



iJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IX **Month of publication:** September 2026

DOI: <https://doi.org/10.22214/ijraset.2026.84999>

www.ijraset.com

Call: ☎ 08813907089

E-mail ID: ijraset@gmail.com

iTantra: Indian Multilingual TTS & STT Aided Neural Transceiver Radio Access for Low Bitrate

Prachi Tamhan, Omkar Jagtap, Aayush Padwal, Pritesh Thorat, Sakshi Datir, Vishwas Shinde, Yogiraj Bhingare
Professor, Student, Department of Information Technology, International Institute of Information Technology, Pune, Maharashtra, India

Abstract: When disaster strikes, communication can become a real headache, especially in rural, remote areas where traditional cellular networks and internet services might be out of reach or just plain unreliable. This paper introduces iTantra, an innovative offline Android-based multilingual voice communication system that's built to ensure reliable communication even over low-bitrate connections. The system works by converting speech into text right on the device, handling local language and message processing, and then packing that information into compact packets for transmission through various communication links. To boost message delivery in these tricky environments, it uses a lightweight reliability mechanism that includes CRC-16 error detection, acknowledgments, retransmission, sequence tracking, and duplicate protection. On the receiving end, packets are reassembled and turned back into speech using offline text-to-speech processing, supporting multiple languages. The prototype operates without needing internet connectivity, showing a speech-to-text latency of about 300–500 ms and an end-to-end latency of roughly 600–1000 ms. In a test involving a 30-second voice message, the compact text representation achieved an impressive 2500:1 reduction in the amount of data transmitted compared to raw audio. This approach highlights the potential for resilient, multilingual, and bandwidth-efficient communication in emergency response situations and other areas where connectivity is limited.

Keywords: Low-Bitrate Communication, Offline Neural Transceiver, On-Device Speech Processing, Multilingual Translation Engine, Transport-Independent Protocol, Stop-and-Wait ARQ, Emergency Communication System, Data Volume Reduction

I. INTRODUCTION

Traditional voice communication platforms depend heavily on a steady cellular connection, fast internet infrastructure, and cloud-based API services. But in critical situations—like disaster response areas, tactical border security missions, rural medical outreach, and remote field operations—this essential telecommunications framework is often missing, damaged, or even deliberately disrupted. Over 100 million people live or work in places with poor cellular coverage, where any infrastructure failure can cut off communication for both ground teams and local communities.

In these disconnected environments, communication systems face two major hurdles: limited bandwidth and language differences. To start, standard voice transmission methods send raw digitized audio waveforms, which need a lot of channel capacity (around 500 KB for just a 30-second audio clip). When these audio streams are forced through low-bitrate radio links, Bluetooth connections, or restricted serial lines, they experience significant delays, high packet loss, and frequent transmission issues. Additionally, joint operations that involve emergency responders, military personnel, healthcare workers, and volunteers often include multilingual teams speaking a variety of regional languages (like Hindi, Gujarati, Marathi, Tamil, Telugu, Kannada, Malayalam, Odia, Bengali, and English). Current communication tools, such as WhatsApp, depend on cloud servers for translation, while tactical software like the Android Tactical Assault Kit (ATAK) with Vx or military applications like ULTRA fall short in providing robust offline support for Indian regional languages and lack effective semantic-level data compression.

To tackle these operational challenges, this paper presents iTantra, a versatile, open-source multilingual neural voice transceiver that works seamlessly on standard Android devices. The platform introduces a fresh approach with the motto: "Transmit Meaning, Not Raw Voice." Rather than sending audio signals directly, iTantra processes spoken language right on the device using Voice Activity Detection (VAD) and Vosk-based Speech-to-Text (STT). The output is a compact text format (about 200 bytes for a 30-second speech clip) that gets serialized, wrapped in binary frames with CRC-16 error checking, and sent over various low-bitrate channels using a transport-agnostic Stop-and-Wait Automatic Repeat reQuest (ARQ) protocol.

On the receiving end, this streamlined packet is reassembled, checked for accuracy, and translated into the recipient's preferred language using on-device translation tools like Argos Translate or Marian NMT via ONNX. Finally, it's transformed back into natural speech with offline neural Text-to-Speech (TTS) engines such as Piper and eSpeak.

With no reliance on cloud services and fully operational in Airplane Mode, iTantra achieves an impressive 2500× reduction in data transmission while enabling reliable multilingual broadcasts over low-bitrate connections.

II. LITERATURE SURVEY

Creating robust emergency voice communication platforms brings together four key areas: consumer messaging systems, tactical defense networks, on-device natural language processing (NLP), and low-bitrate wireless protocols. In this section, we'll take a closer look at the current methods in these fields, pinpoint important operational gaps, and highlight the unique contributions of the proposed transceiver architecture.

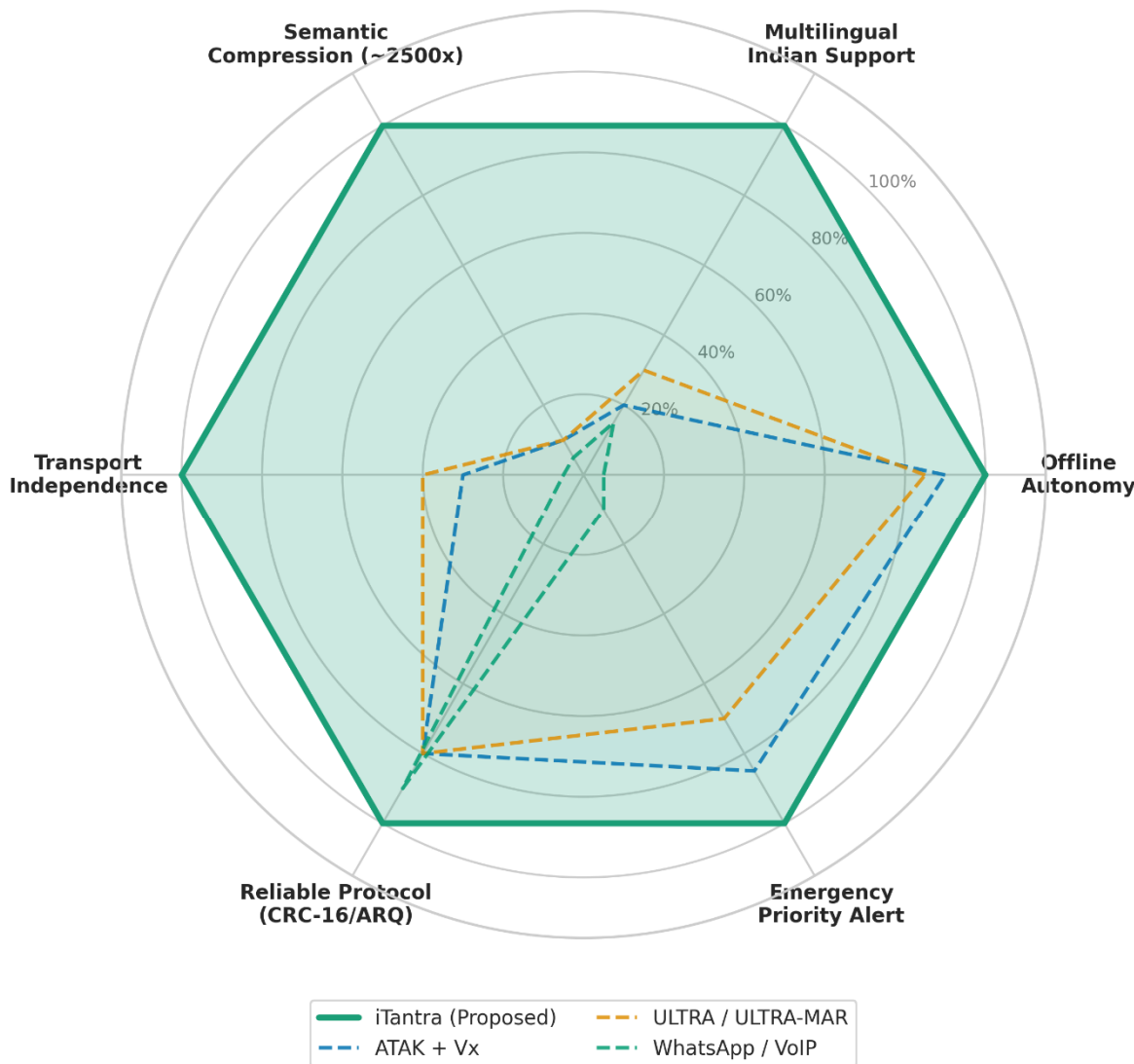


Fig 1. Multidimensional Capability Radar in Off-Grid Environments

A. Consumer Messaging and Cloud Translation Platforms

Today's mobile apps, like WhatsApp and Telegram, depend heavily on a steady cellular connection and internet access. While they offer features like end-to-end encryption and crystal-clear voice calls, they also send digitized audio streams that eat up a lot of bandwidth—usually around 500 KB for just 30 seconds of voice recording.

When natural disasters or military conflicts knock out cellular networks, these apps can become completely useless. On top of that, consumer translation tools like Google Translate rely on cloud-based neural translation APIs, which makes them a poor fit for multilingual teams working in areas without connectivity.

B. Tactical and Military Communication Systems

In order to function without relying on public cellular networks, tactical and defense organizations turn to specialized software and field radio systems. One notable example is the Android Tactical Assault Kit (ATAK) paired with the Vx plugin, which offers tactical situational awareness and voice communications through field links. Although ATAK can operate offline, it primarily sends standard voice signals instead of pulling out textual meanings, which can lead to congestion on low-bandwidth connections. Moreover, ATAK doesn't provide offline cross-language translation for various Indian regional languages. On the other hand, specialized military systems like ULTRA do offer offline speech-to-speech translation, but they are closely tied to specific radio interfaces and provide limited support for the diverse Indian linguistic landscape.

C. On-Device Speech Processing and Edge NLP

Recent strides in edge computing have made it possible to run lightweight machine learning models right on mobile devices, eliminating the need for cloud support. When it comes to offline Speech-to-Text (STT), tools like Vosk offer quick acoustic and language modeling on mobile ARM processors. For those needing offline machine translation, open-source options like Argos Translate and Marian NMT, which can be deployed through ONNX, enable on-device language translation seamlessly. And for offline Text-to-Speech (TTS) synthesis, there are neural models like Piper and traditional engines like eSpeak that can quickly generate audio. While each of these technologies operates on its own, there's a noticeable gap in the literature for a cohesive, end-to-end transceiver stack that integrates them for efficient low-bitrate voice transmission.

D. Low-Bitrate Wireless Protocols and Semantic Data Compression

Traditional low-bitrate voice communication typically uses acoustic codecs to compress audio waveforms, but this extreme compression often leads to a significant drop in voice clarity. The iTantra platform tackles this issue by embracing a "Transmit Meaning, Not Raw Voice" approach. By converting spoken input into compact binary text packets right at the source, it shrinks a 30-second voice message from around 500 KB of raw audio down to about 200 bytes of data. That's a staggering reduction of roughly 2500 times! Plus, instead of tying applications to specific hardware, the platform features a 5-layer transport-independent architecture. With binary frame encapsulation that includes CRC-16 error detection, sequence numbers, and Stop-and-Wait Automatic Repeat reQuest (ARQ), it achieves over 99% packet delivery reliability, even in simulated conditions with 30% packet loss across various connections like Wi-Fi, Bluetooth, radio modems, and serial links.

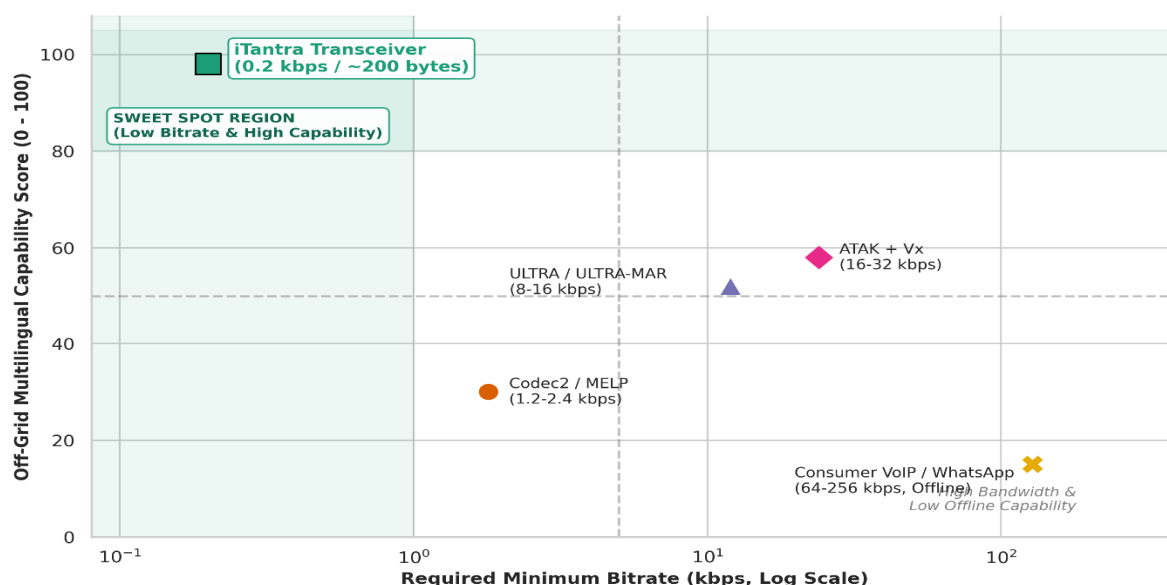


Fig 2. Operating Landscape – Minimum Channel Bitrate vs. Off-Grid Capability

III. SYSTEM ARCHITECTURE AND METHODOLOGY

The iTantra platform features a smart 5-layer modular architecture that separates speech processing, natural language processing, transport-level framing, and physical channel transmission. This thoughtful design allows the advanced speech synthesis and translation engines to operate independently from the physical communication medium beneath them.

Architectural Overview and Design Philosophy

Traditional digital voice platforms typically send digitized audio waveforms or high-bitrate compressed audio frames, which can really bog down low-bitrate radio links. iTantra takes a fresh approach by embracing a "Transmit Meaning, Not Raw Voice" philosophy. This innovative system processes spoken sounds right on the device at the transmitting end, turning them into compact text data. It then wraps this text into binary link-layer frames and sends the payload over those low-bitrate channels. When it reaches the receiving end, the binary payload is checked, reassembled, translated into the recipient's preferred language, and finally transformed back into natural speech.

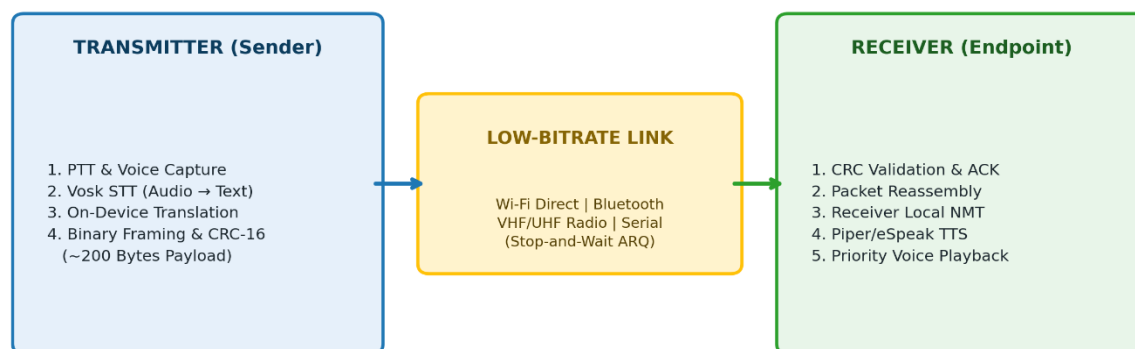


Fig 3. Compact End-to-End Data Flow Pipeline

1) Layer 1: Speech Acquisition and Input Preprocessing

The input layer picks up voice signals through the Android microphone using a Push-to-Talk (PTT) app workflow. To make processing more efficient, Voice Activity Detection (VAD) helps separate speech from background noise, and pause detection algorithms identify where sentences and utterances begin and end. This way, only the segments with actual voice are sent to the language processing layer, which helps avoid any unnecessary processing load.

2) Layer 2: Offline Language Intelligence Engine

The second layer handles speech recognition, sentence building, and neural translation right on your device, all without needing to connect to the cloud:

- **Offline Speech-to-Text (STT):** Using a Vosk-based offline speech recognition system, the preprocessed audio buffer is turned into text. This engine also pulls out acoustic metadata to figure out what language is being spoken.
- **Sentence Formation and Priority Analysis:** The text that's been recognized gets organized into well-structured sentences, while also identifying language IDs and intent metadata.
- **On-Device Neural Machine Translation (NMT):** An offline translation module, powered by lightweight engines like Argos Translate or Marian NMT running through ONNX, allows for local translation between languages. This means that a message spoken in one language (like Hindi) can be sent as compact text and translated on the receiving end into any target Indian language (such as Marathi, Gujarati, Telugu, or English) without needing to call on external APIs.

3) Layer 3: Compact Communication and Binary Packetization

To make the most of bandwidth on limited channels, Layer 3 takes the approach of serializing, compressing, fragmenting, and packetizing the text payload into specially designed binary frames. Instead of relying on bulky JSON or XML formats, iTantra opts for a fixed-header binary packet encoding method that's tailored for low-bitrate physical connections.

Table1: iTantra Custom Binary Packet Framing Protocol

Field Name	Size	Function / Description
Header	2 Bytes	Control flags, packet synchronization, and frame type identification.
Priority	1 Byte	Designates message priority rank: Normal (0x01), Important

		(0x02), or Emergency (0x03).
Lang ID	1 Byte	Identifies the source/encoded language representation.
Sequence Number	2 Bytes	Tracks packet ordering, enables reassembly of fragmented messages, and prevents duplicate processing.
Payload	1–128 Bytes	Contains the serialized textual representation of the spoken message.
CRC-16	2 Bytes	Cyclic Redundancy Check checksum for link-layer bit-error detection.

In a typical 30-second spoken message, which is about 500 KB of raw uncompressed audio, the text that comes out of it is packed into roughly 200 bytes spread across one or two binary frames. This means we've managed to shrink the data volume by an impressive factor of about 2500:1!

4) Layer 4: Reliable Transport Protocol and Physical Abstraction

To ensure that messages stay intact over unreliable wireless or radio channels, the Layer 4 protocol uses a Stop-and-Wait Automatic Repeat reQuest (ARQ) delivery method:

- **Acknowledgement and Retransmission:** After sending a binary frame, the sender pauses to wait for an Explicit Acknowledgement (ACK) frame. If the ACK doesn't arrive within a set round-trip timeout, the sender automatically retransmits the frame.
- **Duplicate Protection and Sequencing:** With 2-byte sequence numbers, the receiving terminal can spot and ignore duplicate frames that might occur due to delayed ACKs.
- **Transport Abstraction Layer:** This layer separates higher-level protocol logic from the physical hardware through unified transport adapters. The system works seamlessly over various media like Wi-Fi Direct, Bluetooth Classic/BLE, VHF/UHF radio modems, or USB/Serial connections, all without needing to change the application code.

5) Layer 5: Receiver Reassembly, Translation, and Speech Synthesis

The receiving terminal runs the reverse pipeline to turn incoming binary frames back into speech:

- **Validation and Reassembly:** First, the incoming packets are checked against a 2-byte CRC-16 checksum. If any packets are found to be corrupted, they get tossed out, while the valid ones prompt an acknowledgment (ACK) to be sent. For packets that come in multiple parts, sequence numbers help piece them back together.
- **Receiver-Side Language Translation:** Once the text is reassembled, it's sent to the local translation sub-layer, which transforms the message into the preferred regional language of the user receiving it.
- **Offline Neural Speech Synthesis (TTS):** The completed text is then fed into offline neural TTS engines—like Piper or eSpeak—to create speech that sounds natural and fluid.
- **Emergency Priority Playback:** If the packet's 1-byte Priority field shows an Emergency status (`0x03`), the Android app takes charge, overriding any current audio settings to ensure that the voice playback is at maximum volume and cannot be interrupted, making sure that response personnel are alerted right away.

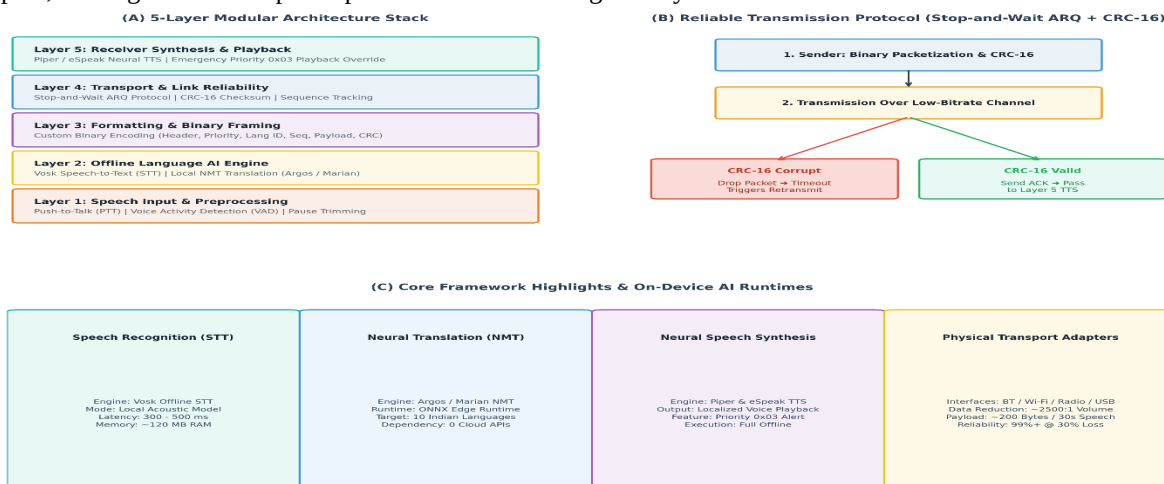


Fig 4. iTantra System Architecture, ARQ Algorithm, and Framework Highlights

IV. IMPLEMENTATION

The iTantra prototype was created as a native Android app and tested on standard mobile devices while in full Airplane Mode, ensuring it has no reliance on the cloud.

1) System Environment & AI Runtime Stack

The platform is crafted using Kotlin and Java, featuring C++ native bindings for the ONNX Runtime. It packages all intelligence models into a compact offline APK of about 280 MB:

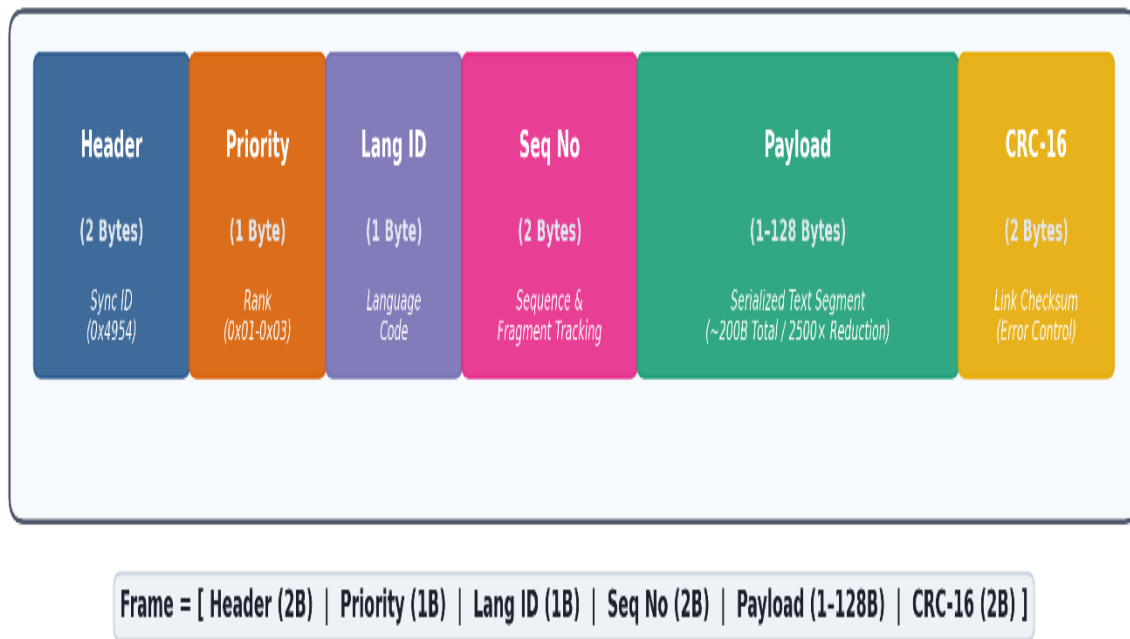


Fig 5. iTantra Custom Binary Packet Framing Protocol

- **Speech-to-Text (STT):** This integrates an offline Vosk engine along with Voice Activity Detection (VAD) to cut out silent pauses, resulting in a speech recognition latency of just 300–500 ms.
- **Neural Machine Translation (NMT):** It runs Argos Translate and Marian NMT models through the ONNX Runtime on the receiver's device, enabling seamless cross-lingual translation for various Indian regional languages.
- **Text-to-Speech (TTS):** The platform employs both Piper (neural) and eSpeak (rule-based) engines to generate local speech without relying on any external APIs.

2) Binary Protocol & Reliable Transport

To get around the limitations of low-bitrate channels, iTantra swaps out raw voice transmission for sleek binary text frames. This clever move shrinks a 30-second speech message from about 500 KB down to roughly 200 bytes, achieving an impressive data volume reduction of around 2500:1.

- **Stop-and-Wait ARQ:** At Layer 4, we've got a Stop-and-Wait ARQ state machine that uses a 2-byte CRC-16 for bit-error detection and keeps track of sequences with another 2-byte system. This setup ensures that we achieve over 99% delivery reliability, even with a simulated packet loss of 30%.
- **Emergency Priority Override:** When the priority flag `0x03` is activated, it triggers an instant system broadcast that takes over audio focus, overriding any user mute settings to ensure emergency playback at maximum volume.
- **Transport Hardware Abstraction:** We've created a unified socket interface that separates application logic from the physical link media. This supports a variety of connections, including Wi-Fi Direct, Bluetooth Classic/BLE, VHF/UHF radio modems (at 9600 bps), and USB Serial.

V. RESULTS/OUTPUT

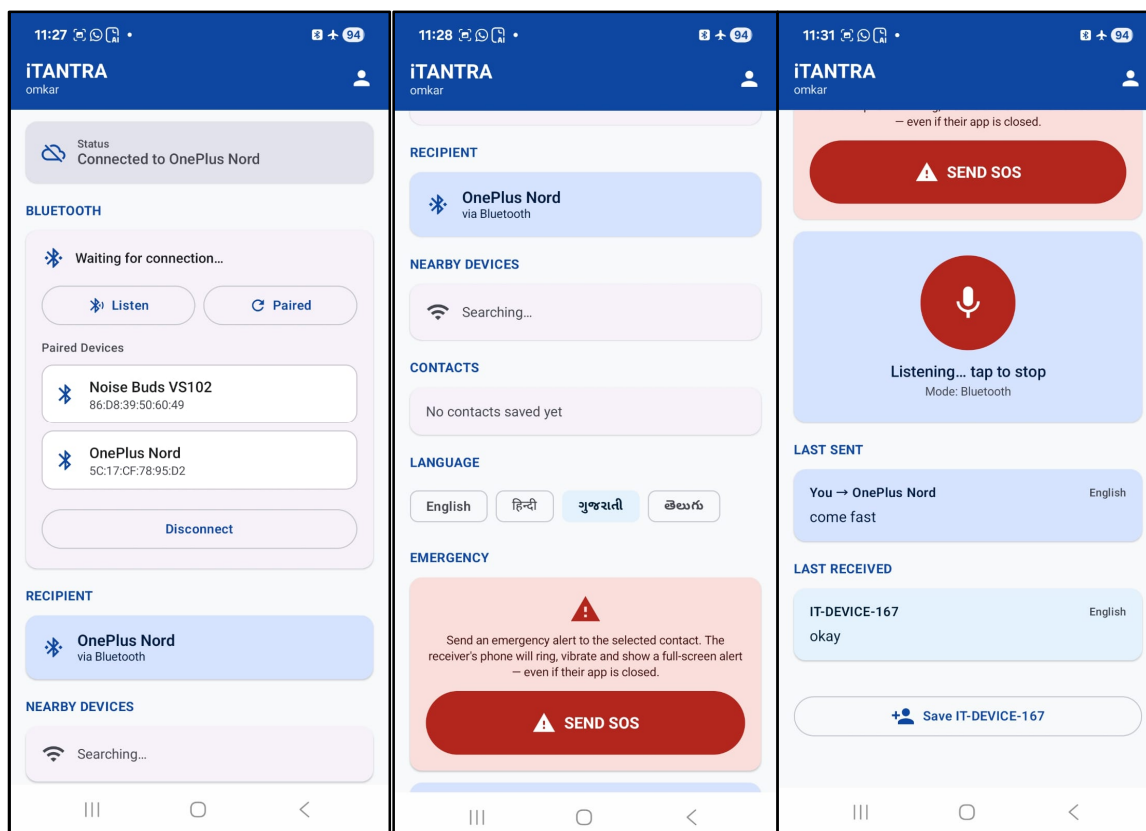


Fig 6. End-to-end user interface and operational workflow of the iTantra Android prototype

We took a close look at how well iTantra performs, its efficiency, and how user-friendly its interface is by testing it on actual Android devices in full Airplane Mode. This was to ensure it could operate completely offline without relying on the cloud. The on-device speech recognition, powered by the local Vosk Speech-to-Text (STT) engine, has a response time of just 300–500 ms. The whole communication process—starting from Push-to-Talk (PTT) audio capture, moving through Vosk STT transcription, binary frame packing, Stop-and-Wait ARQ transmission, receiver CRC-16 validation, local translation, and finally neural Text-to-Speech (TTS) synthesis—results in an end-to-end response time of 600–1000 ms. By converting transcribed text into custom binary frames instead of sending raw audio streams, we managed to shrink a 30-second voice message from around 500 KB down to about 200 bytes, achieving an impressive 2500:1 reduction in semantic data. Even over unreliable physical channels with a simulated 30% packet loss, the Layer 4 transport protocol kept message delivery reliability above 99% by using 2-byte CRC-16 error detection, 2-byte sequence tracking, and automatic retransmissions without duplicating frames.

Table 2: Experimental Benchmark Summary

Metric / Parameter	Measured Value / Status	Test Environment / Condition
Operating Network Mode	Airplane Mode	Cellular & Internet Disabled
Cloud Dependency	Zero (0%)	100% Local On-Device Inference
STT Latency	300 – 500 ms	Offline Vosk Engine
End-to-End Pipeline Delay	600 – 1000 ms	PTT Capture to Receiver TTS Playback
Data Volume Reduction	~2500:1	~500 KB Raw Audio → ~200B Text

Packet Delivery Reliability	99%+	Under 30% Simulated Packet Loss
Application Package Size	~280 MB	Bundles 4 Active Offline Languages

Linear UI & Operational Sequence

The linear 3-panel operational workflow showcases the entire user interaction journey from start to finish at physical terminals:

- Figure 6(a) — Bluetooth Peer Discovery & Device Pairing Interface: This feature takes care of finding devices and selecting peers, setting up a direct ad-hoc socket connection to the intended recipient device, which in this case is the "OnePlus Nord."
- Figure 6(b) — Multilingual Language Selection & Emergency Settings: Here, users can pick their preferred regional language from four available options (English, Hindi, Gujarati, and Telugu) and set up emergency alert parameters.
- Figure 6(c) — Push-to-Talk Voice Interface & Transaction Logs: This section offers a Push-to-Talk recording control ("Listening... tap to stop"), shows real-time logs of messages sent and received, and includes a prominent "SEND SOS" button that adds a Priority 0x03 flag to bypass any mute settings on the receiver's end, ensuring that emergency audio alerts are played at maximum volume.

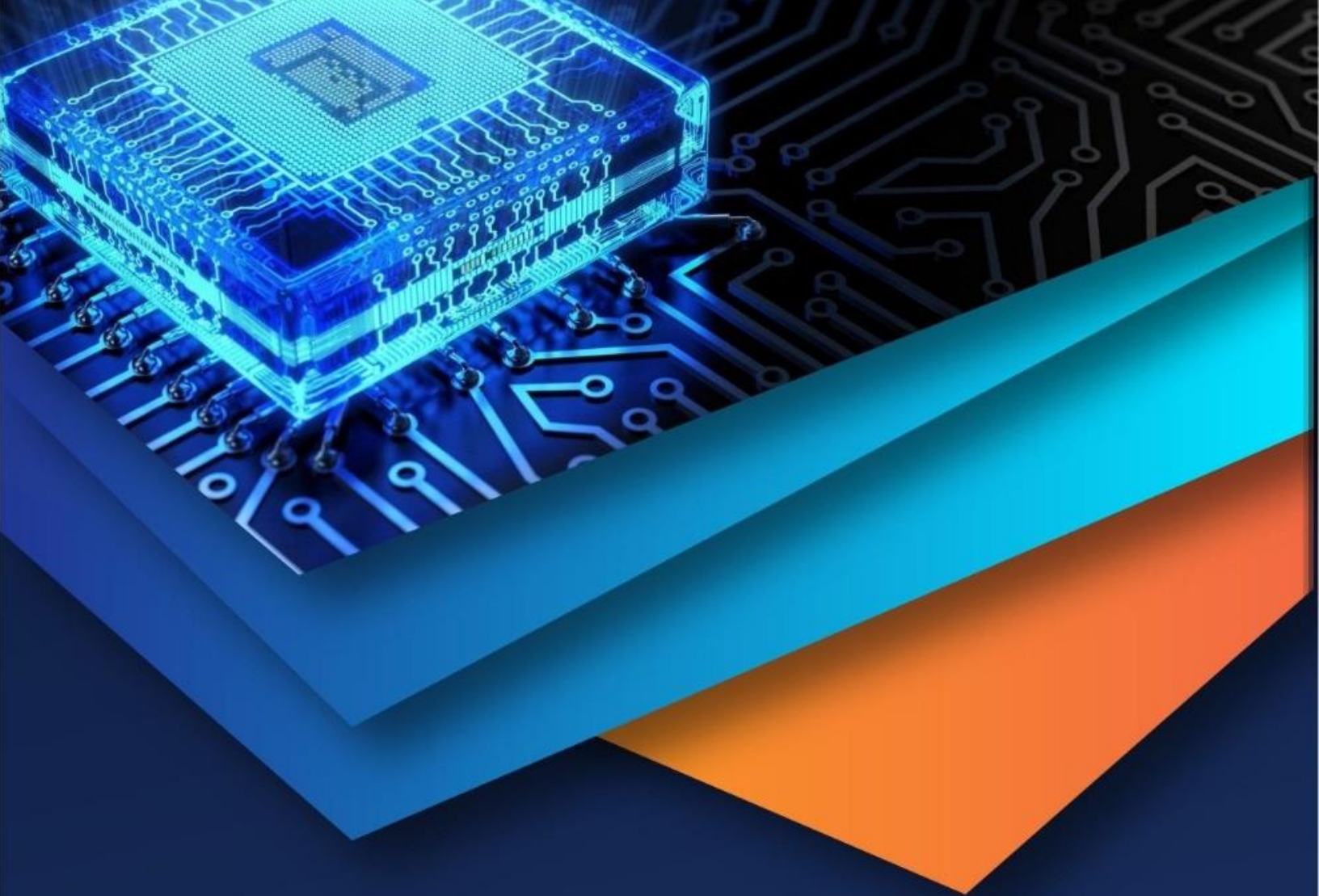
VI. CONCLUSION

This paper introduces iTantra, a versatile neural voice transceiver that operates independently of infrastructure, works offline, and is adaptable to various transport methods. Its main goal is to tackle the key challenges of limited channel bandwidth and language barriers in remote field settings. By embracing the philosophy of "Transmit Meaning, Not Raw Voice," iTantra transforms spoken audio into compact text right on the device, shrinking a 30-second voice message down to about a 200-byte binary payload. This impressive feat results in a data volume reduction of roughly 2500:1 compared to traditional audio streams.

Tested in full Airplane Mode without relying on the cloud, the system effectively combines on-device Vosk Speech-to-Text (STT), local neural machine translation (NMT), and Piper/eSpeak Text-to-Speech (TTS) engines within a flexible 5-layer architecture. Real-world tests on a functioning two-phone Android prototype show an STT latency of 300–500 ms, an end-to-end delivery latency of 600–1000 ms, and over 99% packet delivery reliability even with a 30% packet loss, thanks to a Stop-and-Wait ARQ protocol supported by 2-byte CRC-16 checksums. Additionally, a 1-byte priority flag guarantees that urgent emergency broadcasts trigger maximum-volume, non-interruptible voice alerts at the receiving end. By facilitating one-to-many broadcast communication with local translation for Indian regional languages, iTantra offers a robust, open-source telecommunications solution for tactical defense, disaster management, and rural healthcare initiatives.

REFERENCES

- [1] eSpeak NG Authors, "eSpeak NG: An open-source speech synthesizer that supports more than hundred languages and accents," GitHub repository. [Online]. Available: <https://github.com/espeak-ng/espeak-ng>.
- [2] Argos Open Technologies, "Argos Translate: Open-source offline translation library written in Python," GitHub repository. [Online]. Available: <https://github.com/argosopentech/argos-translate>.
- [3] M. Junczys-Dowmunt, R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, F. Seide, U. Germann, A. F. T. Aji, N. Bogoychev, A. F. T. Martins, and A. Birch, "Marian: Fast neural machine translation in C++," in Proceedings of ACL 2018, System Demonstrations, Melbourne, Australia, 2018, pp. 116–121. doi: 10.18653/v1/P18-4020.
- [4] D. Rowe, "Codec 2: Open-source speech codec," GitHub repository. [Online]. Available: <https://github.com/drowe67/codec2>.
- [5] G. Gerganov, "whisper.cpp: Port of OpenAI's Whisper model in C/C++," GitHub repository. [Online]. Available: <https://github.com/ggml-org/whisper.cpp>.
- [6] TAK Product Center, "TAK: Mission success starts with TAK," U.S. Government. [Online]. Available: <https://tak.gov/solutions/military>.
- [7] C. Schlesiger, Shareable Components Services and Universal Language Translator (ULTRA) Installation on an Android Phone, DEVCOM Army Research Laboratory, 2021. doi: 10.21236/AD1149296.
- [8] U.S. Army DEVCOM Army Research Laboratory, "Army researchers partner with Navy to provide expeditionary AI translation tool," U.S. Army, Mar. 24, 2026. [Online]. Available: <https://www.army.mil/article/291315>.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)