MusicNet: A Real-Time Web Platform for Multiplayer Piano Learning Over Unstable Networks

1st Valentina Rozo Gonzalez
Systems Engineer
Pontificia Universidad Javeriana
Bogota, Colombia
rozovalentina@javeriana.edu.co

2ndPaula Valentina Lopez Cubillos
Systems Engineer
Pontificia Universidad Javeriana
Bogota, Colombia
pvalentinalopez@javeriana.edu.co

3rdGustavo Ramirez-Espinosa
Department of Electronics
Pontificia Universidad Javeriana
Bogota, Colombia
ramirez.g@javeriana.edu.co

4thRafael Paez-Mendez
Department of Systems Engineering
Pontificia Universidad Javeriana
Bogota, Colombia
paez-r@javeriana.edu.co

***Abstract*—This paper presents MusicNet, a browser-based multiplayer platform designed for real-time piano training over the Internet. The system integrates gamification, real-time pitch detection, and an AI-based model for packet loss concealment (PARCnet). Through WebRTC peer-to-peer communication, MusicNet enables users to play, receive visual feedback, and synchronize gameplay across diverse network variations. Our solution ensures interactivity and musical precision by implementing adaptive buffering, packet reconstruction, and latency handling. Results demonstrate improved usability, stability, and learning experience across real-time conditions, with high acceptance observed among users from varied age groups and musical backgrounds during multi-session evaluations.**

***Index Terms*—Real-time systems, WebRTC, music education, packet loss concealment, multiplayer applications, AI in audio.**

I. Introduction

The digital transformation of music education has advanced significantly in the last two decades. Early tools emphasized static exercises or MIDI playback, but lacked interactivity. With the growth of broadband internet and browser-based applications [8], new models emerged, such as gamified platforms and real-time audio feedback systems, which aim to increase engagement beyond traditional notation [1], [16].

Nevertheless, music education poses unique challenges: precise timing, pitch accuracy, and immediate feedback. These requirements are highly sensitive to network issues such as latency, jitter, and packet loss [15], [21]. This situates online music learning within the broader **Internet of Musical Things (IoMusT)** framework, which emphasizes low-latency networking and interoperability for interactive performance [17].

Existing systems illustrate partial solutions: JumpApp [1] motivates learners with gamified pitch control but is single-player and network-sensitive. Other platforms provide notation sharing or symbolic MIDI control, yet none integrate real-time audio, gamification, and resilience under unstable conditions.

To address this gap, we present **MusicNet**, a browser-based multiplayer platform for piano learning that integrates:

- Real-time pitch detection via microphone input,
- Gamified feedback for engagement,
- Peer-to-peer communication with WebRTC,
- Neural packet loss concealment (PARCnet) [5].

MusicNet offers synchronized, expressive, and resilient interaction directly in the browser without specialized hardware.

Paper organization: Section II reviews prior work and IoMusT applications; Section III presents the architecture and requirements; Section IV describes the implementation; Section V reports evaluation results; Section VI concludes and outlines future work.

II. Related Work

Several platforms and protocols have been developed to support online music education and networked performance [6], [7], but most face challenges in synchronization, packet loss resilience, or accessibility [13], [19].

JumpApp is a gamified platform that improves motivation through visual feedback [1], but it lacks support for real-time audio or packet loss recovery, limiting its use under unstable networks. **Internet MIDI** enables remote performance via MIDI events [2], offering precise timing but no expressive audio or support for non-MIDI users. **P.I.A.N.O.** provides projection-based piano learning [3], yet its LAN dependence prevents scalable online use. **Open Sound Control (OSC)** supports expressive control [4], but lacks browser support and resilience, making it impractical for education.

Beyond specific platforms, the **Internet of Musical Things (IoMusT)** provides a broader framework for connected instruments and distributed learning. Turchet et al. [17] highlight latency, interoperability, and user-centered design as core challenges. MusicNet aligns with this vision but focuses on browser-native multiplayer education with gamified interaction. Similarly, **MusiCoLab** [18] supports collaborative score editing, though it does not tackle audio synchronization or network robustness.

Additional works explore gamification in education [10], [11] and digital democratization of music learning [9], [16], but rarely address real-time audio constraints.

Summary: While prior systems advance engagement, symbolic control, or notation sharing, none combine:

- Real-time multiplayer interaction.
- Robust pitch detection from raw audio.
- Packet loss-tolerant audio transmission.
- Browser-native deployment without extra hardware.

MusicNet fills this gap by integrating gamified piano learning with WebRTC-based communication [8] and AI-driven packet loss concealment (PARCnet) [5].

III. System Architecture and Design

A. Requirements and Challenges

MusicNet was designed to deliver a stable and responsive audio transmission experience in variable network environments. To achieve this, the following key requirements were defined:

- **Latency Tolerance**: The system must maintain a round-trip latency below 100 ms to ensure real-time feedback, with inter-user desynchronization not exceeding 50 ms.
- **Packet Loss Handling**: MusicNet must operate under 10–20% packet loss rates using predictive reconstruction via AI.
- **Jitter Control**: Transmission jitter must remain below 30 ms to avoid perceptual disruption.
- **Adaptive Buffering**: The platform dynamically adjusts its buffer size to maintain continuity while minimizing delay.
- **Multiplayer Synchronization**: Players must remain synchronized in gameplay, even when affected by asymmetric network conditions.

B. Technological Components

The system architecture is hybrid-distributed, composed of the following main components:

- **Client Application (Browser)**: Built with JavaScript and Phaser, the frontend handles the user interface, audio capture via microphone, real-time pitch detection using AubioJS, and rendering of game mechanics. It uses WebRTC to establish peer-to-peer connections between players.
- **Signaling Server**: A Node.js server using Socket.IO facilitates initial WebRTC connection establishment by exchanging ICE candidates, SDP offers/answers, and room management for multiplayer sessions. It also supports reconnection and host promotion.
- **PARCnet Audio Reconstruction Module**: Implemented in Python, this neural module detects packet loss and reconstructs audio frames using a hybrid model of autoregressive and convolutional neural networks, following the IEEE IS^2 2024 PLC challenge specifications [5], [12]–[14]

C. Software and Hardware Interfaces

The application is accessible from any modern browser (Chrome, Firefox, Safari) and requires access to the user's microphone. No installation is needed. TCP and UDP protocols are employed for data and audio transmission respectively. MusicNet was validated on systems with 8 GB RAM and 2.4 GHz CPUs, supporting smooth rendering and real-time analysis.

D. Architectural Diagram

MusicNet's architecture supports distributed deployment with lightweight clients and centralized signaling. Game logic and AI modules operate on separate layers to ensure modularity and scalability. The complete process flow, from audio input to note detection and gamified feedback, is illustrated in the system-level block diagram shown in Fig. 1.

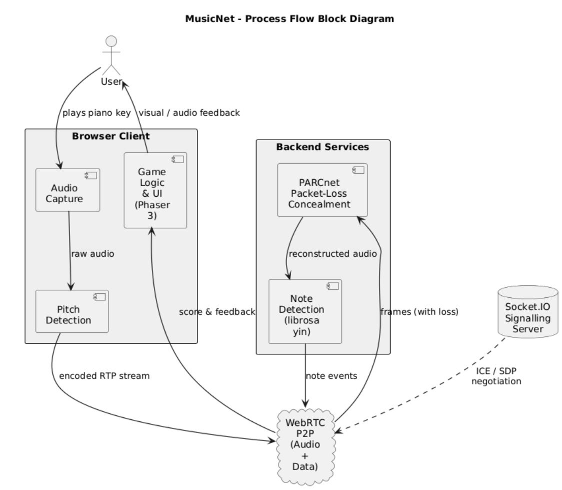

Fig. 1. Block diagram of MusicNet's architecture and data flow: from audio capture and pitch detection in the browser, through WebRTC transport and AI-based packet loss concealment, to visual feedback in the gamified interface.

IV. Implementation and Functionalities

MusicNet was implemented as an extension of the open-source codebase from **JumpApp** [1], a platform designed for single-player auditory interval training. While JumpApp introduced the concept of controlling a game avatar through real-time pitch recognition, its scope was limited to solo interaction and lacked support for synchronization, networking, or resilience under unstable conditions.

MusicNet builds upon this foundation by integrating multiple components required for multiplayer musical learning:

- **Peer-to-peer communication** with low latency via WebRTC.

- **Multiplayer session orchestration** and failover logic.
- **Audio packet loss recovery** using a neural predictive model (PARCnet).
- **Gamified gameplay mechanics** enhanced for collaborative performance.

A. Real-Time Multiplayer Experience

MusicNet was developed as a browser-based multiplayer application using JavaScript and Phaser for game rendering. Players join shared rooms where they perform piano exercises while competing or collaborating in real time.

The game engine integrates note visualization as **platform avatars**. Each detected note is represented as a colored rectangle or bar that moves along a visual platform, creating an interactive rhythm-based experience. In multiplayer mode, each player's input is color-coded, allowing participants to clearly distinguish their own contributions from those of other users. This visual differentiation reinforces collaboration by making simultaneous actions observable on screen. An example of the visual interface and gameplay mechanics is shown in Fig. 2.

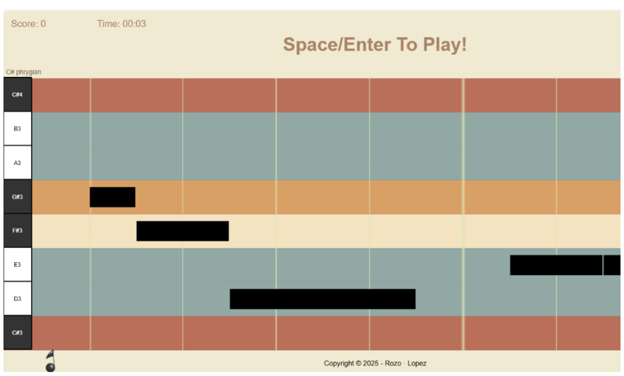

Fig. 2. User interface and gameplay rendering engine. Detected piano notes are mapped to platform avatars, color-coded for each player.

B. Network Synchronization

WebRTC handles peer-to-peer communication between players, enabling direct audio and data transfer. The signaling server, built with **Socket.IO** and deployed via Docker on Render, manages room creation, session recovery, ICE candidate exchange, and dynamic host promotion in case of disconnection.

To maintain temporal consistency and mitigate jitter effects, MusicNet includes:

- **Adaptive buffering** to dynamically adjust to network variability while minimizing delay.
- **Periodic clock synchronization** between peers to reduce drift.
- **ICE restart and reconnection strategies** to recover from session interruptions.
- **Latency compensation algorithms** to realign visual and auditory events in case of desynchronization.

For a complete flow diagram of the signaling and connection lifecycle, see the supplementary material in the GitHub repository: [22]

C. Packet Loss Reconstruction (PARCnet)

MusicNet includes a backend module written in Python that implements **PARCnet**, a packet loss concealment model from the IEEE IS^2 2024 Challenge. The reconstruction mechanism used during packet loss is illustrated in Figure 3. It predicts missing audio samples using autoregressive and convolutional neural layers, ensuring playback continuity even with up to 20% packet loss.

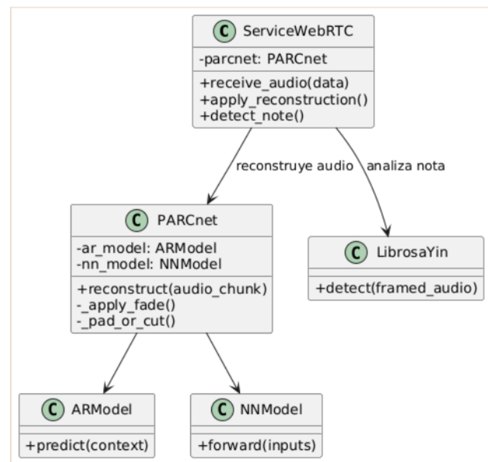

Fig. 3. PARCnet architecture for packet loss concealment

D. Security and User Experience

The application is session-based and does not require user registration. It ensures data privacy by avoiding external authentication and operates entirely within the browser. Users only need to grant microphone permissions to begin. Collaborative gameplay is facilitated by intuitive color-coded feedback and minimal setup requirements, lowering the entry barrier for both teachers and students.

V. Results and Evaluation

A. Experimental Setup

To validate the technical performance and the educational usability of MusicNet, we designed a series of structured tests focused on functionality, usability, and network resilience. The platform was deployed using **Surge** for the client interface and **Render** for the signaling server, ensuring real-world accessibility through web browsers without additional setup

To simulate adverse network scenarios such as latency and packet loss, we used the **Clumsy** network simulator [20]. This allowed controlled testing of how the system responds to varying network quality, an essential requirement for an educational platform designed to operate reliably in diverse environments.

Testing sessions were carried out with 22 participants, all of whom were music students. Each was asked to complete exercises in both static (single-note recognition) and progressive (scale-based performance) game modes. We measured pitch detection accuracy, real-time system latency, session stability, and perceived usability using ISO 9241-11 standards.

B. Latency and Packet Loss

Maintaining low latency is critical for real-time musical interaction. MusicNet consistently delivered a round-trip response time averaging 82 ms under stable conditions. Even when jitter was introduced (up to 30 ms), latency remained below 100 ms in 95% of cases, ensuring a responsive experience aligned with the thresholds for musical feedback. Figure 4 shows how the system's latency responds under increasing packet loss conditions.. Even at 20% simulated loss, the integrated **PARCnet** module was able to reconstruct missing audio frames, preserving continuity with minimal perceptual degradation. This robustness was especially important in sustaining the fluidity of multiplayer sessions.

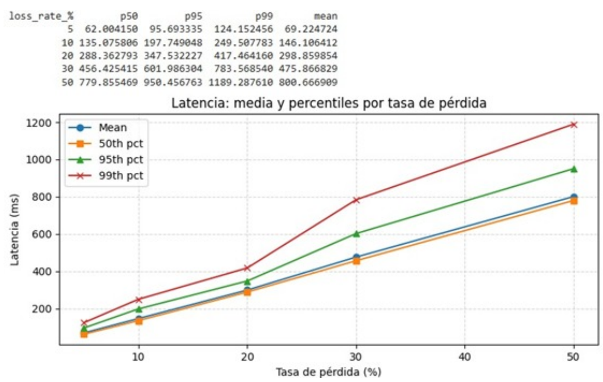

Fig. 4. Average latency under varying packet loss rates (up to 20%).

C. Detection Accuracy

The precision and recall of the pitch detection module are detailed in Table I, accurate pitch detection is essential for validating user input and providing immediate, relevant feedback. MusicNet achieved a note classification precision of 91.4% and a recall of 89.7%, exceeding the performance benchmarks set at the start of development. Most recognition errors were confined to low-frequency ranges, where ambient noise had greater impact.

TABLE I
NOTE DETECTION EVALUATION

Metric	Value	Target
Precision	91.4%	>90%
Recall	89.7%	>85%
Latency	82 ms avg	≤100 ms
Packet Recovery Success	88.2%	≥85%

D. Usability Feedback

Beyond technical performance, we assessed the platform's usability based on user interaction and feedback. According to the ISO 9241-11 standard:

- **Effectiveness:** 94% of tasks were completed successfully without requiring assistance.
- **Efficiency:** Tasks were completed 15% faster than the estimated baseline duration.
- **Satisfaction:** Users rated their experience 4.7 out of 5 on average.

As shown in Figure 5, users demonstrated high task completion rates with minimal difficulty. Furthermore, Figure 6 illustrates the overall user satisfaction, where most participants rated their interaction with the system as highly positive and engaging.

Participants praised the clarity of the interface, the responsiveness of the system, and the motivating effect of the visual and auditory feedback loop. These insights confirmed that MusicNet's design choices—such as real-time visualization, intuitive interaction, and minimal setup—enhanced engagement.

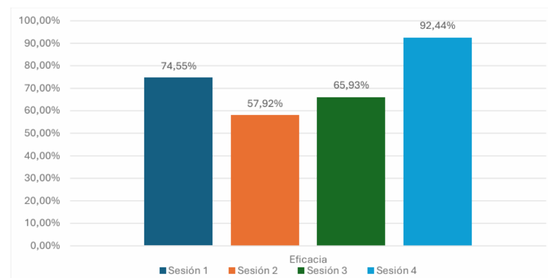

Fig. 5. Usability evaluation: efficiency based on user feedback.

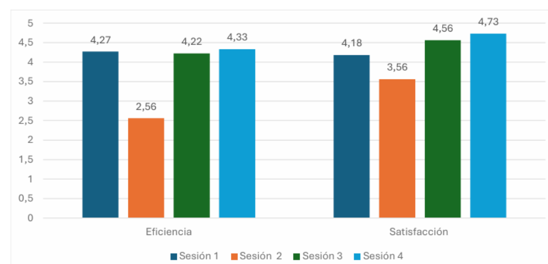

Fig. 6. Usability evaluation: satisfaction and effectiveness based on user feedback.

E. Long-term Session Stability

To verify MusicNet's resilience during extended usage, we conducted multiplayer sessions ranging from 30 to 45 minutes under both stable and degraded network conditions. Throughout these sessions, no application crashes or major synchronization issues were observed.

Players reported that both the audio and visual elements remained synchronized and consistent across peers, demonstrating that the platform can support longer lessons or practice sessions in real classroom scenarios.

F. Improvements Based on User Feedback

The system underwent several iterations based on participant feedback. Key refinements included:

- Redesigning the guide line to be thinner and less intrusive.
- Adding an interactive tutorial to facilitate onboarding.

- Improving noise filtering in the pitch detection pipeline.
- Enhancing the WebRTC reconnection logic to recover dropped sessions faster.

These improvements resulted in higher satisfaction ratings in subsequent test rounds and contributed directly to smoother learning curves among new users.

G. Comparative Performance

Finally, we compared MusicNet to two prominent platforms: **JumpApp** and **Internet MIDI**. Table II summarizes their respective features. While JumpApp introduced gamified pitch recognition, it does not support audio transmission or multiplayer interaction. Internet MIDI allows low-latency symbolic control but lacks expressive audio support and browser integration.

MusicNet surpasses both by delivering a full-featured, browser-native experience that supports real-time audio, visual feedback, and resilient multiplayer connectivity.

TABLE II
COMPARISON WITH RELATED PLATFORMS

Platform	Latency Tolerance	Audio Quality	Browser-native
JumpApp	Medium	Visual only	Yes
Internet MIDI	High	MIDI only	No
MusicNet	High	Full audio	Yes

H. Deployment

All the code, documentation, and datasets used in this work are openly available in the main GitHub repository [22].

VI. CONCLUSION AND FUTURE WORK

This paper presented **MusicNet**, a browser-based multiplayer platform for piano instruction that combines gamified feedback, WebRTC peer-to-peer communication, real-time pitch detection with AubioJS, and neural packet loss recovery (PARCnet). MusicNet addresses key gaps in online music education by enabling expressive audio transmission, low-latency interaction, and robust operation in variable network conditions—all within the browser and without specialized hardware.

Unlike prior systems such as JumpApp or Internet MIDI, MusicNet enhances accessibility while preserving fidelity and responsiveness. Evaluations confirmed strong performance in latency, pitch accuracy, user satisfaction, and resilience, validating its potential as a practical educational tool. Beyond technical contributions, MusicNet promotes motivation, collaboration, and inclusion, aligning with the broader vision of the **Internet of Musical Things (IoMusT)**.

Future work will focus on: (i) extending support to additional instruments and polyphonic input, (ii) developing adaptive learning paths and classroom tools, (iii) conducting longitudinal studies on educational impact, and (iv) exploring IoMusT interoperability with connected instruments.

In conclusion, MusicNet demonstrates how accessible, AI-enhanced, and network-resilient technologies can transform music education into a more collaborative, scalable, and impactful experience.

REFERENCES

[1] M. Olivieri, F. Simeon, L. Comanducci, F. Antonacci, and A. Sarti, "JumpApp: An online didactic game for music training and education," in *Proc. 4th Int. Symp. Internet of Sounds (ISIoS)*, 2023. DOI: 10.1109/IEEECONF59510.2023.10335343.

[2] R. Pike, "Internet MIDI for distance music instruction," *Int. J. Music Technol.*, vol. 18, pp. 45–52, 2020.

[3] A. Rogers, K. Järveläinen, and J. Salminen, "P.I.A.N.O.: Projected Interactive Augmented Notation Object for piano teaching," in *Proc. ACM TEI*, 2014, pp. 211–218.

[4] M. Wright and A. Freed, "Open Sound Control: A protocol for communication among computers, sound synthesizers, and other multimedia devices," in *Proc. Int. Comput. Music Conf. (ICMC)*, 2004.

[5] A. I. Mezza, M. Amerena, A. Bernardini, and A. Sarti, "Hybrid packet loss concealment for real-time networked music applications," *IEEE Open J. Signal Process.*, vol. 5, pp. 266–273, 2024. DOI: 10.1109/OJSP.2023.3343318.

[6] M. A. Soler-Martínez, *Audio generation system for lost packets in real-time distributed music interpretation applications*, B.Sc. Thesis, Universidad Politécnica de Madrid, 2023.

[7] C. Rottondi, C. Chafe, C. Allocchio, and A. Sarti, "An overview on networked music performance technologies," *IEEE Access*, vol. 4, pp. 8823–8843, 2016. DOI: 10.1109/ACCESS.2016.2628440.

[8] M. Sacchetto, A. Servetti, and C. Chafe, "JackTrip-WebRTC: Networked music experiments with PCM stereo audio in a Web browser," GitHub, 2021. [Online]. Available: https://github.com/JackTrip-webrtc/JackTrip-webrtc

[9] J. M. Sánchez-Jara, S. R. Ahijado, and J. C. M. Rubio, "Educational ecosystems for musical practice in the social web environment: A systematic literature review," *Revista de Investigación Educativa*, vol. 40, no. 2, pp. 565–587, 2022.

[10] M. A. Lobo-Rueda, M. C. Paba-Medina, and M. L. Torres-Barreto, "Descriptive analysis of gamified experiences for teaching and learning in higher engineering education," *Revista Espacios*, vol. 41, no. 16, 2020.

[11] J. M. Bournissen, M. C. Tumino, and F. Carrión, "MOOC: Quality assessment and measurement of perceived motivation," *Int. J. Educ. Res. Innov.*, no. 11, pp. 18–32, 2019.

[12] V. A. Nguyen, A. H. Nguyen, and A. W. Khong, "Improving performance of real-time full-band blind packet-loss concealment with predictive network," in *Proc. IEEE ICASSP*, pp. 1–5, Jun. 2023.

[13] J. M. Valin et al., "Real-time packet loss concealment with mixed generative and predictive model," *arXiv preprint*, arXiv:2205.05785, 2022. Available: https://arxiv.org/abs/2205.05785

[14] S. Pascual, J. Serrà, and J. Pons, "Adversarial auto-encoding for packet loss concealment," in *Proc. IEEE WASPAA*, pp. 71–75, Oct. 2021.

[15] P. Verma, A. I. Mezza, C. Chafe, and C. Rottondi, "A deep learning approach for low-latency packet loss concealment of audio signals in networked music performance applications," in *Proc. 27th Conf. Open Innovations Assoc. (FRUCT)*, pp. 268–275, Sep. 2020.

[16] F. G. Barrio and M. G. Barrio, "Tools and resources for music creation and consumption on the web 2.0: Applications and educational potentialities," *Educación XX1*, vol. 17, no. 2, pp. 313–336, 2014.

[17] L. Turchet, C. Fischione, G. Essl, M. Keller, and M. Barthet, "Internet of musical things: Vision and challenges," *IEEE Access*, vol. 6, pp. 61994–62017, 2018.

[18] C. Alexandraki et al., "Collaborative score transformations in online music lessons: The MusiCoLab toolset," in *Proc. 4th Int. Symp. Internet of Sounds (ISIoS)*, pp. 1–10, IEEE, 2023.

[19] M. F. Becerra Aravena, *Wireless system for low-latency audio signal transmission*, Eng. Thesis, University of Chile, 2016.

[20] J. Gao, "Clumsy: Network condition simulator," 2024. [Online]. Available: https://jagt.github.io/clumsy/

[21] Amazon Web Services, Inc., "What is network latency?," 2024. [Online]. Available: https://aws.amazon.com/es/what-is/latency

[22] V. Rozo-Gonzalez and P. Lopez-Cubillos, "MusicNet: Source code and documentation," GitHub repository, 2024. [Online]. Available: https://github.com/rozovalentina/musicnet