

MEGACO-m을 이용한 MG 자원 탐색 메커니즘

엄남경^{U*}, 우성희^{**}, 이지인^{*}, 이상호^{*}
충북대학교 전자계산학과^{*}, 청주과학대학 컴퓨터학과^{**}
{family8, jilee, shlee}@cnlab.chungbuk.ac.kr

Media Gateway Resources Discovery Mechanism
Using MEGACO-m

Um Nam-Kyoung^{U*}, Woo Sung-Hee^{**}, Lee Ji-in^{*}, Lee Sang-Ho^{*}
{family8, jilee, shlee}@cnlab.chungbuk.ac.kr

Dept. of Computer Science, Chungbuk National University^{*},
Dept. of Computer Science, Chongju National Science University^{**}

요약

인터넷 서비스의 폭발적인 수요 증가에 따라 데이터망의 구축에 대한 투자의 필요성이 제기되어, 여러 벤더들이 최적의 망을 구축할 수 있는 플러그인 형태의 개방형 멀티미디어 망 구조에 대한 연구가 활발히 진행되고 있다. 개방형 멀티미디어 망에서는 노드간에 인터페이스로, 표준화된 프로토콜을 구현함으로써 기존의 서비스에 대한 지원 및 신규 서비스의 유연한 수용이 가능하게 된다. 이 논문에서는 기존의 미디어게이트웨이(MG)와 미디어게이트웨이제어기(MGC) 사이의 상호 운용성을 위한 MEGACO 프로토콜을, 미비했던 자원 탐색 및 관리, MG 측의 자원 보고 기능을 추가시켜 MEGACO-m(MEDIA Gateway Control-management)을 제안한다.

1. 서론

개방형 멀티미디어 통합 시스템의 도입은 기존의 서비스의 유연한 수용 및 신규 서비스를 지원하기 위해 필수적이다. 이러한 시스템을 실현하기 위해서는 서로 다른 망을 연동시키기 위한 게이트웨이(Gateway)도 스위칭 기능에서 제어 기능을 분리해 내기 위해 게이트웨이와 게이트웨이 제어기로 분리한다[1]. 다양한 타입의 미디어를 지원할 수 있는 미디어 게이트웨이는 MG(Media Gateway)로 정의되며, 미디어 게이트웨이를 제어하는 장치는 MGC(Media Gateway Controller)라고 정의한다[2]. MG와 MGC 간의 제어는 IETF에서 "RFC3015"에 표준으로 정의되어 있는 "MEGACO(MEDIA Gateway Controller)" 프로토콜을 적용한다. 그러나, 현재 표준으로 나와 있는 "MEGACO" 프로토콜은 MGC에게 MG내의 자원을 보고하는 자원 보고 기능이 미비하다. 따라서 MG의 형상 정보(Configuration information) 등을 MGC에게 알려주며, MG가 가지고 있는 변화하는 자원에 대한 일정한 자원 탐색 및 알림 메커니즘의 연구가 필수적이다.

이 논문에서는 기존의 MEGACO 프로토콜의

자원 관리 및 탐색 기능을 확장시킨 MEGACO-m(management) 프로토콜을 제안하고, 이 프로토콜을 이용한 간단한 자원 관리 흐름도를 기술한다.

이 논문의 구성은 다음과 같다. 2장에서는 기존의 MEGACO 프로토콜에 대해 알아보고, 3장에서는 MG의 자원관리 기능에 대한 요구사항 및 자원 항목을 기술한다. 4장에서는 필요 조건을 충족시키는 MEGACO-m 프로토콜을 제안하고, 이에 대한 간단한 예제를 보여준다. 5장에서는 결론 및 향후 연구 과제를 기술한다.

2. MEGACO 프로토콜

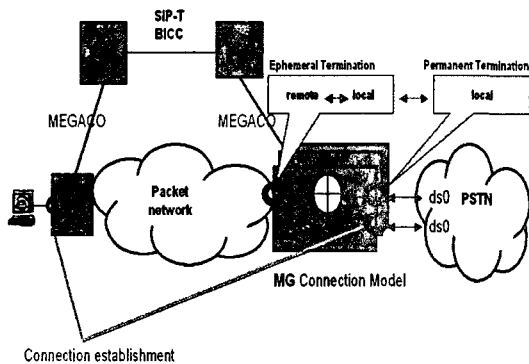
2.1 관련 연구 동향

MEGACO 프로토콜은 IETF에서 처음 시작되어 현재 ITU-T에서도 H.248와 같은 규격으로 표준화 작업중이며 H.323 게이트웨이를 MG와 MGC로 분할하려는 목적을 가지고 있다. IETF에서 MGCP(Media Gateway Control Protocol)로 시작되어 ASN.1/ABNF 등을 이용한 Binary/Text Encoding 방식으로 확장시키고, 단대단(end to

endpoint) 지원에서 단대다중점 (end to multiend -point) 지원으로 확장시켜 음성뿐 아니라 거의 모든 미디어 타입을 지원할 수 있도록 MEGACO (MEdia GATeway COntrol)로 발전시켜 현재 "RFC3015"문서로 규격이 나와 있다[2].

2.2 MG 연결 모델

MGC는 하나의 MG 내의 미디어 채널을 위해 커넥션 및 자원 관리 기능을 수행한다. MG는 미디어를 변환시키고, 관련된 기능들을 지원하는 역할을 한다. 특히 MEGACO는 MG 내에 연결 모델 (Connection Model)을 두며, 하나이상의 스트림을 고정시키거나 통과시키는 논리적인 지점을 나타내는 터미네이션 (Termination)과 그 집합의 연관관계로 나타내는 컨텍스트 (Context)로 구성된다. 또한 각 종단의 터미네이션을 연결시켜주는 위상 (Topology)로 나타낸다.



[그림 1] MG 연결모델 (Connection Model)

2.3 명령어 / 기술자 / 패키지

MEGACO는 MGC와 MG 사이를 [표 1]과 같은 명령어 (Command)로 제어한다. 명령어는 프로토콜 연결 모델의 논리적 요소의 조작 또는 컨텍스트에 터미네이션을 추가, 변경, 삭제하거나 터미네이션에서 다루어질 사건 (Event)들의 내용 및 행동, 사항을 표현한다. [표 1]은 MEGACO의 명령어의 종류를 나타낸다.

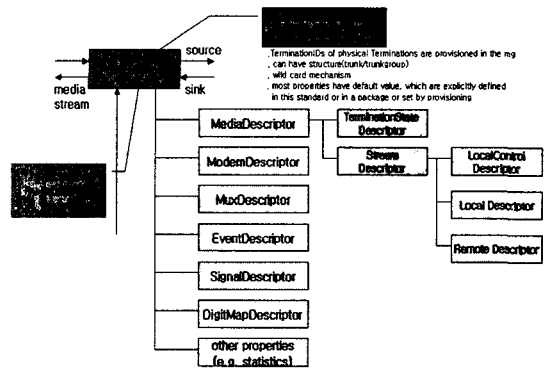
기술자 (Descriptor)는 명령어에 대한 파라미터 (parameter)가 된다. 기술자의 표현 방법은 아래와 같다.

DescriptorName=<someID>{parm=value,....,parm=value}

[그림 2]는 기술자의 종류를 도식화하여 표현하고 있다. 패키지 (Package)는 이와 같이 기술자 내에 실려보낼 수 있는 자원들의 Property, Signal, Event, Statistic과 같은 정보들을 그룹화하여 IANA (Internet Assigned Number Authority)에 등록한다.

표 1] MEGACO 명령어

명령어	설명	방향
Add	컨텍스트에 터미네이션 추가, 널(null) 컨텍스트라면 하나의 새로운 컨텍스트 생성	MGC -> MG
Modify	터미네이션의 속성, 이벤트, 신호 수정	MGC -> MG
Subtract	터미네이션의 연결 해제	MGC -> MG
Move	다른 컨텍스트로의 이동	MGC -> MG
Audit Value	현 상태의 터미네이션의 속성, 이벤트, 신호를 리턴	MGC -> MG
Audit Capability	MG에 의한 터미네이션 속성, 이벤트, 신호를 위한 모든 가능한 값을 리턴	MGC -> MG
Notify	MG에서 현재의 이벤트를 MGC에게 알리기 위함	MG -> MGC
ServiceChange	restart등의 명령을 알리기 위함	MGC->>MG



[그림 2] 기술자의 도식화

또한 하나의 컨텍스트 내에서 동작할 수 있는 명령어의 그룹단위를 액션(Action)이라고 하며, 트랜잭션(Transaction)은 하나이상의 액션을 보낼 수 있는 MG/MGC 간의 최소 메시지 단위이다.

메시지(Message)는 프로토콜 버전과 송신자의 네트워크 주소, 하나이상의 "TransactionRequest"를 포함하여 보낸다. 이에 대한 응답으로 메시지에 "TransactionReply", "TransactionPending"을 보낸다.

3. MG 자원 관리 기능

3.1 자원 탐색 메커니즘의 필요성

각각의 MG는 잠재적으로 하드웨어나 소프트웨어적으로 구현될 수 있는 미디어 자원들을 가진다. 미디어 자원은 MG가 물리적으로 지니고 있는 포트 등의 자원일 수도 있고, 일시적인 비영구적인 자원일 수 있다. 따라서 하나의 MGC는 MG의 집합들의 효과적인 제어를 위해 MG가 가진 자원에 대해 양질의 지식이 필요하다. 프로토콜 요구사항이란 MG가 어떤 자원을 가지고 있는지 MGC에게 발견 가능하게 하는 메커니즘에 대한 것이다. [표 2]는 기존에 연구되고 있는 MG/MGC 간의 자원 탐색 메커니즘에 대한 비교분석표이다[4][5].

표 2] SNMP vs. MEGACO 관리 기능

	SNMP	MEGACO
관리 기능	간격	단순
호처리 기능	무책	강력
처리방법	MG-Agent 로 이용, MG 측에 MIB module 등록하여 사용	기존의 존재하는 명령어는 그대로 유지하며, 기술자만 단지 확장
호처리 기능	무책	강력
제안	간단한 관리 기능만 제공할 경우, MEGACO 를 확장시킨 방법을 사용한다. 즉, ServiceChange 명령어를 시동 내에 사용하여 Profile 을 알려내고, Audit 명령어를 사용하여, MGC는 MG의 자원 정보를 요청한다. 관리기능을 강화할 경우, SNMP를 이용하는 방법을 제안.	

3.2 프로토콜 요구사항

MG를 제어하기 위한 프로토콜의 자원 관리 요구사항은 다음과 같다[3].

- ① MG는 MGC에게 자원에 관련된 터미네이션 정보 및 관리 액션 등에 관련된 이벤트 보고
- ② 터미네이션의 해제 및 관리상의 블록킹 지원
- ③ MGC가 MG를 제어하기 위한 메커니즘 제공
- ④ 기본적으로 터미네이션 이름과 할당된 값을 열거하여 MGC에게 보내는 “단순 메커니즘” 필요
- ⑤ MGC가 소모된 MG 자원에 대한 신뢰성 있는 추적 기법 제공
- ⑥ 컨텍스트 생성 시에 MG의 “Capability” 탐색

위의 여섯 가지 요구사항을 만족하기 위해서, MG는 시동하거나, 이벤트가 발생을 알리거나 또는 MGC 측에서 관리할 자원 정보를 MG에게 요청할 때, MGC에게 MG가 가진 자원 정보들을 송수신할 수 있는 메커니즘이 필요하다.

3.3 MG 자원 관리 항목

다음은 MG 내의 자원 관리 항목이다.

- ① 컨텍스트 정보
- ② 터미네이션 이름 및 할당 값
- ③ MG의 “Capability”
- ④ MG 내의 공통 자원(Common Resource) 정보 및 형상 정보(Configuration Info)
- ⑤ MG가 가지는 링크(Link) 정보 및 통계량
- ⑥ 그 밖의 일반적인 자원 정보

기존의 MEGACO 프로토콜의 핵심 개념은 연결모델의 터미네이션이다. 따라서 터미네이션에 대한 정보는 형상 정보를 포함하는 범위로 볼 수 있다. 컨텍스트는 터미네이션의 관계를 나타낸 것이며, MGC의 제어에 의한 것이므로, 선택 사항이다.

4. MEGACO-m 프로토콜

4.1 MEGACO-m 프로토콜 개요

기존의 MEGACO 프로토콜을 유지하고, 자원 관리 및 이벤트 보고 기능을 확장시킨 프로토콜을 MEGACO-m(management)이라 정의한다. 확장시킨 기능은 다음과 같다.

- ① MG 시동시 점유 자원 알림: “ServiceChange” 명령어 내의 파라미터 보완
- ② MGC가 자원 요청시 보고: “AuditValue” 명령어 내의 기술자 확장
- ③ 정기적인 자원 변화 보고: 패키지를 추가 등. 특히 Notify 명령어 내에 전송
- ④ MG/MGC 간의 관리 전용 기능: MGC에서 MG 측으로 보내는 “Manage” 명령어 추가

4.2 “ServiceChange” 내의 파라미터 보완 기능

“ServiceChange”명령어는 “ServiceChangeMethod”, “Reason”, “Delay”, “Address”, “Profile”, “Version”, “MGCI” 파라미터를 가질 수 있다. MG에 의해 지원되는 미디어 기능을 표현하기 위해서 기존의 “ServiceChange Profile”을 확장하여 MEGACO 가 MG를 어떤 형상으로 구현할 수 있는지 자원을 응답 메시지로 받는다.

4.3 “AuditValue” 명령어 내의 기술자 확장 기능

터미네이션이 가지는 공통 자원 (Common Resource)의 형상을 알기 위해서 AuditValue 명령어를 사용하여 MG의 내부 미디어 기능들을 밝혀내는 것이다. [표 3]에서는 이를 위해 논리적인 최소 모델 (Atomic Model)을 소개한다.

이 해결 방법을 위해서는 MediaFunctions 기술자와 MediaResources 기술자를 확장시킨다. 전자는 미디어 자원들의 정보를 요구할 수 있다. “Property”는 FunctionID, Type, Cardinality, Description, Condition 을 보낸다. FunctionID는 해당 기능의 ID이며, Type은 Atomic Model 의 4가지 기능을 표시한다. Cardinality는 MG에 의해 식별 가능한 자원의 수를 정수형으로 정의한다. Description 값으로는 기능들의 세부사항을 기술한다.

Direction은 Bothway, Oneway, Send/Receive 의 속성들을 갖는다. Condition 값으로는 기능을 구현하기 위한 필수적인 조건들을 ALL 값 등을 이용하여 표현한다.

표 3] 최소 모델

	Transport modification	Media Encoding modification	미디어 발/착신
1 TransportStream	√		
2 EncodingStream		√	
3 MixStream	√	√	
4 TerminateStream			√

1번과 4번을 위한 oneway / bothway, 혹은 4번을 위한 송수신 / bothway 등, 4가지 방식을 적절히 혼합하여 사용

4.4 "Notify" 명령어에 전송될 패키지 추가

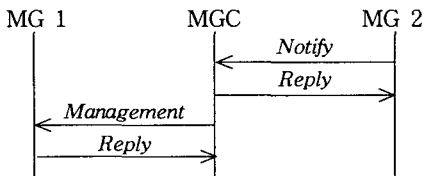
"Notify"는 MG에 이벤트가 생길 때, MGC로 보고하기 위한 명령어이다. MEGACO 프로토콜은 IANA에 등록된 패키지가 들어왔을 때, 이를 처리해주는 기능을 가진다. 따라서, "Notify" 명령어를 확장하기 위해서는 가능한 자원들을 패키지화시켜 등록시켜놓아야 한다.

4.5 "Management" 명령어 추가

"Management"는 MG의 발착신 포트(Port), MG와 MGC 사이의 링크 제어 및 통계량, 자원의 변화에 대한 정보 등을 관리할 수 있는 MGC측에서 MG로 보내는 명령어이다. MG는 이에 대한 응답으로 "Reply" 메시지에 요청한 정보들을 보낸다.

4.6 자원 관리 흐름도 예제

[그림 3]은 MGC와 MG1, MG2 사이에 확장된 MEGACO-m 프로토콜을 이용하여 자원을 요청하고 관리하는 간단한 흐름과 예제이다.



[그림 3] 자원 관리 흐름도

- ① MG2가 MGC에게 "Notify" 명령어를 보내고, MGC는 이에 대한 응답으로 "Reply" 메시지를 보낸다.

```

From MG2 to MGC :
MEGACO/1 MG2
Transaction = 10001 {
Context = - {
Notify = A4444 {ObservedEvents = 2222 {
19990729T22000000 : al/of}}
}
    
```

```

From MGC to MG2 :
MEGACO/1 MGC
Reply = 10001 {
Context = - { Notify = A4444}
}
    
```

- ② MGC는 MG1에게 수정된 자원들에 대한 정보를 요구한다. MG1은 이에 대한 응답으로 해당 자원들에 대한 정보를 실어 "Reply" 메시지를 보낸다.

```

From MG2 to MGC :
MEGACO/1 MG2
Transaction = 10002 {
    
```

```

Context = - {
Management = ROOT {[PortManagement],
[LinkInfo],[Statistic],[ModifiedResources]]}
    
```

```

From MGC to MG2 :
MEGACO/1 MGC
Reply = 10002 {
Context = - {[5555,5556], [], [], []}
}
    
```

5. 결론 및 향후 연구 과제

개방형 망 시스템을 구축하기 위해서 각 인터페이스 역할을 하는 프로토콜에 관한 연구가 활발히 진행되고 있으며, MEGACO-m 프로토콜은 MG와 MGC 간의 관리 기능을 강화한 차세대 제어 프로토콜로 개발 가능하다. MG의 자원을 탐색하는 방법은 IETF나 ITU-T의 규격으로 연구중이며, MG가 관리하기 위한 자원 및 관련 정보를 탐색하는 방법으로 SNMP, CMIP 과 같은 기존의 관리 프로토콜을 이용하여 제어할 수 있으나, MG/MGC 간에 제어 프로토콜인 MEGACO 프로토콜을 그대로 사용하면, 여기에 간단한 기능을 확장시킴으로서 관리가 가능하게 하는 것은 네트워크의 오버헤드를 줄이고, 호처리 기능을 하는 프로토콜을 유지시킴으로서 경제적인 면에서도 효과를 볼 수 있다.

이 논문에서는 기존의 MEGACO 명령어 및 명령어내의 기술자를 MGC가 MG의 자원을 탐색할 수 있는 기법에 대해서 알아보았다. 향후에는 MEGACO 프로토콜과 제안하는 MEGACO-m 프로토콜을 구현하여 비교 분석하여 효율성을 검증해내는 것이 연구 과제이다.

[참고문헌]

- [1] 이정규, 이순석, 김영부, 전경표, "개방형 멀티서비스 통합 교환 기술", 한국통신학회지 17권 2호, pp.153~166, 2000. 2.
- [2] F.Curve, N.Greene, A.Rayhan, "Megaco Protocol Version 1.0", RFC 3015
- [3] N. Greene, M. Ramalho, B. Rosen "Media Gateway Control Protocol Architecture and Requirements", RFC2805
- [4] T.Levy 외 2인, "Megaco/H.248 Media Gateway Resources Discovery", Internet Draft, 2000.11
- [5] SNMPv3, IEEE Communication 4th Quater 1998 Vol1 No1, William Stallings