Stochastic Models and Simulation: Queueing Foundations

Lecture 2 – Birth–Death Models, M/M queues, and Beyond

Sebastian Müller

Lecture 2

From Poisson Counts to Queues

- Lecture 1: Poisson counts, thinning, simulation boilerplate.
- Today: turn arrival counts into full service systems.
- Core questions:
  - How do waiting times/queue lengths arise from stochastic primitives?
  - What closed-form results exist for Markovian queues?
  - How do we simulate when formulas are unavailable?

Building Blocks

Definition

A queueing system is described by arrivals, service mechanism, number of servers, capacity, and service discipline.

- Kendall notation $A/S/c/K/m/Z$ with defaults ∞ capacity, infinite population, FIFO.
- States often captured by customer count $N(t)$.
- Stability requires arrival rate λ smaller than total service capacity.

Little's Law

Theorem

$L = \lambda W$, holds for any stable queue in steady state.

- L: expected number in system, λ: effective arrival rate, W: expected sojourn time.
- Corollaries: $L_q = \lambda W_q$, throughput equals arrival rate when stable.
- Applies to simulations: we validate estimates by cross-checking Little's identities.

Birth–Death Chains

- Many queues map to continuous-time Markov chains with transitions $n \to n+1$ (birth with rate λ_n) and $n \to n-1$ (death with rate μ_n).
- Balance equations yield stationary probabilities π_n when $\sum_n \pi_n = 1$.
- For homogeneous rates: $\pi_n = \pi_0 \prod_{k=1}^{n} \frac{\lambda_{k-1}}{\mu_k}$.
- Convergence criterion: $\limsup_{n\to\infty} \prod_{k=1}^{n} \lambda_{k-1}/\mu_k < 1$.

Interpreting State Probabilities

- π_n answers: fraction of time system holds n customers, or probability an arriving job sees n in steady state (PASTA).
- Performance measures as expectations: $L = \sum n\pi_n$, blocking probability $= \pi_K$ for finite-capacity queues.
- Insight: small changes in utilisation can dramatically shift mass toward high n when ρ close to 1.

M/M/1 Recap

- Arrival rate λ, single server with rate μ.
- Stability: $\rho = \lambda/\mu < 1$.
- Performance:

$$L = \frac{\rho}{1-\rho}, \qquad L_q = \frac{\rho^2}{1-\rho},$$
$$W = \frac{1}{\mu-\lambda}, \qquad W_q = \frac{\rho}{\mu-\lambda}.$$

- Queue-length distribution: geometric $\mathbb{P}[N = n] = (1-\rho)\rho^n$.
- Time in system: exponential with mean $1/(\mu - \lambda)$.

M/M/1 Intuition

- ρ is utilisation: proportion of time server is busy.
- As $\rho \uparrow 1$, mean wait grows like $\frac{1}{1-\rho}$: diminishing returns of adding load.
- Memoryless service $\Rightarrow$ past does not inform remaining service time (strong assumption!).
- Sensitivity analysis: 10% error in ρ translates to large swings in W_q when near saturation.

M/M/c with Infinite Buffer

- Model: M/M/c, or Erlang-C. c parallel servers, Poisson arrivals λ, exponential service μ.
- Offered load and utilisation: $\rho = \lambda/(c\mu)$; stability requires $\rho < 1$.
- Birth–death structure: $\lambda_n = \lambda, \quad \mu_n = \min(n, c)\,\mu$.
- Erlang-C waiting probability (with $a = \lambda/\mu = c\rho$):

$$P_{\text{wait}} := C(c, \lambda/\mu) := \frac{\dfrac{(c\rho)^c}{c!}\,\dfrac{1}{1-\rho}}{\displaystyle\sum_{k=0}^{c-1} \frac{(c\rho)^k}{k!} + \frac{(c\rho)^c}{c!}\,\frac{1}{1-\rho}}.$$

  Probability that arrival is forced to join the queue.
- Performance via Little's Law: $L_q = P_{\text{wait}}\,\dfrac{\rho}{1-\rho}, \quad W_q = L_q/\lambda,$ $W = W_q + 1/\mu, \quad L = \lambda W.$

M/M/c Intuition

- Adding servers reduces wait times dramatically when ρ close to 1.
- Diminishing returns: each additional server helps less than the previous one.
- Key design question: balance cost of servers vs. cost of customer waiting.
- Example: call centers, cloud computing, hospital wards.

M/M/c/K and Blocking

- Finite capacity K: arrivals finding system full are lost.
- The M/M/c/c is known as Erlang-B model. No queueing, just blocking.

M/M/c/K Details

- Let offered load $a = \lambda/\mu$ and $\rho = a/c$. Stationary probabilities:
$$\pi_n = \pi_0 \begin{cases} \dfrac{a^n}{n!}, & 0 \le n \le c, \\ \dfrac{a^n}{c!\, c^{\,n-c}}, & c \le n \le K, \end{cases} \quad \text{with}$$
$$\pi_0^{-1} = \sum_{n=0}^{c-1} \frac{a^n}{n!} + \frac{a^c}{c!} \, \frac{1-\rho^{\,K-c+1}}{1-\rho}.$$
- Blocking probability: $\pi_K = \dfrac{a^c}{c!} \, \pi_0 \, \rho^{\,K-c}$. Accepted arrival rate: $\lambda_a = \lambda \pi_K$.
- Exact mean times and counts:
$$W_q = \frac{\pi_0 \, \rho \, (c\rho)^c}{\lambda \, (1-\rho)^2 \, c!}, \qquad W = W_q + \frac{1}{\mu}.$$
$$L_q = \lambda_a \, W_q, \qquad L = \lambda_a \, W = \frac{\lambda_a}{\mu} + L_q.$$

Case Study: ICU Triage

- Beds correspond to servers; rooms limited $\Rightarrow$ M/M/1/K (or M/M/c/K).
- Key decisions: number of surge beds, transfer policies, triage thresholds.
- Blocking corresponds to diverting patients; quantify expected diversions per day.
- Simulations incorporate surge arrivals, length-of-stay variance, priority rules.

Combining theory + simulation informs contingency planning with quantitative evidence.

Simulation Outputs to Track

- Time series: queue length $N(t)$, utilisation, waiting time trajectories.
- Distributional summaries: histograms, quantiles, tail probabilities.
- Diagnostics: Little's Law gaps, autocorrelation, warm-up bias detection.
- Sensitivity: rerun with perturbed λ, μ to gauge robustness.

Use these views to communicate findings to non-technical stakeholders.

When Exponentials Fail

- Real systems often exhibit general service times or bursty arrivals.
- Example: M/G/1 (Poisson arrivals, general service); closed-form results via Pollaczek–Khinchine formula.
- G/G/1: few general formulas; rely on approximations and simulation.
- Renewal theory helps when inter-arrivals have finite mean; some heavy-tail cases have infinite variance.

Why Non-Markov Models Matter

- Customer patience distribution drives abandonment behaviour (call centers, web services).
- Service-time variance dominates mean wait (cloud functions, healthcare lengths of stay).
- Regulatory/compliance constraints require tail guarantees, not just averages.
- Simulation allows scenario testing when analytic formulas break down.

Pollaczek–Khinchine Snapshot

- For M/G/1 with service time S (mean $\mathbb{E}[S]$, variance $\mathrm{Var}(S)$):

$$W_q = \frac{\lambda \mathbb{E}[S^2]}{2(1-\rho)}, \quad W = W_q + \mathbb{E}[S].$$

- Reveals sensitivity to service variance; heavy-tailed S inflates W_q massively.
- Distributional results harder; simulation gives empirical quantiles/tails.

Example: Lognormal Service

- Same mean service as exponential, but lognormal with $\sigma = 0.6$ doubles $\mathbb{E}[S^2]$.
- Pollaczek–Khinchine predicts W_q nearly doubles: variability is as important as mean.
- Notebook visualises histograms, time-averages, and tail probabilities for exponential vs. lognormal.
- Use quantitative bounds (CLT/Bernstein) to report uncertainty in estimated averages.

Markov-Modulated Poisson Processes (MMPP)

- Arrival rate driven by a background CTMC with states $1, \ldots, m$ and generator Q.
- In state i, arrivals follow a Poisson process with intensity λ_i.
- Captures burstiness/seasonality while retaining tractable structure (phase-type arrivals, matrix-analytic methods).

Useful for modelling traffic with bursts, e.g., telecom networks or demand spikes in e-commerce.

Hands-On: Comparing Models

The accompanying notebook features:

1. Plain-Python and SimPy M/M/1 and M/M/c simulations with validation against theory.
2. Non-Markov queue example (M/G/1 with lognormal service) including:
    - Empirical waiting-time histograms vs. Pollaczek–Khinchine predictions.
    - Confidence intervals / concentration bounds for estimated means.
    - Visual comparisons of exponential vs. heavy-tail behaviour.
3. Template for students to extend to G/G/1.

Takeaways

- Birth–death models yield closed forms for many performance metrics.
- Simulations complement theory: diagnostics, sanity checks, non-Markov cases.
- Next lecture: renewal theory and regenerative processes for general systems.