한 헤더 정보를 보낸다.
- Decompressor가 ACK를 보내기 전까지 IR State 대기

• FO State

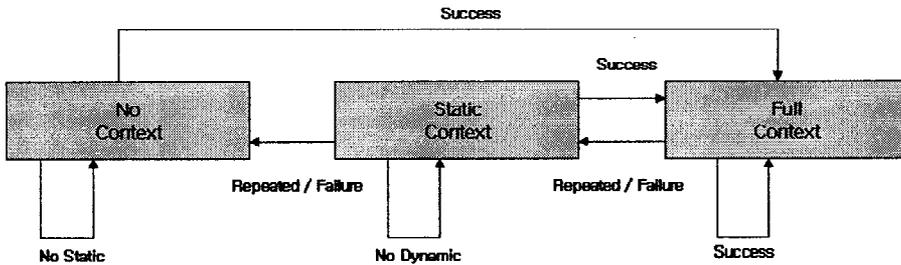
- 패킷 스트림의 불규칙적인 Fields를 능률적으로 전달
- 부분적 context를 만들
- Dynamic Field 정보의 부분적 압축 전송

• SO State

- Compressor의 최적화 상태
- 주어진 SN을 통한 예측으로 완벽한 헤더 압축
- Compressor는 Decompressor가 SN으로부터 다른 Field까지 요구되는 모든 파라미터를 얻을 수 있음

[그림 3.8]은 Decompressor에서의 상태 전이를 나타낸다. Decompressor에서는 전체 헤더를 받은 NC 상태에서 STATIC Fields, CID or STATIC Fields, DYNAMIC Field, CID를 저장한다. 성공적으로 저장 한 뒤에 FC 상태로 전이 한다. 다음 패킷이 CID, irregular Fields를 받을 경우 Success 메시지와 함께 FC 상태를 유지하며, Decompression이 실패 할 경우 NACK 패킷을 Compressor에서 보내며 SC 상태로 변하여 Compressor에게 DYNAMIC Field를 요구하게 된다. Compressor가 CID, DYNAMIC Field를 전송 할 경우 다시

FC 상태로 전이하게 된다.



[그림 3.8] Decompressor에서 상태 전이

• NC State

- Decompressor의 가장 낮은 상태
- Decompressor가 성공할 경우 FC 상태로 전이한다.
- Decompressor가 압축된 헤더를 성공적으로 Decompression 할 수 없음을 의미

• SC State

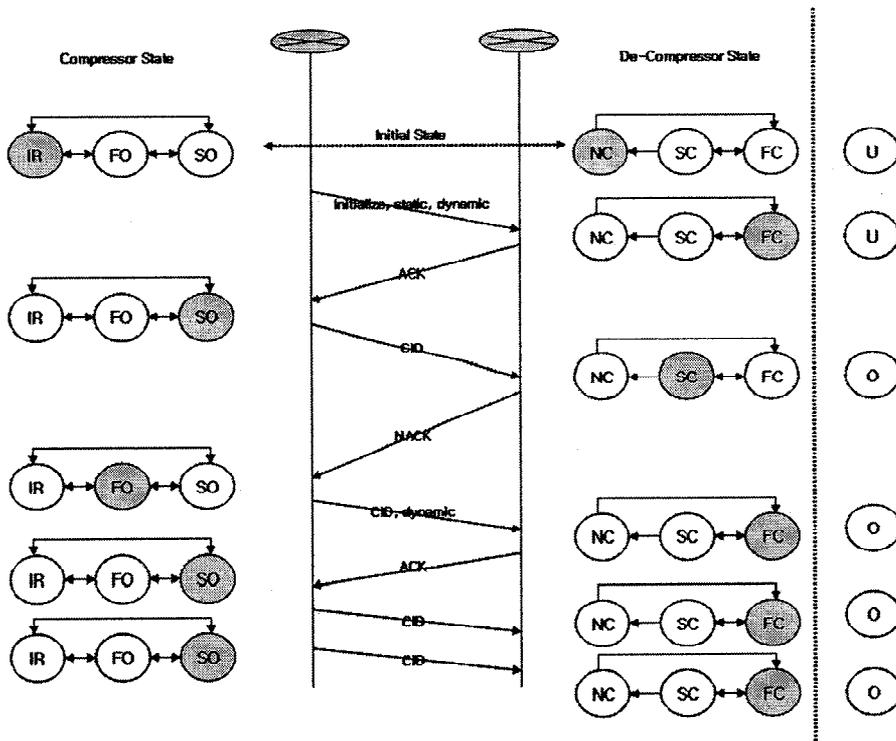
- IR 패킷 수신을 통해 SC 상태로 전이한다.

• FC State

- FO 패킷 수신을 통해 FC 상태로 전이한다.
- 최적화된 헤더 압축 패킷 수신 후 각 상태에 저장된 상태정보를 통하여 정보를 얻게 된다.

Compressor와 Decompressor 사이에는 3가지 상태로 변하게 되며, 작동 할 수 있는 3가지 모드가 있다. U-Mode(Unidirectional Mode)는 패킷이 오직 한 방향 (Compressor에서 Decompressor)으로 전송되며 주기적인 Timeout에 기초를 두고 있다. 시작 에러 복구를 위한 주기적인 Refresh와 Feedback이 없기 때문에 압축은 비효율적이다. ROHC 알고리즘은 U-Mode로 시작한다. O-Mode(Bi-directional

Optimistic Mode)는 U-Mode와 유사하지만 Feedback 채널이 Send error Recovery request와 Decompressor로부터 Compressor로 중요한 context Update ACK로 사용된다는 것이다. 주기적인 Refresh들은 사용되지 않지만 적당한 견고성을 유지하는 동안에 최적의 압축 효율과 적은 수의 Feedback 채널 사용을 위해 설계 되었다. R-Mode(Bi-directional Reliable Mode)는 앞의 두 가지 모드와는 구현과 기능에 관하여 가장 큰 차이점은 Feedback 채널 과다한 사용이다. Compressor와 Decompressor 사이에 모든 context Update에 ACK를 보내게 된다. 따라서 패킷 손실에 대한 부분은 완벽한 견고성으로 context 손실을 최소화 한다. [그림 3.9]는 ROHC 초기화 과정 및 동작절차를 나타낸다. Compressor는 ROHC 알고리즘에 의해 초기에 CID, STATIC, DYNAMIC 헤더 Fields와 초기화 Field를 보낸다. Decompressor는 이 패킷을 수신 한 후 FC State 전이 한 후 ACK 신호를 응답한다. De-Compressor는 자신이 O-Mode로 간다는 것을 표시한다. Compressor는 ACK 수신한 뒤 O-Mode로 전이 한 후 Compressor는 SO State로 전이한다. 다음 패킷과 함께 Compressor는 헤더와 패킷 Payload로 구성된 CID를 보낸다. 그러나 Decompressor는 수신한 패킷 헤더를 Decompresses하지 못하여 패킷을 폐기 한 후 SC State로 전이한다. 전이와 동시에 NACK 신호를 송신 한 후 NACK를 수신한 Compressor는 FO State로 전이한다. Compressor는 다음 패킷에 CID, Dynamic 헤더 Field를 구성하여 보낸다. Decompressor가 Compression된 헤더를 Decompresses하는데 성공하고 FC State로 바꾼 후 ACK로 응답한다. Compressor는 ACK를 받은 후 SO State로 전이하고 다음 패킷(CID, Payload Data)을 전송한다. ROHC는 O-Mode에서 Compressor는 매 패킷마다 각각의 ACK를 전송하지 않고 또한 기다리지 않는다.



[그림 3.9] ROHC 초기화 과정 및 동작절차

IP 패킷의 경우는 총 20byte이다. IP version과 Header Length와 같은 Field는 패킷에 상관없이 고정적 값을 가지고 있으므로 uncompressed_value로 구분한다. Tos, TTL 값은 처음 보내어진 패킷 이후로 변하지 않기 때문에 static으로 구분한다. length, checksum Field 경우 패킷을 순차적으로 받을 경우 추론이 가능하기 때문에 inferred_ipv4_length, inferred_ipv4_header_checksum으로 구분한다. ip_id 경우 예측이 불가능하기 때문에 irregular으로 구분한다. 하지만 ip_id경우는 CID를 포함시켜 전송하기 때문에 꼭 필요한 필드는 아니어서 제외시켜도 된다. draft-brinkmann-rohc-3095 문서에 따라서 아래와 같이 코드로 구분된다.[46]

```

uncompressed_format = version,      %[ 4 ]
                             hdr_length,  %[ 4 ]
                             tos,         %[ 6 ]
                             ecn_flags,   %[ 2 ]
                             length,      %[ 16 ]
                             ip_id,       %[ 16 ]
                             df,          %[ 1 ]
                             mf,          %[ 1 ]
                             rf,          %[ 1 ]
                             offset,      %[ 13 ]
                             ttl,         %[ 8 ]
                             protocol,    %[ 8 ]
checksum,                     %[ 16 ]
src_addr,                      %[ 32 ]
dst_addr;                       %[ 32 ]

```

```

version      ::= uncompressed_value(4, 4);
hdr_length   ::= uncompressed_value(4, 5);
tos          ::= static;
ecn_flags    ::= static;
length       ::= inferred_ipv4_length;
ip_id        ::= irregular(16);
ip_id_rnd    ::= static;
ip_id_nbo    ::= static;
df           ::= static;
mf           ::= uncompressed_value(1, 0);
rf           ::= static;
offset       ::= uncompressed_value(13, 0);
ttl          ::= static;
protocol     ::= static;
              % in RTP/UDP/IP profile always 17 (UDP)
checksum     ::= inferred_ipv4_header_checksum;
src_addr     ::= static;
dst_addr     ::= static;

```

UDP 패킷의 경우는 총 8byte이다. udp_src_port, udp_dst_port는 첫 패킷

이후 변하지 않는 Field 이기 때문에 static으로 구분한다. udp_length Field는 패킷을 순차적으로 받을 경우 추론이 가능하기 때문에 inferred_udp_length로 구분한다. udp_checksum과 같은 불규칙한 Field는 irregular로 구분한다. draft-brinkmann-rohc-3095 문서에 따라서 아래와 같이 코드로 구분된다.

```

uncompressed_format =  udp_src_port,    %[ 16 ]
                        udp_dst_port,    %[ 16 ]
                        udp_length,      %[ 16 ]
                        udp_checksum;    %[ 16 ]

udp_src_port    ::= static;
udp_dst_port    ::= static;
udp_length      ::= inferred_udp_length;
udp_checksum    ::= irregular(16);

```

RTP 패킷의 경우는 총 12byte이다. version, padding 등은 첫 패킷 이후 변하지 않는 Field 이기 때문에 static으로 구분한다. seq_number, timestamp Field는 패킷을 순차적으로 받을 경우 추론이 가능하기 때문에 inferred_rtp_seq_number, inferred_rtp_timestamp 등으로 구분한다. draft-brinkmann-rohc-3095 문서에 따라서 아래와 같이 코드로 구분된다.

```

uncompressed_format = ip_headers,
                        udp_header,      %[ 8 ]
                        version,          %[ 2 ]
                        padding,          %[ 1 ]
                        rtpext,           %[ 1 ]
                        csrc_count,       %[ 4 ]
                        marker,           %[ 1 ]
                        payload,          %[ 7 ]
                        seq_number,       %[ 16 ]

```

```

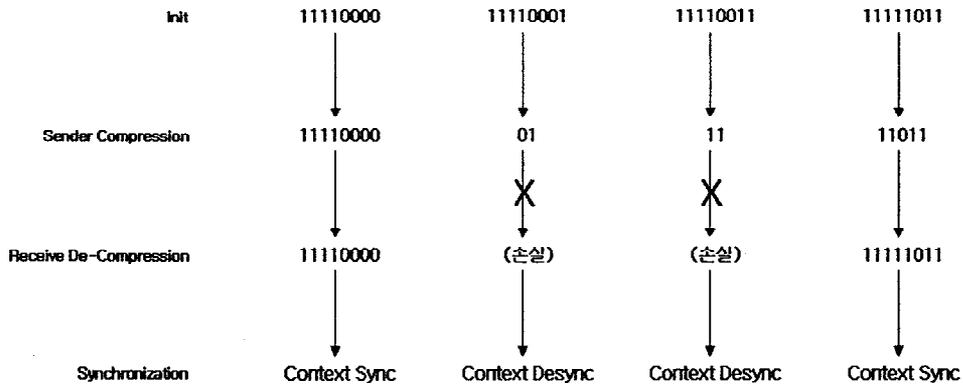
timestamp,          %[ 32 ]
ssrc,               %[ 32 ]
csrc;

ip_headers          ::= ip_headers(id1, rnd1, nbo1,
                                   id2, rnd2, nbo2);

%
udp_header          ::= udp_header(udpcs, udpcs_ctx);
%
version             ::= uncompressed_value(2, 2);
padding             ::= static;
rtptext            ::= static;
csrc_count          ::= static;
marker              ::= static;
payload type        ::= static;
seq_number          ::= inferred_rtp_seq_number;
timestamp           ::= inferred_rtp_timestamp;
ssrc                ::= static;
csrc                ::= static;

```

ROHC 알고리즘은 Dynamic 헤더 Field 압축률을 높이기 위해 W-LSB(Window-Least Significant Bit)를 이용한다. W-LSB Encoding 방법은 여러 개의 연속된 패킷이 손실되는 경우 장점을 찾을 수 있다. 손실된 패킷을 발견 할 경우 context는 즉시로 재동기화 될 수 있다. W-LSB Encoding은 헤더의 LSB k-Bit를 전송한다. k-LSB를 수신 한 후 Decompressor는 헤더 Field의 실질적 값을 재구축 가능하다. [그림 3.10]은 W-LSB를 응용한 예이다.



[그림 3.10] W-LSB 동적 Header Field 동작과정

멀티미디어 전송에서 가장 중요한 프로토콜은 RTP이다. 특히 헤더에 포함된 Sequence Number, Timestamp는 음성 데이터의 순서와 첫 번째 옥테트가 샘플링 된 순간부터의 순차적 증가를 가진 Field로써 실시간 데이터에서 중요한 부분에 속한다. PTT 서비스 경우 실시간 음성 데이터 전송으로써 패킷이 손실되거나 오류가 발생할 경우 ROHC 동작과정에서 압축상태가 낮은 상태로 이동하게 된다. 이동된 상태에서 현재 STATIC, DYNAMIC Field를 전송해야 하기 때문에 음성이 끊기는 현상이 발생되게 된다. 이와 같이 실시간성 데이터 전송 시 패킷의 손실이나 오류 등을 방지하기 위하여 고안된 A+ROHC(Advanced ROHC)기법을 제안한다.[47]

Sequence Number는 RTP 패킷이 송/수신 될 때마다 1씩 증가한다. 수신측은 이 Field를 이용하여 패킷 손실을 감지하고 패킷 순서를 재 저장한다. 16bit를 가진 Sequence Number의 압축 과정은 Modulo-7 연산을 통하여 정수부분과 나머지부분의 bit를 W-LSB Encoding을 이용하여 보내게 된다.

$$Sequence\ Number = SEQR \times 7 + SEQ7$$

(SEQR : 7로 나눈 몫, SEQ7 : 7로 나눈 나머지)

위와 같은 수식을 통하여 간략화 될 수 있으며 SEQ7은 압축헤더의 첫 3Bit에 표시되어 전송된다. 이때 3Bit를 패킷 담아 보내기 전에 1개의 reordering을 처리하기 위해 사용하는 코드의 순서번호는 [-1, 5]범위를 다루고 있다. 이러한 기존의 방법을 확장시킨 방법으로는 SEQR, SEQ7부분을 고려하여 다중 패킷 손실에 대한 강인성을 제공할 수 있다.

$$S(n') = S(n' - 1) - SEQR_{LSB}(n' - 1) \times 7 - S(n' - 1) \% 7 + SEQ_{LSB}(n') \times 7 + S(n') \% 7$$

SEQR_{LSB} : SEQR의 LSB

S(n) : 현재 순서번호

S(n-1) : 이전 순서번호

위와 같이 식을 이용할 경우 확장된 Header의 순서번호의 변화는 [-3, (7×2×SEQR_NB) - 4]를 가진다. SEQR_NB는 De-Compressor에 전송된 SEQR Bit 수이다. Wrap-around에서 패킷 손실이 발생하면 순서번호의 Decoding 된 값이 잘못될 확률이 증가하는데 이러한 현상을 피하기 위해서 동적 패킷이나 압축 패킷의 확장을 사용해서 순서번호를 코드화 하였다. Wrap-around란 16Bit 크기의 순서번호가 가질 수 있는 최대값인 65535에서 다시 0으로 넘어갈 때 SEQR_{LSB}의 값이 잘못 인식되는 것을 의미한다. 이와 같은 확장된 SN로 인하여 다중 패킷에 대한 손실을 이전의 압축 방법보다 더 효율적으로 줄일 수 있다.

Timestamp는 RTP 패킷의 첫 번째 옥테트가 샘플링 된 순간을 나타낸다. 그 샘플링 시점은 일정하게 증가하는 Clock으로부터 생성된다. 이것은 실시간 데이터의 동기화와 지터 계산에 이용된다. 초기 값은 Sequence Number와 동일하게 무작위 수로 설정된다. 16Bit를 가진 Field인

Timestamp는 TS의 변화는 고정적이다. G.711 Codec, 64 Kbit/s, 30ms 경우 TS의 간격은 $64 \text{ Kbit/s} / 8 \times 0.03 = 240\text{Byte}$ 라는 값이 나온다. 즉 G.711 Codec의 경우 TS변화폭은 고정적으로 240Byte씩 증가하게 된다. 이와 같은 Timestamp Field값은 SN와 Payload Data Size에 의해서 충분히 유추될 수 있다. 안정된 SN를 통하여 손실을 줄인 RTP 패킷에서 SIP Module에서는 SN에 맞추어 Payload Data를 디코딩 한 후에 스피커를 통하여 음성 데이터를 흘려보내게 된다. TS는 고정적인 값을 가지고 있기 때문에 기존의 5-LSB Bit 전송되던 Field를 생략할 수 있다. 단, 묵음처리를 할 경우 SN는 +1 간격으로 늘어나는 반면에 TS 간격은 \times TS Interval 배수로써 늘어가기 때문에 이에 대한 부분을 고려하여 1 Byte의 Field가 필요 하게 된다. 1 Bit는 묵음처리로 인한 갑자기 늘어난 TS의 유무를 판별하기 위한 Flag이며 7 Bit는 TS Interval 배수 Field이다. 따라서 첫 Bit가 1일 경우 TS는 다음과 같은 공식에 의해 계산된다.

$$TS = TSQ \times TSInterval + TSR$$

TS : Timestamp, TSQ

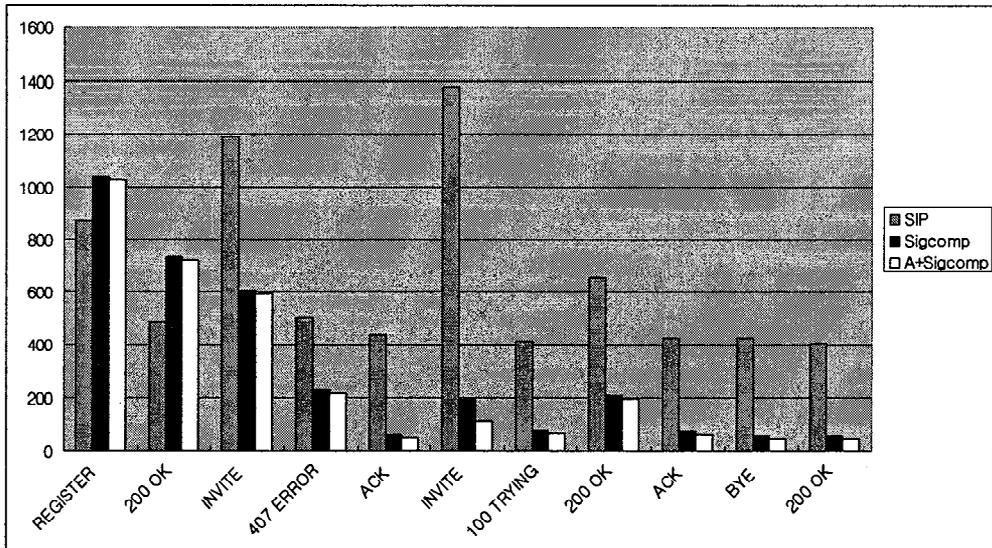
TSQ : Timestamp Quotient

TSR : Timestamp Residual

제 4장 성능 평가

4.1 호 처리 향상 성능 평가

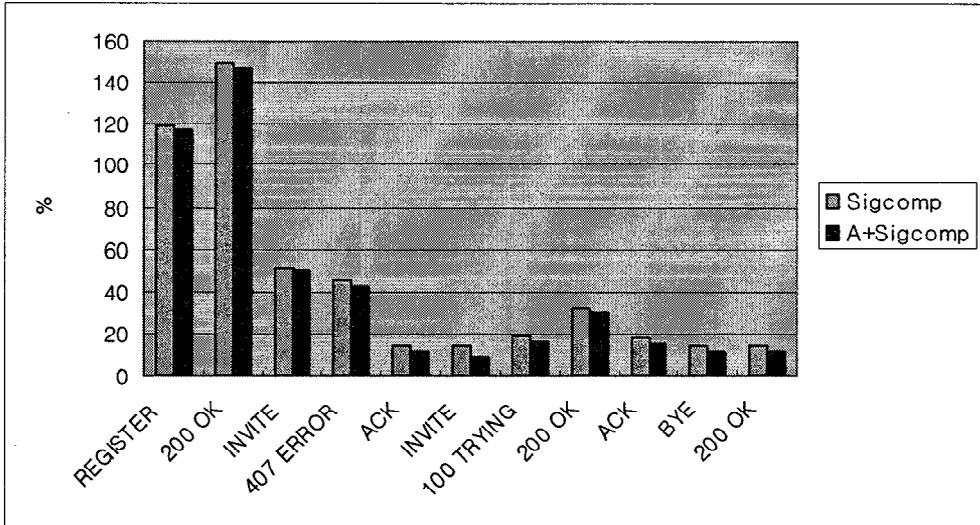
SIP 메시지 압축 크기를 확인하기 위하여 Open Sigcomp에서 오픈된 소스를 이용하여 압축전의 SIP 메시지, Sigcomp 기법을 이용하여 압축된 메시지, A+Sigcomp 기법을 이용하여 압축된 메시지 순으로 [그림 4.1]에 비교 해 보았다.



[그림 4.1] 패킷 크기 비교

최초 서버에 접근하여 등록 할 시의 메시지는 REGISTER, 200OK이다. 이 경우 SIP 메시지 크기에 비해 Sigcomp, A+Sigcomp 기법을 이용하여 압축한 메시지가 더 크을 알 수 있다. 이 경우는 최초 메시지 전송 시 Compressor와 Decompressor 사이의 현재 메시지 상태를 미리 state에 저장 해 둬으로써 다음 압축률을 높이기 위해서이다. 그 이후 INVITE 메시지에서 압축률이 증가함을 볼 수 있다. SIP 원본 데이터에 대해 Sigcomp와 A+Sigcomp 패킷 오버헤드를 아래의 수식으로 표현

할 수 있다. [그림 4.2]는 아래 수식을 이용하여 Sigcomp와 A+Sigcomp 패킷 오버 헤드를 비교 해 보았다.

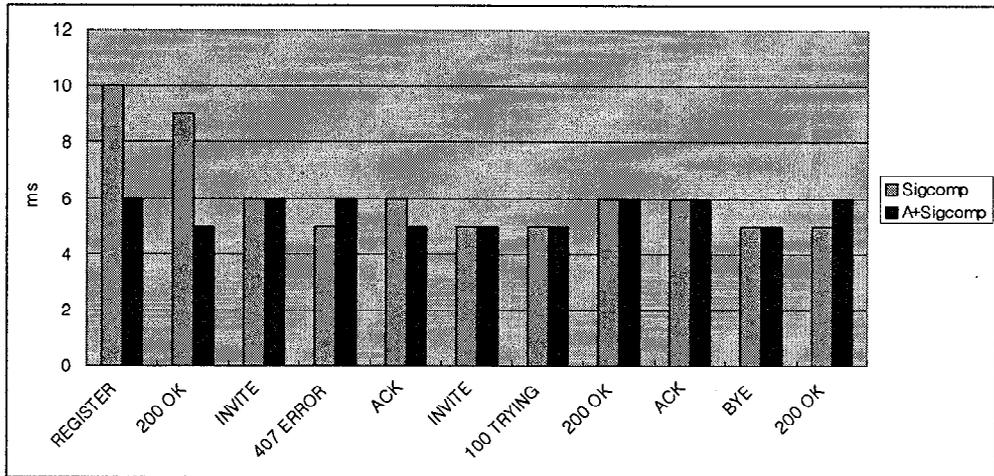


[그림 4.2] 패킷 오버 헤드 비교

$$Packet\ Overhead\ Rate = \frac{Compression\ Data\ Size}{Original\ Data\ Size} \times 100$$

동일한 형태를 가진 메시지를 계속적으로 보낼수록 압축률이 높아짐을 알 수 있다. state에 저장된 내용을 검색하여 압축 및 복원 과정을 거치기 때문에 압축률이 높아짐을 알 수 있다. 위의 패킷 크기를 비교 해 본 결과 A+Sigcomp 기법을 이용한 메시지가 가장 압축률이 좋다는 것을 알 수 있다.

이번 실험에서는 메시지 압축 시 소요되는 프로세스 시간을 알아보았다. SIP 메시지 경우 압축하지 않은 상태에서 메시지를 보내기 때문에 프로세스 시간을 0으로 측정하였다. 압축 시간 측정은 한 PC에서 SIP 메시지를 압축 한 후에 복원까지의 시간을 측정하였다. [그림 4.3]은 Sigcomp와 A+Sigcomp의 메시지 압축 처리 시간을 비교하였다.

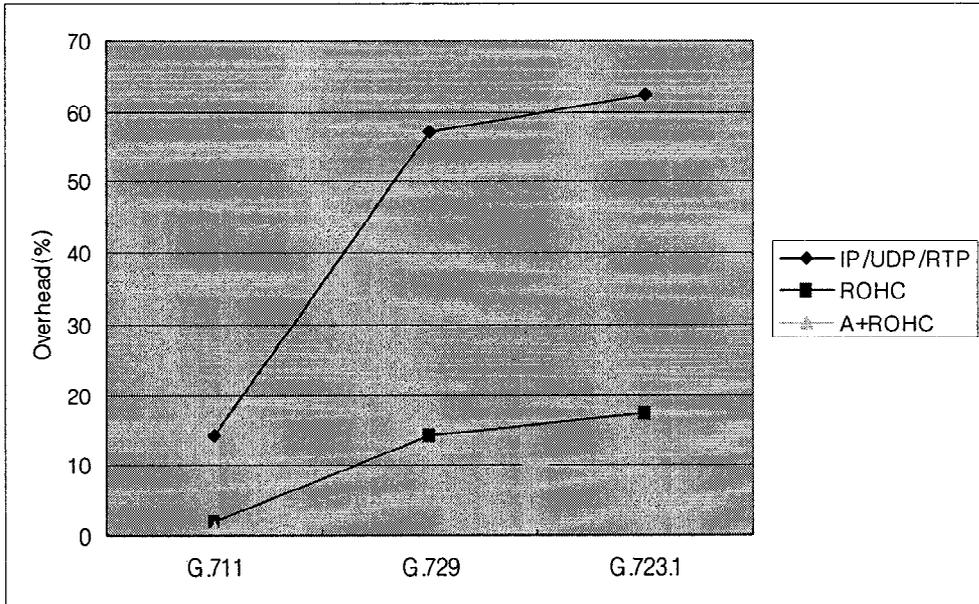


[그림 4.3] 메시지 압축 처리 시간 비교

REGISTER, 200OK 메시지 경우 [그림 4.2]에서 보면 압축률이 압축하지 않은 SIP 메시지 보다 크다는 것을 알 수 있었다. 약 10, 9ms 정도의 처리시간을 보인 Sigcomp 기법과는 달리 A+Sigcomp 기법의 경우는 SIP Directory를 사용함으로써 기존의 Sigcomp 기법보다 약 4ms 이상이 단축된 것을 볼 수 있다. 초기 state 저장 과정 이후는 비슷한 처리 시간을 가지는 것을 볼 수 있다.

4.2 통화 성능 향상 성능 평가

IP/UDP/RTP 패킷 오버 헤더 측정시 ROHC 기법에서 3가지 상태를 고려해야 한다. IR 상태는 전체 헤더를 전송하며, FO 상태는 부분적 동적 헤더를 받은 상태이다. 가장 최적 압축률 상태인 SO 상태는 1byte CID와 함께 이전 패킷과의 현재 패킷간의 차이를 k-LSB 인코딩을 통하여 헤더에 실어 보내게 된다. 따라서 ROHC 기법과 A+ROHC 기법에서 대상이 되는 압축 헤더 데이터 크기는 SO 상태의 데이터 크기가 된다.



[그림 4.4] SO 상태에서 오버 헤드 비교

IP/UDP/RTP 패킷 데이터 크기는 IP(20byte), UDP(8byte), RTP(12byte) 헤더 합인 40byte이다. SO 상태에서 ROHC 기법의 경우 IP(20bit), UDP(4bit), RTP(8bit) 헤더 합인 5byte이다. A+ROHC 기법의 경우는 IP(4bit), UDP(4bit), RTP(4bit) 헤더 합인 3byte이다. 압축 기법에는 CID의 1byte 크기의 헤더가 가장 앞에 붙게 된다. [그림 4.4]는 IP/UDP/RTP, ROHC, A+ROHC 패킷 헤더의 크기와 코덱별 Payload Data값을 아래의 수식을 이용하여 패킷 오버헤드 값을 퍼센트로 표시하였다.

$$Packet\ Overhead\ Rate = \frac{Compression\ Data\ Size}{Original\ Data\ Size} \times 100$$

모든 Payload Data의 경우 전송속도를 30ms로 제한하였으며 G.711의 경우 64kbit/s로 240byte, G.729의 경우 8kbit/s로 30byte, G.723.1의 경우 6.3kbit/s로 24byte의 Payload Data를 가진다. 음성 데이터가 클수록 패킷 오버헤드 값이 감소

하는 것을 볼 수 있다.

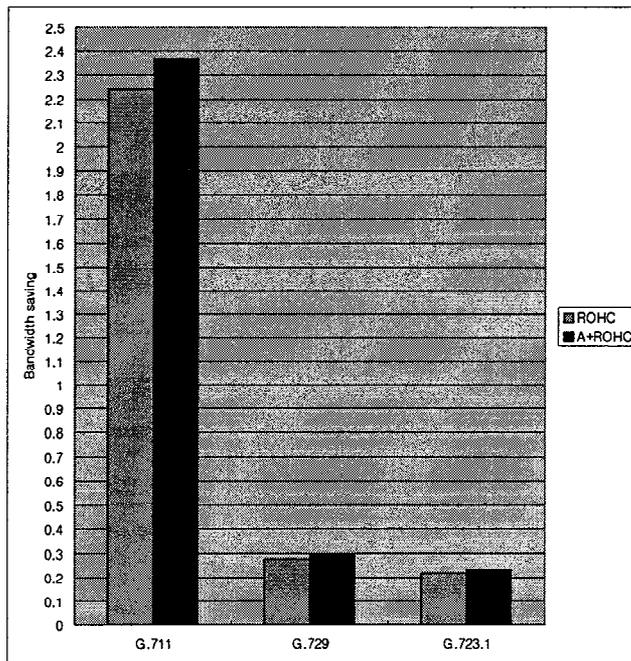
패킷 오버헤더는 대역폭의 효율 상에서도 같은 효과를 나타낸다. 오직 헤더 전송에만 필요한 대역폭이 헤더 압축이 없을 때 전체 서비스에 필요한 대역폭에서 차지하는 비율로 대역폭의 점유율을 표시할 수 있다. 아래 수식을 이용하여 절약할 수 있는 대역폭의 값을 구할 수 있다.

$$\begin{aligned}
 \text{Bandwidth}_{\text{saving}} &= \text{Bandwidth}_{\text{wc}} - \text{Bandwidth}_{\text{c}} \\
 &= (\text{HeaderSize} + \text{PayloadDataSize}) \times F_{\text{sf}} \\
 &\quad - (\text{CompressionHeaderSize} + \text{PayloadDataSize}) \times F_{\text{sf}} \\
 &= (\text{HeaderSize} - \text{CompressionHeaderSize}) \times F_{\text{sf}}
 \end{aligned}$$

$\text{Bandwidth}_{\text{wc}}$: 헤더 압축을 하지 않았을 때 전체 서비스에 필요한 대역폭

$\text{Bandwidth}_{\text{c}}$: 헤더 압축을 했을 때 전체 서비스에 필요한 대역폭

F_{sf} : 소스 프레임 율



[그림 4.5] 절약할 수 있는 대역폭 비교

절약할 수 있는 대역폭 수식을 이용하여 ROHC, A+ROHC 기법을 비교하였다. 음성 데이터가 클수록 절약되는 대역폭이 많음을 알 수 있다.

제 5장 결론 및 향후 연구

본 논문에서는 IMS 망에서 PTT 서비스 구현 시 호 설정 시간의 지연과 통화 성능의 향상을 위하여 프로토콜 압축 방법을 제안하였다. PTT 서비스는 1:N의 통화가 순간적으로 이루어 져야 하기 때문에 기존의 1:1 음성 및 영상 서비스 보다 많은 대역폭을 요구하게 된다. IP 기반 IMS 망에서 PTT 서비스의 호 설정 프로토콜인 SIP의 경우 전송 헤더에 비해 메시지 길이가 길다는 단점을 가지고 있다. 음성 통화 프로토콜인 IP/UDP/RTP의 경우 헤더의 길이가 음성 데이터보다 크다는 단점을 가지고 있다. 높은 에러율과 긴 왕복소요시간을 갖고 있는 무선망에서는 큰 패킷은 부적합하다고 판단하였다. 따라서 호 설정 프로토콜인 SIP 메시지를 A+Sigcomp 기법을 이용하여 메시지를 압축하였고 음성 통화 프로토콜은 A+ROHC 기법을 이용하여 헤더 크기를 줄임으로써 전송 시의 효율성을 높였다.

Sigcomp 기법을 이용할 경우 SIP 메시지 압축 시 초기에 state를 저장하는 과정에서 처리시간이 많이 지연되므로 SIP Directory 방법을 이용한 A+Sigcomp 기법을 이용하여 초기 state 저장 시간을 단축시켰다.

음성 통화 프로토콜인 IP/UDP/RTP의 경우 ROHC 기법을 적용함으로써 헤더의 크기를 줄였다. 하지만 ROHC 기법의 경우 패킷 오류 및 손실이 발생할 경우 Full 헤더를 전송해야 하는 단점이 있다. 패킷 오류나 손실이 발생할 경우를 최소화하기 위해서 A+ROHC 기법을 제한하였다. A+ROHC 경우 Sequence number와 Timestamp의 확장된 LSB 방식을 이용함으로써 비트에러 발생 시 패킷 폐기가 최소화하였다.

본 논문에 이은 향후 연구로는 제안된 기법을 적용하여 IMS망 PTT 서비스를 구현하는 것이다. 또한, 호 설정 및 통화 내용에 대해서는 보완 기술을 적용하여 안정된 서비스를 제공하는 것이다.

참고문헌

- [1] 이홍제, "통신서비스 시장 현황 및 전망", 정보통신정책, 2003
- [2] TR 23.981, "IP Multimedia Subsystem (IMS); Stage 2", March 2005
- [3] NOKIA, "Push to Talk over Cellular-stay connected"
- [4] ATLAS Research Group, "법인시장을 겨냥한 Verizon Wireless의 Push-To-Talk 시장진입", 2003.09
- [5] 김도경, "해외 이동통신업계의 PTT 도입 현황 및 시사점", 정보통신정책 제16권 6호 통권 344호, 2004.04
- [6] Knowledge Research Group, "IMS 동향", March 2006
- [7] ERICSSON, "IMS-IP Multimedia Subsystem", October 2004
- [8] Lucent Technologies, "IP Multimedia Subsystem(IMS) Service Architecture"
- [9] Gonzalo camarillo, Miguel A.Garcia-Martin, "The 3G IP Multimedia Subsystem(IMS)"
- [10] 위성근, "IMS기반 차세대통신망에서의 이동품질 서비스에 관한 연구", 2005. 06
- [11] 류승관, "IMS 기반 이동통신망에서의 IPv4/IPv6 연동에 관한 연구", 2005.12
- [12] 김규연, "PoC(Push-To-Talk over Cellular) 시스템의 설계와 구현
- [13] 심갑진, "TRS망 구성 및 서비스 계획", 1996
- [14] SKTelecom, "해외 사업자들의 PTT 서비스 현황 및 전망", 2004
- [15] 정성구, "VoIP 기반의 CDMA PTT(Push to Talk) 서비스 구현방안, 2003.05
- [16] R. Price Siemens/Roke Manor, C. Bormann TZI/Uni Bremen, "Signaling Compression (SigComp)", RFC3320, January 2003
- [17] TSG-SA-WG2, TSG-GERAN and TSG-RAN, "Header compression for optimized voice bearers", Joint Meeting, 3GPP, Aug. 2001

- [18] Van Jacobson, "Compressing TCP/IP Headers for Low-Speed Serial Links", RFC 1144, February 1990
- [19] M.Degermark, B.Nordgern and S.Pink, "IP Header Compression", RFC 2507, February 1990
- [20] Stephen Casner, Van Jacobson, "Compressing RTP/UDP/IP Headers for Low-Speed Serial Links", RFC 2508, February 1999
- [21] C.Borman et. al, "Robust Header Compression(ROHC)", RFC 3095, July 2001
- [22] L-E. Jonsson, Ericsson, "RObust Header Compression (ROHC): Terminology and Channel Mapping Examples", RFC 3759, April 2004
- [23] KTF, "Always On Standard", 2004.04
- [24] 정의명, "이동통신망 기반 Push-to-Talk 서비스의 호 설정 성능 향상에 관한 연구", 2005.06.
- [25] KTF, "SDB Standard", 2004.04
- [26] 류봉석, "CDMA 1x망에서 PoC 서비스의 Call Setup Delay 개선 방법에 관한 연구", 2004
- [27] Audio-Video Transport Working Group:1889 H. Schulzrinne, GMD Fokus:S. Casner, Precept Software, Inc.:R. Frederick, Xerox Palo Alto Research Center:V. Jacobson, Lawrence Berkeley National Laboratory, "RTP: A Transport Protocol for Real-Time Applications",RFC:1889, Jan. 1996
- [28] Motorola, "M-PTT Protocol-Developer's Guide", Motorola Inc., Mar.2004
- [29] Verizon Wireless, "Push to Talk (PTT) System Call Flows Version 1.1", Verizon Wireless, Feb. 2004
- [30] ITU-T H.323, "Packet-based Multimedia Communication Systems", September 1999

- [31] OpenH323 Project, <http://www.openh323.org>
- [32] 차세대 VoIP 음성 코덱 위원회, “차세대 VoIP 음성 코덱 규격(안)”, 2003
- [33] 이미숙, 김도영, 김대웅, “음성 코덱의 특성과 협대역 음성 코덱 표준 기술 분석”, 2005
- [34] 천재윤, “VoIP(Voice over Internet Protocol)기반에서 음성품질확보 방안에 관한 연구”, 연세공학대학원, Nov. 2002
- [35] Cisco System Korea, “Voice over IP 백서”, Cisco Systems, 2000
- [36] LG CNS 연구센터, “VoIP설계 및 구현 능력 확보”, LG CNS, Jun. 2002
- [37] 김현옥, 김연구, 이성범, 이명성, “IMT-2000 이동통신 원리 - CDMA 2000 중심으로”, 진한도서, 2001.05
- [38] TIA/EIA/IS-707A, “Data service option standard for spread spectrum systems”, 1999.05
- [39] 김충남, “차세대 무선인터넷 서비스”, 전자신문사, 2003.01
- [40] J. Rosenberg, “SIP : Session Initiation Protocol”, RFC 3261, June 2002
- [41] IETF SIP Working Group,
[Http://www.ietf.org/html.charters/sip-charter.html](http://www.ietf.org/html.charters/sip-charter.html)
- [42] OpenSigComp project, <http://www.opensigcomp.org/>
- [43] Deutch, P. , “DEFLATE Compressed Data Format Specification version 1.3”, RFC 1951, May 1996
- [44] M. Garcia-Martin, Ericsson, C. Bormann, “The Session Initiation Protocol (SIP) and Session Description Protocol(SDP) Static Dictionary for Signaling Compression (SigComp)”, RFC 3485, February 2003
- [45] 트라이콤텍, “<http://www.tricomtek.com>”
- [46] E. Brinkmann, C. Bormann, “RTP-ROHC in ROHC Formal Notation”,

draft-brinkmann-rohc-3095, October 18, 2004

- [47] 김경신, 강문식, “차세대 IP 체계에서 효율적인 데이터 전송을 위한 확장된 ROHC 알고리즘”, 2005.10

ABSTRACT

A Protocol Compression Method for Improving Call Setup and Call Bandwidth of Push-To-Talk Service over IMS

Kim, Soo-Hee
Major in Computer Engineering
Dept. of Computer Engineering
Graduate School
Hansung University

In the world most of mobile communication Co. were suggested formation management that can unify wire/wireless, voice/movie/data, and communication/broadcasting/subnet in 3GPP by the possibility that is transmission of fast multimedia data through wireless packet network. it seems that communication companies will be able to develop more easily and cost-efficiently various multi media service and highly new service by using IMS network. one of these new service is PTT depended on IMS network.

PTT is service through communication network developing internet environment to unify wire/wireless for offering voice and communication using text for one to one or group. PTT can communicate immediately with speaking pushing switch just meaning of word, so it can offer relatively fast service when compared with waiting time of general call.

In the IMS network for servicing PTT have call setup same as TRS network and Mobile network that is compatibilization now and should be enough call setup and using frequency resource of call. When

servicing PTT in IMS network, call setup use SIP and after completing it, call setup use RTP. SIP and RTP is protocol that is made for multimedia communication and in case SIP have big message over 1000 byte, in case RTP we can know that size of header is bigger than size of voice data. In this paper, suggested ***A+SigComp***, ***A+ROHC*** that is algorithm, protocol compression method, for call setup time and call bandwidth that is improve ability of PPT service and appraised ***A+SigComp***, ***A+ROHC*** ability by experiment