FINAL PROJECT SUBMISSION (GROUP -05)

DONE BY :- NEHA SAHA (2251181) PRIYANSHU MONDAL (2251222)
ANKIT OJHA(2251220) ARKASNATO BHATTACHARYYA(2251064)

TOPIC :
UDP Payment Simulator Using JavaFX

1. Introduction

In the contemporary digital economy, instantaneous and secure peer-to-peer (P2P) payment solutions are fundamental. Traditional blockchain networks, while highly secure, often suffer from scalability and latency issues. This makes them less optimal for microtransactions or high-frequency transactions.

To address these limitations, off-chain solutions such as the **Bitcoin Lightning Network** have emerged, enabling fast, scalable, and cost-effective micropayments by shifting transaction processing off the main blockchain.

This project, **UDP Payment Simulator**, replicates the behaviour of such off-chain payment networks by:

- Using **UDP** for low-latency communication.
- Implementing a **JavaFX-based GUI** for user-friendly interaction.
- Simulating crucial aspects like **payment channel management**, **transaction acknowledgment**, **packet loss**, and **latency measurement**.

The goal is to model real-world decentralized payment behaviours in a controlled simulation environment.

---

2. Objective

The main objectives of this project are:

- **Simulation**: Model decentralized peer-to-peer microtransactions over an unreliable network (UDP).
- **GUI Development**: Create a professional and intuitive graphical interface using JavaFX.
- **Acknowledgment Handling**: Implement mechanisms for payment confirmation and timeout detection.
- **Packet Loss Simulation**: Emulate real-world network unreliability by introducing artificial packet drops.
- **Latency Tracking**: Measure and log the round-trip time of payments for analysis.

DONE BY :- NEHA SAHA (2251181) PRIYANSHU MONDAL (2251222)
ANKIT OJHA(2251220) ARKASNATO BHATTACHARYYA(2251064)

3. Problem Statement

Modern blockchain-based payment systems, though secure, are inherently slow due to the consensus mechanism. For micropayments, these delays and transaction fees are prohibitive.

The project addresses the problem by:

- Simulating an off-chain fast transaction environment.
- Analyzing how unreliable networks (like UDP) affect transaction reliability.
- Testing strategies to mitigate transaction failures (acknowledgment, timeout handling).

This simulation helps understand how decentralized payment systems ensure both **speed** and **reliability** despite network challenges.

---

4. Technologies Used

Technology	Purpose
Java 11+	Core programming language for backend and logic.
JavaFX	Graphical User Interface (GUI) framework.
UDP Sockets (java.net)	Peer-to-peer communication channel.
CSV Files	Persistent logging of transaction data.
Traffic Simulation	Packet loss and delay simulation within UDP communication.

---

5. Project Features

- **Open/Close Payment Channels**: Simulation of enabling and disabling the ability to transact.
- **Send Payment Requests**: Send payments via UDP packets formatted as lightweight JSON objects.
- **Receive Acknowledgments**: Confirm transaction delivery with acknowledgment packets.

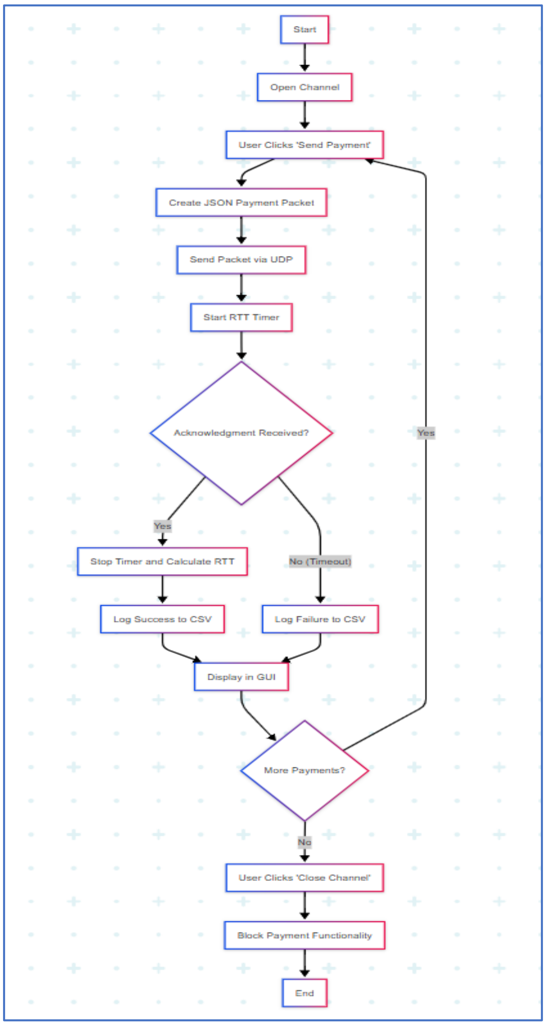
DONE BY :- NEHA SAHA (2251181)   PRIYANSHU MONDAL (2251222)
ANKIT OJHA(2251220)   ARKASNATO BHATTACHARYYA(2251064)

- **Packet Loss Simulation**: Randomly discard 20% of outgoing packets to emulate real-world network instability.
- **Latency Measurement**: Record time between sending a request and receiving an acknowledgment.
- **Transaction Logging**: Save all transaction attempts (successes and failures) into a structured CSV file.
- **GUI with Status Feedback**: Provide real-time feedback to users via an auto-updating, scrollable log display.

---

6. Working Principle

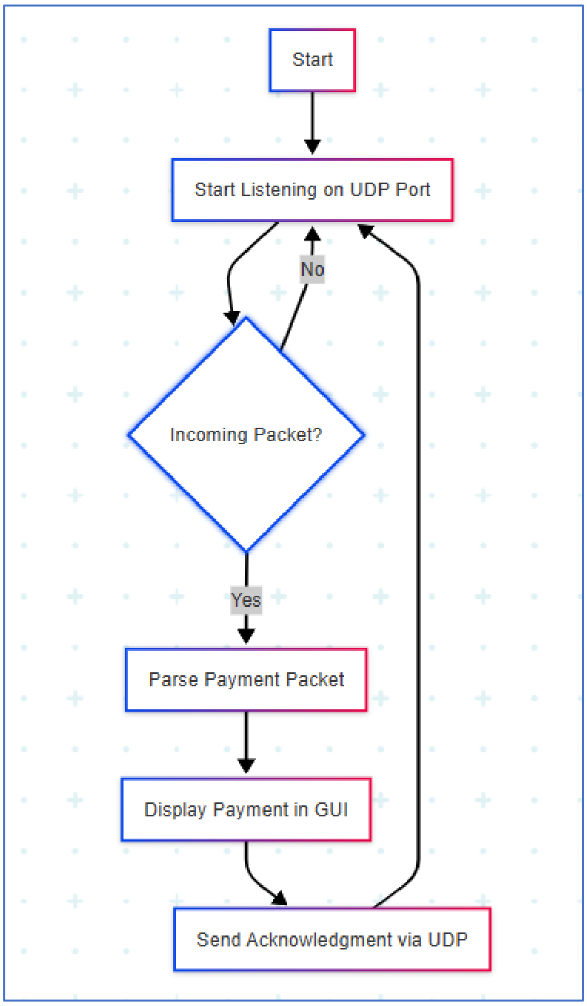
6.1 Sender App Workflow



DONE BY :- NEHA SAHA (2251181) PRIYANSHU MONDAL (2251222)
ANKIT OJHA(2251220) ARKASNATO BHATTACHARYYA(2251064)

1. **Channel Initialization**:
   Upon "Open Channel" command, sender can start sending payments.
2. **Payment Request Sending**:
   A JSON packet containing payment details is created and sent via UDP.
3. **Waiting for Acknowledgment**:
   A timer starts to measure RTT. If acknowledgment is received within timeout, success is logged. Otherwise, transaction failure is logged.
4. **Channel Closure**:
   After payments are done, the "Close Channel" command halts further transmission.

6.2 Receiver App Workflow

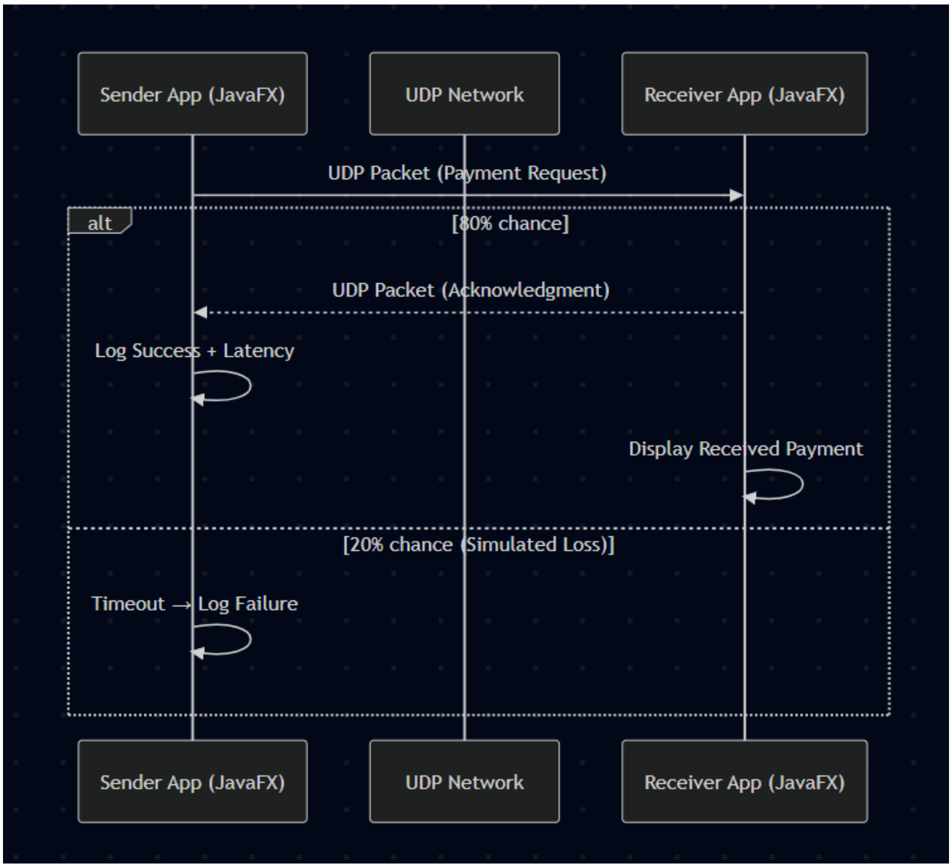


DONE BY :- NEHA SAHA (2251181)   PRIYANSHU MONDAL (2251222)
ANKIT OJHA(2251220)   ARKASNATO BHATTACHARYYA(2251064)

1. **Listening for Packets**:
   The receiver stays in a loop, continuously monitoring the UDP port for incoming payment packets.
2. **Acknowledgment Handling**:
   Upon receiving a payment packet, the receiver sends back a simple acknowledgment packet immediately.
3. **User Interface Updates**:
   Received payments and sent acknowledgments are logged and displayed on the receiver's GUI in real-time.

---

7. System Architecture Diagram



---

8. Outputs shown in a separate file attached after.

FINAL PROJECT SUBMISSION (GROUP -05)

DONE BY :- NEHA SAHA (2251181) PRIYANSHU MONDAL (2251222)
ANKIT OJHA(2251220) ARKASNATO BHATTACHARYYA(2251064)

9. Why UDP is Used

UDP is selected as the communication protocol because:

- **No Handshakes**:
  Minimal connection overhead ensures fast communication.
- **Low Latency**:
  UDP allows quicker message delivery without waiting for network setup.
- **Lightweight Headers**:
  Less data overhead compared to TCP improves transmission speed.
- **Resembling Lightning Network Characteristics**:
  Similar to Bitcoin's Lightning Network philosophy where speed is prioritized over reliability (handled cryptographically, not by the transport layer).

Real-World Comparison:

- **Lightning Network**: Guarantees reliability through cryptographic contracts.
- **UDP Payment Simulator**: Simulates fast, possibly unreliable communication, where responsibility lies at application logic.

---

10. Risk of Money Loss Discussion

In this simulation:

- Packet loss results in missing transactions without real-world financial loss.
- CSV log records failed attempts for analysis.

In actual Lightning Network systems:

- Loss of acknowledgment does not necessarily mean monetary loss because of **Hashed Timelock Contracts (HTLCs)**.
- If a payment is not confirmed within a timelock, it is **automatically refunded** to the sender.
- Cryptographic rules ensure **either payment completion or no fund movement**.

Thus, real systems are significantly more resilient than the simple UDP simulator.

DONE BY :- NEHA SAHA (2251181)   PRIYANSHU MONDAL (2251222)
ANKIT OJHA(2251220)   ARKASNATO BHATTACHARYYA(2251064)

11. How This Project Relates to Real Bitcoin Lightning Network

Simulator Feature	Lightning Network Concept
Channel Management	Channel opening/closing
Payment Sending	Off-chain transaction
Acknowledgment Processing	Cryptographic receipt
Packet Loss Simulation	Node failure or unstable routing paths
Latency Measurement	Tracking transaction finalization time
Transaction Logging	Maintaining payment records (commitment states)

---

12. Future Improvements

- **Add Encryption**:
  Introduce DTLS (Datagram TLS) to encrypt UDP traffic.
- **Retry Mechanism**:
  Implement retransmissions for unacknowledged packets based on timeout.
- **Multi-client System**:
  Extend to support concurrent multi-peer payment sessions.
- **Data Visualization**:
  Integrate charting libraries to plot transaction statistics (e.g., success/failure rate, latency distribution).
- **Blockchain Integration**:
  Connect with a Bitcoin testnet to simulate on-chain channel settlements.
- **Mobile Compatibility**:
  Extend GUI to Android or lightweight platforms using JavaFX Mobile.

DONE BY :- NEHA SAHA (2251181) PRIYANSHU MONDAL (2251222)
ANKIT OJHA(2251220) ARKASNATO BHATTACHARYYA(2251064)

13. Conclusion

The **UDP Payment Simulator** successfully models the key principles of decentralized, real-time payment systems, focusing on speed, packet loss handling, and acknowledgment verification.
It serves as an educational tool to understand network behaviour in unreliable environments and offers a foundational prototype that can be enhanced into a real-world payment layer system with further cryptographic and blockchain integrations.

By providing insights into transaction behaviour over unreliable mediums, this project bridges academic learning with modern decentralized finance concepts.
