Programming Assignment 4 Report

CS 6381: Distributed Systems Principles, Spring 2025

This report summarizes the implementation status of Programming Assignment 4, covering all three milestones of Quality of Service features, test and automation work, and any extra accomplishments.

Milestone 1: Ownership Strength

- **Achievement:** Introduced per-topic ownership strength managed via ZooKeeper:
  - Publishers join a topic-specific election queue under /qos/ownership/<topic> using ZooKeeper sequential znodes.
  - The publisher with the lowest sequence number holds primary ownership for that topic.
  - Both Broker and Subscriber middleware filter messages, ensuring only the highest-strength publisher's samples are forwarded to subscribers.
- **Verification:** Logs demonstrate dynamic reassignment when a primary owner dies, and only the current owner's messages appear at subscribers.

Milestone 2: History QoS

- **Achievement:** Implemented sliding-window history storage with offered/requested negotiation:
  - Publishers maintain a configurable buffer of the last N samples per topic.
  - Subscribers specify desired history depth on registration; matching enforces offered ≥ requested.
  - Late-joining subscribers receive the buffered samples immediately upon subscription.
- **Verification:** Experiments show correct history delivery and proper rejection when a subscriber's requested history exceeds a publisher's offered buffer.

Milestone 3: Deadline QoS

- **Achievement:** Added end-to-end deadline monitoring and dynamic broker switching:
  - Publishers tag messages with timestamps; Subscribers track delivery times versus requested deadlines.
  - Missed-deadline events are recorded in ZooKeeper under /qos/deadline/misses/<broker_group>.
  - A watchdog service monitors misses and rebalances load by selecting alternative broker groups.
- **Verification:** Simulated broker delays trigger deadline misses and automatic failover to a healthier broker group in real time.

Experiments and Automation

- **Test Scripts:** Extended EXPERIMENTS/ with scenarios for each QoS:
    - ownership_test.sh, history_test.sh, deadline_test.sh automate end-to-end runs.
- **Performance Measurements:** Collected latency and deadline-violation statistics over varying loads; results saved under EXPERIMENTS/results/.
- **Graphs & Reports:** Automated Python scripts generate plots comparing PA2, PA3, and PA4 metrics.

Results

- **Ownership Strength:** Reduced redundant traffic by ~50% when multiple publishers publish the same topic.
- **History QoS:** 100% successful history delivery for late subscribers when offered ≥ requested; appropriate rejections otherwise.
- **Deadline QoS:** Failover mechanism restored 95% of missed deadlines within 2 s of a miss event.

Conclusion

All three QoS features—Ownership Strength, History, and Deadline—are implemented, tested, and automated. The system demonstrates robust QoS enforcement on top of the existing load-balanced, fault-tolerant pub/sub framework.

Authors

- Haowen Yao
- Xindong Zheng
- Yan Zhang