

성과 유연성을 고려한 VOIP AGW 소프트웨어 설계 및 구현

고석갑^o 김영한

송실대학교 정보통신공학과 네트워크 연구실
{softgear^o, yhkim}@dcn.ssu.ac.kr

Design and Implementation of VOIP AGW Software for Performance & Flexibility

Seokkap Ko^o Younghan Kim

DCN Lab, Information-Telecommunication Engineering, Soongsil University

요 약

VOIP 액세스 게이트웨이는 원활한 VOIP 서비스를 위해 높은 호 처리 능력을 가져야 한다. 또한, 여러가지 VOIP 프로토콜을 지원 가능하여야 하고, 고품질 서비스를 위한 QoS 기능과 고신뢰성 및 고가용성을 가져야 하는 등 각종 요구사항을 만족하여야 한다.

본 논문에서는 일반적인 VOIP 액세스 게이트웨이의 요구사항을 살펴보고, 하드웨어 블록 구조를 설명한 후, 높은 호 처리 성능을 가지면서 확장 및 변경 가능한 소프트웨어 구조를 설계하고 구현하였다.

1. 서 론

현재 가장 많이 사용되는 VOIP 프로토콜로는 H.323[1], SIP[2], MGCP[3], Megaco(H.248)[4]가 있다. 일반적으로 VOIP 액세스 게이트웨이는 Firmware 변경 등을 통하여 여러 가지 VOIP 프로토콜을 지원하고 있다. VOIP 프로토콜 변경 및 하드웨어 변경에 따른 전체 소프트웨어의 변경을 최소화 할 수 있는 유연성 있는 소프트웨어 구조 설계가 필요하다. 또한, 액세스 게이트웨이는 고품질 서비스를 위한 QoS 기능과 고신뢰성 및 고가용성을 가져야 하는 등 여러 가지 구조적, 기능적, 성능적 요구사항을 가지고 있다. 특히 중용량 이상의 VOIP 게이트웨이에서는 높은 호 처리 능력이 요구된다.

본 논문에서는 VOIP 액세스 게이트웨이의 일반적인 요구사항을 살펴보고, 하드웨어 구조를 설계한 다음, 높은 성능을 가지면서도 확장 변경이 용이한 유연성 있는 VOIP 소프트웨어 구조를 설계한 후, 임베디드 리눅스 상에서 구현한다.

2. 요구사항

VOIP 액세스 게이트웨이의 요구사항은 일반적으로 구조적 요구사항, 기능적 요구사항, 그리고 성능적 요구사항으로 구성된다. 구조적 요구사항에는 시스템 형상 및 인터넷망과의 인터페이스, 전화망(POTS, ISDN 등)과의 인터페이스, 그리고 VOIP 프로토콜 등 VOIP 운용서버와의 연동을 위한 인터페이스, 시스템 관리를 위한 Management 인터페이스 등이 있다. 기능적 요구사항은 VOIP 서비스를 운용하는데 필요한 기능에 대한 것으로, 지원해야 하는 VOIP 프로토콜 및 관련 표준과 프로토콜 측면에서의 서비스 및 장비의 구조를 설명한다. 추가적으로 서버 이중화 및 등록 절차, 자동 설정 절차에 대한 내용이 있다. 성능 요구사항은 시스템의 호 처리 능력을 설명하는 것으로, 최대 포트 수, 음질 수준, 시간당 호 처리 능력(BHHCA), 동시 연결 가능 채널 수 등이 있다. 또한, 높은 신뢰성을 가진 서비스를 유지하기 위한 장비 이중화, 전원 이중화, 보호 복구 기능 등이 요구된다. 실제 VOIP 액세스 게이트웨이의 요구사항들은, 인터넷 전화 서비스 업체마다 요구사항이 다르고, 지원하는 VOIP 프로토콜이 다르며 계속 변화하므로, VOIP 액세스 게

이트웨이 소프트웨어를 설계할 때에는 다른 VOIP 프로토콜 이식이 쉽도록 유연성 있고, 업그레이드가 가능하도록 설계를 해야 한다.

3. 하드웨어 구조

VOIP 액세스 게이트웨이 시스템의 하드웨어를 구성하는 것은 VOIP 프로토콜 및 여러 가지 기능을 수행하는 CPU, 그리고, 아날로그 음성신호를 디지털 신호로 변경하고 인코딩/디코딩을 수행하는 CODEC와 DSP칩, 전화인터페이스를 제어하는 SLIC회로, 그리고 네트워킹을 위한 스위칭칩, 전원부 등이다. 중용량 이상의 시스템에서는 제어 보드(Control Board)와 가입자 보드(Subscriber Board), 그리고 전원 및 감시 보드(Power Board)를 착탈 가능한 형태로 구성한다. 이는 시스템의 업데이트 및 고장 수리 중에 서비스 중단을 최소화하기 위한 구조이다.

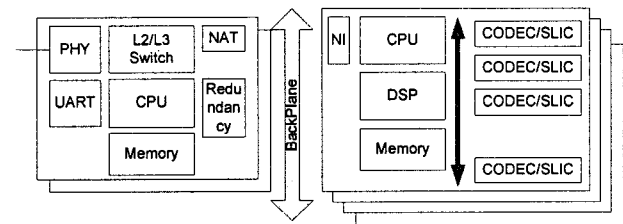


그림 1 하드웨어 구조

그림1은 VOIP 액세스 게이트웨이 시스템의 하드웨어 구조를 보인다. 제어 보드에는 호 처리를 위한 CPU와 네트워킹을 위한 스위칭칩이 위치한다. 제어 보드는 고장 및 오작동 등을 대비하여 이중화가 가능한 구조로 되어 있으며, 이중화 동기를 위한 채널로 서로 연결되어 있다. 가입자 보드에는 CODEC, DSP, SLIC회로가 위치한다. 가입자 보드와 제어 보드 간에는 제어 신호와 디지털 음성 데이터를 전달하기 위한 버스 또는 통신 채널이 필요하다. 운영 중에 가입자 보드 착탈을 쉽게 하

기 위해서 이더넷을 이용해 이러한 통신 채널을 구성할 수 있다. 가입자 보드의 주요 회로는 여러 가지 기능이 내장된 VOIP 용 SoC를 사용하여 구성한다. 가입자 측의 아날로그 신호는 CODEC를 통해 디지털 신호로 바뀌며, DSP를 통해 인코딩 과정을 거쳐 RTP 패킷으로 변환된다. RTP 패킷은 Back Plane 채널을 통해 제어 보드로 전달되고, 필요에 따라 NAT가 수행된 후, 스위칭 모듈을 통해 Uplink로 전달된다.

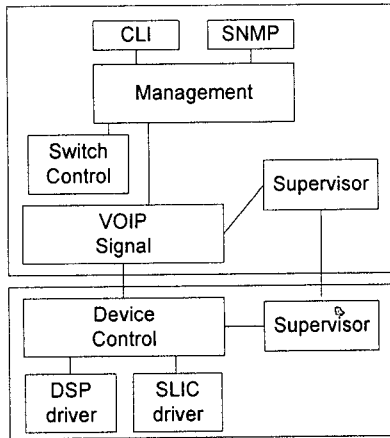


그림 2 소프트웨어 구조

4. 소프트웨어 구조

VOIP 액세스 게이트웨이 소프트웨어의 전체 구조는 그림2와 같다. 시스템의 제어 보드에는 CLI(Command Line Interface), SNMP 등 시스템을 관리하기 위한 인터페이스 및 관리 (Management) 모듈이 위치한다. 그리고, 시스템 전체 및 각 소프트웨어 모듈의 운영상태를 감독하는 감시(Supervisor) 모듈이 있다. SIP, H.323, MGCP, H.248(Megaco) 등 VOIP 시그널을 처리하는 VOIP 시그널 모듈은 가입자 보드에 위치할 수도 있으나, 통합적인 관리 및 운용을 위하여 보통 제어 보드에 위치한다. VOIP 시그널 모듈은 각 가입자 보드를 제어 하고 모든 VOIP 시그널을 처리하므로 시스템에서 성능적인 병목이 될 수 있으므로 하드웨어 설계 및 소프트웨어 구현시 주의를 하여야 한다. 시스템 전체적으로 하나의 IP를 사용하기 위해서는 NAT 기능 하드웨어가 추가 될 수 있으며, 이 경우 VOIP 시그널 모듈은 이를 고려하여 동작해야 한다. 가입자 보드에서는 DSP, CODEC, SLIC 등 하드웨어를 제어하는 장치 제어 (Device Control) 모듈이 동작한다. DSP 드라이버, SLIC 드라이버는 보통 OS 커널 내에 구현되며, 장치 제어 모듈이 시스템을 등을 이용하여 각 드라이버를 제어한다. 또한 가입자 보드에는 가입자 보드 내의 각 소프트웨어 모듈을 감시하는 감시 모듈이 있으며, 감시 결과를 제어 보드 내의 감시 모듈에게 알리는 동작을 수행한다.

그림 3은 VOIP 시그널 모듈의 구조를 보인다. 불필요한 Context 스위칭을 줄이고 리소스 Polling을 최소화하기 위해, Thread 개수를 최소화하였으며 Message Queue를 사용하여 각 Thread 간에 통신을 구성하였다. VOIP 시그널 모듈의 주

Thread는 Timer Thread와 UDP Receiver Thread로부터 Message를 받아 동작을 수행한다. 그리고 DNS Resolving 관련 라이브러리가 Blocking 함수이므로 이를 보완하기 위해 별도의 Thread를 두어 처리가 가능하도록 하였다. Timer Thread는 별도로 타이머 만을 관리하며 Time Out이 되면 메시지를 만들어 주 Thread로 전송하는 기능을 수행한다. 그러나 주 Thread가 이 Timeout 메시지를 받기 전에 해당 타이머를 Cancel할 경우가 있으므로, 이를 고려하여 메시지가 완전히 전달되면 리스트에서 타이머 엔트리를 제거하도록 하였다.

VOIP 프로토콜 모듈은 VOIP 서버 또는 상대방 장비와 메시지 송수신을 위한 전송(Transport) 모듈, 그리고 재전송 및 메시지를 생성과 분석을 수행하는 트랜잭션(Transaction) 모듈, 이벤트에 대한 처리와 동작, 부가서비스 등을 수행하는 호 처리(Call Control) 모듈로 구성된다. 그리고 NAT 제어 모듈, 다른 모듈과 통신하기 위한 IPC 모듈, 그리고 내부적인 통계 자료 및 설정을 관리하기 위한 관리 모듈을 포함하고 있다. VOIP 시그널 모듈의 주 Thread는 각 VOIP 시그널 모듈 내의 이벤트에 따른 동작을 수행하기 위하여 StateMachine을 가지고 있어, 이를 중심으로 동작을 결정, 수행한다. 그리고 이중화를 위하여 항상 VOIP 상태를 대기중인(Standby) 제어 보드와 동기화를 수행한다.

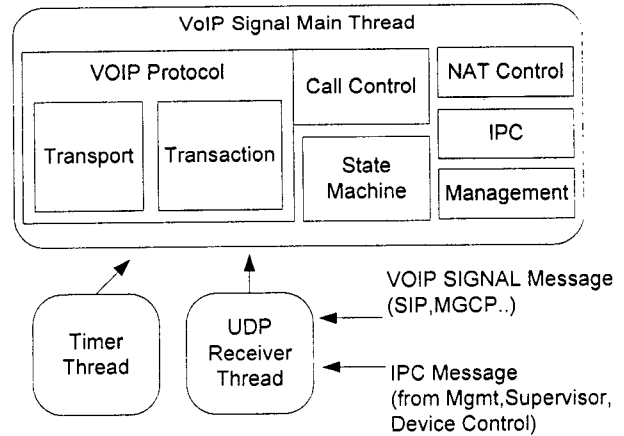


그림 3 VOIP 모듈 구조

VOIP 시그널 모듈은 VOIP 시그널 프로토콜이 어떤 것이냐에 따라 구조 차이가 있다. MGCP와 Megaco(H.248) 같은 경우에 실제적인 호 처리 및 부가서비스 기능들이 별도의 장비인 MGC(Media Gateway Controller)에 있으므로, 호 제어(Call Control) 모듈이 없거나 간단하게 구현하면 된다. 하지만, SIP나 H.323 프로토콜의 경우, 호 제어를 직접 수행해야 하고, 장치 제어 이벤트에 따른 동작, 부가 서비스 등을 모두 호 제어 모듈에 구현해 넣어야 하므로, 호 제어 모듈이 상당히 크게 된다.

많은 가입자 보드가 있는 중,대용량 VOIP 게이트웨이에서는 VOIP 시그널 모듈의 기능 일부를 분산하여 가입자 보드에서 처리하도록 구성할 수도 있다. 이러한 유연성 있는 설계를 위

하여 IPC 설계 및 정의가 중요하다. 장치 제어 모듈과 VOIP 시그널 모듈 간의 메시지 통신을 위한 IPC 설계시 다음 사항을 고려하여야 한다.

- VOIP 시그널 프로토콜 종류 및 장치 종류에 독립적으로 설계해야 한다. VOIP 프로토콜 종류에 따라 IPC가 바뀐다면, 그 소프트웨어 변경 작업이 무척 크게 되므로 비효율적이다. 일반적으로 Megaco 또는 MGCP 메시지와 의미가 대응되는 형태로 IPC 메시지를 구성하는 것이 좋다. MGCP와 Megaco 프로토콜은 장치제어를 고려한 프로토콜이며, VOIP SoC 역시 이를 고려하고 설계되어 있다. SIP, H.323의 경우에는, 호 제어 모듈을 통해 IPC 메시지를 이용하여 장치를 제어할 수 있다. 장치 종류에 독립적으로 설계하는 것은 어느 정도 한계가 있지만 최대한 독립적으로 설계하여야 추후 하드웨어 변경시 소프트웨어 변경을 최소화 할 수 있다.

- 신뢰성이 있어야 한다. 비록 장비 내부의 통신을 위한 IPC이지만, BackPlane을 통한 보드 간의 통신이므로 에러율을 완전히 무시할 수는 없다. 또한 순간적으로 급격히 메시지가 전달되는 과정에서의 버퍼 오버 플로우로 인한 IPC 메시지 손실도 고려해야 한다.

- 실시간성이 있어야 한다. 신뢰성을 위해 TCP와 같은 비실시간성 전송 프로토콜을 쓸 수 없다. VOIP 시그널 프로토콜 역시 어느 시간이 지나면, 효과가 상실되는 실시간 프로토콜이다. 그러므로 VOIP 시스널 프로토콜을 받아 처리하는 IPC 역시 실시간성을 갖아야 한다. 즉, 어느 시간이 지나면 Timeout이 발생하고 이에 맞는 동작을 할 수 있어야 한다.

- 간결해야 한다. IPC가 복잡하다면 차라리 MGCP나 Megaco를 그대로 쓰는 것이 낫다. IPC메시지 인코딩, 디코딩이 간편하고 빨라야 하며, 메시지 길이도 간결한 것이 좋다. IPC처리가 시스템의 성능을 좌우할 수 있다.

본 논문에서는 VOIP 시그널링 모듈과 장치 제어 모듈간의 통신을 위한 IPC 모듈은 UDP상에서 동작하도록 하였다. API를 통해 IPC모듈의 재전송주기와 횟수를 설정할 수 있도록 하였다. 장치 제어 모듈이 이벤트를 통보하는 경우, 이 이벤트가 VOIP 시그널 모듈을 통해서 VOIP 서버 또는 MGC로 전달하는 과정이 필요하므로, 이를 고려하여 IPC Timeout을 다소 크게 설정하였다. 반대로 VOIP 서버 또는 MGC에서 시그널이 전달될 경우에는 VOIP 시그널 프로토콜에서 먼저 Timeout 나지 않도록, IPC Timeout을 이보다 작게 설정하였다. IPC 메시지를 전달하는 채널은 별도로 정의한 Ethernet Type을 사용하거나, Tagged VLAN을 이용할 수 있다.

VOIP 액세스 게이트웨이 소프트웨어 설계시 고려해야 할 부분 중 가장 큰 것이 호 처리 능력이다. 성능을 좌우하는 것 중 하나가 메모리 관리이다. 일반적으로 OS에서 제공하는 동적 메모리 할당 함수는 다소 느리다. 특히 VOIP 액세스 게이트웨이 소프트웨어에서는 메모리 할당 패턴이 규칙적이며 어느 정도 예측가능하다. 그러므로, 자체 메모리 할당 메커니즘을 갖는 것이 좋다. 또한 임베디드 메모리 최적화[5]를 위한 기법을 적용하여 코딩하는 것이 좋다. 표1은 OS에서 제공하는 동적

메모리 함수인 malloc를 사용했을 때의 수행시간과 어플리케이션에서 별도로 구현한 배열을 사용했을 때의 수행시간을 보인다. 이 시험은 실제 구현한 액세스 게이트웨이 소프트웨어를 구동하면서 얻은 메모리 할당 관련 로그를 사용하였다. 시뮬레이터에 이 로그를 이용하여 동일 패턴을 10000회 반복하면서 수행시간을 측정하였다. 표에서 수행시간은 실제 수행시간(ms)이며, 상대시간은 실제 수행시간에서 null함수 수행시간을 제외한 시간이다. null함수 수행시간은 malloc, free 대신 아무 동작도 하지 않았을 때의 수행결과이다. 시험결과 약 6배 정도 빠른 성능을 볼 수 있었다. 이는 실험을 통해서, ARM9을 사용한 장비에서 하나의 호를 처리하는데, 약 1.1 ms 더 빠르게 할 수 있었으며, PPC8245를 사용한 장비에서 약 0.5 ms 더 빠르게 처리할 수 있었다.

표 1 메모리 할당 함수의 평균 수행시간

	NULL	Malloc		Array		비율
	수행시간	수행시간	상대시간	수행시간	상대시간	
PC	434	921	487	515	81	16.63
ARM	44970	58920	13950	47759	2789	19.99
PPC	21609	28792	7183	23469	1360	18.93

추가적으로 VOIP 액세스 게이트웨이에서 성능을 좌우하는 것 중의 하나가 Log 시스템이다. 시험결과 Log 및 Trace를 동작시켰을 때 4배 이상의 성능저하가 나타났다. Log루틴을 설계할 때 시스템 성능에 미치는 영향을 고려하여야 한다. 본 논문에서 Log 루틴은 운영자가 각 모듈별/수준별 제어가 가능하도록 하였고, 별로 로그 데몬이 로그를 관리하도록 하여 성능저하를 줄였다.

5. 결론

본 논문에서는 VOIP 액세스 게이트웨이의 일반적인 요구사항과, 하드웨어 설계를 보이고, 성능과 유연성을 고려한 VOIP 액세스 게이트웨이 소프트웨어를 설계하고 구현하였다. VOIP 시그널 변경시 VOIP 시그널 모듈 중 일부만 변경하면 되며, 하드웨어 변경시에도 Management 모듈만 변경하면 된다. 앞으로 다른 구조와의 비교 성능 평가와 지속적인 성능 개선을 위한 연구가 필요하다.

6. 참고문헌

- [1] SIP: Session Initiation Protocol, IETF RFC3261, 2002.
- [2] H.323: Packet-based multimedia communications systems, ITU-T H.323, 2003.
- [3] MGCP: Media Gateway Control Protocol, IETF RFC 3435, 2003.
- [4] H.248: Gateway Control Protocol, ITU-T H.248, 2000
- [5] Memory System Optimization of Embedded Software, WAYNE WOLF, IEEE 2003.