Case Study: Resolving Transaction Delays at Upstox (Apr–May 2025)

1. Executive Summary

Between April and May 2025, Upstox faced a sharp spike in fund transfer delays across NEFT, IMPS, and UPI channels. Transactions that usually settled in minutes were now taking hours — in some NEFT cases, up to 24 hours. This led to:

- A rise in support tickets and user complaints
- Negative sentiment across Reddit and X (formerly Twitter)
- A noticeable drop in trading volume on both Pro and Investor platforms

This case study identifies root causes, presents a SQL-powered investigation, proposes data-backed solutions, and outlines a real-time dashboard for continuous monitoring.

2. Problem Overview

Metric	Normal (Expected)	During Apr–May 2025	Δ Change
Avg.. NEFT Delay	30 mins	3.2 hrs (194 mins)	+540%
Support Tickets	Baseline	↑ 3.4x	+240%
Trading Volume	100%	88% (drop)	-12%
Retry Attempts	0	↑ up to 3	✓

3. Root Cause Analysis (RCA)

1. NEFT Delay Spike

- **Avg. delay**: 194 mins (3.2 hours), max up to 24 hrs
- **Expected delay**: 30 mins or less
- **Distribution**: 75% of NEFT transactions exceeded 1 hour
- **Root Cause**:
    - Banking partners (especially **HDFC**, **Axis**, and **IDFC**) exhibited slower batch processing during peak hours.
    - Manual queueing of NEFT transfers during weekends and holidays is not handled optimally.
    - Server latency and a missing retry timeout policy caused multiple retries for NEFT transactions.

2. Bank-Specific Failures

- Top offenders: HDFC (83.1 mins avg), Axis (78.7), IDFC (72.8)
- Root Cause:
    - Some partner banks faced internal outages or server latency issues.
    - Lack of SLAs or automated health checks from Upstox's side.

3. Retry Policy Issues

- Observation: Higher retry attempts were associated with increased delays (up to 3 attempts)
- Root Cause:
    - The retry logic was not intelligent — it retried too soon or too often without assessing the root cause.
    - No backoff mechanism or delay-time penalty calculation

4. Transaction Method Health

- UPI & IMPS: Generally stable (7.2 mins & 18.8 mins average), but a spike in failures on Apr 28–30
- Root Cause:
    - Third-party PSP downtime was not accounted for in the routing logic.
    - UPI/IMPS routing was non-adaptive and didn't switch to healthier paths dynamically.

5. Platform-Driven Delays

- **Findings**:
    - Most delays occurred during **peak hours (9 AM – 11:30 AM)** and **end-of-day settlements.**

- **Root Cause**:
    - Platform load spikes are not backed by scalable infrastructure.
    - Real-time monitoring for fund transfer performance is missing.

4. Fixes (Immediate Implementation: Within 1 Week)

1. Bank Routing Optimization

a. Dynamically route transactions through partner banks with the lowest average delays (e.g., Yes Bank, ICICI).
b. Avoid banks with known outages (e.g., HDFC, Axis) during peak hours.

2. Retry Logic Overhaul

a. Implement an **exponential backoff strategy**: wait 5 minutes → before retrying.
b. Limit retries to a **maximum of 2** to prevent compounding delays.

3. Auto-Escalation Mechanism

a. Introduce a system that flags transactions pending >45 mins (NEFT) or >15 mins (IMPS/UPI) to the Ops team.
b. Auto-log these into Zendesk/JIRA for faster resolution.

4. User Communication Layer

a. Send **real-time delay notifications** via in-app alerts, SMS, or email.
b. Display live status and estimated time to credit for transparency.

5. Quick Bank Performance Snapshot (Internal Tool)

a. Build a lightweight dashboard tracking average. Delay, failure rate, and SLA breach% % per bank.

5. Fixes (Strategic Improvements: 2–4 Weeks)

1. **Real-Time SLA Dashboard (Bank × Method)**

    a. Developed a full-scale dashboard using **Tableau / Power BI / QuickSight** to monitor:

        i. Average delay per bank and method
        ii. Failure/pending count
        iii. Retry trends and escalations

2. **Smart Failover System**

    a. For NEFT transactions below ₹2L, **auto-switch to IMPS** during bank downtime or NEFT window closures.

3. **ML-Based Delay Prediction Engine**

    a. Train a model using historical transaction data to predict potential delays based on:

        i. Time of day
        ii. Bank
        iii. Transaction amount
        iv. Retry history

4. **Partner Bank SLA Contracting and Alerts**

    a. Formalize Service Level Agreements (SLAs) with partner banks.
    b. Trigger alerts if SLA thresholds are breached (e.g., 95% of NEFTs must settle <60 minutes).

5. **Platform Infrastructure Scaling (Peak Load)**

    c. Auto-scale microservices to handle transaction processing during high-traffic periods (9–11 AM & EOD).
    d. Optimize DB write latency and API responsiveness under load.

6. Solution Prioritization & Trade-offs:

To further strengthen this case study from a Product Analyst perspective, the following elements would be incorporated into the project lifecycle:

1. **Solution Prioritization & Trade-offs (Using RICE Framework):** Solutions would be rigorously prioritized using the RICE framework (Reach, Impact, Confidence, Effort) to determine their sequence of implementation. This ensures that the most impactful and feasible solutions are tackled first, delivering maximum value to Upstox and its users.
    - **RICE Score Calculation:** `(Reach * Impact * Confidence) / Effort`
        - **Reach (R):** Number of users affected (e.g., 1 = Few, 2 = Some, 3 = Many, 4 = Most, 5 = All)
        - **Impact (I):** Magnitude of positive effect on goals (e.g., 1 = Minimal, 2 = Low, 3 = Medium, 4 = High, 5 = Critical)
        - **Confidence (C):** How certain we are of success (e.g., 1 = Low, 2 = Medium, 3 = High)
        - **Effort (E):** Resources needed (e.g., 1 = Days, 2 = Weeks, 3 = 1-2 Sprints, 4 = 3-4 Sprints, 5 = Months)

Solution	Reach (R)	Impact (I)	Confidence (C)	Effort (E)	Score	Priority
Retry Logic Overhaul	**4**	**5**	**3**	**2**	**30**	**1st**
Bank Routing Optimization	**5**	**4**	**3**	**2**	**30**	**2nd**
User Communication Layer	**5**	**4**	**3**	**1**	**60**	**High**
Auto-Escalation Mechanism	**3**	**3**	**3**	**2**	**13.5**	**3rd**
Smart Failover System	**3**	**5**	**2**	**3**	**10**	**4th**
ML-Based Delay Prediction Engine	**4**	**5**	**1**	**5**	**4**	**Lowest**

Platform Infrastructure Scaling	**5**	**5**	**2**	**5**	**10**	**Lower**
Partner Bank SLA Contracting	**5**	**4**	**2**	**4**	**10**	**Medium**

- **Prioritization Rationale:**
  - User Communication Layer stands out as a quick win due to high reach/impact and low effort, addressing immediate user frustration.
  - Retry Logic Overhaul and Bank Routing Optimization are top technical priorities as they directly address core system inefficiencies and impact a large number of delayed transactions.
  - Smart Failover System and ML-Based Delay Prediction are high-impact, long-term strategic initiatives, but require more effort and have slightly lower initial confidence due to complexity.
  - Platform Infrastructure Scaling and Partner Bank SLA Contracting are critical foundational efforts that may not show immediate user-facing impact, but are essential for long-term stability and prevention.

- **Anticipation of Trade-offs:**
  - Implementing exponential backoff for retries, while beneficial overall, might mean a slightly longer initial wait for a very small percentage of users compared to instant retries.
  - Auto-switching NEFT to IMPS (for <₹2 Lakh) might incur marginally higher per-transaction costs for Upstox, a trade-off balanced against significantly improved user experience and reduced support burden.

Refined Success Metrics & Monitoring:

- **Specific Targets & Baselines:** Beyond the general KPIs, each implemented solution would have precise success metrics with specific targets and baselines. For instance, "Reduce the proportion of NEFT transactions with >1 retry attempt from 9.6% to <2% within 2 months of retry logic deployment."
- **A/B Testing Strategy:** For features like new routing algorithms or retry backoff strategies, a conceptual A/B testing approach would be designed.

This would involve rolling out the new logic to a small, controlled segment of users initially to validate its positive impact on KPIs before a broader deployment.

7. Dashboard Mockup (KPIs to Track)

KPIs:

Metric	**Description**
Avg Delay by Method	NEFT, UPI, IMPS avg. time (mins)
Delay Heatmap by Bank	Color-coded bank-wise performance
% Transactions Delayed	>30 mins delay flag
Status Trend Line	SUCCESS, FAILED, PENDING over time
Retry Attempts Funnel	Distribution of retries (0–3)

7. Business Impact

Projected Outcome after Implementation:

Metric	**Before**	**After**
Avg.. NEFT Delay	194 mins	45 mins
Pending Transactions	100	<10
Support Tickets (Apr)	3000+	<800
User Churn	High	Reduced by 40%
Trading Volume Recovery	-12%	Back to 98%+

8. Conclusion

This project effectively diagnosed Upstox's transaction delay issues using data analysis, validating root causes, and quantifying impacts. By translating raw data into actionable insights and presenting them via a dashboard, it provides a clear roadmap for resolution.