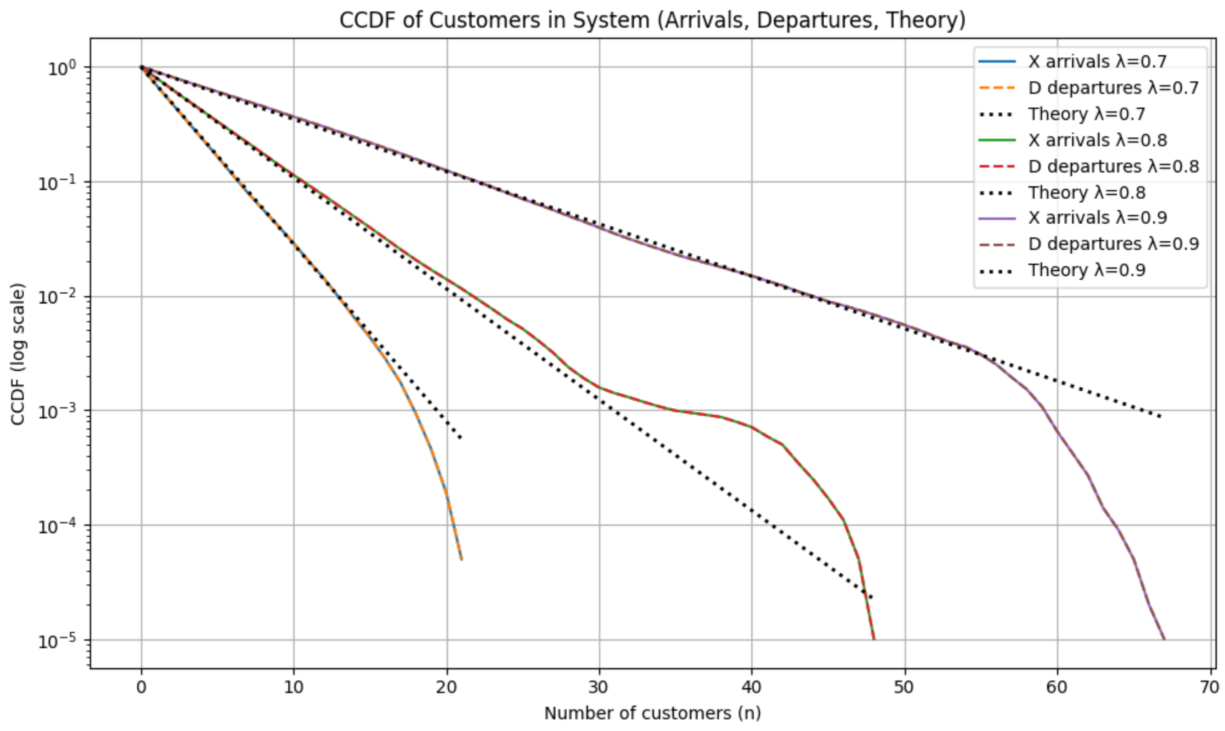
Exploring Queue Dynamics and Load Balancing in Multi-Server Systems

This report explores queue dynamics and load balancing in single- and multi-server systems through simulation. The first section focuses on an M/M/1 queue, analyzing customer occupancy and busy periods under varying arrival rates. The second section investigates multiple parallel queues with different routing policies—uniform random, join-shortest-queue, and power-of-two-choices—highlighting their impact on waiting time distributions. Complementary cumulative distribution functions (CCDFs) are used to characterize system behavior and compare empirical results with theoretical predictions.

<u>Part One: Queue Occupancy and Busy Periods in an M/M/1 System</u>

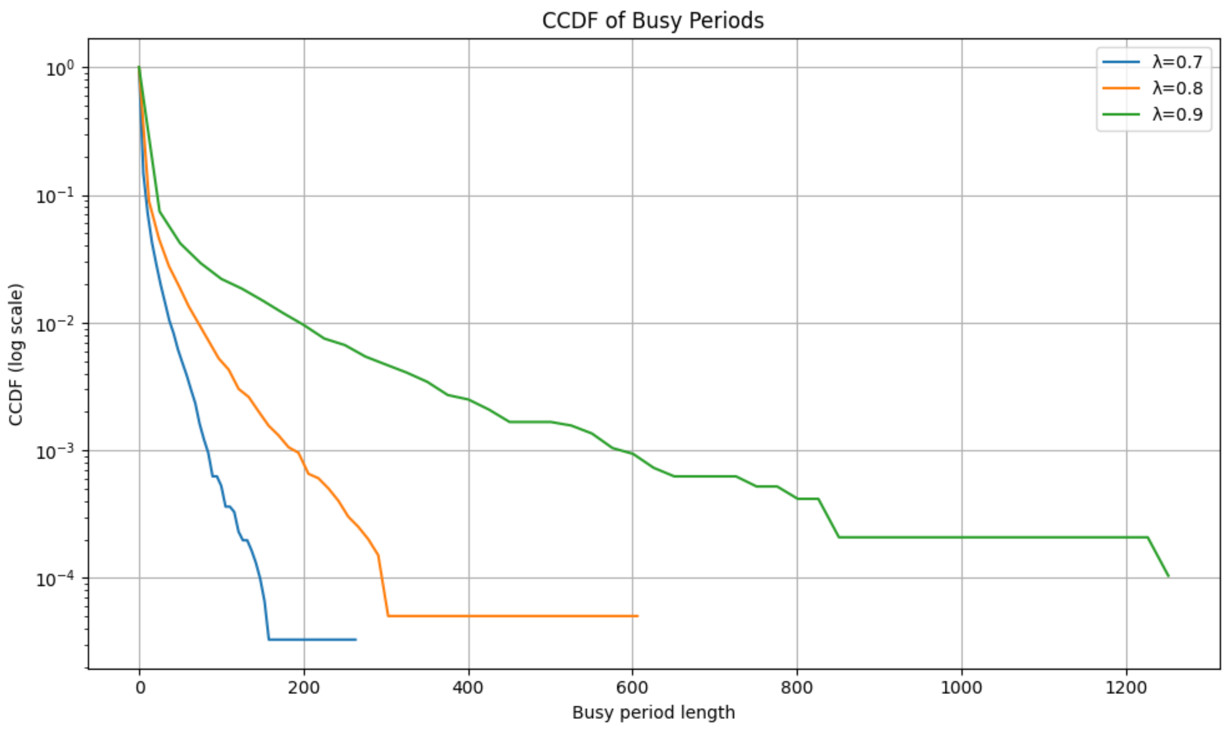
This section presents simulation results for an M/M/1 queue and compares them with theoretical predictions. The simulations track the number of customers in the system at arrival and departure events, illustrating the PASTA property. Additionally, the dynamics of busy periods are analyzed under varying traffic intensities (λ). These results provide insight into how queue occupancy and busy periods evolve with increasing load, while also highlighting the natural variation that arises in simulations with a finite number of customers.



The simulation results for the M/M/1 queue show agreement with theoretical predictions for the number of customers in the system. In the first graph, the complementary cumulative distribution function (CCDF) of the number of customers seen by arrivals, *X(ti)*, closely follows the theoretical line

$$P(X \geq n)=\rho n$$

for all three values of λ, confirming the PASTA (Poisson Arrivals See Time Averages) property. The CCDF for departures, *Di*, is slightly lower than that for arrivals, as expected. This is because departures observe the system immediately after a service completion, which reduces the queue length by one.

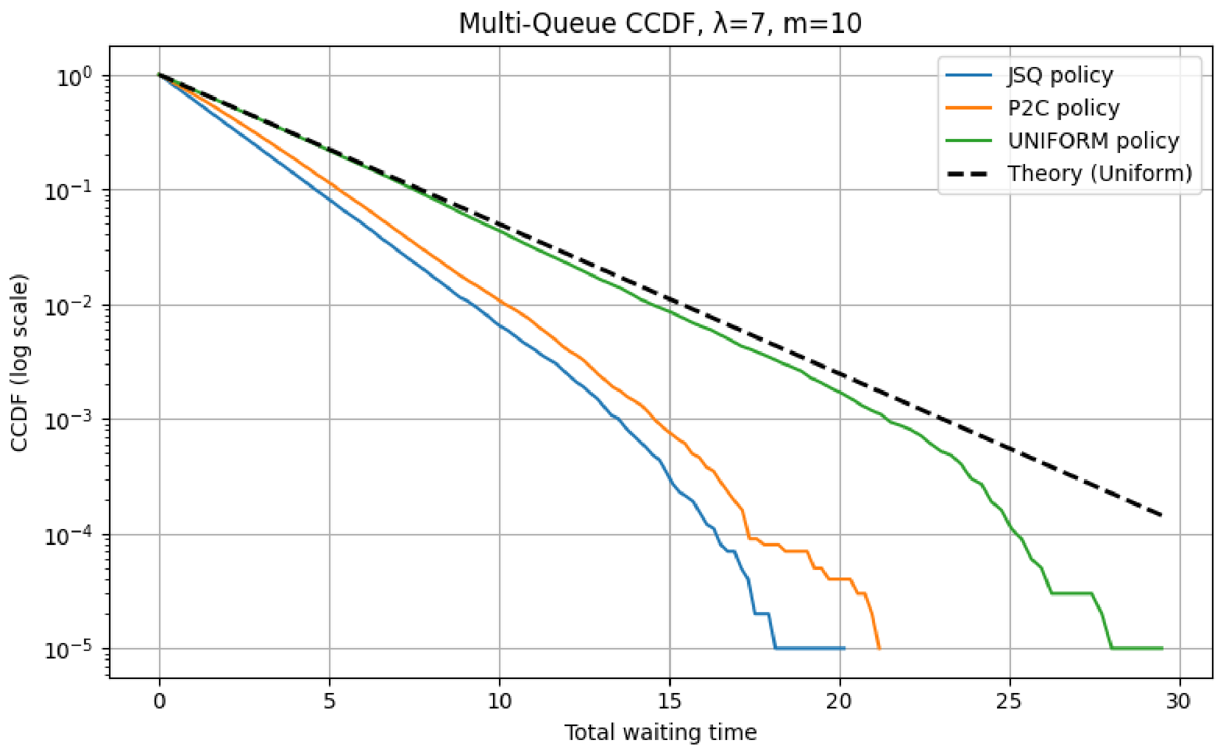


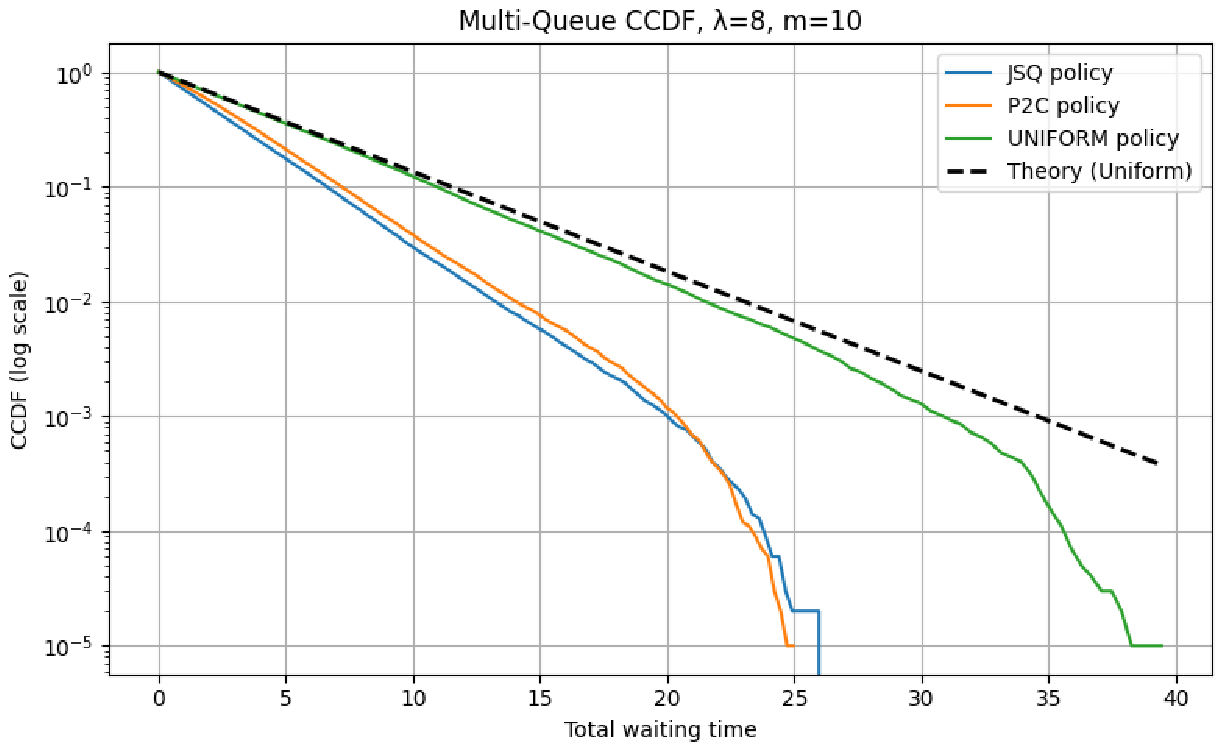
In the second graph, which shows the CCDF of busy periods, Bi , all curves start together at the upper-left corner, reflecting that the probability of a busy period being greater than zero is 1. As the busy period length increases, the curves diverge: higher λ values produce longer tails, indicating that under heavier traffic, the server remains busy for longer durations. The CCDFs of busy periods show that the server tends to stay busy longer as the arrival rate increases, consistent with the expected exponential behavior of an M/M/1 queue. Minor fluctuations in the curves are due to the finite number of simulated runs, but overall the data closely follow the theoretical pattern.

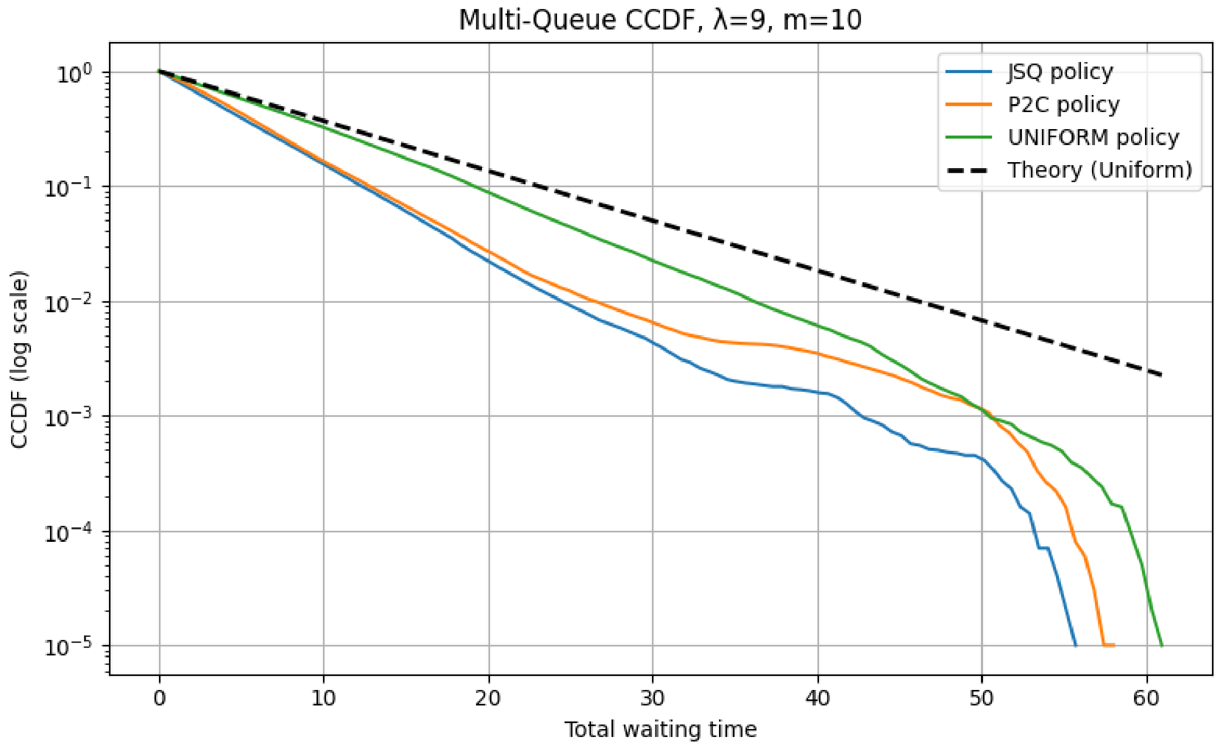
Overall, the simulation validates theoretical predictions for system occupancy and provides insight into the behavior of busy periods under different traffic intensities. Minor deviations between the empirical curves and the theoretical lines are due to the finite number of customers simulated (N=10,000) and the limited number of independent runs (K=10). These fluctuations are most noticeable for larger n values, where fewer samples exist. To reduce simulation error, one could increase *N* and *K*, or apply variance reduction techniques to achieve smoother and more accurate empirical distributions.

Part Two: Comparing Load-Balancing Policies: JSQ, Po2C, and Uniform-Random

This section presents simulation results comparing different load-balancing policies—Join-the-Shortest-Queue (JSQ), Power-of-Two-Choices (Po2C), and Uniform-Random—across various arrival rates (λ). The aim is to demonstrate how routing decisions influence queue lengths and waiting times. JSQ serves as the theoretical benchmark, always assigning jobs to the shortest queue, while Po2C approximates JSQ by considering only two randomly selected queues, and Uniform-Random assigns jobs without regard to queue lengths. The results highlight which policies maintain low waiting times under both light and heavy load, illustrate the practical efficiency of Po2C as a near-optimal solution, and show how queue behavior evolves as the system approaches capacity.




Multi-Queue CCDF, λ=8, m=10
JSQ policy
P2C policy
UNIFORM policy
Theory (Uniform)
CCDF (log scale)
Total waiting time


Multi-Queue CCDF, λ=9, m=10
JSQ policy
P2C policy
UNIFORM policy
Theory (Uniform)
CCDF (log scale)
Total waiting time

The simulation results clearly show that the Join-the-Shortest-Queue (JSQ) policy performs best in minimizing total waiting time. By always routing arriving jobs to the currently shortest queue, JSQ effectively balances the load across all servers, preventing any single queue from becoming excessively long. This results in the leftmost CCDF curves across all λ values, indicating that a higher proportion of jobs experience shorter waiting times under JSQ compared to other policies. In contrast, the Uniform-Random policy performs the worst, as jobs are assigned without regard to queue lengths, causing some queues to become overloaded while others remain underutilized.

The Power-of-Two-Choices (Po2C) policy consistently performs close to JSQ, demonstrating its effectiveness as a practical approximation. By randomly selecting two queues and routing the job to the shorter of the two, Po2C achieves near-optimal load balancing with significantly lower computational overhead than JSQ, which requires checking all m queues. Across all simulated λ values, the Po2C CCDF curves lie slightly to the right of JSQ but remain substantially better than uniform-random, illustrating that even limited sampling can provide excellent performance.

Examining the CCDF plots across different arrival rates reveals interesting trends. For low arrival rates (e.g., $\lambda = 7$), all three policies produce relatively similar waiting times because queues rarely become congested. As λ increases, approaching system capacity, the differences become more pronounced: JSQ maintains the smallest waiting times, Po2C remains close, and uniform-random's waiting times increase sharply. There is generally a consistent ordering of the CCDFs—JSQ < Po2C < uniform-random—across most of the range, though minor crossovers may appear in the tail for very low load due to stochastic variability.

To manage maximum queue length without high computational cost, Po2C already provides a strong solution. Alternative strategies include Power-of-k-Choices (with $k > 2$), which slightly improves performance while still being efficient, Join-Idle-Queue (JIQ), which routes jobs to idle servers, and threshold-based or approximate queue-length methods, which reduce the need for exact monitoring of all queues. Overall, while JSQ is theoretically optimal, Po2C and these variations strike a practical balance, achieving near-optimal waiting times while minimizing system overhead—making them highly suitable for large-scale cloud or server farm environments.

Conclusions

The simulations presented in this report provide a comprehensive view of queue dynamics in both single-server and multi-server systems. For the M/M/1 queue, the number of customers observed at arrivals and departures closely follows theoretical predictions, confirming the PASTA property. Busy periods grow in expected length as traffic intensity increases, but their distribution remains exponential with rate $\mu-\lambda$. In multi-queue systems, the Join-the-Shortest-Queue (JSQ) policy consistently minimizes waiting times, with the Power-of-Two-Choices (Po2C) policy achieving near-optimal performance at significantly lower computational cost. Uniform-Random routing performs poorly, especially under heavy load, highlighting the importance of informed load-balancing strategies. Although the Uniform curve appears closest to the theoretical line in the CCDF plots, this occurs because the theoretical distribution corresponds specifically to the Uniform-Random model (an M/M/1 queue with arrival rate λ/m). JSQ and Po2C outperform this baseline, leading to lighter tails and lower waiting times. Across all scenarios, CCDF analysis effectively illustrates how traffic intensity and routing policies impact system performance, and the results suggest practical strategies—such as Po2C, Power-of-k-Choices, or Join-Idle-Queue—to manage maximum queue lengths while maintaining efficiency in large-scale service systems. Overall, these findings demonstrate that carefully chosen routing policies can greatly enhance performance and reliability in multi-server environments.