OmniClip: A Universal Clipboard for All Platforms

Dr. Jeevitha B K[1], Adithya D P[2], Anish A[3], Karthik K[4], Karthika Krishna K[5]
[1]*Associate Professor, Department of Computer Science and Engineering, VCET, Puttur, Karnataka*
[2,3,4,5]*7th Semester, Dept. of Computer Science and Engineering,*
Vivekananda College of Engineering and Technology, Puttur, Karnataka, India

***Abstract*—In modern computing, users frequently operate across multiple systems such as Windows, macOS, and Linux. However, the clipboard a fundamental productivity tool remains confined to a single device. OmniClip provides a cross-platform solution that enables real-time clipboard and file synchronization over a local network without internet or cloud dependency. Built using Flutter and Dart, it employs UDP for device discovery and TCP for reliable data transfer, ensuring privacy and low latency. The modular design supports text and file sharing across all major desktop platforms, improving workflow continuity and productivity.**

***Index Terms*—Clipboard synchronization, cross-platform application, UDP, TCP, Flutter, Dart, Riverpod**

I. INTRODUCTION

Users often work across multiple devices and operating systems, yet clipboard data remains isolated per system. Existing tools rely on third-party cloud services, introducing latency and privacy concerns. OmniClip addresses this limitation by enabling clipboard and file synchronization through a local network using lightweight communication protocols.
The system automatically discovers nearby devices, allowing users to copy data on one device and paste it on another seamlessly. Implemented in Flutter for uniform cross-platform performance, OmniClip ensures responsive interaction and reliable synchronization using Dart's asynchronous architecture.

II. LITERATURE SURVEY

Previous research informed the networking and design principles of OmniClip. Nanavati et al. [1] optimized Flutter performance through widget-level improvements, validating its use for high-efficiency applications. Veluswami et al. [2] improved TCP reliability for high-bandwidth systems, influencing OmniClip's hybrid TCP model. NewTek Inc. [3] introduced Reliable UDP (RUDP), combining UDP speed with delivery assurance, inspiring OmniClip's discovery mechanism.
Lovrić [4] demonstrated Flutter's near-native performance across platforms, while Anderson and Roberts [5] highlighted Riverpod's scalability for real-time state management. Abiona and Oladapo [6] confirmed the efficiency of Wi-Fi Direct peer discovery, supporting OmniClip's auto-detection via UDP.
These studies collectively reinforce OmniClip's design combining cross-platform performance, network reliability, and reactive state management for efficient synchronization.

III. METHODOLOGY

The development of OmniClip: A Universal Clipboard for All Platforms was carried out using a systematic and modular approach that emphasizes cross-platform compatibility, reliability, and real-time performance. The system was implemented using Flutter and Dart, integrating both UDP and TCP communication models to achieve fast device discovery and dependable data transfer. The implementation phase transformed the proposed design into a working prototype through several functional modules, each responsible for a distinct operation in the synchronization process.

A. System Overview

OmniClip enables users to share clipboard content across multiple desktop platforms through a local LAN or Wi-Fi network. Each instance of OmniClip discovers peers using UDP broadcasts and exchanges clipboard data via TCP sockets. The system operates entirely offline, ensuring both speed and privacy.

B. Functional Architecture

The architecture comprises interconnected modules that manage device discovery, clipboard monitoring, and data transfer

- Device Discovery Module: Broadcasts and listens for UDP packets to identify peers within the same network.
- Clipboard Monitoring Module: Detects clipboard updates and triggers synchronization requests.
- Data Transfer Module: Uses TCP sockets for reliable, ordered data transmission.
- State Management Module: Tracks connected devices, transfer progress, and synchronization completion.
- User Interface Module: Built with Flutter, provides a simple interface for managing connections and transfers.
- Deployment Module: Packages the system for Windows, macOS, and Linux to ensure smooth execution.

Together, these modules enable efficient and realtime clipboard sharing across multiple platforms.

C. Software Stack

Component	Technology Used
Frontend	Flutter
Programming Language	Dart
Networking	UDP and TCP sockets
Architecture Pattern	Modular event- driven architecture
Development IDE	Visual Studio Code
Operating Systems Supported	Windows, macOS, Linux

Flutter and Dart enable native performance using a single codebase. Dart's asynchronous features ensure non-blocking network communication for a smooth, responsive interface.

D. Data Flow and Processing Steps

The data flow begins when a user copies data on one device and proceeds through the following stages:

- Device Discovery: Devices on the same network broadcast their presence via UDP and maintain a list of active peers.
- Clipboard Detection: The system monitors clipboard events and detects new copy actions.
- Data Serialization: The copied content is formatted for network transfer.
- Connection and Transfer: The sender establishes a TCP connection with the selected peer and transmits the data reliably.
- Clipboard Update: The receiver deserializes the content and updates its local clipboard.

This process ensures real-time, reliable synchronization and data integrity across connected devices.

E. Deployment Strategy

OmniClip runs natively on Windows (.exe), macOS (.app), and Linux (.snap) platforms, and is also compatible with virtual machine environments. Its lightweight design ensures quick startup, minimal resource usage, and smooth performance across all supported systems.

IV. SYSTEM DESIGN

The system design defines the structure, communication flow, and components required to achieve real-time clipboard synchronization across multiple platforms. OmniClip ensures interoperability and reliability using UDP and TCP protocols, operating entirely offline without external servers.

A. Overview

OmniClip enables cross-platform clipboard synchronization within a local network. When a user copies text or files, the system detects the change, serializes the data, and transfers it through a TCP connection. UDP is used for device discovery, while TCP ensures ordered and reliable data delivery. The modular architecture allows easy maintenance, scalability, and efficient performance across platforms.

B. System Architecture

The architecture follows a layered and modular design consisting of:

- User Interface Layer: Built using Flutter for user interaction and device visualization.
- Application Layer: Coordinates system logic, device discovery, and synchronization actions.
- Network Layer: Uses UDP for discovery and TCP for data transfer.
- Clipboard Layer: Detects and captures new clipboard content.

- Synchronization Layer: Manages transfers, prevents duplication, and ensures reliability.

Each layer communicates through well-defined interfaces to maintain modularity and scalability. Figure 1 illustrates the layered architecture of OmniClip and its component interactions.

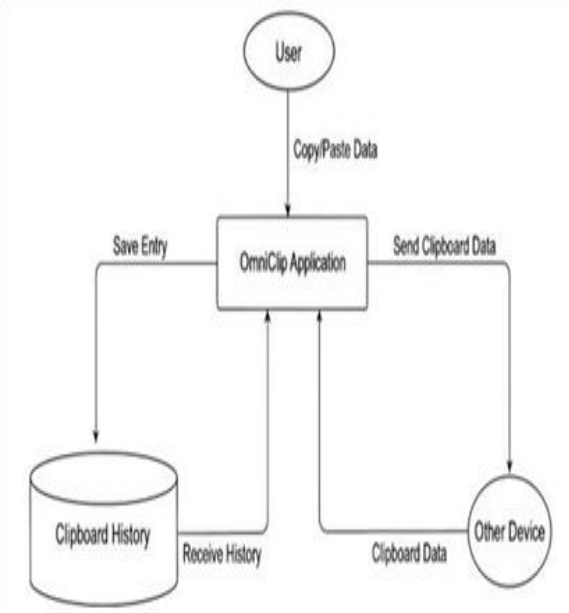

Fig. 1. System Architecture of OmniClip

C. Process Flow

The OmniClip process flow begins with device discovery, clipboard monitoring, data transfer, and synchronization confirmation. Each instance starts a UDP listener on a fixed port, periodically broadcasting its device name and IP. Other instances receive these packets and update their peer list automatically.

- The application starts and activates a UDP listener on a specific port.
- Each instance broadcasts its device name and IP periodically for peer discovery.
- Other instances receive these packets and update the device list.
- When data is copied, the clipboard listener detects and serializes it.
- A TCP connection is established with the selected peer for transfer.
- The receiver reconstructs the data, updates its clipboard, and confirms synchronization.

This event-driven process enables real-time and reliable communication between devices on the same network.

D. Use Case Model

The use case model describes the interaction between the user and the OmniClip system. The main actor in the system is the user, who performs synchronization-related actions.

Primary Use Cases:

- Copying data to the clipboard.
- Detecting connected devices on the same network.
- Selecting a target device for synchronization.
- Sending clipboard data to the target device.
- Receiving clipboard data from another device.

These interactions enable smooth communication between the user and system through the application interface.

V. RESULTS AND DISCUSSION

The OmniClip: A Universal Clipboard for All Platforms system was developed, implemented, and tested across Windows, macOS, and Linux. Results show that clipboard and file synchronization can be efficiently achieved in real time over a local network using UDP and TCP communication protocols. The system's performance was evaluated for responsiveness, accuracy, reliability, and platform compatibility cross-platform compatibility.

A. Functional Validation

All modules were tested individually and as an integrated system. The application successfully detected peers, monitored clipboard changes, and synchronized data between devices.

- Device Discovery: The UDP-based discovery module efficiently identified all active OmniClip instances within the same LAN, automatically listing available peers without manual setup.
- Clipboard Synchronization: The system detected new copy events and transferred text or file data accurately and without delay.
- Cross-Platform Execution: The same Flutter build operated smoothly across Windows, macOS, and Linux.
- User Interface: The interface displayed connected devices, transfer progress, and synchronization completion clearly and responsively.

Figure 2 shows the OmniClip home page, where the user can initiate device discovery and view available peers within the same local network

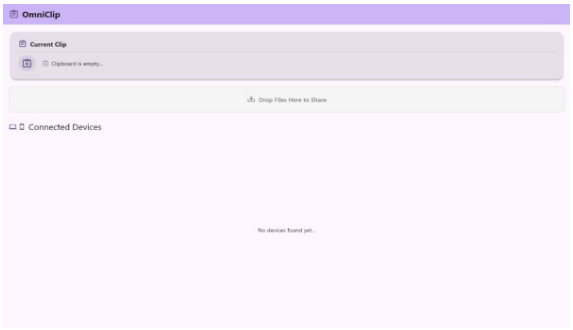

Fig. 2. Home Page Interface of OmniClip

Figure 3 illustrates the device discovery window displaying all active OmniClip instances detected on the local network.

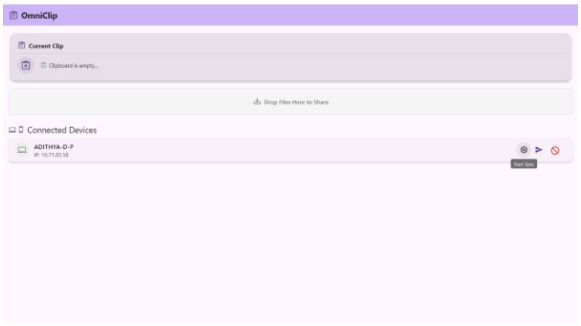

Fig. 3. Device Discovery Window of OmniClip

Figure 4 presents the clipboard synchronization screen, confirming successful data transfer and realtime clipboard update between devices.

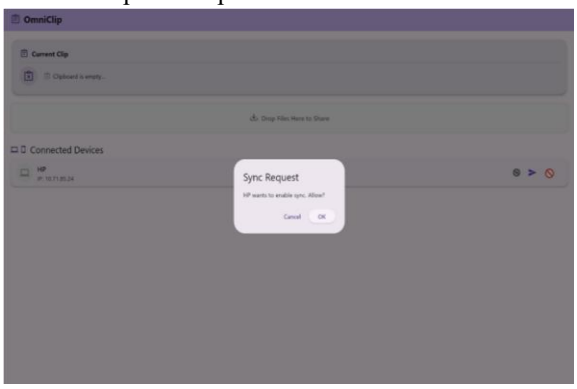

Fig. 4. Clipboard Synchronization Window

B. Performance Analysis

Performance testing confirmed that the system is fast, reliable, and lightweight.

- Discovery Speed: Average peer detection time was under 2 seconds.
- Transfer Latency: Text synchronization averaged 1–3 seconds and 3–6 seconds for small files, depending on network conditions.
- Accuracy: No data corruption or packet loss was observed during testing.
- Resource Usage: The application maintained stable CPU and memory usage, even during concurrent transfers, confirming that the implementation is lightweight and efficient.

C. Discussion

The system effectively enables seamless clipboard and file sharing across platforms using a hybrid UDP–TCP model. UDP ensures instant peer discovery, while TCP provides secure and complete data delivery. Built with Flutter and Dart, OmniClip maintains consistent performance and appearance across devices. Its event-driven design ensures asynchronous, lag-free synchronization. The only limitations are its reliance on local network connectivity and the absence of encryption, which can be addressed in future work. Overall, OmniClip demonstrates efficient and private real-time synchronization suitable for multi-device workflows.

VI. CONCLUSION

OmniClip successfully enables cross-platform clipboard sharing without cloud dependency. By combining UDP and TCP communication within a local network, the system ensures fast, reliable, and private synchronization. Developed in Flutter and Dart, it delivers a simple, responsive interface for efficient real-time data sharing across devices, improving convenience and productivity in multiplatform environments.

VII. FUTURE WORK

The current version of OmniClip efficiently supports clipboard and file synchronization across desktop platforms through local network communication. Future enhancements can focus on extending the system's functionality and improving data security. Planned improvements include implementing end-toend encryption for secure data transmission, enabling synchronization over the internet, and integrating the system with mobile platforms such as Android and iOS. Additionally, features like clipboard

history management, multi-device group sharing, and intelligent content recognition can be incorporated to enhance usability. These advancements will make OmniClip a more comprehensive, secure, and versatile solution for real-time data sharing across all user environments.

REFERENCES

[1] A. Nanavati et al., "Critical Review and Fine-Tuning Performance of Flutter Applications," International Journal of Mobile Computing and Application Development, vol. 12, no. 3, pp. 45–52, 2024.

[2] S. J. R. S. Veluswami et al., "Improvement of Transmission Control Protocol for High Bandwidth Applications," International Journal of Advanced Networking and Applications, vol. 13, no. 5, pp. 489–495, 2021.

[3] NewTek Inc., "Reliable UDP (RUDP) Protocols in Broadcasting," Technical White Paper, USA, 2020.

[4] A. Lovrić, "Evaluation of the Cross-Platform Framework Flutter Using Real-World Applications," Journal of Software Engineering and Applications, vol. 16, no. 4, pp. 215–225, 2023.

[5] J. Anderson and D. Roberts, "State Management Patterns in Distributed Flutter Applications Using Riverpod Architecture," Software Practice and Experience, vol. 53, no. 1, pp. 12–20, 2023.

[6] O. Abiona and T. Oladapo, "Wi-Fi Direct Peer Discovery and Connection Reliability in Real-Time Systems," IEEE Internet of Things Journal, vol. 9, no. 7, pp. 12583–12591, 2022.