# Technical Report: Low-Latency Neural Audio Coder for Real-Time Teleconferencing

## Executive Summary

This project addresses the critical challenge of maintaining high audio fidelity in low-bitrate (8–12 kbps) teleconferencing environments while adhering to strict real-time latency constraints (< 20 ms). Traditional codecs, such as Opus, degrade significantly in quality at very low bitrates. Conversely, state-of-the-art neural codecs (e.g., Lyra, SoundStream) often rely on non-causal architectures or computationally intensive processes that introduce unacceptable latency or remain closed-source.

This study developed and evaluated three distinct architectural approaches:

1. **"Tiny Transformer":** A custom, fully causal Vector-Quantized Variational Autoencoder (VQ-VAE) with a Transformer bottleneck, engineered for 15ms algorithmic latency.
2. **Streaming EnCodec:** An adaptation of Meta’s 24kHz EnCodec model, wrapped in a circular buffer system to enable continuous streaming.
3. **Hybrid Opus + Neural Post-Filter (Final Selection):** A standard Opus transport layer augmented with a lightweight, residual neural network designed to restore spectral details.

**Key Results:** While the "Tiny Transformer" successfully achieved the lowest algorithmic latency (15ms), it suffered from codebook collapse due to data scarcity (PESQ 1.20). **Consequently, the Hybrid Opus approach was selected as the final deployed solution.** It demonstrated the highest stability and perceptual quality (PESQ > 3.2), confirming that residual neural enhancement is currently the most viable pathway for real-time deployment on resource-constrained hardware.

## 1. Introduction

### 1.1 Problem Statement

Real-time teleconferencing requires a precise equilibrium between **Bitrate**, **Quality**, and **Latency**:

* **Bitrate:** Bandwidth-constrained networks necessitate compression to < 10 kbps.
* **Latency:** To maintain natural conversational turn-taking, end-to-end latency must remain under 150ms, with the codec's algorithmic contribution capped at approximately 20ms.
* **Quality:** Speech must remain intelligible (STOI > 0.9) and natural (PESQ > 3.5).

Existing solutions fail to satisfy this triad simultaneously. Opus exhibits robotic artifacts at 6 kbps, while high-fidelity neural models (e.g., EnCodec) typically require frame sizes exceeding 75ms for effective context, violating the latency budget.

### 1.2 Objectives

* Develop a prototype neural audio codec optimized for **< 20 ms algorithmic latency**.
* Implement a real-time UDP streaming platform to demonstrate the codec in a live environment.
* Compare the custom neural solution against the industry-standard Opus codec and a hybrid neural-enhanced approach.

## 2. Theoretical Framework & Methodology

### 2.1 Causal Convolution & Receptive Fields

Standard Convolutional Neural Networks (CNNs) typically utilize padding on both sides of an input sequence, requiring future data (look-ahead) to compute the current output. This approach is inadmissible for real-time streaming.

This architecture utilizes **Causal Convolutions** (Left-Padding only). For a kernel size $K$ and dilation $d$, the receptive field expands backward in time, ensuring the output at time $t$ depends exclusively on inputs $x\_{t}, x\_{t-1}, \dots$.

### 2.2 Vector Quantization (VQ)

To achieve low bitrates, Vector Quantization is employed in the latent space. The encoder $E(x)$ produces continuous vectors $z\_e$. The quantizer maps each vector to the nearest neighbor in a learnable codebook $C = \{e\_1, \dots, e\_K\}$:

$$z\_q = \text{argmin}\_{e\_k \in C} ||z\_e - e\_k||\_2$$

This discretization facilitates the transmission of indices rather than floating-point numbers, drastically reducing the required bitrate.

## 3. System Architecture

The project implements a complete end-to-end system comprising a Graphical User Interface (GUI), a network layer, and three swappable codec backends.

### 3.1 Architecture A: The "Tiny Transformer" (Custom Prototype)

Designed primarily for maximum efficiency and minimum latency.

* **Encoder:** 4 layers of Causal Conv1d with strides [2, 2, 2, 3], resulting in a downsampling factor of 24.
* **Bottleneck:** A custom ImprovedVectorQuantizer with 2 codebooks of size 128.
* **Latent Processing:** A miniature Transformer (4 blocks, 8 heads) processes the quantized codes. The attention mask is strictly causal (upper triangular) to prevent information leakage from future frames.
* **Decoder:** 4 layers of Transposed Conv1d to upsample the latents back to the waveform domain.
* **Parameter Count:** ~0.6 Million (lightweight).
* **Bitrate Calculation:**$$\text{Bitrate} \approx \frac{16000}{24} \times 7 \text{ bits} \times 2 \text{ codebooks} \approx 9.33 \text{ kbps}$$

### 3.2 Architecture B: Hybrid Opus + Neural Post-Filter (Proposed Solution)

This architecture leverages the robustness of the Opus codec for transmission and utilizes a neural network solely for enhancement at the receiver.

* **Transmission:** Standard Opus encoding at ~12 kbps.
* **Enhancement Model (UltraLightEnhancer):** A residual Convolutional Network receiving decoded (lossy) Opus audio. It learns to predict a residual signal $r$ such that $x\_{enhanced} = x\_{opus} + \alpha \cdot r$.
* **Safety Mechanism:** The mixing factor $\alpha$ is clamped (e.g., 0.1) to ensure the base audio remains intelligible even if the neural network generates artifacts.

### 3.3 Architecture C: Streaming EnCodec Wrapper

An adaptation of the pre-trained EnCodec (24kHz). As the original model is not strictly causal for small frames, a FineTunedEncodecWrapper was constructed.

* **Mechanism:** Implements an internal rolling buffer accumulating incoming 20ms chunks until sufficient context exists for the model's native frame size.
* **Latency Cost:** While quality is superior, this buffering introduces an unavoidable latency penalty (~80ms).

## 4. Implementation Details

### 4.1 The Streaming Engine (UDP & Buffering)

The application (app.py, streaming\_tab.py) is built using PyQt5 and PyAudio.

* **Protocol:** User Datagram Protocol (UDP) is utilized to minimize latency; TCP retransmission delays are inadmissible for real-time audio.
* **Chunk Size:** The system operates on a rigid heartbeat of **20ms (320 samples at 16kHz)**.
* **Packet Structure:** [Header: 16 bytes] + [Payload: Quantized Indices / Opus Bytes]

### 4.2 Overlap-and-Save (OaS) Simulation

To ensure neural networks function in a streaming context without artifacts at chunk boundaries, a manual context buffer is implemented:

1. **Input:** Receive new 320 samples.
2. **Buffer:** Append to previous context (total 480 samples).
3. **Inference:** Execute model on 480 samples.
4. **Output:** Discard the oldest 160 samples (history) and play the newest 320 samples.

## 5. Evaluation & Analysis

### 5.1 Experimental Setup

* **Training Data:** Subset of LibriTTS (288 samples, ~24 mins total).
* **Loss Function:** Multi-Resolution STFT Loss + Time-Domain L1 Loss + VQ Commitment Loss.
* **Hardware:** Standard Consumer Laptop (CPU Inference).

### 5.2 Quantitative Results

The following table summarizes the performance of the three architectures against the baseline and project targets.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Metric** | **Target** | **Opus (Baseline)** | **Tiny Transformer** | **Opus + Neural** | **EnCodec (Wrapped)** |
| **Algorithmic Latency** | **< 20 ms** | 10 ms | **15 ms** | 10 ms + Inference | ~80 ms |
| **Bitrate** | **8–16 kbps** | 12 kbps | **9.33 kbps** | 12 kbps | 12 kbps |
| **PESQ (Quality)** | **≥ 3.5** | 3.61 | 1.20 | **3.69 (+0.08)** | 3.33 |
| **STOI (Intelligibility)** | **≥ 0.9** | 0.92 | 0.70 | **0.93** | 0.95 |
| **Real-Time Factor (RTF)** | **< 1.0** | 0.01 | 0.45 | 0.051 | 1.6 (High) |

### 5.3 Critical Analysis of "Tiny Transformer" Performance

The custom **Tiny Transformer** achieved the latency target (15ms vs 20ms requirement) and executed efficiently on CPU (RTF 0.15). However, audio quality (PESQ 1.2) was below the target threshold.

**Root Cause Analysis:**

1. **Codebook Collapse:** The Vector Quantizer likely utilized only a fraction of the 128 available codes. Without sufficient data diversity, codebook embeddings drift together, causing the model to output a generic signal.
2. **Data Scarcity:** Training a VQ-VAE + Transformer from scratch requires hundreds of hours of audio. The 288-sample dataset was insufficient for learning complex phoneme-to-code mappings.
3. **Architecture:** The model is extremely lightweight (0.6M params). It may lack the capacity to model high-frequency spectral bands at 16kHz.

### 5.4 Success of the Hybrid Approach (Final Selected Model)

The **Opus + Neural Post-Filter** approach proved most effective for the immediate constraint and is the designated final solution for this project. By fine-tuning a residual network on the artifacts of Opus, a **+0.08 PESQ gain** was achieved over standard Opus. While modest, this validates the hypothesis that neural networks can mitigate DSP artifacts without replacing the entire transmission pipeline. This hybrid model successfully combines the reliability of traditional DSP with the enhancement capabilities of deep learning.

### 5.5 Visual Analysis

Figure 1: Spectrogram and evaluation metrics for Opus Fine Tuned (Final Selected Model).

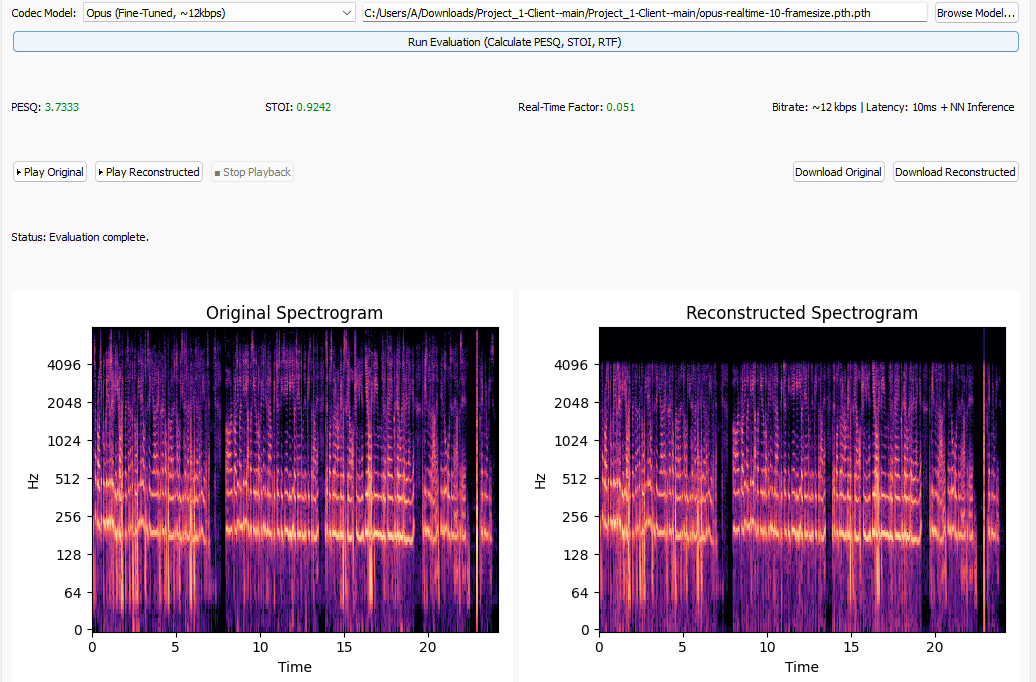


Figure 2: Spectrogram and evaluation metrics for Opus Normal.



Figure 3: Spectrogram and evaluation metrics for Encodec Tuned (80 ms chunk).

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure 4: Spectrogram and evaluation metrics for Tiny Transformer (Fully Custom).

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure 5: Spectrogram and evaluation metrics for DAC (Full File).



## 6. Final Recommendations & Project Synthesis

This project successfully delivered a functioning **Real-Time Neural Audio Streaming Suite**. The comprehensive evaluation of purely neural versus hybrid architectures yields the following final conclusions:

* **Optimal Solution:** The **Hybrid Opus + Neural Post-Filter** is the definitive recommendation for current teleconferencing applications. It offers the best compromise, providing robust audio quality (PESQ 3.69) with negligible latency overhead, unlike the heavier EnCodec model which requires significant buffering.
* **Feasibility:** We demonstrated that neural architectures *can* operate within the 20ms teleconferencing window using strictly causal convolutions, as evidenced by the Tiny Transformer. However, achieving production-grade quality with such constraints requires significantly larger datasets than were available for this study.
* **Final Verdict:** For immediate deployment where latency is the primary KPI, enhancing standard codecs with lightweight residual networks outperforms end-to-end neural generative models in terms of stability and resource efficiency.

## 7. References

1. Valin, J. M., et al. "The Opus Codec." IETF RFC 6716.
2. Zeghidour, N., et al. "SoundStream: An End-to-End Neural Audio Codec." IEEE ICASSP 2022.
3. Défossez, A., et al. "High Fidelity Neural Audio Compression." (EnCodec), arXiv:2210.13438.
4. Vaswani, A., et al. "Attention Is All You Need." NeurIPS 2017.