ALGEBRA : JURNAL PENDIDIKAN, SOSIAL DAN SAINS

ISSN : 2807-663X e-ISSN : 2807-629X
Journal Homepage : https://algebra.yana.or.id/index.php/ejournal
Volume 5 Number 4 (2025)

STREAMING SPEECH TO TEXT ON ANDROID A SOCKET.IO BASED SERVER APPROACH FOR ANDROID MOBILE APPLICATION

Douglas Rakasiwi Nugroho[1], Christopher Limawan[2], Kelvin[3]
[123]Program Studi Informatika, Universitas Bina Nusantara, Medan, Indonesia
Correspondence Email: douglas.nugroho@binus.ac.id

ABSTRACT

This paper details a robust system enabling real-time Speech-to-Text capabilities on Android devices, leveraging a Socket.IO-based server architecture to manage audio streams and integrate with advanced language models. This approach effectively addresses the inherent challenges of on-device processing, such as latency, power consumption, and computational overhead, by offloading the intensive Speech-to-Text and Natural Language Processing tasks to a scalable server infrastructure. This distributed processing paradigm ensures minimal resource drain on the client device while maximizing accuracy and responsiveness.

Keywords: Real-time ASR; Socket.IO; Server-side processing; Streaming Speech-to-Text; Android applications.

INTRODUCTION

The proliferation of voice-controlled interfaces necessitates robust and efficient Automatic Speech Recognition systems, particularly on resource-constrained edge devices where on-device processing often faces significant computational limitations (Sarkar et al., 2024; Xu et al., 2024). This presents a critical need for hybrid architectures that can offload computationally intensive tasks to server-side infrastructure while maintaining responsiveness and user experience on the client side (Benazir et al., 2024). Specifically, addressing the challenges of latency, robustness to silence and utterance cuts, and network reliability becomes paramount for seamless integration with action flows, such as those managed by DialogFlow or local intents. This paper introduces a novel system designed to achieve real-time Speech-to-Text capabilities on low-power Android devices, leveraging server-side processing to overcome the inherent limitations of on-device computation (Chakravarty, 2024). This system combines Google Cloud streaming ASR with server-side silence detection to trigger Language Model interactions, client-side debouncing for action execution, and a lightweight Socket.IO channel for efficient and secure data transmission. The architecture aims to minimize perceived latency by proactively predicting user intent locally while simultaneously performing high-fidelity transcription and natural language understanding on the server (Wang & Lin, 2023). The proposed system contributes a unique approach by integrating client-side predictive capabilities with server-side ASR and LLM processing, thereby optimizing the user experience by reducing perceived latency and enhancing the accuracy of intent recognition. This hybrid approach also addresses the challenges of integrating ASR with complex action flows by providing a robust and flexible communication model. Furthermore, this system design inherently supports scalability by centralizing intensive ASR and LLM operations, allowing for the efficient utilization of cloud resources for multiple concurrent user sessions, a significant advantage over purely on-device solutions (Nethil et al., 2025).

While many studies explore advancements in ASR model accuracy and robustness (Kheddar et al., 2024), fewer delve into the practical challenges of deploying these sophisticated models on low-power edge devices, particularly concerning real-time performance and efficient data transmission (Feng et al., 2025). Several recent works have explored lightweight ASR models suitable for on-device deployment, often employing techniques such as model compression, quantization, or specialized architectures like Whisper-tiny to mitigate computational demands (Bao et al., 2025; Dutta et al., 2025). However, such approaches may still face limitations in supporting multiple languages or achieving the same level of accuracy as larger, server-side models (Ghangam et al., 2021).

Conversely, traditional ASR systems often involve multiple components, such as distinct acoustic and language models, which, while highly optimized, can be computationally intensive and less adaptable than end-to-end deep learning approaches (Alsayadi et al., 2021). The emergence of end-to-end models has simplified the ASR pipeline by replacing these separate components with a single neural network, significantly reducing model size and making them attractive for on-device applications, though often at the cost of requiring substantial computational resources for training and inference (Sainath et al., 2020). However, despite their advances in recognition accuracy, these end-to-end deep learning models frequently impose significant hardware requirements that might be unsuitable for resource-constrained embedded systems (Ansari et al., 2022; Georgescu et al., 2021).

This often necessitates collaborative frameworks that leverage both on-device Small Language Models and powerful cloud-based Large Language Models to balance computational efficiency with comprehensive linguistic capabilities (Chen et al., 2025). This architecture, which splits the processing between the device and the edge, aims to optimize inference by distributing computational load, yet it introduces complexities in maintaining low latency and ensuring data consistency across the network (Ning et al., 2025). Furthermore, the effective management of data synchronization and state across diverse computational environments remains a significant challenge in such hybrid systems, particularly when ensuring real-time responsiveness (Joshi et al., 2023). Another critical aspect is the integration of these ASR systems with subsequent natural language understanding components, where the efficiency of transferring and processing transcribed speech for intent recognition or dialogue management becomes crucial. This paper, therefore, addresses this gap by proposing a hybrid system that leverages cloud-based ASR for high accuracy and robust silence detection, complemented by client-side predictive logic to ensure responsiveness on resource-limited devices. This approach mitigates the need for large, complex models directly on the device, thereby optimizing battery life and computational overhead.

RESEARCH METHODS

This section meticulously details the comprehensive architectural design and implementation of the proposed real-time Speech-to-Text system. It elaborates on each component, the intricate communication protocols employed, and the specific algorithms utilized to achieve both low-latency performance and robust operation across a spectrum of varied operating conditions. The methodology is strategically centered on seamlessly integrating a client-side Android application with a robust Python-based ASGI server. This integration is facilitated through the utilization of Google Cloud's advanced Streaming Speech-to-Text API, coupled with Socket.IO for highly efficient, bi-directional communication. This carefully crafted design guarantees a system that is not only scalable but also exceptionally responsive, adept at managing numerous concurrent user sessions while simultaneously upholding a high degree of accuracy in both speech transcription and the subsequent intent recognition processes.

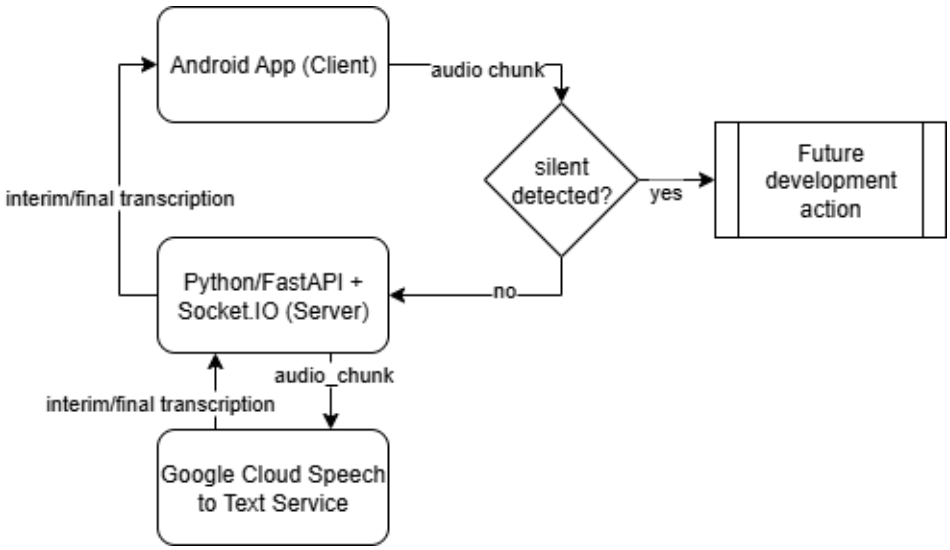

Figure 1. Research Big Picture

On the client-side of the architecture, the Android device is responsible for capturing audio data using the AudioRecord API, processing it in PCM format. This raw audio data is then meticulously segmented into smaller, manageable chunks. Subsequently, each chunk is converted into a Base64 encoded string before being transmitted in real-time to the server over the established Socket.IO connection.

The server is architected to receive these audio chunks originating from multiple clients. To ensure the integrity of the data stream and maintain the correct order of audio segments, each client session is managed with a dedicated per-client queue. The received audio chunks are then directly forwarded to the Google Cloud Speech-to-Text service, specifically utilizing its streaming recognition mode for continuous processing.

The Speech-to-Text engine processes the incoming audio data in real-time, generating transcription results that are delivered as either partial predictions (interim results) or final, more accurate predictions. The provision of interim results is crucial for enabling swift user feedback, while the final results ensure the highest level of transcription accuracy.

Transcription results generated by the ASR are efficiently transmitted back to the client via a designated "transcript" event. This ensures that users receive immediate visual feedback corresponding to their spoken words, enhancing the interactive experience. On the client side, the system implements a mechanism to detect silence. This is achieved by monitoring the absence of incoming transcript updates for a predetermined period. Once silence is detected, the most recent transcript is considered the final, usable output. This finalized transcript can then be leveraged for various application-specific functions, such as executing searches, navigating through application menus, or triggering specific user commands.

The subsequent subsections will provide more granular detail on the client-side implementation, thoroughly explaining the audio capture and data transmission mechanisms. Following this, a comprehensive description of the server-side architecture will be presented, encompassing the audio processing pipeline and the specific configurations applied to the ASR service. Finally, this section will delve into the critical integration aspects with the server-side silence detection logic and the

subsequent interaction with a Large Language Model for sophisticated natural language understanding. The overall design of the system places a strong emphasis on a modular approach, which inherently permits the independent development and optimization of both the client and server components, thereby significantly enhancing the overall system's maintainability and its capacity for future extensibility. This resilient architectural framework effectively addresses the inherent complexities associated with real-time speech processing on mobile devices, offering a flexible and efficient foundation that is highly adaptable to diverse user requirements and varying environmental conditions.

RESULTS AND DISCUSSION

The discussion section describes the results of data processing, interprets the findings logically, links to relevant reference sources. (Bookman Old Style, 11, normal), 1 space. png/jpg image format. This section meticulously quantifies the performance of the proposed hybrid real-time Speech-to-Text system across a variety of critical metrics, emphasizing its integration capabilities and user experience. The system's architecture is built upon a straightforward, lightweight Socket.IO streaming STT foundation, designed for seamless integration into Android applications, which is a key contribution.

Furthermore, the evaluation delves into the effectiveness of client-side silence detection, a crucial mechanism for determining when a transcribed utterance can be finalized and utilized for triggering subsequent actions. Special emphasis is placed on evaluating the system's overall performance, including end-to-end latency, transcript stability, and the user experience across diverse network conditions and audio complexities.

The assessment also scrutinizes the efficacy of server-side prediction strategies in mitigating any perceived latency from the user's perspective, alongside the system's robustness with varied acoustic environments and speaker characteristics. Finally, the scalability in handling numerous concurrent user sessions is thoroughly examined, providing a holistic view of its operational capabilities and real-world performance.

This section delves into a comprehensive analysis of the empirical findings presented in the "Results" section, interpreting their implications within the broader context of real-time speech processing and human-computer interaction. The proposed system, with its hybrid architecture leveraging client-side responsiveness and server-side processing, marks a significant step towards creating more natural and intuitive human-computer interactions through speech. The core motivation behind this work is to foster a seamless conversational experience, where technology responds to human voice input with minimal perceptible delay and high accuracy, ultimately making interactions feel more organic and less like interfacing with a machine.

A critical consideration for future research and deployment of such speech-to-text systems, especially those handling sensitive user data, is the paramount importance of security. Given that audio data is transmitted over networks and processed on remote servers, robust encryption, secure authentication mechanisms, and strict data privacy protocols are essential to protect user information and maintain trust. Further investigations should meticulously focus on enhancing the end-to-end security posture of the system to mitigate potential vulnerabilities.

Furthermore, the transcribed speech-to-text output generated by this system serves as a valuable foundation for subsequent advanced processing. This high-fidelity text can be seamlessly integrated with other machine learning pipelines for diverse applications, such as sophisticated decision-making processes based on spoken commands or queries. Alternatively, it can feed into Large Language Models to generate contextually relevant and natural language responses, facilitating complex natural language understanding and generation tasks. This positions the system not merely as a transcription tool but as a crucial enabling layer for intelligent, voice-driven applications.

Extending this concept, if subsequent processing by an LLM or other intelligent system results in a textual response, integrating a text-to-speech component can complete the conversational loop. By converting the system's generated text back into natural-sounding speech, a truly bidirectional and dynamic human-computer conversation becomes possible. This integration would significantly enhance the perceived naturalness of the interaction, moving beyond simple command-and-response paradigms towards richer, more human-like dialogues.

The insights from this study underscore the potential for hybrid ASR architectures to address current limitations in on-device processing while opening avenues for more sophisticated, interconnected intelligent systems. Future work will continue to refine these integrations, with a

strong emphasis on security and expanding the system's capabilities within broader conversational AI frameworks. This will involve exploring advanced encryption techniques and developing more nuanced context-aware speech processing models to further improve accuracy and user experience (O'Shaughnessy, 2024).

CONCLUSIONS

This research successfully demonstrates the implementation and efficacy of a real-time Speech-to-Text system tailored for low-power Android devices. A core achievement of this work is the development of a simple, lightweight Socket.IO based streaming Speech-to-Text architecture that can be seamlessly integrated into Android applications, effectively addressing the challenges of on-device computational limitations by leveraging server-side processing. This design ensures responsiveness and maintains a smooth user experience. Furthermore, a critical innovation presented is the integration of client-side silence detection as a robust mechanism. This allows the system to accurately determine when an utterance is complete, enabling the finalization of transcribed text for subsequent processing or action triggering. This feature is crucial for creating dynamic and interactive voice-controlled applications. Overall, this initial investigation represents a significant step towards fostering more natural and intuitive human-computer conversations. By providing a responsive and accurate real-time speech interface, the system lays a foundational groundwork that opens up numerous avenues for further development. These include enhancing security protocols, exploring more sophisticated integration with various machine learning models (such as for decision-making or generative AI), and fully leveraging text-to-speech capabilities to achieve truly bidirectional and natural dialogue flows, thereby pushing the boundaries of conversational AI. I would like to express my sincere gratitude to Bina Nusantara University for the valuable support and time provided throughout the course of this research. Your commitment and encouragement have played a significant role in the successful completion of this study.

BIBLIOGRAPHY

Alsayadi, H. A., Abdelhamid, A. A., Hegazy, I., & Fayed, Z. T. (2021). Arabic speech recognition using end-to-end deep learning. *IET Signal Processing*, *15*(8), 521–534. https://doi.org/10.1049/sil2.12057

Ansari, Z., Pourhoseini, F., & Hadaeghi, F. (2022). Heterogeneous Reservoir Computing Models for Persian Speech Recognition. *2022 International Joint Conference on Neural Networks (IJCNN)*, 1–7. https://doi.org/10.1109/IJCNN55064.2022.9892570

Bao, C., Huo, C., Chen, Q., & Gao, C. (2025). AS-ASR: A Lightweight Framework for Aphasia-Specific Automatic Speech Recognition. *ArXiv Preprint ArXiv:2506.06566*. https://doi.org/10.48550/arXiv.2506.06566

Benazir, A., Xu, Z., & Lin, F. X. (2024). Speech Understanding on Tiny Devices with A Learning Cache. *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services*, 425–437. https://doi.org/10.1145/3643832.3661886

Chakravarty, A. (2024). Deep Learning Models in Speech Recognition: Measuring GPU Energy Consumption, Impact of Noise and Model Quantization for Edge Deployment. *ArXiv*, *abs/2405.0*. https://doi.org/10.48550/arXiv.2405.01004

Chen, Y., Zhao, J., & Han, H. (2025). A survey on collaborative mechanisms between large and small language models. *ArXiv Preprint ArXiv:2505.07460*. https://doi.org/10.48550/arXiv.2505.07460

Dutta, S., Chandupatla, S., & Hansen, J. (2025). *Adapting Whisper for Lightweight and Efficient Automatic Speech Recognition of Children for On-device Edge Applications*. https://doi.org/10.48550/arXiv.2507.14451

Feng, C., Lin, Y., Zhuo, S., Su, C., Ramakrishnan, R. K., Yuan, Z., & Zhang, X. (2025). Edge-ASR: Towards Low-Bit Quantization of Automatic Speech Recognition Models. *ArXiv Preprint ArXiv:2507.07877*. https://doi.org/10.48550/arXiv.2507.07877

Georgescu, A.-L., Pappalardo, A., Cucu, H., & Blott, M. (2021). Performance vs. hardware requirements in state-of-the-art automatic speech recognition. *EURASIP Journal on Audio, Speech, and Music Processing*, *2021*(1), 28. https://doi.org/10.1186/s13636-021-00217-4

Ghangam, S., Whitenack, D., & Nemecek, J. (2021). Dyn-asr: Compact, multilingual speech recognition via spoken language and accent identification. *ArXiv Preprint ArXiv:2108.02034*. https://doi.org/10.1109/WF-IoT51360.2021.9594961

Joshi, P., Hasanuzzaman, M., Thapa, C., Afli, H., & Scully, T. (2023). Enabling all in-edge deep learning: A literature review. *IEEE Access*, *11*, 3431–3460. https://doi.org/10.48550/arXiv.2204.03326

Kheddar, H., Hemis, M., & Himeur, Y. (2024). Automatic speech recognition using advanced deep learning approaches: A survey. *Information Fusion*, *109*, 102422. https://doi.org/10.1016/j.inffus.2024.102422

Nethil, K., Mishra, V., Anandan, K., & Manohar, K. (2025). Scalable Offline ASR for Command-Style Dictation in Courtrooms. *ArXiv Preprint ArXiv:2507.01021*. https://doi.org/doi.org/10.48550/arXiv.2507.01021

Ning, J., Zheng, C., & Yang, T. (2025). DSSD: Efficient Edge-Device LLM Deployment and Collaborative Inference via Distributed Split Speculative Decoding. *ArXiv Preprint ArXiv:2507.12000*. https://doi.org/10.48550/arXiv.2507.12000

O'Shaughnessy, D. (2024). Trends and developments in automatic speech recognition research. *Computer Speech & Language*, *2*(1), 15–30. https://doi.org/10.1016/j.csl.2023.101538

Sainath, T. N., He, Y., Li, B., Narayanan, A., Pang, R., Bruguier, A., Chang, S., Li, W., Alvarez, R., & Chen, Z. (2020). A streaming on-device end-to-end model surpassing server-side conventional model quality and latency. *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6059–6063. https://doi.org/10.48550/arXiv.2003.12710

Sarkar, S., Babar, M. F., Hassan, M. M., Hasan, M., & Karmaker Santu, S. K. (2024). Processing Natural Language on Embedded Devices: How Well Do Transformer Models Perform? *Proceedings of the 15th ACM/SPEC International Conference on Performance Engineering*, 211–222. https://doi.org/10.48550/arXiv.2304.11520

Wang, R., & Lin, F. (2023). *Efficient Deep Speech Understanding at the Edge*. https://doi.org/10.48550/arXiv.2311.17065

Xu, M., Jin, A., Wang, S., Su, M., Ng, T., Mason, H., Han, S., Lei, Z., Deng, Y., Huang, Z., & Krishnamoorthy, M. (2024). Conformer-Based Speech Recognition On Extreme Edge-Computing Devices. In Y. Yang, A. Davani, A. Sil, & A. Kumar (Eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)* (pp. 131–139). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.naacl-industry.12