B.Tech. Project

on

Digital Payment Apps

End-Term Presentation
May 2025

Supervised by:
Prof. Tarun Mangla

Submitted By :
Tushman Khalse 2019CS10411
Shreejeet Golhait 2019CS10351

Introduction

- **UPI apps are the backbone of India's digital economy**, enabling millions of secure, instant transactions daily.
- **The rapid adoption of UPI emphasizes the need to optimize transaction performance**, especially under varying network conditions.
- **We aim to analyze network traffic and interactions during UPI transactions**, identifying which steps take the most time.
- **By evaluating UPI apps under different conditions**, we establish benchmarks to improve transaction efficiency.
- **Our goal is to recommend measures to optimize transaction times**, ensuring UPI apps perform efficiently and minimize delays.

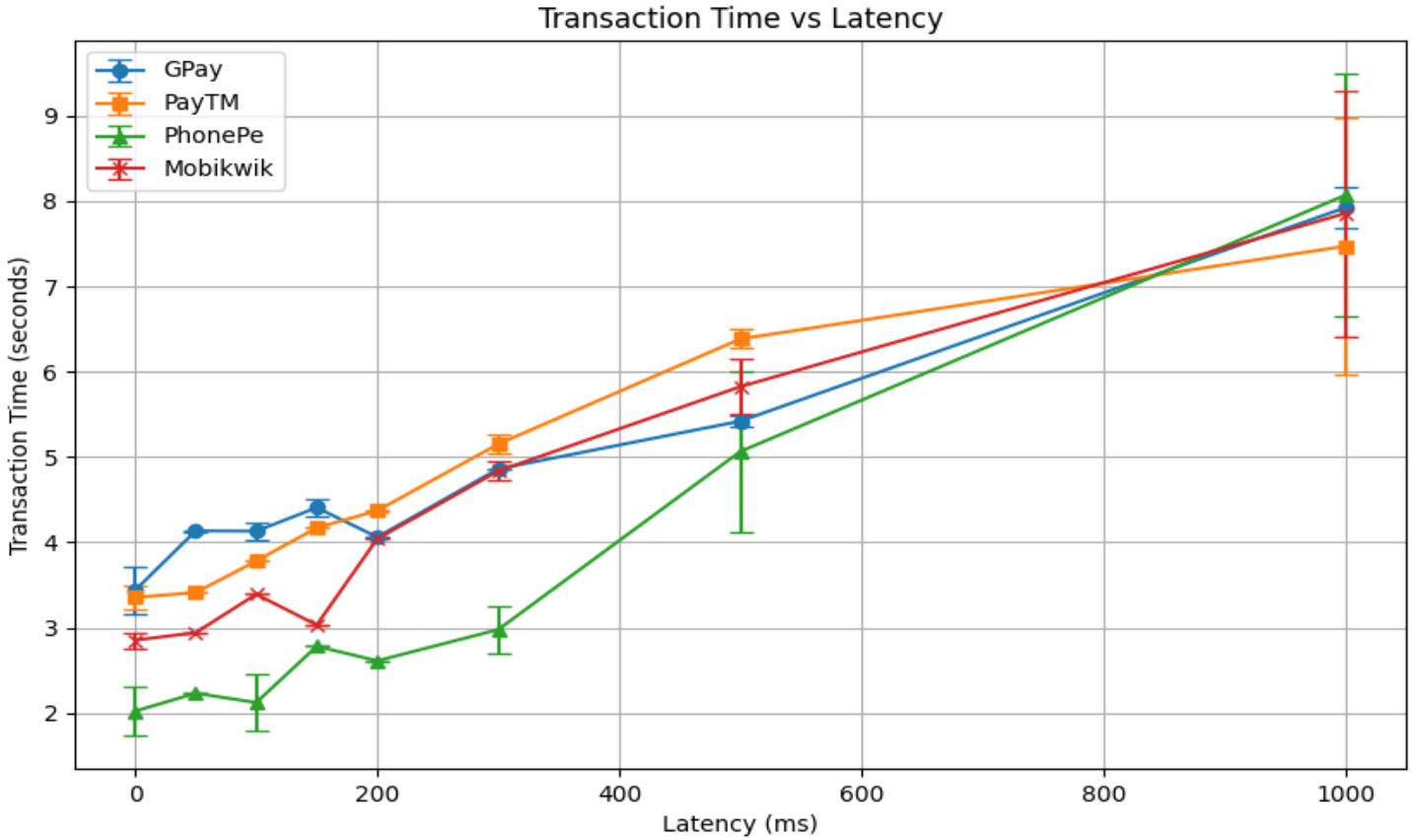
Project Statement

- Building upon prior work presented challenges in defining a clear path forward.
- Initially explored both performance and privacy aspects of digital payment apps, with preliminary work in both areas.
- **Mid-Term Problem Statement**: Automation of digital payments across major UPI apps, followed by performance analysis under varying network conditions.
- **Post Mid-Term Goal**: Analyze encrypted transaction data from network captures (pcap files).
- **Final Objective**: Develop a framework to determine UPI transaction times and delays solely from pcap file analysis.
- **Broader Vision**: Enable large-scale analysis of campus network data to extract UPI transaction performance metrics.

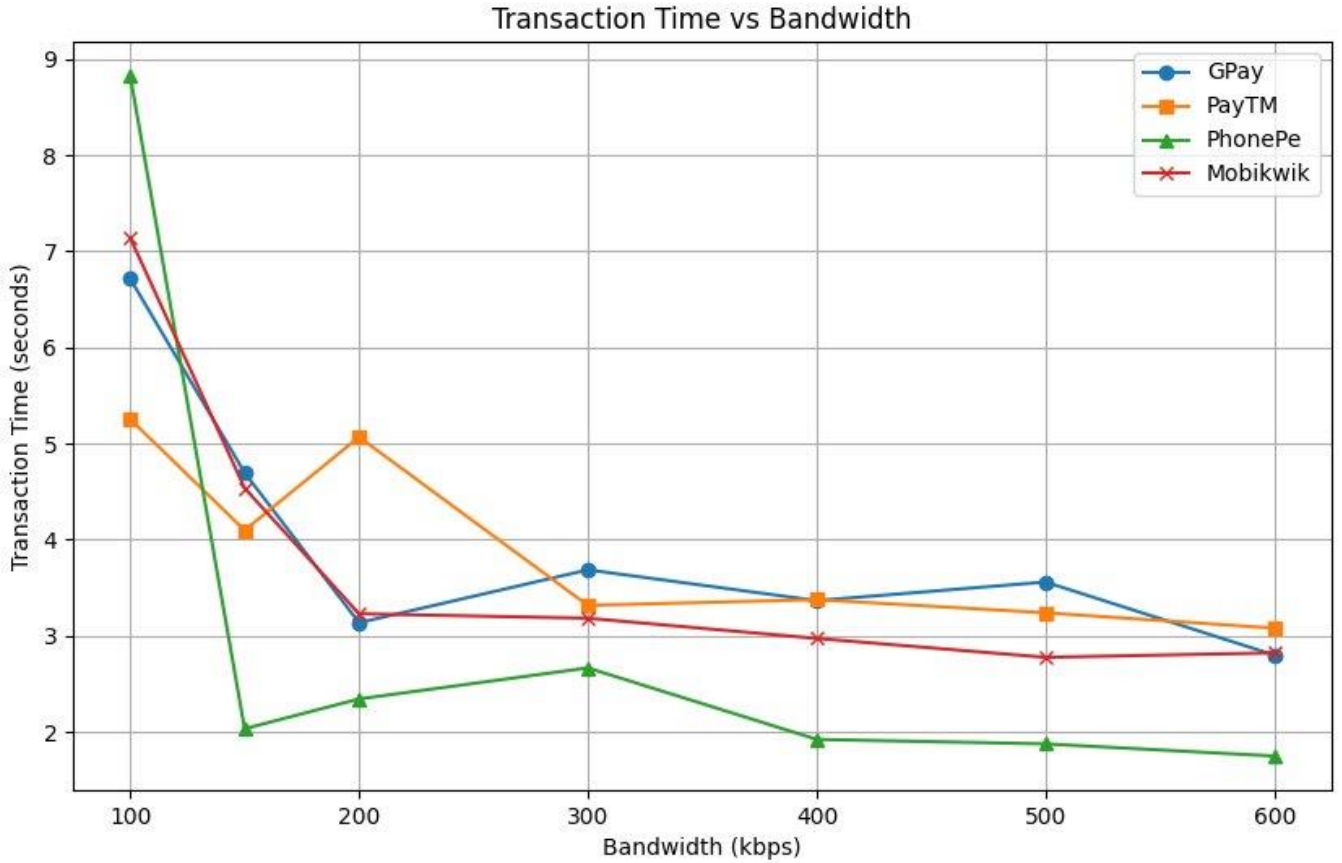
Work done before Mid-Term

- Collected **decrypted** transaction data for two UPI Apps
- Manipulating **Network Conditions** and Measuring Their Impact
- Collected encrypted network data corresponding to different network scenarios.
- Plotted graphs comparing transaction times with **error bars**
- Tried to understand performance under critical **latency**, **bandwidth** and **packet loss** conditions.

Work done before Mid-Term



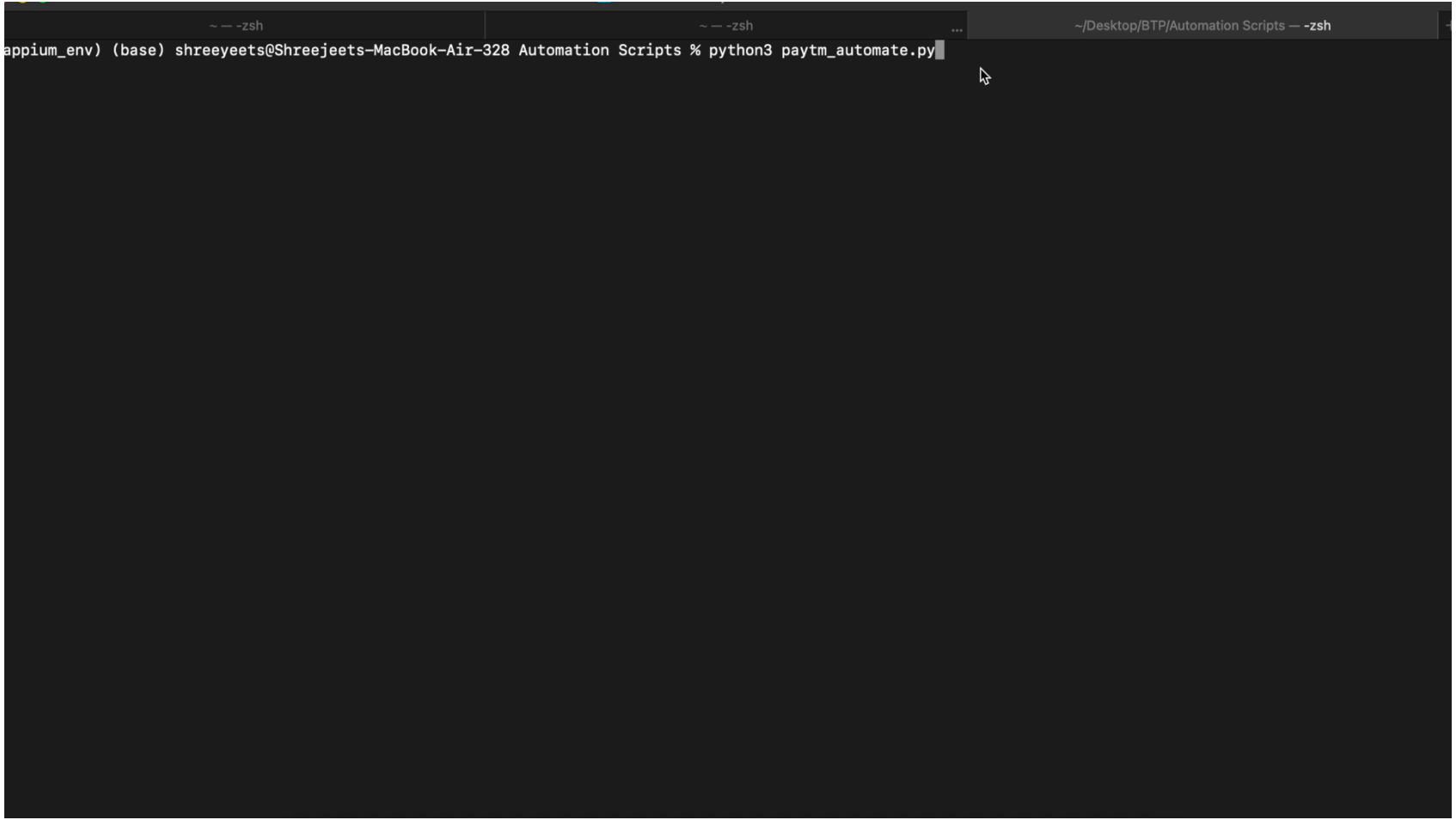
Work done before Mid-Term



Methodology

1. **Implemented Automation of UPI Apps:**

- **Environment Setup** : Used an Appium server and scrcpy to establish connection.
- **App Launch**: Used Appium capabilities to initialize and launch the app on the device.
- **Transaction Automation**: Perform UPI payment steps, including selecting contacts and entering details, through automated scripts (analysing elements with Appium Inspector).
- **Performance Tracking**: Measure transaction times by recording start and end timestamps for each payment.


~ — -zsh
~ — -zsh
~/Desktop/BTP/Automation Scripts — -zsh
appium_env) (base) shreeyeets@Shreejeets-MacBook-Air-328 Automation Scripts % python3 paytm_automate.py

Methodology

2. **Analysis of encrypted packet captures:**

- **SNI Identification:** Developed a Python script to extract all Server Name Indication (SNI) values from TLS handshakes within captured network traffic.

- **Traffic Correlation (Manual):** Manually identified uplink and downlink traffic bursts within pcap files and correlated them with specific SNIs, hypothesizing these corresponded to UPI transactions.

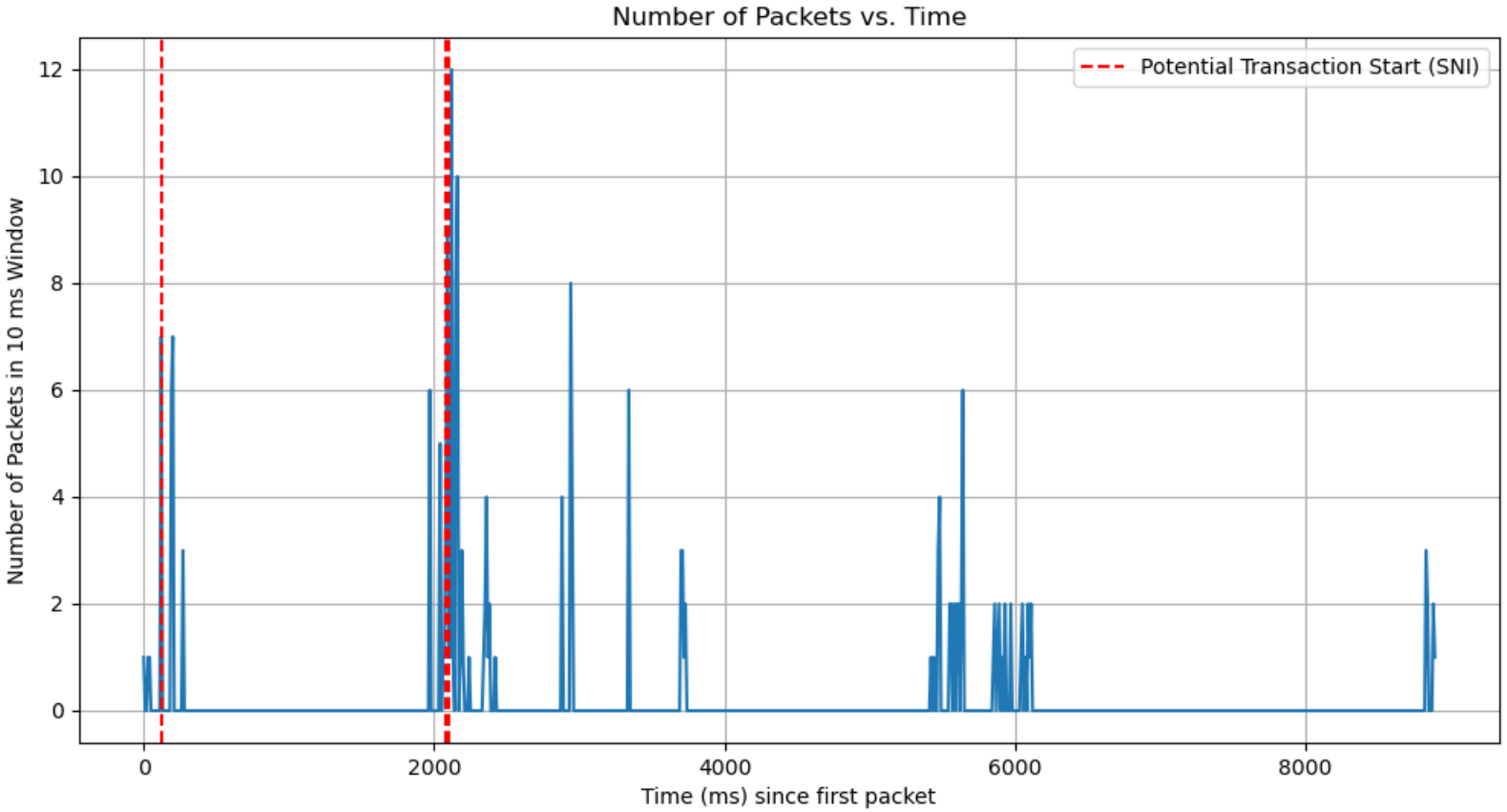
- **Verification with Existing Data:**
  - Analyzed previously captured pcap files (from network quality analysis).
  - Cross-referenced identified SNIs with transaction timestamps recorded during the automation phase.

- **Final Validation (Decrypted Data):** Compared decrypted network traffic for specific UPI applications to definitively identify SNIs associated with transaction-related calls

Methodology

2. **SNI's Obtained Corresponding to transactions:**

GPay * **india-paisa-pa.googleapis.com** at start of transaction * **paymentsincentives-pa.googleapis.com** at end of transaction	**PayTM** * **upi.paytm.com** at start of transaction * digitalapiproxy.paytm.com at end of transaction
PhonePe * **apicp2.phonepe.com** (only one SNI at start and end) - two calls on entering upi pin page - one call at the start of transaction - one call 1s after end of transaction	**Mobikwik** * **appapi.mobikwik.com** - only one sni at end of transaction - same sni on entering upi pin page - long packets at start of transaction detected but no specific SNI


Number of Packets vs. Time
Potential Transaction Start (SNI)
Number of Packets in 10 ms Window
Time (ms) since first packet
0
2
4
6
8
10
12
0
2000
4000
6000
8000

Methodology

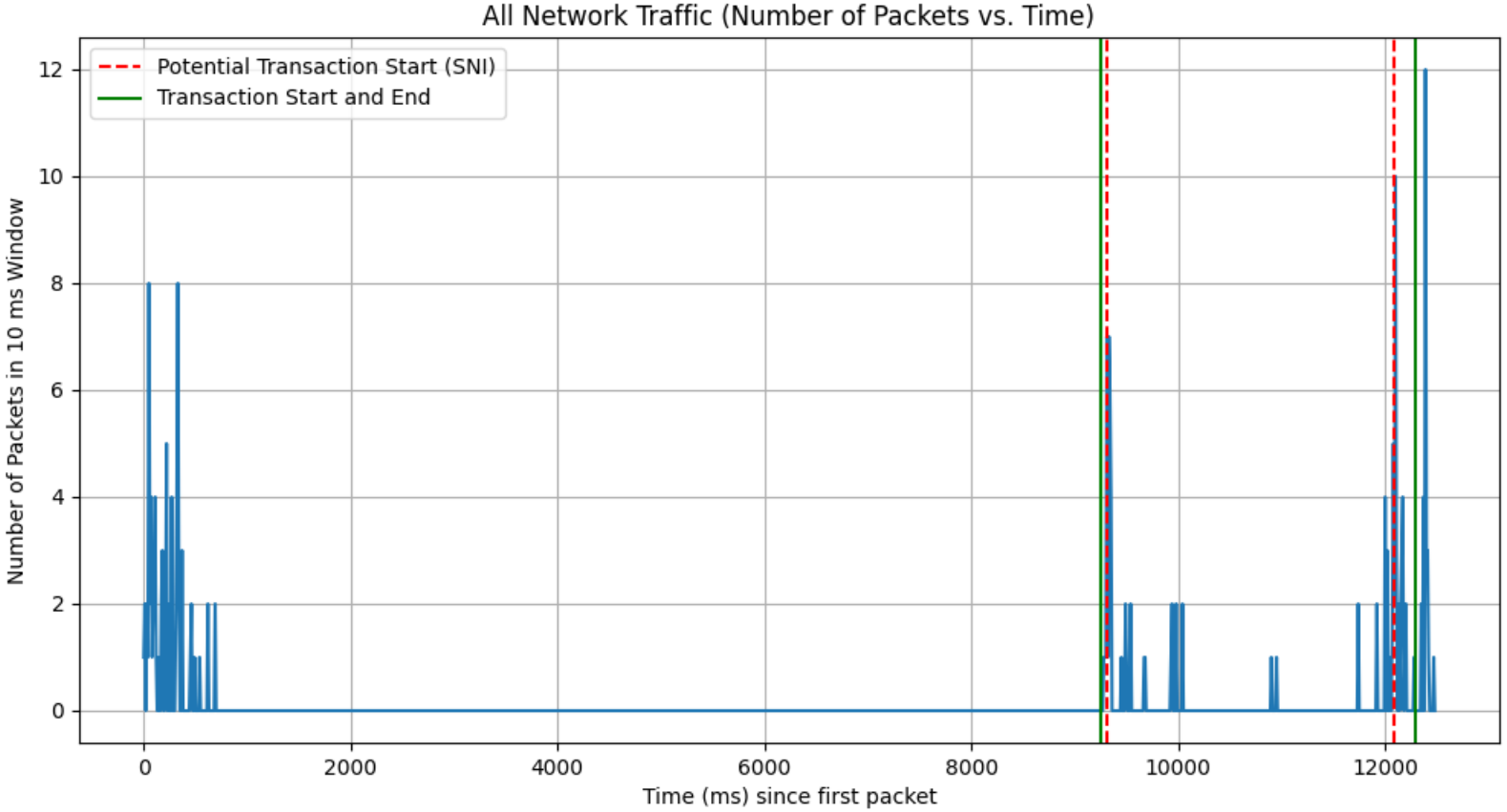
3. **Combination and Automation:**

➢ **Concurrent Capture & Measurement:** Automated UPI payments while simultaneously capturing network traffic (pcap) and recording **observed** transaction times.
Initial Packet Overview: Generated packet count vs. time graphs for each transaction.

➢ **Computational Time Extraction:** Used observed time and pcap files as input for a script that:
- Identifies SNIs around the transaction time.
- Matches them against verified UPI SNIs.
- Extracts relevant transaction packets.
- Calculates the **computational** transaction time.

Network Delay Calculation: Determined network delay as the difference between computational time and observed transaction time.

➢ **Result:** Obtained specific transaction packets for network analysis and quantified network delay.


All Network Traffic (Number of Packets vs. Time)
Potential Transaction Start (SNI)
Transaction Start and End
Number of Packets in 10 ms Window
Time (ms) since first packet
0
2
4
6
8
10
12
0
2000
4000
6000
8000
10000
12000

Methodology

4. Campus Network PCAP Filtering by SNI:

- Used a script that automates the process of sifting through network traffic captures to isolate transaction data.
- Identifies and extracts network traffic (full flows) where the **TLS** Server Name Indication (SNI) matches a predefined list of allowed domains (e.g., **paytm.com, phonepe.com**).
- Saves the filtered traffic into new .pcap files in a dedicated output directory (./filtered_pcaps/), limiting the size of each output file.
- Originally intended for deployment on the campus network to facilitate targeted traffic analysis; however, this phase was not realized within the project's timeframe.

Future Scope

- **Infrastructure Upgrade Justification:** Use data to support necessary network improvements in specific locations.
- **UPI Adoption Patterns on Campus:** Understand frequency and timing of digital payment usage.
- **Predicting Transaction Success:** Build models to forecast successful payments based on network conditions.
- **Network Bottleneck Identification:** Pinpoint congested areas or poor performance affecting UPI using network delay data.
- **UPI App Performance Comparison:** Analyze how different apps perform under varying network conditions on campus.