

主 題

MGCP/MEGACO 와 소프트스위치 기술

강 태 익

차 례

- I. 서론
- II. MGCP
- III. MEGACO
- IV. 소프트스위치
- V. 결론

I. 서론

1960년대에 시작된 디지털 음성통신은 회선 교환 방식을 이용하여 음성통신을 지원하는 대표적인 기술로 발전하였으며, 높은 신뢰성과 품질을 보장함으로써 현재까지 광범위하게 사용되고 있다. 회선 교환방식은 호 연결 후 회선을 독점 사용하기 때문에 대역폭이 낭비되고 회선 사용이 비효율적이므로 회선 당 비용이 많이 소요되는 문제점을 안고 있다. 이러한 문제점을 해소하는 방안으로 인터넷 망으로 음성통신을 제공하는 인터넷 텔레포니 서비스가 등장하게 되었으며, 인터넷 전화 서비스는 패킷 교환 방식을 이용하여 대역폭의 효율적 점유와 음성 압축으로 회선 비용 절감 효과를 가져와 급격한 성장을 보이고 있다.

인터넷 전화는 공중 인터넷 망과 사설 IP망을 포함하는 IP망 상에서 전화서비스를 제공한다. 이것은 전통적인 전화사업자의 통신법과 Business(사업) 모델에 도전함으로써 변혁을 일으켰으며, 통신시장에

다양한 서비스를 제공할 수 있는 통신사업자의 출현을 가능하게 하였다. 현재의 인터넷 전화 사업자(ITSP, Internet Telephony Service Provider)는 공중통신망(PSTN)과의 상호운용을 달성함으로써, 일반가입자 및 기업고객에게 PC투폰, 폰투폰 서비스를 제공하고 있다. ITSP는 Best Effort QOS(Quality of Service)를 보장할 수 있는 자체망을 구성하며 UMS(Unified Messaging Service), ICW(Internet Call Waiting) 등 새로운 서비스를 제공하고 음성포털과 기업고객 등 새로운 시장에 진입을 위해 노력하고 있다.[1]

초기의 인터넷 전화는 소프트웨어 호환성, 컴퓨터의 압축, 망 지연, 통화 품질의 저하 등 여러 가지 단점들 때문에 성공 가능성이 없었다. 그러나 공중전화 교환 망(PSTN)과 인터넷 망을 연결하는 "게이트웨이"의 개발과 상호 운용성을 보장함으로써 급속히 성장하게 되었다. 게이트웨이는 VoIP(Voice over IP) 기술을 이용하여 회선교환망과 패킷교환망의 상호 운용성을 가능하게 한다. 또한 VoIP는 IP

(Internet Protocol)를 이용하여 음성과 데이터를 하나의 통신망으로 전달하는 기술이다. 즉 전화 또는 전화망에서 인입한 음성 신호를 보코더를 통해 음성 압축(G.729, G.732.1)하고 패킷 데이터화하여 IP 망을 통해 전송하고, 수신단에서 수신된 음성 패킷을 원래의 음성신호로 복원하는 기술을 말한다. 현재는 음성, 데이터, 멀티미디어 등을 통합하여 IP망 상에서 제공하는 다양한 서비스 및 응용서비스를 총칭하는 광의의 뜻으로 사용하고 있다[2][3].

1990년대 후반에 인터넷의 열풍에 편승하여 VoIP 기술은 음성, 데이터와 비디오 신호를 IP 기반으로 하는 하나의 Network으로 통합시켰으며 장비의 운용관리의 용이성과 저렴한 비용절감 등을 가져왔다. 그러나 인터넷 텔레포니 서비스는 공중 인터넷망의 속성인 데이터의 전송지연, Jitter, Packet 손실 이외에도 QOS(quality of service), 신뢰성, Interworking 등이 VoIP 서비스를 가로막는 장벽이 되고 있다. IETF와 네트워크 장비 제조업자들은 문제점을 해결하기 위해 WFQ(Weighted Fair Queue), Diffserv, MPLS, RSVP 등 여러 방안들을 제안하고 있다[4]. VOIP 표준 Protocol 인 MGCP/MEGACO, SIP등은 여러 표준화 단체에서 진행 중이며 차세대 이동망인 3GPP, 3GPP2는 물론 차세대 통신망(NGN, next generation network)에서도 활발히 연구되고 구현 단계에 이르렀다.[5][6][7]

본 논문은 IP와 ATM 네트워크를 기반으로 하는 차세대 통신망에서 VoIP/VoATM 서비스를 제공하는 Media Gateway 시스템과 MGC에 사용하는 MGCP/MEGACO 프로토콜의 기본 개념

과 차세대 핵심망 요소인 소프트웨어에 대하여 고찰한다. 2절에서는 MGCP프로토콜의 망구성과 VoIP 서비스를 제공할 수 있는 게이트웨이 종류 및 기본호 흐름에 대하여 살펴보고, 3절에서는 MEGACO 프로토콜의 기본 개념과 망 구성에 대하여 고찰하고, 4절에서는 소프트웨어가 요구되는 차세대 통신망 구성도, 소프트웨어의 구조와 소프트웨어 API, 응용 서버와의 관계를 정립하고, 마지막으로 결론을 맺는다.

II. MGCP

1. MGCP 개념

MGCP는 네트워크 상의 각각의 게이트웨이를 외부에서 제어하기 위한 프로토콜로서 1999년 IETF에서 정의하였다. MEGACO WG은 VoIP 서비스를 위한 Network 구성에서 미디어 게이트웨이를 외부 요소인 MGC(Media Gateway Controller)로 제어할 수 있는 구조와 요구사항에 대하여 기술하였다.[8][9] 또한 ITU-T SG16과 함께 H.248 표준문서를 공동으로 작업하고 있다.[10] MGCP/MEGACO 표준문서는 게이트웨이의 종류, MG를

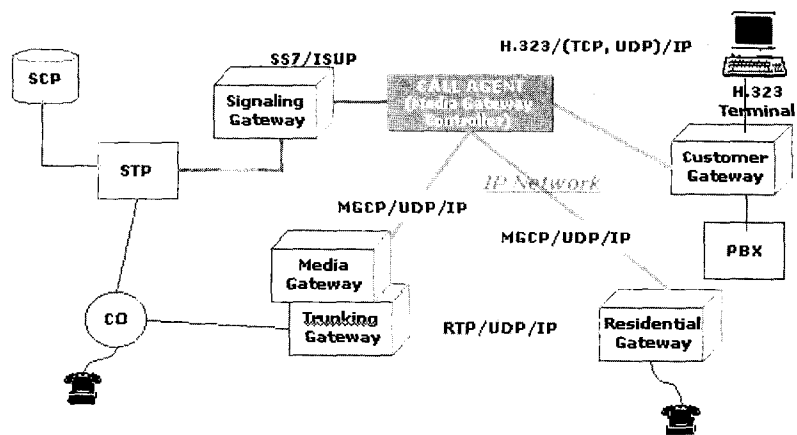


그림 1. MGCP 네트워크 구성도

제어하는 MGC, 패킷망 접속, 회선 교환망의 다양한 신호체계와의 접속 등 VoIP 서비스를 제공할 수 있는 시스템 체계를 기술하고 있다.(그림 1).

그림 1.에서 살펴보면 미디어 게이트웨이는 회선 교환망의 정보와 인터넷 또는 IP망 상에서 운반되는 데이터 패킷을 서로 변환하는 망요소이며, MGC는 지능적인 호 제어를 하기 위해 게이트웨이 밖에서 여러 개의 게이트웨이를 제어하는 망요소로서 MG내의 미디어 채널(음성 또는 데이터)에 관련된 호 상태를 제어한다. MG의 종류로는 사설 교환망에 접속된 사용자 VoIP 망에 접근할 수 있도록 지원하는 Access Gateway, 회선 교환망과 VoIP 패킷 교환망과 접속하는 Trunking Gateway, SS7 또는 R2 등 신호링크를 중단할 수 있는 Signaling Gateway, 아날로그 회선을 패킷망에 접근할 수 있는 Residential Gateway 등이 있다.[6][7].

MGCP는 H.323과 달리 MG나 Endpoint에 대해 단지 미디어 스트림을 처리하기 위한 역할을 담당하도록 하며, 이들간의 Call Control을 위한 Intelligence는 MGC(Media Gateway Controller)에서 관리하도록 정의하고 있다. 따라서 MGC는 MG를 관리하기 위한 외부 서버에 해당하며, 자신이 관리해야 할 MG 및 Endpoint들의

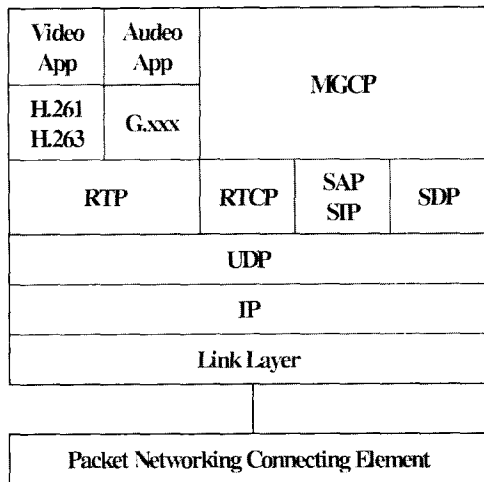


그림 2. MGCP Protocol Stack

구성에 대한 정보와 함께 각 구성요소가 지닌 상태를 지속적으로 유지함으로써 MG와 Endpoint에 대한 행동을 지시하고 관리할 수 있게 된다.

MGCP는 MGC와 MG간의 Call Control Signaling을 위한 명령어만을 정의하고 있다. MG 간에 실제 음성통신을 위해 이루어지는 오디오 스트림에 대해서는 RTP와 SDP를 부가적으로 이용하고 있으며, MG와 Endpoint간에 이루어지는 Event와 Signaling의 처리에 대해서도 정의하지 않고 있다. 다음은 MGCP Protocol의 구조에 관해 도식화하고 있다(그림 2).

2. MGCP 호 흐름(Call Flow)

MGCP는 호 제어를 위해 명령어가 이용되며 호 제어를 위한 명령어는 MGC와 MG사이의 Signaling에만 국한된다. MGCP 프로토콜에서의 명령어는 RQNT (Notification Request), NTFY(Notify), CRCX (Create Connection), DLCX(Delete Connection), MDCX(Modify Connection, AUEP (Audit Endpoint), AUCX(Audit Connection), RSIP(Restart In Progress)등이 있다.

MGCP에서는 기본적인 Call Flow를 처리하기 위해 위에서 정의한 명령어를 다음과 같은 순서로 이용한다(그림 3).

MGC는 자신의 관리영역에 있는 MG에 대해 해당 MG와 연결되어 있는 Endpoint의 Off-hook Event를 받아들일 수 있도록 RQNT(1)를 전달한다. RQNT(1)를 받은 MG는 자신에게 연결된 Endpoint로부터 Off-hook이 발생하면 이를 파라미터와 함께 NTFY(2)를 이용하여 MGC로 전달한다.

Off-hook이 발생한 후 MGC는 Endpoint에게 RTP를 준비하도록 하는 CRCX(3)를 MG에게 보낸다. 이 때 MGC는 Endpoint로부터 Digit

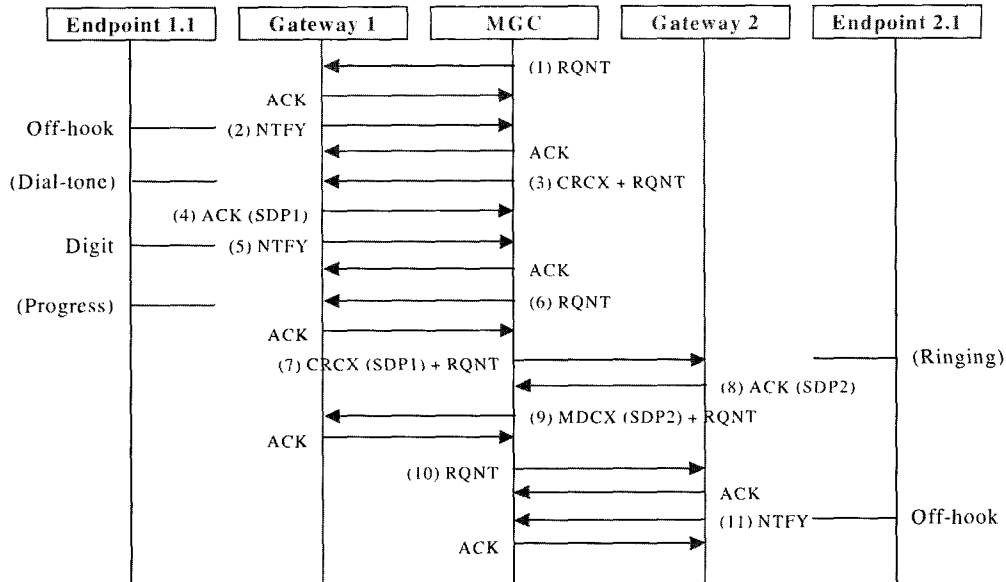


그림 3. MGCP Call Flow

Event의 발생을 전달 받기 위해 RQNT에 Digit Map을 포함하여 전달하며, 역시 RQNT를 통해 Dial-tone Signal을 Endpoint로 보낼 수 있도록 한다. 해당 Endpoint에서 RTP Session을 준비 하면 MG는 CRCX에 대한 ACK(4)상에 RTP Session에 대한 파라미터를 포함하여 MGC로 전달 한다. RTP Session에 대한 파라미터는 SDP로 기술한다.

Endpoint로부터 Digit이 발생하면 MG에서는 Digit Map을 이용하여 저장하고, Digit Map에서 정의한 규칙에 일치하는 번호가 발생했을 때 NTFY(5)를 이용하여 MGC로 해당 번호를 전달한다. Destination Endpoint의 번호를 받은 MGC는 Source Endpoint에 대해 RQNT(6)를 이용하여 Progress tone에 대한 Signal을 발생시키도록 MG에 지시한다. 그 후 MGC는 Destination Endpoint가 RTP Session을 준비할 수 있도록 MG에 대해 CRCX(7)를 전달한다. 이 때 MGC는 ACK(4)에서 받은 SDP를 함께 전달하여 Destination Endpoint가 RTP 연결을 수용할 수

있는지 판단할 수 있도록 한다. Destination Endpoint에서 RTP Session의 준비가 끝난 후 MG는 해당 RTP Session의 파라미터를 ACK(8)상에 포함하여 MGC로 전달한다.

Destination Endpoint로부터 SDP를 받은 MGC는 양단의 Endpoint의 RTP Session Parameter에 대한 Negotiation 작업을 수행한다. 그리고 수정된 RTP Session Parameter를 MDCX(9)를 이용하여 Source Endpoint로 전달 하여 RTP Session을 열도록 지시한다. MGCP는 이와 같은 일련의 과정을 통해 호 흐름을 제어할 수 있다.

III. MEGACO/H.248

MEGACO/H.248 표준은 여러 기관에서 공동으로 표준화 작업을 진행 중이며, IETF 또는 ETSI, MSF(Multiservice Switching Forum), ITU-T SG16에서는 프로토콜 요구 사항들을 제안

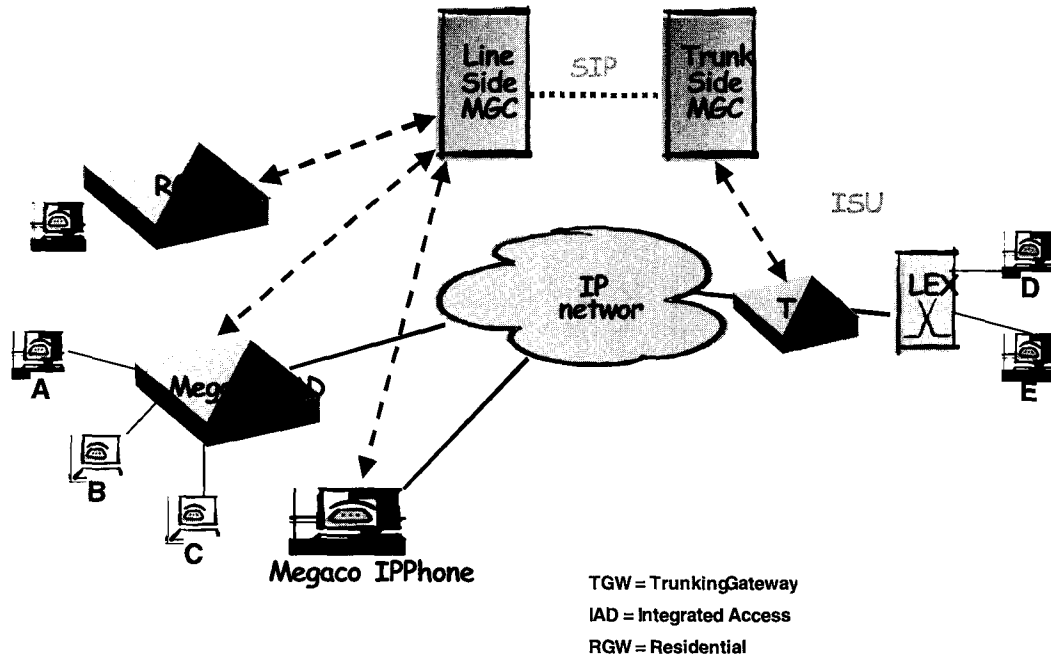


그림 4. MEGACO/H.248 네트워크 구성도

하고 있다.[10][11]

MEGACO/H.248은 MGCP 표준과 많은 공통점을 가지고 있으며 Media Gateway 또는 Media Gateway Controller는 근본적으로 같은 개념으로 사용하고 있다.[8][9] 그러나 Protocol 구현 방법이나 Media Gateway를 구성하는 요소들은 차이점을 보이고 있다. 그림 [4]는 MEGACO/H.248 프로토콜을 사용하여 통신망을 구성한 한 예이다.

1. MEGACO 프로토콜 구성

MEGACO의 Connection model은 Media Gateway Controller에 의해 제어되는 Media Gateway 내의 논리적인 entity 또는 object로 기술하고 Media Gateway를 Termination과 Context로 구분한다. Termination은 media stream parameter 와 bearer parameter를 가지고 있으며 stream의 생성과 소멸에 관여한다.

Context는 termination의 집합과의 관계를 나타내며, 서비스가 없는 채널과 같이 모든 termination과 상관 관계가 없을 때 Null context로 정의한다. 그림 [5]는 Media Gateway내의 Connection Model을 나타내고 있다.

가. Context

Context는 MG내의 Termination간의 논리적인 연결구성 요소이며, Termination들간의 Topology나 Media Mixing, Switching Parameter등 Context내에서 Termination들 사이의 관계를 나타낸다. Context는 여러 Termination들간의 연결에 이용되므로 두개 이상의 Termination들을 포함하고, Event의 발생 여부를 검출하거나 연결된 망을 지원하기 위한 Null Context의 경우는 하나의 Termination을 포함한다.

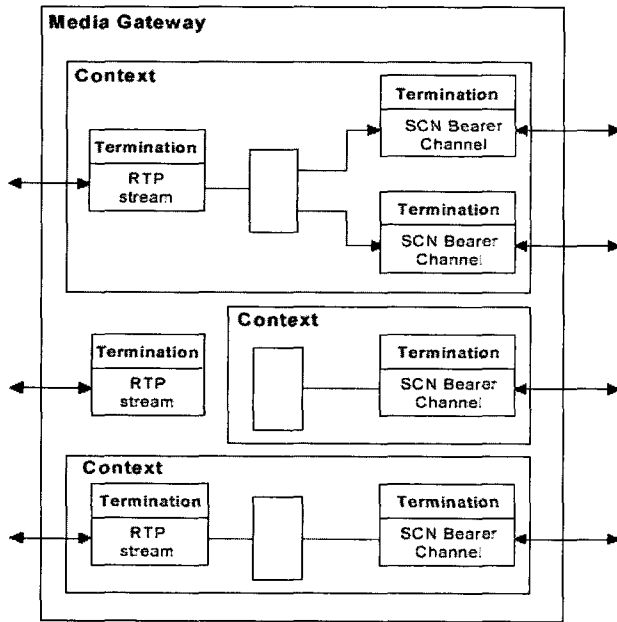


그림 5. MEGACO Connection Model

다. Add 명령어는 Termination을 Context에 연결시키기 위해서 Add 명령어가 특정 Context ID를 가리킬 경우 지정된 Context로 Termination을 연결시키고, 그렇지 않은 경우에 MG는 새로운 Context를 만들어서 Termination을 연결한다. 반대로 Context에서 Termination을 삭제하거나 연결을 끝낼 때는 Subtract Command가 이용되며 다른 Context로 이동시킬 때는 Move Command를 이용한다.

나. Termination

Termination은 MG내에서 Media Stream, Control Stream들을 생성하고 전송하는 논리적인 Entity이다. Termination은 명령어 내에 포함되어 있는 descriptor에 의해 성격이 규정되고, 생성시 MG로부터 할당된 Termination ID를 갖으며 Termination의 종류는 크게 두 가지로 구분된다.

Physical entity를 나타내는 TDM 채널과 같은 permanent Termination은 반영구적으로 게이트웨이에 존재하고, 이 Termination은 MG가 MGC에 등록시 생성되어 계속 Null Context내에 존재한다. Ephemeral Termination은 RTP flow와 같은 서비스를 지원하는 기간에만 존재한다. 이는 Add 명령어에 의해 생성되어 Context에 연결되고 Subtract 명령어에 의해 사라진다.

Termination은 MGC에 Noti 메시지를 기동하는 사전 즉 Event를 검출하거나 MG에 의해 Action이 일어난 경우를 알 수 있다. 또한 Call 성립 후 이용된 서비스의 Statistics 데이터들은 Termination에서 모아져서 Call이 끝날 때 MGC에 보고되어 Billing 정보 등으로 이용되며, 그

외에 MGC에서 AuditValue Command를 보내서 Statistics 데이터를 요구할 때도 MGC로 보고된다. 생성된 Termination의 Property는 MGC에서 보낸 Modify Command에 의해 계속 변경될 수 있으므로 MGC에서는 MG가 검출해야 할 Event들이나 수행해야 할 Signal들을 MG에 지시한다. Termination ID는 MG에서 선택되며 ALL과 CHOOSE 두 가지 종류의 Wildcard가 이용될 수 있다. ALL Wildcard는 Multiple Termination들을 한번에 지정할 때 이용되며 CHOOSE Wildcard의 경우는 주어진 조건을 만족하는 Termination ID 중 하나를 선택할 때 이용된다. 이 경우 MGC에서는 MG에 한 Trunk Group내의 한 Circuit을 선택한다. "Root"는 특수 Termination의 일종이며 패키지로 정의되고 특성(Property), 사건(Event), 시그널(Signal)과 통계(statistics)를 갖고 있다.

다음의 [그림 6]은 Context내에서 Call Waiting 수행을 위한 시나리오를 나타낸 것이다; 현재

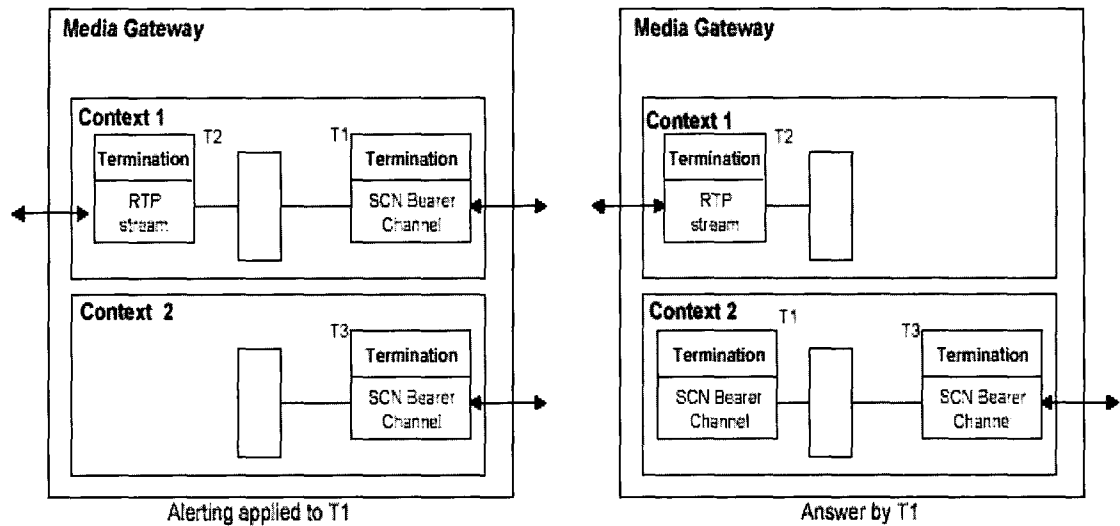


그림 6. Call Waiting Scenario

T1과 T2가 Context 1에 연결되어 있으며, T1과 연결을 요청하는 T3가 T1의 응답을 기다리고 있다. T1이 T2와의 연결을 잠시 보류하고 T3에 응답한다면 T1은 Move 명령어에 의해 Context 1에서 Context 2로 옮겨지고 T3와의 호가 이루어진다.

2. MGCP 명령어 체계

MGC와 MG간의 메시지를 교환하는 방법은 명령어로 이루어지는데 명령어들은 Transaction내에 그룹으로 나누어지고 각각의 Transaction는 Transaction ID로 구분된다. Transaction은 한 개 이상의 Action들로 구성되어 있고, 또한 Action은 여러 명령어들로 이루어져 있다. 각각의 Action은 한 개의 Context내에 제한적인 동작을 수행하는 명령어의 결합으로 이루어져 있으므로 각각의 Action은 ContextID를 언급해야 한다. 만약

ContextID를 제공 받지 못한 Action은 Context 밖으로 Termination이 조정되거나 MGC가 신규 Context를 생성하도록 MG에 요구한 경우이다. 아

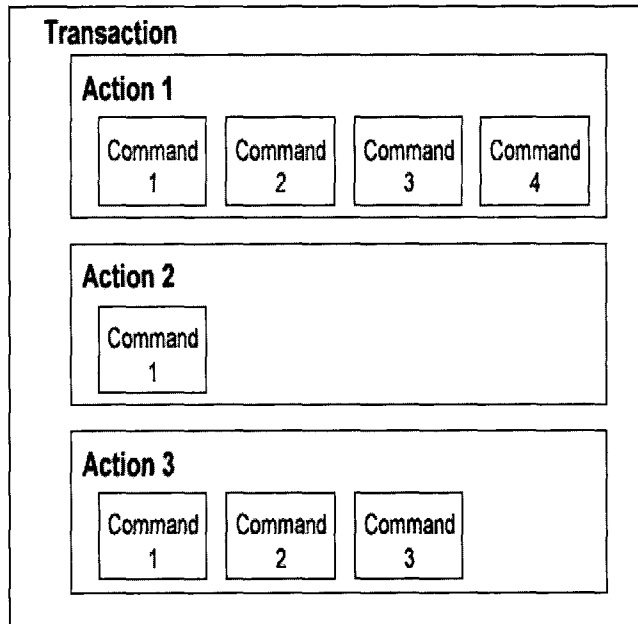


그림 7. Transaction, Action, Command의 구성

래 그림 (7)은 Transaction의 구성을 나타내고 있다.

Transaction은 Transaction Request의 형태로 보내지고 처리 후에는 Transaction Reply로 응답된다. Transaction내에 포함된 명령어는 순서대로 처리되도록 보장되며, 명령어 처리 중 문제가 발생되면 Transaction내에서 이후의 명령어 처리는 중단한다. 명령어가 Wildcard Termination ID를 포함하고 있으면 Wildcard 와 일치하는 모든 Termination을 시도하고, 만약에 일치하는 Termination ID가 포함될 경우에는 Termination 응답을 해야 한다. 마찬가지로 Termination 수행시에 에러가 발생한 경우는 다음의 명령어는 진행되지 않는다. 명령어가 성공적으로 수행된 경우는 Return 값을 전달하고 실패한 경우는 명령어 에러 Descriptor로 나타낸다.

가. Transaction API

TransactionRequest는 송신자에 의해 호출된다. 각각의 요청(Request)은 한 개의 Transaction을 할당하며 여러 개의 Action을 포함하고 각각의 Action은 특정 Context나 Context당 한 개 이상의 명령어를 언급한다. Transaction ID는 수신자의 Transaction Reply 또는 Transaction Pending 의 값을 설정해야 한다. Transaction Reply는 수신측에서 보내고 한 개의 Transaction 당 한 개의 응답(Reply) 소환이 수행된다. 또한 응답(Reply)도 특정 Context와 Context당 여러 개의 반응(Response)을 언급하는 Action들로 구성되어 있다 명령어 수행중에 에러가 발생하였거나 응답중에 에러가 발생한 경우는 그 이후의 처리는 중단된다. Reply의 Transaction ID는 수신했던 Request 메시지의 ID와 동일해야 하며, 각각의 Response Parameter들은 처리된 명령어에 대한 리턴값이나 에러 발생시에는 Error Descriptor를

나타낸다.

TransactionPending은 수신자에게 호출되고 사실상 처리 중이나 완성되지 않은 경우이며 송신자가 TransactionRequest를 상실하는 것을 방지해 준다.

```
TransactionRequest(TransactionID {
    ContextID { Command_Command},
    ContextID { Command_Command} )

TransactionReply(TransactionID {
    ContextID { Response_Response},
    ContextID { Response_Response} } )
```

나. Command

명령어(Command)는 프로토콜 Connection model의 Context나 Termination 등과 같은 논리적인 Entity를 조작한다. 명령어는 Termination을 Context에 추가하거나, Termination을 수정하거나, Context에서 Termination의 연결을 제거하거나 Termination과 Context 의 특성들을 감사(Audit) 한다. 그래서 명령어는 Context와 Termination의 Property들을 완전하게 제어할 수 있다. 대부분의 Command들은 MGC에서 MG를 제어하는데 이용되며 Notify 명령어는 MG에서 MGC로, ServiceChange는 Termination이 서비스를 완료했거나 서비스를 벗어났을때 MG가 MGC에게 알려준다. MEGACO 프로토콜에는 ADD, Modify, Move 등 8 종류의 명령어가 있으며 모든 서비스의 수행은 명령어의 조합에 의해 이루어진다.

각각의 명령어에 파라미터를 Descriptor로 명명하고 이것은 이름과 항목의 나열로 구성한다. 많은 명령어들은 공통적인 Descriptor를 공유하고 있으며 명령어 파라미터는 여러 개의 descriptor로 구성되어 있다. Descriptor의 텍스트 형식은 다음과 같

이 표현 된다.

```
DescriptorName=<someID>
  {param=value, param=value_}.
MUX = H.221(MyT3/1/2, MyT3/2/13, Myt3/3.6.)
```

Parameter값들은 Fully-Specified, Under-Specified, Over-Specified의 형태로 나타낼 수 있다. Fully-Specified Parameter의 경우는 한 개의 값을 지정하여 명령어 응답자에게 특정 파라미터를 사용하도록 한다. Under-Specified는 명령어 응답자에게 "Choose Value"을 선택하게 하며, Over-Specified Parameter의 경우는 가능성이 있는 값을 제안하여 그 중 상대측에서 원하는 Parameter를 선택할 수 있도록 한다.

MEGACO 프로토콜의 구성 요소들은 앞서 언급한 요소들 이외에도 Termination의 Property, Event, Signal, Statistics등을 나타내는 패키지 정의, 그리고 MG의 Cold Start로 시작하여 MGC

에 ServiceChange 메시지를 등록하는 제어 접속 기능 등이 있다. 또한 Transport 요소는 MG와 MGC간에 신뢰성 있는 Transaction를 전달하기 위하여 UDP 또는 TCP를 사용하는데, MGC는 TCP 와 UDP를 동시에 지원해야 하며 MG는 TCP 또는 UDP 또는 TCP/UDP를 동시에 지원할 수 있다.

IV. 소프트웨어

현재의 통신망은 1960년대 이후에 공중 통신망을 대표하는 회선교환 방식에서 패킷 교환 방식으로, 패쇄형에서 개방형으로, 음성, 데이터와 비디오 각각을 전송하는 전용망에서 음성, 비디오와 데이터가 함께 수용하는 통합망(converged network)으로 발전하고 있다.[2][10] 그림 [8]과 같이 향후 차세대 통신망은 미디어 전송과 시그널링, 호 제어, 응용 프로그램이 분리되어 지원하는 개방형, 분산형 구조로 전

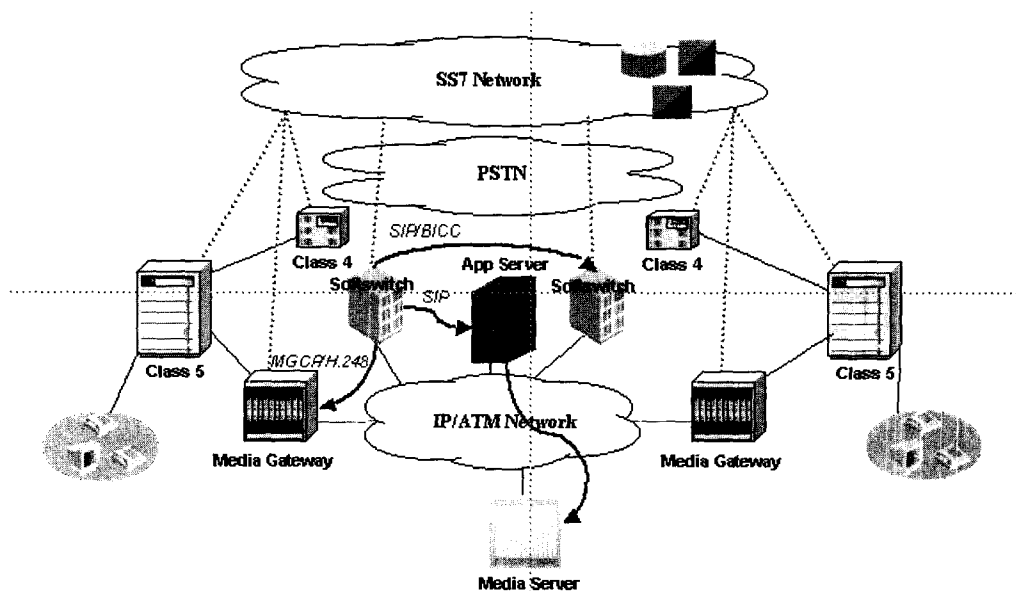


그림 8. 소프트웨어 망 구성도

환될 것으로 예측하고 있다. 차세대 통신망에서의 개방형 구조는 한 노드 당 수많은 가입자를 수용할 수 있으므로 통신사업자에게 네트워크의 효율성과 확장성을 제공한다. 차세대 통신망에서는 공중 전화망과 패킷망과의 음성, 비디오와 데이터의 미디어 변환은 미디어 게이트웨이(media gateway, MG), 시그널링(SS7, ISUP, INAP등) 변환은 시그널링 게이트웨이(signaling gateway, SG), 이들과의 호 제어 및 가입자의 시그널링 제어는 미디어 게이트웨이 제어기(media gateway controller, MGC)에 의해 이루어지고 또한 서비스와 특징 서비스(Features)는 IP망에 접속되어 있는 응용 서버(Application Server)에 의해 제공된다. 이와 같이 MG는 서로 다른 네트워크간의 다양한 서비스를 처리하기 위해 미디어 변환기능을 수행하고, MGC 또는 소프트스위치는 여러 종류의 프로토콜을 지원하고 시그널링을 제어하는 기능을 수행한다.

그림 [8]처럼 차세대 통신망의 핵심 요소인 소프트스위치는 Call Agent, Call Server 또는 MGC(media gateway controller)등으로 불려지고 있으며 다음과 같은 기능을 제공해야 한다고 규정하고 있다. [2][5][7]

- 다양한 미디어 게이트웨이 또는 IP 종단점(endpoint)에 대해 호 제어를 제공하며 종단간 RTP connection을 제공한다.
- 각 호에 적절한 서비스 및 부가 서비스의 처리를 선택할 수 있다.
- 시그널링과 가입자 정보를 근거로 네트워크내에 호 라우팅(call routing)을 제공한다.
- 호 제어 기능을 타 네트워크로 전환할 수 있는 능력을 갖는다.
- 과금, 장애, 네트워크 관리 등 망 운용에 필요한 기능과 접점을 제공한다.
- 통신 네트워크에 확장성과 신뢰성을 제공한다.
- 다양한 프로토콜(MGCP, H.248, SIP, H.323, ISUP, SS7등)을 지원한다.

소프트스위치는 차세대 통신망에서 호 제어, 프로토콜 변환등 중추적인 역할을 수행함과 동시에, IP 네트워크상에 접속되어 있는 여러 망 요소들과 결합하여 기존의 서비스는 물론 다양한 형태의 새로운 서비스들을 제공하게 된다. 이와 같은 개방형 구조와 API 형태는 장비 업체, 응용 소프트웨어 업체와 통신사업자들에게 진입 장벽을 낮추고 경쟁적으로 Solution을 제공함으로써 차세대 통신망은 융통성과 신뢰성이 보장될 것이다.

소프트스위치는 차세대 통신망에서 전략적인 망 요소로서의 역할을 수행한다. 그래서 차세대 망에 진입하려는 장비 업체들은 소프트스위치의 Solution 확보나 소프트스위치와의 연동을 위한 연합을 구성하고 있다. 또한 많은 업체와 표준화 단체는 자신들의 연구한 결과를 토대로 소프트스위치의 구조를 제안하고 있으며 특히 MSF(multiservice switching forum)이 제안한 Multiservice 스위치의 구조가 개방형, 분산형이고 여러 통신 프로토콜을 수용하고 있는 ISC (international softswitch consortium)의 소프트스위치 구조와 유사하다.[6][7]

ISC내의 Architecture WG는 소프트스위치의 구조를 그림 (9)와 같이 나타내고 있다. 즉 소프트스위치의 구조는 신호(signaling)의 물리적인 종단, 스위칭 기능을 담당하는 전달 평면, 프로토콜 또는 신호 처리(signaling processing) 기능, 베어러 연결 제어 기능, 게이트 키퍼/프락시 호처리 기능들을 담당하는 제어 평면, 세션 제어와 서비스 논리 기능, 변환과 라우팅 기능들을 담당하는 응용 평면, 가입자의 자료 관리와 자료 저장, 과금등 관리하는 데이터 평면, 마지막으로 네트워크 운영 및 제어, 네트워크 관리(administration), 통계 및 사용자 감시와 장애 경고를 수행하는 네트워크 운용(management)을 수행하는 관리 평면으로 구성되어 있다. 각 평면내의 여러 기능들이 타 평면과의 기능별 연관 관계는 참조점으로 정의하고 있으며 각 평면 내에서 수행해야 할 기능과 필요한 프로토콜도

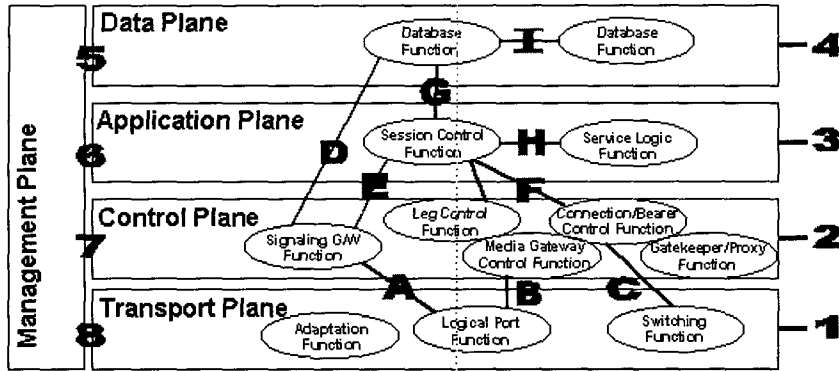


그림 9. 소프트웨어 스위치의 구조 및 참조 기능

언급하고 있다.[10]

소프트스위치는 소프트웨어 API(application program interface)를 통하여 응용 서버(Application Server)에 존재하는 서비스를 제공 받을 수 있으며 향후의 신규 서비스도 다양한 형태의 응용 서버로 제공 받을 수 있다. 소프트웨어 API는 여러 가지의 특징을 가지고 있다. 첫째는 소프트웨어의 모듈과는 독립적인 서비스 구조를 가지고 있으며 원격 데이터 서버와의 상호 작용을 담당하는 근접 응용 서버를 지원해야 한다. 둘째는 원격 서버와 정합을 위한 다양한 프로토콜이 사용될 것이며 번호 체계, 과금, CDR(call routing record), 사용자 인증과 대역폭 할당 등을 지원한다. 셋째는 호의 연결, 사건의 통지, 연결 중단 및 응용 모듈 등록이나 해제 등도 지원한다.

한편 응용 서버는 다양한 프로토콜(AIN/IN, JAIN, CPL, PARLAY등)을 수용하고 있으며 SIP 프로토콜로 소프트웨어와 통신하며 고급 서비스(enhanced service)의 수행과 관리를 담당하는 플랫폼이다. 이는 서비스 생성과 구현을 위한 API를 제공하며 5 가지의 요구 사항을 만족해야 한다. 1)응용 서버와 네트워크 구조와 분리, 2)신호 프로토콜(SS7, ISDN, CAS, H.323, SIP) 또는

MG 제어(MGCP, H.248) 과의 분리 3) 동시 통화 지원 4) 응용 서버에게 Directory Number를 사용하는 호 가입자의 정보 통지 5)NANP/E.164를 사용하는 호와 콜레그(Call Leg)를 언급하여 소프트웨어가 호 라우팅을 할 수 있도록 해야한다.

또한 응용 서버는 미디어 서버와 연동하여 미디어 자원을 가입자가 요구했을때 제공 가능하도록 해야 하며, 미디어 서버는 회의통화, IVR, TTS, FAX 등의 자원을 수용하여 RTP로 MG에 공급한다. 이 상과 같이 소프트웨어는 다양한 서비스를 제공하는 서버들과 연동하여 가입자가 원하는 서비스를 지원할 수 있다.

V. 결론

인터넷의 급속한 성장은 통신, 방송, 출판 등 여러 분야에서 변화를 일으켰고, 특히 음성통신 시장의 새로운 변혁을 가져왔다. 현재 데이터 통신사업자가 데이터 통신망을 통하여 얻는 수입(minute revenue)은 음성통신 시장의 수입의 10분의 1에 그치고 있지만, 음성과 비디오, 데이터가 통합 운영되는 차세대 통신망에서는 10년 이내에 동등한 수준

또는 그 이상으로 증가할 것으로 예측하고 있다. 이에 따라 전통적인 음성 통신이 유선망 또는 이동망에서 벗어나 All-IP 네트워크로 전환하는 차세대 통신망의 구조와 운용에 대하여 활발히 연구되고 있으며, 부분적인 시범 서비스도 이루어지고 있다. 본 논문에서는 차세대 통신망에서 VoIP/VoATM 서비스를 제공할 수 있는 MGCP 와 MEGACO 프로토콜에 대하여 권고안을 중심으로 각각의 프로토콜 구성과 기능 및 일반 호 처리 절차에 대하여 살펴보았으며, 차세대 통신망에서 중추적인 역할을 수행하는 소프트웨어의 구조와, 신규 서비스를 제공하는데 필요한 소프트웨어 API가 가지고 있는 여러 특성, 그리고 이와 같은 서비스를 제공 가능하게 하는 응용 서버와 미디어 서버에 대하여도 고찰하였다. 향후에는 차세대 통신망에서 여러 형태의 통신망 요소들이 연동할 때 발생할 수 있는 QoS 문제와 연동시의 각 통신망 요소간의 요구 사항에 대하여 연구가 필요하다.

참고문헌

- [1] Ovum, "IP Telephony: Exploiting Market Opportunities", Peter Hall, 2001/4/20
- [2] 유승화, "인터넷 전화", 전자 신문사, 2002
- [3] 박銀娥, "인터넷 전화의 발전과 공중통신 사업자들의 대응", 격주간통신정책동향 10권16호, pp 1~28, 1996
- [4] 김영환, "VOIP를 위한 QoS기술", KRNet 2000, 8차 Computer Networking Conference (<http://www.krnet.or.kr>), pp 253~259, 2000
- [5] 강태익, "Softswitch and MGCP", KRNet 2000, 8차 Computer Networking Conference (<http://www.krnet.or.kr>), pp 260~263, 2000
- [6] 강태익, "MGCP 기반 VoIP 시스템", VoIP Forum Network Solution 세미나 (<http://www.knra.or.kr>), pp 41~68, 2000
- [7] 강태익, "MGCP/MEGACO 표준기술", VoIP Forum 표준기술 Workshop, pp 133~150, 2000
- [8] <http://www.ietf.org/rfc/rfc2805/>
- [9] <http://www.ietf.org/html.charters/megaco-charter.html>
- [10] <http://www.softswitch.org>
- [11] <http://www.msforum.org>
- [12] Ramnath A. Lakshmi-ratan, "The Lucent technologies Softswitch-realizing the promise of convergence", Bell Labs Technical Journal, April-June, 1999.



강 태 익

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 책임연구원(실장) <주요 관심 분야> VOIP, 음성/영상 신호처리, ATM, 차세대 지능망