멀티룸 주거형 AI 에이전트의 효율적 통신 구조 설계

황선태, 박혁인, 이건희
HDC LABS

iamsuntae@hdc-labs.com, hyeokout@hdc-labs.com, Gunhee_Lee@hdc-labs.com

Efficient Communication Architecture for Multi-room Residential AI Agent System

Hwang Sun Tae, Park Hyeok In, Lee Gun Hee
HDC LABS

요 약

홈네트워크 환경에서 세대 내 음성 기반 AI 에이전트 적용은 IoT 접근성 증대와 사용자 개인화 서비스를 위한 핵심 기술이다. 그러나 세대 구조의 다양성과 멀티룸 환경은 홈네트워크를 고려한 설계와 다중 세션 관리, 서버-클라이언트 간 효율적 통신, 응답 지연 최소화 등의 기술적 도전과제가 존재한다. 본 논문에서는 멀티룸 주거형 AI 에이전트 시스템의 효율적인 통신 구조를 제안한다. 제안하는 시스템은 파일 기반 Baseline 캐싱, 파일 안정성 검증, 화자인식/STT 병렬 처리, 세션 갱신 메커니즘, Proactive 중단 메커니즘 등 5 가지 핵심 최적화 기법을 적용하여 통신 효율성을 향상시킨다. 이론적 분석을 통해 제안 기법들의 시간 복잡도 개선을 검증하였으며, 분산 아키텍처 기반의 확장 가능한 홈네트워크 기반 음성 AI 시스템 구현 가능성을 제시한다.

Ⅰ. 서 론

스마트홈 기술의 발전과 함께 음성 인터페이스 기반 AI 에이전트에 대한 수요가 급증하고 있다. 특히 주거 환경에서 여러 방에 걸쳐 일관된 음성 서비스를 제공하는 멀티룸 AI 에이전트 시스템은 사용자 경험 향상에 핵심적인 역할을 한다[1].

그러나 멀티룸 환경에서 AI 에이전트를 운영하는 데에는 여러 기술적 도전과제가 존재한다. 첫째, 다중 방에서 동시에 발생하는 세션을 효율적으로 관리해야 한다. 둘째, 중앙 서버와 각 방의 에지 디바이스 간 통신 오버헤드를 최소화해야 한다. 셋째, 사용자 요청에 대한 응답 지연을 최소화하여 자연스러운 대화 경험을 제공해야 한다[2].

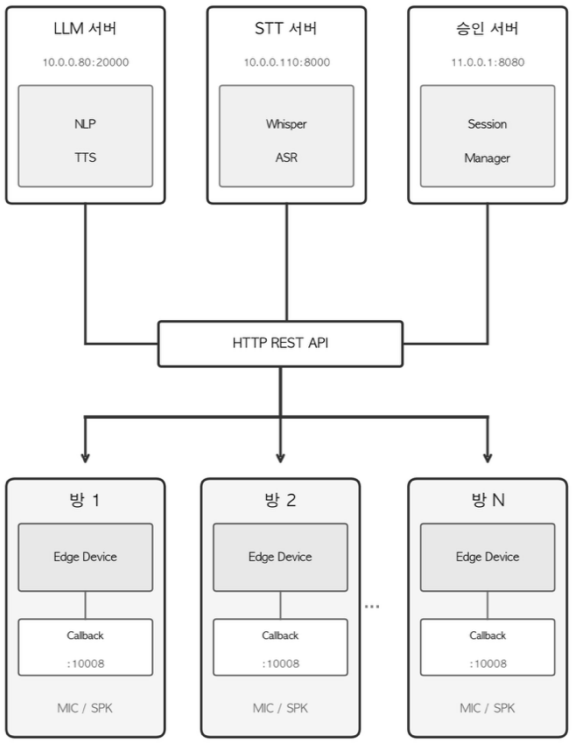
본 논문에서는 이러한 도전과제를 해결하기 위한 효율적인 통신 구조를 제안한다. 제안하는 시스템은 분산 아키텍처를 기반으로 하며, 5 가지 핵심 최적화 기법을 통해 통신 효율성을 크게 향상시킨다. 본 논문의 주요 기여점은 다음과 같다:

- 멀티룸 AI 에이전트를 위한 분산 통신 아키텍처 설계
- O(1) 시간 복잡도의 파일 기반 Baseline 캐싱 기법
- 화자인식과 STT 의 병렬 처리를 통한 응답 지연 감소
- Reactive/Proactive 양방향 통신 메커니즘

Ⅱ. 본론

Ⅱ.1. 시스템 아키텍처

제안하는 시스템은 중앙 서버 클러스터와 방별 에지 디바이스로 구성된 분산 아키텍처를 채택한다. 그림 1 은 전체 시스템 구조를 나타낸다.



[그림 1] 시스템 아키텍처 다이어그램

중앙 서버 클러스터는 LLM 서버, STT 서버, 제어 서버로 구성되며, HTTP REST API 를 통해 각 방의 에지 디바이스와 통신한다. 각 방에는 마이크와 스피커가 장착된 에지 디바이스가 배치되어 음성 입출력을 담당한다.

시스템은 두 가지 통신 모드를 지원한다: (1) Reactive 모드는 사용자의 웨이크워드 감지로 시작되는 일반적인 대화 흐름이며, (2) Proactive 모드는 서버에서 먼저 사용자에게 정보를 전달하는 능동적 통신이다.

콜백 서버는 LLM 처리 결과를 수신하여 해당 방의 에지 디바이스에 오디오 파일로 전달하며, 파일 기반 메시지 큐 방식을 통해 비동기적이고 신뢰성 있는 통신을 구현한다.

Ⅱ.2. 효율성 최적화 기법

Ⅱ.2.1 파일 기반 Baseline 캐싱

콜백 디렉토리에서 새로운 오디오 파일을 감지할 때 중복 처리를 방지하기 위해 Set 자료구조 기반의 Baseline 캐싱 기법을 적용한다. 수식 (1)과 같이 현재 파일 집합에서 Baseline 집합을 차집합 연산하여 새 파일만을 O(1) 시간에 감지한다.

$$new_files = current_files - baseline \quad (1)$$

이 기법은 파일 시스템 순회 비용을 최소화하고, 중복 처리로 인한 리소스 낭비를 방지한다.

Ⅱ.2.2 파일 안정성 검증

네트워크를 통해 수신 중인 파일의 부분 처리를 방지하기 위해 안정성 검증 메커니즘을 도입한다. 파일 크기를 100ms 간격으로 확인하여 3 회 연속 동일한 크기가 확인되면 파일 전송이 완료된 것으로 판단한다.

이 적응적 대기 방식은 고정 대기 시간 대비 불필요한 지연을 줄이면서도 불완전한 파일 처리를 방지한다.

Ⅱ.2.3 화자인식/STT 병렬 처리

사용자 음성 입력 후 화자 인식과 STT(Speech-to-Text) 처리가 순차적으로 진행되면 응답 지연이 발생한다. 본 시스템에서는 ThreadPoolExecutor 를 활용하여 두 작업을 병렬로 수행한다.

$$T_total = max(T_speaker, T_stt) \ll T_speaker + T_stt \quad (2)$$

수식 (2)와 같이 병렬 처리를 통해 이론적으로 약 50%의 처리 시간 단축이 가능하다.

Ⅱ.2.4 세션 갱신 메커니즘

장시간 대화 시 세션 타임아웃을 방지하기 위해 매 대화 턴마다 세션 갱신 API 를 호출한다. 이를 통해 유휴 시간(idle time)이 초기화되어 안정적인 장시간 대화를 지원한다.

Ⅱ.2.5 Proactive 중단 메커니즘

서버에서 능동적으로 메시지를 전달하는 Proactive 모드와 사용자 요청(Reactive)이 충돌할 경우, Threading.Event 기반의 중단 신호를 통해 우선순위 기반 요청 처리를 구현한다. 사용자 요청이 우선순위를 가지며, Proactive 세션은 즉시 중단되어 자연스러운 사용자 경험을 제공한다.

Ⅱ.3 이론적 분석

표 1 은 제안하는 최적화 기법들의 시간 복잡도 개선을 정리한 것이다.

최적화 기법	기존 방식	제안 방식
파일 감지	O(n) 순회	**O(1) Set**
화자인식+STT	$T_1 + T_2$	**$\max(T_1, T_2)$**
안정성 검증	고정 대기	**적응적 대기**

[표 1] 최적화 기법별 시간 복잡도 비교

제안하는 분산 아키텍처는 HTTP REST 기반 경량 프로토콜을 사용하여 통신 오버헤드를 최소화한다. 파일 기반 메시지 큐 방식은 네트워크 불안정 상황에서도 메시지 손실을 방지하며, 비동기 콜백 방식을 통해 서버 부하를 분산시킨다.

Ⅲ. 결론

본 논문에서는 멀티룸 주거형 AI 에이전트 시스템의 효율적인 통신 구조를 제안하였다. 분산 아키텍처를 기반으로 5 가지 핵심 최적화 기법을 적용하여 통신 효율성을 향상시켰다.

파일 기반 Baseline 캐싱은 O(1) 시간 복잡도로 새 파일을 감지하고, 병렬 처리를 통해 화자인식/STT 처리 시간을 약 50% 단축하였다. 세션 갱신 및 Proactive 중단 메커니즘은 안정적이고 자연스러운 사용자 경험을 제공한다.

향후 연구에서는 실제 주거 환경에서의 성능 측정 데이터를 기반으로 제안 기법들의 효과를 정량적으로 검증할 예정이다..

참 고 문 헌

[1] C. Irugalbandara et al., "A Secure and Smart Home Automation System with Speech Recognition and Power Measurement Capabilities," Sensors, vol. 23, no. 13, pp. 5784, Jun. 2023.

[2] A. Alshammri et al., "IoT-Based Voice-Controlled Smart Homes with Source Separation Based on Deep Learning," J. Sensors, vol. 2023, Article ID 1911385, Mar. 2023.

[3] M. Vidal et al., "Reducing Communication Overhead in the IoT-Edge-Cloud Continuum: A Survey on Protocols and Data Reduction Strategies," Internet of Things, vol. 29, 2025.

[4] Y. Zhang et al., "Research and Design of Smart Home Speech Recognition System Based on Deep Learning," in Proc. IEEE ICCCS, Chengdu, China, 2020, pp. 306-310.