

MGCP 기반의 게이트웨이 시스템에서 다양한 MGC와의 호환성을 위한 기법

정희원 강재경*, 김현규**, 오은록**, 강태익**, 김철주***

A study on AX-Gateway System Design and Construction for Interfacing between MGCP and H.323

Jae-kyung Kang*, Hyeon-gyu Kim*, Eun-rog Oh*, Tae-ik Kang*, Chul-ju Kim**

Regular Members

요약

MGCP는 IETF에서 제안한 VoIP기술의 표준 프로토콜로서 다른 프로토콜과의 연동이 용이하도록 설계되었으며, 현재 다양한 이기종 시스템과 프로토콜로 구성되어 있는 망을 통합하기 위한 적절한 해결책으로 부각되고 있다. 본 논문에서는 MGCP 호 제어를 담당하는 외부 시스템(MGC)으로부터 VoIP 게이트웨이를 제어하기 위해 사용된다. 또한 MGCP기반의 다양한 게이트웨이 시스템에서 다양한 MGC에 대해 호환성을 향상시키기 위한 방법에 대해 소개한다. 특히 메시지의 흐름(call flow)이나 인코딩 방식과 같은 각 MGC에 종속적인 부분에 있어서 호환성을 지원하기 위해 고려되어야 할 사항과 이를 해결하기 위한 방법에 대해 소개하고 성능 평가를 하였다.

ABSTRACT

MGCP is a standard protocol by IETF, which is designed to facilitate inter-working with other protocols. Since the nature of protocol, it can be known as a proper solution to integrate heterogeneous networks consisted of various protocols and systems. In this paper MGCP is used for controlling VoIP gateways from external call control elements(MGCs) and introduces a method to improve interoperability for various MGC in a gateway based on MGCP. It presents implementation issues to provide the interoperability for MGC dependent parts such as call flows or message encoding, and a system architecture to resolve the issues.

I. 서론

최근 인터넷을 이용하여 전화를 할 수 있는 인터넷 전화 서비스가 확산되면서 기존의 장거리 전화와 국제 전화 이용자의 사용패턴에 급격한 변화를 가져왔다. 1995년 2월 VocalTec사가 인터넷 전화 서비스를 시도한 이후 다양한 부가 기능과 저렴한 요금을 기반으로 VoIP(Voice over IP)서비스의 이용자는 급증하였다. 전 세계적인 인터넷 전화 사용자는 1995년 말 약 50만명에 불과하였으나 2000년

말에는 1500만명을 넘어서었다.^[1]

이처럼 급증하는 요구에 부응하여 ITU-T 및 IETF/OMTC 등에서 VoIP 기술의 표준화에 대한 연구가 이루어졌다. ITU-T SG16은 VoIP에 대한 최초의 표준안으로서 H.323^[2,3,4] 표준을 제안하였다. H.323 표준은 QoS가 보장되지 않는 LAN상에서 실시간 음성, 데이터 및 비디오를 전송하기 위한 프로토콜로 정의되어 있으며, 시장 선점으로 인해 현재 VoIP 프로토콜로서 가장 대중적인 위치를 차지하고 있다. 그리고 IETF-MMUSIC WG에서는 Web

* LG전자 CTO 부문 Network Systems & Solutions Team Solution 기술 Gr.(hankang@lge.com)

** LG전자 디지털시스템사업부 미디어통신실,

*** 서울시립대학교 전자전기컴퓨터공학부

논문번호 : 010215-0821, 접수일자 : 2001년 8월 21일

과 HTTP의 개념과 유사한 SIP(Session Initiation Protocol)[5]를 제정하였으며, 이것은 Mobile 개념을 지원함으로써 무선통신 분야에 보다 적절히 이용될 수 있다는 특징을 장점으로 하고 있다. 또한 IETF에서는 VoIP의 표준으로 MGCP(Media Gateway Control Protocol)^[6,7,8]를 제안하였으며, 이것은 특유의 Simplicity를 통해 다른 프로토콜과의 연동을 쉽게 할 수 있다는 점을 장점으로 부각시켰다. 현재 MGCP는 ITU-T와 IETF 공동 작업을 통해 MEGACO/H.248[9] 표준으로 재정의되고 있다.

현재 구현된 AX-Gateway는 Radvision사의 Stack을 이용하여 개발되었으며, Radvision사의 MGC,를 이용한 자체 시험과 클라렌트(Clarent)사의 MGC와의 연동시험을 통해 기존의 전화시스템이나 휴대폰 등과의 통화가 가능한 상태이다. 연동시험에 성공한 기종은 아날로그 8 포트 형상의 AX-1000M이며, 그 이외에도 AX-400M, AX-2100M 등 시장의 수요에 따른 다양한 용량의 라인 업을 구축하기 위해 제품의 특성을 연구, 개발하고 있다.

초기에 구현한 AX-Gateway 시스템은 H.323을 바탕으로 설계되었으며, 새로운 시장의 요구에 부응하여 MGCP를 새로운 VoIP 관리 프로토콜로 채택하게 되었다. 그 결과 기존의 H.323을 기반으로 작성된 모든 관리 모듈과의 인터페이스를 원활히 하기 위해 MGCP 모듈의 인터페이스를 H.323과 연동이 가능하도록 구현하게 되었다.

AX-Gateway가 H.323 프로토콜과의 연동을 위한 구현을 포함한 것은 기존의 AX-Gateway 시스템의 모든 소프트웨어 관리 모듈이 H.323 기반으로 구성되어 있었기 때문이다. AX-Gateway에 MGCP를 손쉽게 포팅하기 위해 기존 시스템의 모듈을 그대로 이용하고자 했으며, 기존 모듈과의 연동을 위해서는 내부적으로 H.323 방식의 연동이 필요했던 것으로 AX-Gateway가 외부의 H.323 게이트웨이에 대한 인터페이스도 지원할 수 있도록 구현되었다. 그러나 이러한 방법은 구현상 가능하지만 MGCP의 본연의 목적에 위배되는 점이 있으므로 H.323 방식의 연동은 단지 내부적인 목적만으로 제한함을 밝혀둔다.

본 논문에서는 내부의 H.323 방식의 연동과 이를 위해 유지되는 state machine을 포함하는 것은 물론이고 AX-Gateway가 지닌 구조와 타 시스템과의 연동을 위해 필요한 사항 및 구현 방법 등에 대해 소개하기 위해 다음과 같은 형태로 구성되었다. II장에서는 IETF에서 제안한 MGCP 프로토콜의 구성, 명령어 및 메시지 흐름 등의 사항에 대해 소개

하고, 본 논문에서 제안하는 방식과의 차이점을 설명한다. III장에서는 Radvision사의 MGCP 프로토콜 스택에 대한 내용과 Endpoint Manager를 통해 지원되는 추상화 등에 대해 살펴본다. 그리고 IV장에서는 AX-Gateway 시스템에서 MGCP 모듈을 구성하고 있는 MGC와 MG의 데이터 정의, 초기화 과정 등을 설명하고 다른 MGC와의 연동을 위한 설계 방법, 특히 각 MGC에 종속적인 메시지 흐름 및 인코딩 방식 등에 대해 설명한다. 마지막으로 V장에서는 MGCP를 적용한 AX-Gateway 시스템과 클라렌트사와의 연동 결과를 나타내고 자체 성능 평가 시험 결과를 나타낸다. 또한 현재 연구 진행사항과 보완사항, 그리고 추후 연구사항에 대해 언급하고 결론을 제시한다.

II. MGCP

MGCP는 네트워크 상의 각각의 게이트웨이를 외부에서 제어하기 위한 프로토콜로서 1999년 IETF에서 정의하였다. MGCP는 회선 교환망의 신호를 패킷 교환망의 신호로 전환하는 MG(Media Gateway)와 각각의 MG를 제어하기 위한 MGC(Media Gateway Controller)로 구성되어 있으며 MGC는 다른 서비스와의 호환성 역시 제공할 수 있도록 정의되어 있다 [그림1].

MGCP는 H.323과 달리 MG나 Endpoint에 대해 단지 미디어 스트림을 처리하기 위한 역할만을 담당하도록 되어 있으며 이들간의 호 제어를 위한 Intelligence는 MGC(Media Gateway Controller)에서 관리하도록 정의하고 있다. 따라서 MGC는 MG를 관리하기 위한 외부 서버에 해당하며, 자신이 관리해야 할 MG 및 Endpoint에 대한 행동을 지시하고 관리할 수 있다.

MGCP 기반의 시스템은 그림 1과 같이 구성되어 있다. 각 구성요소별로 그 기능을 살펴보면 다음과 같다. Call Agent는 MGC로서 자신에게 연결된 미디어 게이트웨이(MG)를 관리하기 위한 서버에 해당한다. 내부적으로는 각 게이트웨이의 상태정보를 보관하고 있으며 이를 바탕으로 MGCP 메시지를 이용하여 게이트웨이 시스템의 동작을 지시한다.

Call Agent는 주로 소프트웨어 모듈의 형태로 구현된다. MG는 음성통화를 위한 지원을 가지고 있으며 Call Agent로부터 통제를 받는다. MGCP 메시지는 Call Agent에서 MG를 관리하고 동작을 지시하기 위한 명령어를 의미한다. RQNT, NTFY,

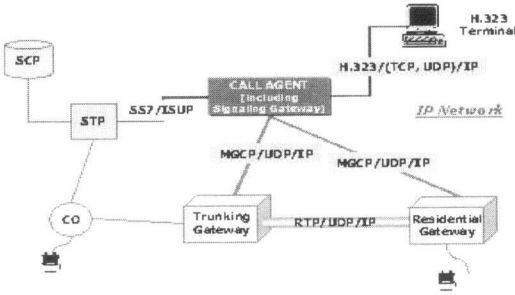


그림 1. MGCP Architecture

CRCX, MDCX 등과 같은 명령어들이 사용된다.

MGCP 기반의 시스템에서는 Call Agent와 MG 간의 관리를 위한 프로토콜로 MGCP를 사용하지만 음성통화를 프로토콜은 RTP(Real Time Protocol)를 사용한다. 따라서 음성통화가 이루어질 때는 RTP에 기반한 stream이 전달된다. 즉, MGCP는 MGC와 MG간의 호 제어 신호(call control signaling)를 위한 명령어만을 정의하고 있으며, MG간에 실제 음성통신을 위해 이루어지는 오디오 스트림에 대해서는 RTP와 SDP를 부가적으로 이용하고 있다. 또한 MGCP는 MG와 Endpoint간에 이루어지는 Event와 Signaling의 처리에 대해서도 정의하지 않고 있다. 다음 그림은 MGCP 프로토콜의 구조를 도식화하여 나타낸 것이다[그림2].

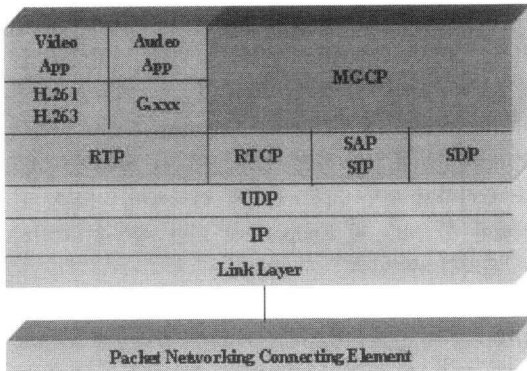


그림 2. MGCP Protocol Stack

MGCP는 Call Control을 위해 다음과 같은 명령어를 이용한다. Call Control을 위한 명령어는 MGC와 MG사이의 Signaling에만 국한된다. MGCP에서 Call Control을 위해 사용되는 명령어는 다음과 같다.

▶ RQNT(Notification Request)

MGC에 연결된 특정 Endpoint에서 일어나는

Off-hook 등의 Event를 MG에서 감시하도록 명령한다. 만약 요청한 Event가 Endpoint에서 발생할 경우 MG는 NTFY를 이용하여 MGC에 알려야 한다. MGC에서 MG로 전달된다.

▶ NTFY(Notify)

RQNT에 의해 감시하도록 정의된 Event가 일어났을 때 해당 Event의 종류와 특성을 MGC로 알린다. MG에서 MGC로 전달된다.

▶ CRCX(Create Connection)

MG에 연결된 특정 Endpoint로부터 Off hook event가 발생했을 때 해당 Endpoint에 RTP Session을 열도록 지시하는 역할을 한다. 그리고 Digit Event가 발생하였을 때 Digit에 해당하는 Destination Endpoint에 대해서도 RTP Session을 열도록 지시한다. MGC에서 MG로 전달된다.

▶ DLCX(Delete Connection)

MG에 연결된 특정 Endpoint로부터 On-hook event가 발생했을 때 해당 Endpoint에 RTP Session을 닫도록 지시한다. MGC에서 MG로 전달된다.

▶ MDCX(Modify Connection)

MG에 연결된 특정 Endpoint로부터 Digit Event가 발생하여 Destination Endpoint와 CRCX를 통해 연결되었을 때, 기존의 Source Endpoint가 지닌 RTP Session에 대해 Destination의 RTP Session 정보를 가질 수 있도록 변경하는 작업을 한다. Source와 Destination간의 RTP Session에 대한 Parameter Negotiation은 MGC에서 일어난다. MGC에서 MG로 전달된다.

▶ AUPE(Audit Endpoint)

MG에 연결된 특정 Endpoint에 대한 감사를 수행한다. MGC에서 MG로 전달된다.

▶ AUCX(Audit Connection)

MG에 연결된 특정 Connection에 대한 감사를 수행한다. MGC에서 MG로 전달된다.

▶ RSIP(Restart In Progress)

MG에 특정 Endpoint가 새로이 등록하였을 때 MGC로 등록 정보를 전달하는 역할을 한다. MG에

서 MGC로 전달된다.

그림 3은 하나의 음성통화를 위해 MGC와 게이트웨이 사이에 전달되는 MGCP 메시지 흐름에 관한 것으로 권고안(Implementation Guide)에 정의되어 있다.^[9]

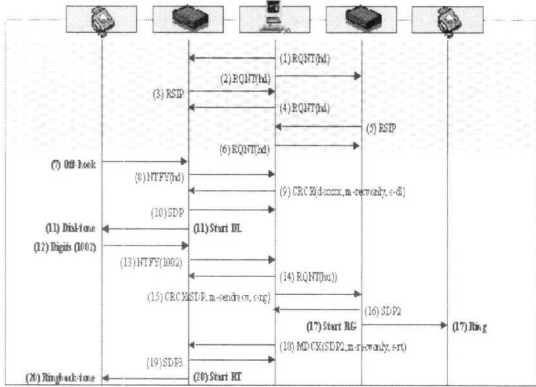


그림 3. 권고안에서 제안한 MGCP Call Flow

MGCP에서는 기본적인 Call Flow를 처리하기 위해 위에서 정의한 명령어가 그림 3과 같은 순서로 실행된다. 구체적인 메시지의 흐름과 이에 따른 동작은 다음과 같다. MGC는 자신의 관리영역에 있는 MG에 대해 해당 MG와 연결되어 있는 Endpoint의 Off hook event를 받아들일 수 있도록 RQNT(1)를 전달한다. RQNT(1)을 받은 MG는 자신에게 연결된 Endpoint로부터 Off hook이 발생하면 이를 파라미터와 함께 NTFY(2)를 이용하여 MGC로 전달한다. 특정 Endpoint는 RSIP(3)을 통해 MGC에 등록되며, MGC는 RQNT(4)를 전달함으로써 Endpoint의 Off hook event를 받아들일 수 있도록 한다. RQNT(4)를 받은 MG는 자신에게 연결된 Endpoint로부터 off hook(7)이 발생하면 이를 파라미터와 함께 NTFY(8)를 이용하여 MGC로 전달한다. Off hook이 발생한 후 MGC는 Endpoint에게 RTP를 준비하도록 하는 CRCX(9)를 MG에게 보낸다. 이 때 MGC는 Endpoint로부터 Digit Event의 발생을 전달받을 수 있도록 하기 위해 RQNT에 Digit Map을 포함하여 전달하며, 역시 RQNT를 통해 Dial-tone signal을 Endpoint로 보낼 수 있도록 한다. 해당 Endpoint에서 RTP Session을 준비하면 MG는 CRCX에 대한 ACK(10)상에 RTP Session에 대한 파라미터를 포함하여 MGC로 전달한다. RTP Session에 대한 파라미터 및 Endpoint의 자원정보는

SDP를 이용하여 기술되어 있다.

Endpoint로부터 Digit이 발생하면 MG에서는 Digit Map을 이용하여 저장하며, Digit Map에서 정의한 규격에 일치하는 번호가 발생했을 때 NTFY(13)을 이용하여 MGC로 해당 번호를 전달한다. Destination Endpoint의 번호를 받은 MGC는 해당 Endpoint가 RTP Session을 준비할 수 있도록 MG에 대해 CRCX(15)를 전달한다. 이 때 MGC는 ACK(16)에서 받은 SDP를 함께 전달하며 Destination Endpoint가 RTP Session을 수용할 수 있는지의 여부를 판단할 수 있도록 한다. Destination Endpoint에서 RTP Session의 준비가 끝난 후 MG는 해당 RTP Session의 파라미터를 ACK(16)상에 포함하여 MGC로 전달한다. Destination Endpoint로부터 SDP를 받은 MGC는 양단의 Endpoint의 RTP Session 파라미터에 대한 Negotiation 작업을 수행한다. 그리고 수정된 RTP Session 파라미터를 MDCX(18)를 이용하여 Source Endpoint로부터 전달하여 RTP Session을 열도록 지시한다. MGCP는 이와 같은 일련의 수행과정을 통해 Call Control을 구현한다.

그림 4는 AX-Gateway 시스템에서 하나의 음성통화를 위해 MGC와 게이트웨이간에 전달되는 MGCP 메시지 흐름에 관한 것이며, 이는 현재 운용중인 특정 시스템에서 채택한 내용이다. 권고안에서 제시한 방법과의 차이점은 음성통화를 위한 자원정보의 수집에 있어서 권고안에서는 음성통화가 이루어질 당시에 CRCX에서 처리하는 반면, 이 시스템에서는 시스템이 시작할 때 AUEP를 통해 처리한다는 측면이 있다.

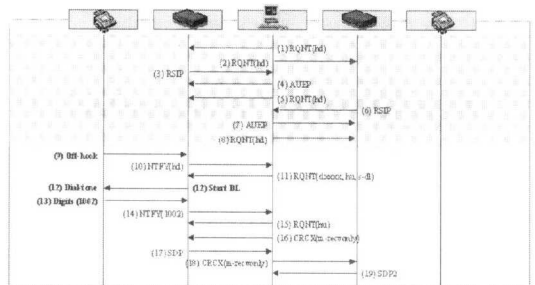


그림 4. AX-Gateway에서 제안한 MGCP Call Flow

그림 5는 권고안에 정의된 MGCP 메시지의 예를 나타낸다. 그림 5에서 첫번째 메시지(1)는 그림 3의 RQNT(4)에, 두번째 메시지(2)는 CRCX(9)에, 세번째

째 메시지(3)는 CRCX(9)에 대한 SDP(10)에 각각 해당된다.

```

(1)
09/14/01 06:39:15>User Thread[SEND]pqt1 10005 aahs0@165.243.134.242 MGCP 1.0
X: 1005
R: L/hd(N)

(2)
09/14/01 06:39:25[MgcpMgc_jube32427_0]SENDrrcs 10012 aahs0@165.243.134.242 MGCP 1.0
C: 20962
X: 1012
L: p:20, a:PCMU;G7231;G729;PCMA
M: recvonly

(3)
09/14/01 06:39:25[MgcpMgc_jube32427_0]RECEIVE|200 10012 OK
I: 0015A07A08

v=0
m=audio 30000 RTP/AVP PCMU
c=IN IP4 165.243.134.242
    
```

그림 5. 권고안에서 제시한 MGCP 메시지

그림 6은 AX-Gateway 시스템에서 사용되는 메시지의 예를 나타낸 것으로 권고안에서 제시한 방법과의 차이점은 각 그림의 메시지(1)에서 볼 수 있듯이 포맷에 있어서 패킷지의 선언 유무(ex. Msg.5-1: "R: L/hd(N)", Msg.6-1: "R: hd"), 그리고 각 그림의 메시지(3)에서 Codec을 정의하기 위한 Syntax가 다른 점(ex. Msg.5-3: "m=audio 30000 RTP/AVP PCMU", Msg.6-3: "m=audio 30008 RTP/AVP 0") 등이다.

```

(1)
2001/06/21 17:21:07 5821 TX:
RQNT 3010 aahs1@gjw.yang MGCP 1.0 NCS 1.0
R: hd
X: ABCD8010

(2)
2001/06/26 10:37:17 9412 TX:
CRCX 15387 aahs1@gzykoo MGCP 1.0 NCS 1.0
C: 9999100700000000000000156350
L: p:20, a:PCMU, esn
M: recvonly

(3)
2001/06/26 10:17:17 9442 RX:
200 15387 OK
I: b

v=0
c=IN IP4 211.237.75.106
m=audio 30000 RTP/AVP 0
    
```

그림 6. AX-Gateway에서 제안한 MGCP 메시지

III. RADVision Protocol Stack

현재 MGCP의 구현을 지원하는 Protocol Stack으로서 RADVision에서 제공하는 MGCP Toolkit이 널리 응용되고 있다⁷⁾. 다음은 RADVision에서 MGCP를 지원하기 위한 구조를 보여준다(그림.7).

RADVision은 MGCP Stack에 대한 추상화를 제공하기 위해 Event와 Signal을 처리하기 위한 Handler를 제공하여 시스템의 구현에 대한 편의를 제공한다. 그리고 메시지 처리에 있어서도 MGCP 고유의 메시지 포맷과 이를 간략히 정의한 자신들

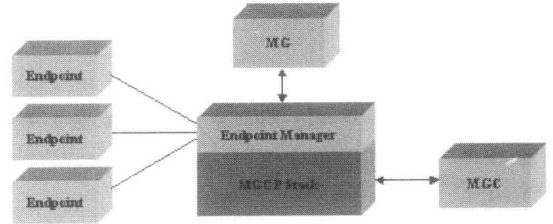


그림 7. RADVision Architecture to support MGCP

의 Message Handling API를 지원하고 있다.

그림[7]과 같이 RADVision은 Endpoint Manager를 이용하여 MGCP에 대한 별도의 지식이 없어도 MG를 구성할 수 있도록 지원한다. Endpoint Manager에서는 Endpoint에서 전달되는 Call 처리에 있어서 MGCP 명령어가 아닌 기존의 전화시스템에서 이용되는 Event와 Signal의 형태로 인터페이스를 제공한다. Endpoint Manager는 다음과 같은 절차를 이용하여 하나의 Endpoint를 등록하고 Stack으로부터 전달되는 Callback Event 및 Signal을 받게 된다.

- ① Endpoint에 대한 Capability를 설정한다.
- ② Endpoint에 대한 인스턴스를 만든다.
- ③ Endpoint에 대해 Callback Function을 등록한다. Callback Function은 Stack으로부터 해당 Endpoint에 대해 Connection과 관련된 Callback Event (Open, Modify, Close) 및 Endpoint로 전달되는 Signal(Dial tone, Ring, Ring-back tone, Busy tone)을 받기 위해 이용된다. Callback Function으로 이용되는 함수들은 별도로 정의되어야 한다.
- ④ 준비된 Endpoint를 Stack에 등록한다.

이러한 절차를 통해 Stack으로부터 특정 Endpoint에 대해 발생하는 Connection과 관련된 Callback Event 및 Signal을 받을 수 있게 된다. Offhook이나 Onhook과 같이 Endpoint에서 일어나는 Event를 Stack으로 전달하기 위해서는 다음과 같은 메소드를 이용한다.

- ⑤ rvMgcpEndpointProcessEvent()를 이용하여 Event를 Stack으로 전달한다.

IV. MGCP Module on AX-Gateway System

하나의 MGCP 모듈은 회선교환망의 신호를 패킷 교환망의 신호로 전환하는 MG(Media Gateway)와 각각의 MG를 제어하기 위한 MGC(Media Gateway

Controller)로 구성되며, 다음과 같은 구조로 설계되었다.

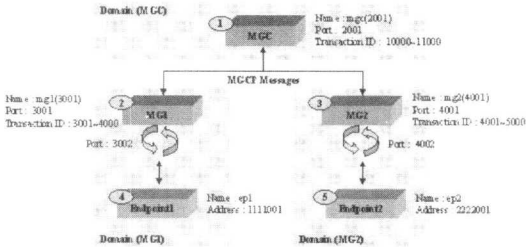


그림 8. MGCP System for AX-Gateway

MGC(1)는 MG 및 각 MG에 연결된 Endpoint에 대한 구성정보와 상태정보를 가지고 있다. MGC는 각 Endpoint에 대한 상태정보를 바탕으로 해당 Endpoint를 제어하기 위한 명령어를 MG로 전송한다. AX-Gateway 시스템의 MGCP 모듈에서는 MGC의 Stack에 대한 포트를 2001로 설정하였으며, Transaction ID의 범위로 10000 ~ 11000을 이용하고 있다. MG(2,3)는 Endpoint(4,5)로부터 Off-hook이나 On-hook과 같은 Event를 받으며, 이를 처리하기 위해 NOTIFY 명령어를 MGC로 전송한다. MGC는 MG로부터 받은 명령어를 바탕으로 상태정보를 변경시킨 후, MG에게 CRCX, DLCX와 같은 명령어를 전달하여 MG가 Event를 발생시킨 Endpoint를 제어할 수 있도록 한다. 명령어를 전달받은 MG는 Endpoint에게 Dial-tone이나 Ringback-tone과 같은 Signal을 발생시킬 것을 명령하고, 자신은 RTP Connection에 대한 처리 작업을 수행한다. MG는 Stack에 대한 포트와 Endpoint 및 External Unit (TRCP, DSP, RTP)과 인터페이스를 위한 포트를 유지하며, Transaction ID는 포트를 기준으로 할당하게 된다. MGC는 각각의 MG 및 MG에 연결된 Endpoint를 제어하기 위한 역할을 하며, 이들에 대한 구성정보와 상태정보를 유지한다. MGC가 각각의 Endpoint에 대해 유지하는 정보는 다음과 같다 [그림.9].

하나의 Endpoint에 대해 유지하는 정보는 크게 Endpoint 자체의 식별정보(Endpoint-related data), 연결관리정보(Connection related data), 그 외 시스템 데이터와 플래그 정보 등으로 구성된다. Endpoint의 식별정보는 Endpoint의 주소(IP, Port, ...)와 Endpoint가 지원하는 패키지(D, G, L, ...), 현재 자신이 서비스 할 수 있는지 등의 상태정보, 그리고 자신이 원격 Endpoint로서 동작하는지 등의

```

1.2 mgcpMgc: Endpoint list definition in mgcpMgc.h...

typedef struct {
    /* Endpoint-related data
    mgcpTpAddr address;
    char* defaultIPflag;
    int anno=ep;
    mgcpTpStateEp epState;

    /* Connection-related data
    char connId[32];
    unsigned int transId;
    char dialString[32]; int dialStringLength;
    mgcpTpStateCall callState;
    RvDpMMsg *dpMMsg;

    /* System data and Flag
    RvTimer timer; RvTimerId timerId; RvMutex mutex;
    RvBool occupiedEntry; RvBool earlyAnswer;
} mgcpTpEp;

mgcpTpEp mgcpMgcListEp [mgcpMgcCstNumEp];
    
```

그림 9. MGC data Definition

정보를 포함한다. 연결관리정보는 현재 Endpoint가 연결된 Connection 관련정보를 의미하며, Connection ID, Transaction ID, 상대방에 대한 Dial String, Connection 중 자신이 속해 있는 상태정보, 자신의 SDP 정보 등을 포함한다. MGC의 각 정보는 다음과 같은 절차를 거쳐 초기화됨으로써, 서비스를 위한 준비를 수행한다[그림.10].

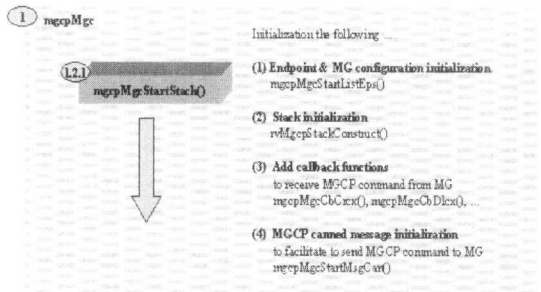


그림 10. Initialization Steps of MGC

MGC의 초기화는 함수 mgcpMgcStartStack() 내에서 이루어진다. 초기화는 우선 구성파일로부터 Endpoint에 대한 정보를 읽어 들이고, 이에 대한 리스트를 만드는 작업부터 수행된다. Endpoint에 대한 구성정보는 다음과 같은 형식으로 이루어져 있다[그림.11].

File Name : config/ep.cfg

MGC			MG			EP			
Name	IP	Port	Name	IP	Port	Name	Address	IP	Port
cal	165.243.134.90	2001	mg1	165.243.134.90	3001	ep1	1111001	165.243.134.90	3002
cal	165.243.134.90	2001	mg2	165.243.134.90	4001	ep2	2222001	165.243.134.90	4002

그림 11. Configuration Table

Endpoint에 대한 리스트가 만들어지면, MGC Port 및 Transaction ID Range 등의 정보와 함께

rvMgcpStack Construct()를 이용하여 Stack을 초기화한다. Stack 초기화 이후 MG로부터 전달되는 MGCP 명령어를 받기 위한 Callback Function들을 Stack에 등록한다. 그리고 MGC에서 MG로 전달하기 위한 명령어를 Canned Message라는 간략한 형태로 초기화한다. 초기화 과정이 완성되고 나면, MG로부터 다음과 같은 과정을 통해 MGCP 명령어를 주고 받게 된다[그림.12].

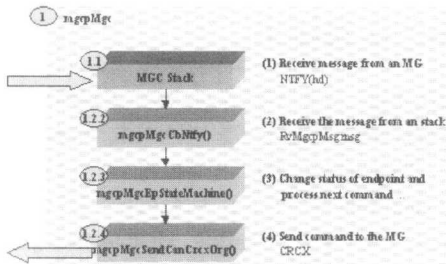


그림 12. MGCP Command Processing in MGC

그림[12]의 과정은 MG로부터 Off-hook 이벤트에 대한 NTFY 명령어가 전달되었을 때를 가정하였다 (1). 이벤트가 Stack을 통해 전달되면, Stack은 NTFY에 해당하는 Callback Function인 mgcpMgcCbNtfy()를 실행하며, 인자로 RvMgcpMsg를 전달한다(2). RvMgcpMsg에는 이벤트를 발생시킨 Endpoint에 대한 정보와 이벤트에 대한 자세한 정보가 포함되어 있다. 전달된 RvMgcpMsg를 바탕으로 전달된 이벤트와 함께 다음 상태를 유추해 CRCX 명령어를 MG로 전송한다. MGCP기반의 시스템에서 게이트웨이는 MGC의 동작에 절대적으로 의존해야만 하며, 이는 MGCP 프로토콜의 고유의 특성이기도 하다. 그러나 MGC는 MGCP가 허용하는 범위내에서 메시지의 흐름이나 패키지의 선언 유무, 인코딩 방식 등에 있어서 MGC를 구현한 각 업체들간에 있어서 구현상의 미묘한 차이점을

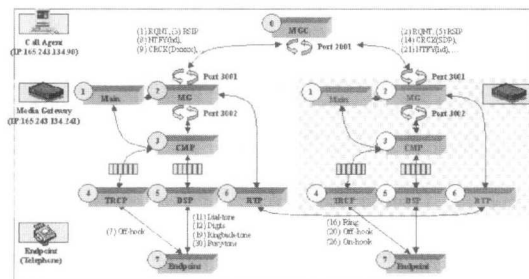


그림 13. AX-Gateway에서의 MGCP S/W 구성도

가지고 있으며 이러한 차이점은 게이트웨이 시스템의 호환성을 저하시키거나 호환이 되지 않는 원인이 되고 있다. 가지고 있으며 이러한 차이점은 게이트웨이 시스템의 호환성을 저하시키거나 호환이 되지 않는 원인이 되고 있다.

그림 13은 위에서 제시한 문제점을 극복하기 위한 방안으로서 본 논문에서 제안한 방식을 적용하여 현재 사용되고 있는 AX-Gateway 시스템의 소프트웨어 구성도이다. 각 소프트웨어 구성 모듈은 다음과 같이 이루어져 있다.

▶ Main(13-1) : 시스템의 초기화를 수행한 후, 사용자의 명령을 받아들이고 해당 루틴을 수행하는 부분.

▶ MG(13-2, Media Gateway) : MGCP Stack을 이용하여 MGC와 명령어를 주고 받기 위한 모듈이다.

▶ CMP(13-3, Central Management Process) : 시스템의 구성정보를 바탕으로 시스템을 관리하기 위한 모듈.

▶ TRCP(13-4, Trunk Control Process) : Trunk와 관련된 Event와 Signal을 처리하는 부분이다.

▶ DSP(13-5, Vocoder Process) : 음성을 G.711, G.723, G.729와 같은 Codec 형식으로 압축하거나 복원하기 위한 모듈이다.

▶ RTP(13-6, Real-time Transfer Protocol Process) : 실제 음성통화에 이용되는 스트림을 생성하고 입, 출력하기 위한 모듈.

▶ MG(13-2, Media Gateway) : MGCP Stack을 이용하여 MGC와 명령어를 주고 받기 위한 모듈이다.

▶ CMP(13-3, Central Management Process) : 시스템의 구성정보를 바탕으로 시스템을 관리하기 위한 모듈.

▶ TRCP(13-4, Trunk Control Process) : Trunk와 관련된 Event와 Signal을 처리하는 부분이다.

▶ DSP(13-5, Vocoder Process) : 음성을 G.711, G.723, G.729와 같은 Codec 형식으로 압축하거나 복원하기 위한 모듈이다.

▶ RTP(13-6, Real-time Transfer Protocol Process) : 실제 음성통화에 이용되는 스트림을 생성하고 입, 출력하기 위한 모듈.

그림 14는 그림13의 소프트웨어 모듈 중에서 MG에 해당하는 내용을 세분화 하여 도식화한 것이다. 그림 9의 구성에서는 다양한 MGC와의 호환성

에 문제를 일으킬 수 있는 부분을 소프트웨어 모듈화하여 시스템의 일반적인 부분과 분리시켜 놓았다.

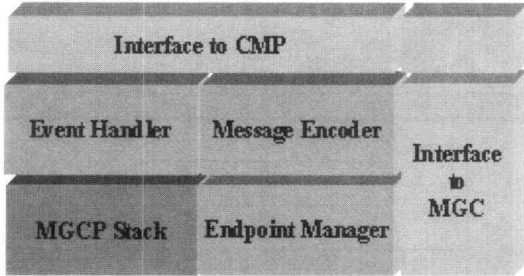


그림 14. AX-Gateway에서의 MG Mode 구조

호환성을 위한 소프트웨어 모듈로는 메시지의 흐름을 제어하기 위한 모듈과 메시지의 인코딩을 정의하기 위한 모듈 등이 있다. 각 구성 모듈을 살펴보면 다음과 같다.

- ▶ MGCP Stack : MGCP 명령어를 처리하기 위한 Stack에 해당.
- ▶ Event Handler : CMP와 MG간에 Event나 Signal의 전달을 중재하기 위한 인터페이스 루틴.
- ▶ Endpoint Manager : MG의 State Machine을 유지하고 있으며, 이를 바탕으로 MGC로부터 전달된 MGCP 메시지에 따라 State를 변경. 또한 이 State를 중심으로 Endpoint에 대해 어떠한 Event나 Signal이 전달되어야 할 지 파악.
- ▶ Message Encoder : 메시지의 인코딩 방식을 처리하기 위한 모듈.

그림 14의 모듈 중에 다양한 MGC와의 호환을 위해 고려되어야 할 부분은 Endpoint Manager와 Message Encoder 부분이다. MGC에 따라 구현이 달라질 수 있는 부분은 메시지의 흐름이나 메시지의 Encoding방식이기 때문이다.

현재 구현된 게이트웨이 시스템에서 Endpoint Manager는 별다른 설정 없이 다양한 MGC로부터 전달되는 메시지의 흐름을 수용할 수 있도록 설계되었다. (메시지의 다양한 흐름에 대해서는 그림 3과 그림 4를 참고한다.) MGC마다 메시지의 흐름에는 미묘한 차이가 있더라도 MGCP공유의 Semantic을 유지하기 위한 범위 내에서 구현되기 때문에, 동일한 State Machine으로도 각기 다른 메시지의 흐름을 수용하는 것이 가능하다.

이에 반해 Message Encoder는 게이트웨이 시스템에서

Encoding 방식에 대한 설정을 실시간으로 정의할 수 있는 방법을 제공한다. 사전에 어떤 MGC와 연동이 될 지 알 수 없기 때문에, 메시지의 Encoding 방식에 대해 시스템의 구성메뉴를 이용하여 별도로 정의할 수 있도록 한다.

본 논문에서 제안한 MGCP Software구조를 AX-Gateway 시스템에 적용하여 자체적으로 테스트한 환경을 간략히 나타내면 다음과 같다[그림15].

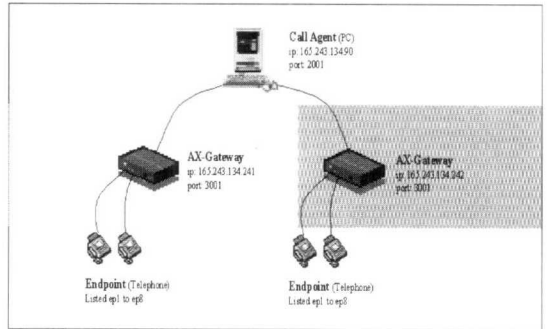


그림 15. Test-bed of MGCP Software

그림 16은 클라린트의 MGC와 연동시험을 위한 테스트 환경을 나타낸 것이다.

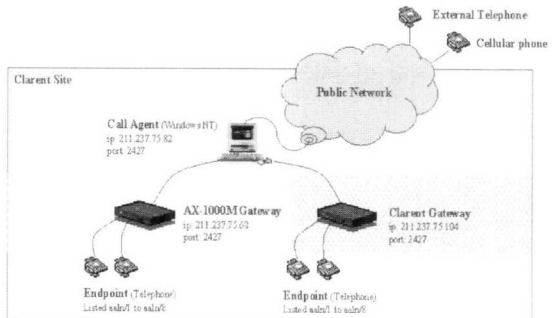


그림 16. 클라린트사 MGC와의 연동

클라린트의 MGC는 windows NT 기반에서 운용되었으며 이를 통하여 AX-Gateway에 연결된 단말과 Public Network를 통해 외부에 연결되어 있는 단말과의 음성통화 테스트를 하였다.

시스템의 성능측정은 통화품질, 지연, 최대 동시 통화수를 측정하였다. 통화음질을 측정하기 위해서 HP VQT를 사용하여 구간별 PSQM과 지연시간을 측정하였고, 최대 동시 통화수는 내신간의 통화와 AX-Gateway 2대간의 통화수를 측정하였다. 본 측정에 사용된 codec의 사양은 표1과 같다.

표 1. Codec 사양

Codec	Frame 크기(ms)	대역폭(Kbps)
G.723 5.3K	30	5.3
G.723 6.3K	30	6.3
G.729A	10	8
G.711 a-law	10	64

▶ 통화음질과 지연

본 성능 측정은 그림17과 같이 A,B,C의 세 구간으로 나누어 실험을 하였으며 네트워크에 의한 영향을 최소화하기 위하여 가장 단순한 형태의 연결을 하였다. 시스템의 성능 목표는 평균PSQM 3.0이하, 최대PSQM 6.0이하, 지연시간은 150ms이하로 하였다.

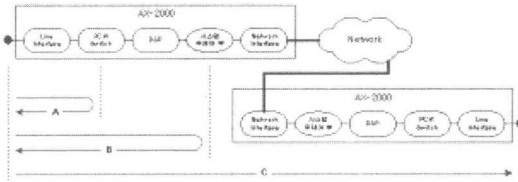


그림 17. 통화음질과 지연측정 구간

① A Region

한 대의 AX-Gateway에서 FXS 2 Port에 HP VQT를 연결하고, FXS 2 Port간의 경로를 PCM Switch로 직접 연결하였다.

표 2. 구간 A 측정 결과

평균 PSQM	최대 PSQM	Delay(ms)
1.03	2.10	1.375

② B Region

한 대의 AX-Gateway에서 2개의 FXS Port간의 내선통화 환경을 구성하여 측정하였다.

표 3. 구간 B 측정결과

Codec	평균 PSQM	최대 PSQM	Delay (ms)	성능 목표
G.723 5.3k	2.98	5.91	125.25	만족
G.723 6.3k	2.56	4.71	125.25	만족
G.729A	2.7	4.76	78.625	만족
G.711 A law	1.08	2.1	98.75	만족

③ C Region

두 대의 AX-Gateway에 대하여 측정하였다. 측정

은 각 시스템의 FXS Port에서 하였으며 Codec의 종류와 한 패킷당 음성 프레임의 수를 변화시키면서 측정하였다.

표 4. 구간 C 측정결과

Network	Codec	Frame/Packet	평균 PSQM	최대 PSQM	Delay (ms)	성능 목표
LAN	G.723 5.3k	1	2.6	4.96	130.375	만족
		2	2.64	5.54	146.75	만족
		3	2.63	5.43	199.375	불만족
	G.723 6.3k	1	2.55	4.98	124.875	만족
		2	2.63	5.41	161.375	불만족
		3	2.7	5.65	186.125	불만족
	G.729A	1	2.6	4.45	132.75	만족
		2	2.68	4.97	166.25	불만족
		3	2.72	5.29	198.875	불만족
G.711 A law	1	2.99	4.77	136.875	만족	
	2	2.57	4.73	153.0	불만족	
	3	2.99	5.12	192.875	불만족	
Serial	G.723 5.3k	1	2.87	5.57	142.25	만족
		2	2.92	5.05	163.0	불만족
		3	2.85	5.42	186.75	불만족
	G.723 6.3k	1	2.56	5.12	130.875	만족
		2	2.63	5.09	171.375	불만족
		3	2.6	4.88	181.625	불만족
	G.729A	1	2.65	4.38	77.625	만족
		2	2.67	5.24	93.5	만족
		3	2.62	4.46	104.375	만족
Modem	G.723 5.3k	1	2.95	5.49	190.0	불만족
		2	2.89	5.25	244.25	불만족
		3	2.87	5.99	273.5	불만족
	G.723 6.3k	1	2.63	5.19	202.125	불만족
		2	2.6	5.07	229.625	불만족
		3	2.99	4.93	281.875	불만족
	G.729A	1	2.75	5.04	136.5	만족
		2	2.99	4.55	146.875	만족
		3	2.64	5.54	165.25	불만족

구간 A의 음질과 구간 B의 음질의 차이는 G.711의 경우에는 거의 없었으며 나머지 codec에서는 약 2.7 배의 차이를 보였다. 네트워크 연결을 단순하게 하였기 때문에 G.711을 제외한 다른 codec에서는 구간B,C에서 음질이 비슷하게 측정되었다. 음성지연은 구간 A와 B가 많은 차이를 보였으며, 구간 B,c에서는 모뎀일 경우에 많은 차이를 나타내었다. 음질은 성능목표를 만족시켰으나 지연은 프레임/패킷 비율과 네트워크의 종류에 따라 만족시키지 못하는 경우가 발생했다. 특히 Modem이나 Serial 포트를 사용할 경우에 품질 저하가 발생함을 알 수 있었다.

▶ 최대 동시 통화 사용자의 수

본 측정은 시스템 구성을 변경하면서 구성에 따라 동시에 통화할 수 있는 최대 호 수를 측정하였다. 측정시 네트워크의 대역폭은 시스템 모든 포트에서 동시에 호가 진행이 가능하도록 설정하였다. 측정은 내선통화와 AX-Gateway 2대간 통화로 나누어 실시하였다. 최대 동시 통화 호수는 평균

PSQM 6이하. 지연 250ms이하를 만족시키는 범위 내에서 최대한로 동시 통화가 가능한 호수로 결정하였다. 본 측정에 사용된 환경에서 내선통화일 경우 최대한로 시도할 수 있는 호 수는 32 call이다.

표 5는 내선통화시 최대 동시 통화호 수의 측정 결과를 나타낸다.

① 내선통화

표 5. 내선통화 측정결과

Codec	동시 통화 호 수	평균 PSQM	Delay (ms)	성능 목표
G.723.53k	14	2.89	125.25	만족
	15	2.87	125.25	만족
	16	2.87	125.25	만족
	17	2.86	125.25	만족
G.729A	6	2.63	78.625	만족
	7	2.66	78.625	만족
	8	2.63	78.625	만족
	9	2.63	78.625	만족
	10	2.86	178.625	만족

② AX-Gateway 2대간 통화

표 6. 2대간 통화

Network	Codec	Frame/Packet	동시 통화 호 수	평균 PSQM	Delay (ms)	성능 목표
LAN	G.723.53k	1	15	2.64	124.92	만족
			16	2.64	117.35	만족
			17	2.65	104.96	만족
			18	2.68	134.92	만족
			19	3.59	303.125	불만족
			20	2.65	130.125	만족
		2	26	2.7	124.625	만족
			29	2.65	137.125	만족
			38	2.6	141	만족
			29	3.25	263.96	불만족
			28	2.6	151.275	만족
			29	2.36	159.25	만족
	G.729A	3	30	2.67	149.96	만족
			31	2.81	174.375	만족
			32	2.64	168.375	만족
			4	2.83	95.46	만족
			5	2.67	83.71	만족
			6	2.65	130	만족
		1	7	2.6	121.17	만족
			8	5.21	359.625	불만족
			7	2.65	151.79	만족
			8	2.7	137.71	만족
			9	2.64	142.375	만족
			10	2.6	130.125	만족
2	11	3.34	276.71	불만족		
	9	2.64	193.46	만족		
	10	2.66	180.79	만족		
	11	2.6	171.21	만족		
	12	2.58	162.83	만족		
	13	4.48	271.08	불만족		

시스템 구성에 따라 많은 차이를 보였으며, 같은 구성이라도 측정시마다 조금씩 차이가 발생하였다. 이는 측정시 네트워크에 걸리는 부하에 따른 영향이라고 판단된다.

표 6의 측정결과를 분석하여 볼 때 한계점에 이르기까지는 호가 1개 접속되어 있을 때 측정결과가 유사하게 유지되지만 한계점을 넘어서면 측정값이

급격하게 나빠짐을 알 수 있다.

또한 한계점을 넘어서면 패킷 손실도 급격하게 증가함을 알 수 있었다. 최대 동시 통화 호 수는 그림 18과 같다.

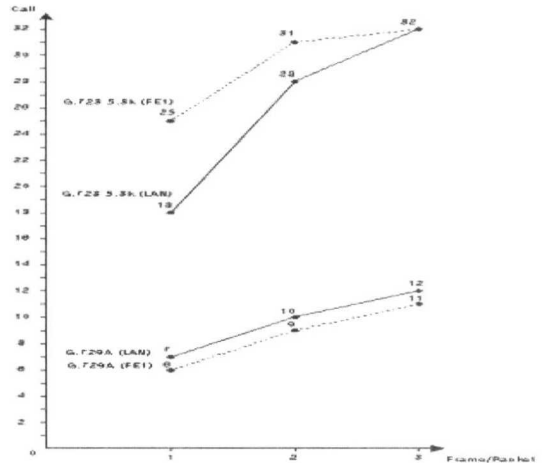


그림 18. 최대 동시 통화 호 수

V. 결론

본 논문에서는 MGCP 기반의 게이트웨이 시스템에서 다양한 MGC에 대해 호환성을 지원하는 방법에 관한 것으로, 각 MGC에서 미묘한 차이를 보일 수 있는 메시지 흐름이나 Encoding 방식 등에 대한 문제점을 지적하고 이를 소프트웨어 모듈화를 통해 해결하기 위한 방법을 소개하였다.

즉, 다른 시스템과의 호환성을 높이기 위한 새로운 MG 소프트웨어의 구조, 메시지의 흐름을 제어하기 위한 소프트웨어 모듈의 정의와 구조, 메시지의 인코딩 방식을 제어하기 위한 소프트웨어 모듈의 정의와 구조, 소프트웨어 모듈에서 메뉴를 통해 메시지의 인코딩 방식을 실시간으로 설정하기 위한 방법 등을 제안하고 구현하였으며 성능평가를 하였다.

제안된 소프트웨어 구성과 모듈을 이용하면 다음과 같은 효과를 기대할 수 있다.

- ▶ MGC 고유의 메시지 흐름에 독립적으로 게이트웨이를 구현할 수 있다.
- ▶ MGC 고유의 메시지 Encoding 방식에 독립적으로 게이트웨이를 구현할 수 있다.
- ▶ 소프트웨어 모듈화를 통해 추가적인 수정사항의 발생에 용이하게 대응할 수 있으며, 다른 시스템

에 대한 재사용 또한 용이하게 할 수 있다.

▶ 다른 MGCP 시스템의 MGC와 호환성에 있어서 유연하게 대처할 수 있다.

본 논문에서 제시하고자 하는 구조는 현재 본사에서 개발중인 VoIP 시스템인 AX-Gateway 시스템을 기반으로 하였다.

AX-Gateway 시스템에 MGCP를 구현하는 작업은, 기존에 개발한 H.323을 기반으로 한 시스템에서 작성된 관리 모듈을 그대로 이용하기 위해서 H.323을 지원하기 위해 MG내에 자신에게 연결된 Endpoint에 대한 상태를 유지하게 하였다. 따라서 H.323을 바탕으로 설계된 기존의 시스템과의 연동을 위해, MGCP의 MG 모듈 설계에 있어서 H.323의 Request와 Indication 형태의 인터페이스와 연동될 수 있게 함으로써 MG에서 MGC로 Call Connection을 수행하기 전에 MG차원에서 해당 요청이 적절한 것인지 판단할 수 있게 되어 Call Agent의 부하를 줄여 Traffic의 절감과 시스템의 안정성 및 성능향상을 높일 수 있는 부가적인 효과도 갖게 된다.

현재까지 MGCP를 적용한 시스템 개발상황은 Real-time OS기반(pSos, VxWorks)의 AX-Gateway System에 적용을 완료하였고 클라이언트의 MGC와 연동시험을 마쳤다. 현재 성능 향상을 위한 테스트와 각 사이트별 특성에 따른 필드 adaptation을 진행 중에 있다. 향후 개발 방향으로 AUDIT 기능에 대한 추가 사항, 다른 밴더의 MGC와의 호환성 시험 및 SIP, MEGACO의 적용 등에 대해 연구를 지속적으로 진행할 계획이다.

참 고 문 헌

[1] Ovum, "IP Telephony: Exploiting Market Opportunities", Peter Hall, April 20th, 2001.

[2] ITU-T, "ITU-T Recommendation H.323 Draft V4", Nov, 2000.

[3] Asim Karim, "H.323 and Associated Protocols", February 7th, 2000

[4] RADVision, "RADVision's H.323 Protocol Stack Primer", 1999. Reference Site: www.radvision.com

[5] IETF MMUSIC, "SIP, Session Initiation Protocol", September, 1999 Reference Site:

http://www.cs.columbia.edu/sip/

[6] IETF, "RFC 2705, Media Gateway Control Protocol Version 1.0", October, 1999

[7] RADVision, "RADVision's MGCP Media Gateway Toolkit Programmer Guide", 1999.

[8] Bill Douskalis, "IP Telephony, The Integration of Robust VoIP Services", pp.210-221, Prentice Hall PTR, 2000.

[9] http://www.softswitch.org/attachments/mgcp-guide.txt

[10] IETF, "RFC 3015, Megaco Protocol Version 1.0", November, 2000

강 재 경(Jae-kyng Kang)

정희원



1988.03 ~ 1992.02 :

서울시립대학교 전자공학과

1992.03 ~ 1994.02 :

서울시립대학교 대학원

전자공학과 석사

1994.03 ~ 2001.02 :

서울시립대학교 대학원

전자공학과 박사수료

1993.12 ~ 1997.03 : 롯데 전자 기술연구소

1997.04 ~ 1998.12 : LG정보통신 정보시스템연구단

멀티미디어실 주임연구원

1998.12 ~ 2000.9 : LG정보통신 정보시스템연구소

미디어S/W 주임연구원

2000.9 ~ 2001.2 : LG전자 디지털네트워크연구소

미디어통신실 주임연구원

2001.3 ~ 2001.11 : LG전자 디지털네트워크연구소

미디어통신실 선임연구원

2001.12 ~ 현재 : LG전자 CTO부문 Network Systems & Solutions Team Solution 기술 Gr. 과장

김 현 규(Hyeon-gyu Kim)



1997년 2월 : 울산대 전자계산과 (공학사) 졸업

2000년 2월 : 울산대 전자계산과 (공학석사) 졸업

2001년 1월 ~ 2001년 12월 : 한국국방연구원 연구원

2002년 2월 ~ 현재 : LG전자 기업통신연구소 주임연구원

<주관심 분야> VOIP, 분산객체

이메일 : jube3@lge.com

오 은 록(Eun-rog Oh)



??~현재 : LG전자
기업통신연구소 책임연구원
<주관심 분야> VOIP, 음성/영상
신호처리, ATM, 차세대 지
능망
이메일 : eroh@lge.com

강 태 익(Tae-ik Kang)



1979년 2월 : 연세대 전자공학과
(공학사) 졸업
1981년 2월 : 연세대 전자공학과
(공학석사) 졸업
1990년 1월 : Polytechnic
University(Ph.D) 졸업

1981년 3월~1984년 6월 : 육군사관학교 전임 교수
1990년 5월~현재 : LG전자 기업통신연구소
책임연구원
<주관심 분야> VOIP, 음성/영상 신호처리, ATM,
차세대 지능망
이메일 : tikang@lge.com

김 철 주(Chul-ju Kim)



- 1981 3. 1984
Tokai University of Japan,
전자공학과(박사)
- 1979 3. 1981
Tokai University of Japan,
전자공학과(석사)
- 1966 2. 1973
한양대학교 전기공학과(학사)
1984년 4월~현재 : 서울시립대학교 전자전기컴퓨터
공학부 교수
1997년 12월~1998년 2월 : 일본 동경 공업 대학교
정밀 공학 연구소 객원교수
1989년 7월~1990년 6월 : 일본 sophia대학 이공학부
객원교수
1979년 4월~1984년 3월 : 일본 sophia대학 이공학부
고체 물리 연구실 연구원