# Conclusion

This research investigates a blockchain-based queuing system that models the transaction process under different combinations of user priority and impatience, while incorporating ON/OFF operational states that reflect the stochastic availability of block generation and consensus phase. The blockchain queuing system comprises two queues in series: a customer queue, where users wait to be grouped into a block, and a block queue, where grouped users await consensus. The system supports a partial batch generation mechanism, allowing up to users to be grouped at once. The ON/OFF states represent the operational availability of the system; during OFF periods, block generation and consensus are suspended, while arrivals continue. We model four scenarios: (1) single-class customers without impatience, (2) two-class customers without impatience, (3) Single-class customers with impatience, and (4) two-class customers with impatience.

For each scenario, we construct a multi-dimensional Markov chain that captures the state of the system. Balance equations are formulated and solved iteratively until the steady-state probability distribution is obtained. Based on these probabilities, key performance metrics are calculated, including throughput, blocking probability, and average waiting times. To validate our analytical findings, we develop a discrete-event simulation implemented in C++. The simulation strictly adheres to the queuing logic and service discipline of each scenario to ensure consistency with the analytical model.

In conclusion, based on our simulation and analytical results, we have made several interesting observations. First, as the system adopts a non-preemptive priority discipline to ensure that low-priority customers still receive fair service, high-priority customers outperform low-priority customers even with the same block generation rate. Second, the introduction of the impatience mechanism in scenarios with impatience leads to improvements in most performance metrics compared to scenarios without impatience. Notably, however, remains unaffected by the impatience rate. Third, in every scenario, we observe that as increases, initially grows but eventually stabilizes around constant value. This is because the system can achieve a balance between and without necessarily reaching the maximum batch size. Fourth, under the priority-base scenario, increases with but the increase gradually slows. This is attributed to the limitation imposed by arrival rate. Fifth, when increasing the under the priority-base scenario, and remains constant. However, shows a down trend. This is because a higher increases the proportion of high-priority customers in the system. This shift in priority composition also affects other metrics, such as , which decreases accordingly. Sixth, under the priority-base scenario, lead to an upward trend in . This is because faster departure of high-priority customers frees up block queue capacity, allowing more low-priority customers to be included in the consensus process per cycle.

Future research directions include incorporating a more realistic voting mechanism into the consensus phase to better reflect practical blockchain operations. In addition, further extensions may involve optimizing the batch policy dynamically based on the real-time queue state, and enhancing the impatience modeling to capture the total time until departure, including both queueing and consensus delays.