Queueing Theory

Example: Single server, infinite buffer. Arrivals and departures only happen at discrete time steps (like clock ticks). Step size is δ seconds.

Assume: δ is very small, only 1 packet arrives and departs within one δ.

Define: The system state is the # of packets in the system including the packet in the server

At time $t = n\delta$, assume there are 2 packets in the system, 1 getting service and 1 waiting in the queue. What are the possible system states at time $t = (n+1)\delta$?

i A new arrival and no departure $\longrightarrow$3 packets

ii A new arrival and the packet in service departs $\longrightarrow$ 2 packets

iii No arrival and no departure $\longrightarrow$ 2 packets

iv No arrival and the packet in service departs $\longrightarrow$ 1 packet

Observe: The # of packets in the system at $t = (n+1)\delta$ (the next state) depends on:

- the # of packets at $t = n\delta$ (present state)
- the probability of arrival and the probability of departure within δ

The system is memoryless. The next state depends only on the present state and present arrivals/departures. This is called MARKOVIAN PROPERTY.

What do we want to know?

$\Rightarrow$ $E[N_s]$: The expected (average) # of packets in the system at steady state.

Define Π_i: The steady state probability that the system is in state i (has i packets).

$$E[N_s] = \sum_{i=0}^{\infty} i \cdot \Pi_i \qquad \text{Note:} \sum_{i=0}^{\infty} \Pi_i = 1$$

Questions:

Q1) How to model the discrete time system?

Q2) How to find Π_i?

Q3) How to get the real-life continuous time model from the discrete time model?

Q4) Remember Kendall's notation A/B/m/K/M . What kind of queue do we model with this approach?

A1) Model: Discrete time Markov Chain.

Define: probability of arrival within δ : p

probability of departure within δ : q

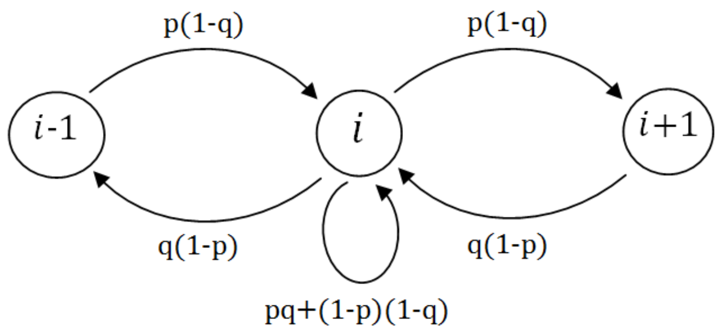


Figure 1: Discrete time Markov Chain

Given the system at time step $n\delta$. The system can go to state i from states $i-1$ and $i+1$ or can stay at state i.

$$\Pi_i = \Pi_i[pq + (1-p)(1-q)] + \Pi_{i-1}[p(1-q)] + \Pi_{i+1}[q(1-p)]$$

Rearrange this to get a "balance equation".

$$\underbrace{\Pi_i[(1-p)q + (1-q)p]}_{\text{Prob. of leaving state } i} = \underbrace{\Pi_{i-1}[p(1-q)] + \Pi_{i+1}[q(1-p)]}_{\text{Prob. of entering state } i \text{ from state } i-1 \text{ or } i+1}$$

A2) How to compute Π_i's:

Write such balance equations and use $\sum \Pi_i = 1$ to compute Π_i's.

A3) Getting the real-life continuous time model:

For time step λ, probability of packet arrvial is p.
Define $\lambda = p/\delta$: average packet arrival rate.
Similarly, $\mu = q/\delta$: average packet departure rate. Therefore:

$$p = \lambda\delta \quad \text{and} \quad q = \mu\delta$$

Rewrite the balance equation:

$$\Pi_i[q - pq + p - pq] = \Pi_{i-1}[p - pq] + \Pi_{i+1}[q - pq]$$
$$\Pi_i[\mu\delta - \mu\lambda\delta^2 + \lambda\delta - \mu\lambda\delta^2] = \Pi_{i-1}[\lambda\delta - \mu\lambda\delta^2] + \Pi_{i+1}[\mu\delta - \mu\lambda\delta^2]$$

For continuous model δ is very small as $\delta \to 0$. Thus we can ignore 2^{nd} order δ terms.

$$\Pi_i[\mu\delta + \lambda\delta] = \Pi_{i-1}[\lambda\delta] + \Pi_{i+1}[\mu\delta]$$

Up to now we expressed the system probabilities because it was discrete time steps. For continuous time, $\delta \to 0$ case, probabilities are replaced by rates. So we divide the balance equation by δ to take the time derivative.

$$\underbrace{\Pi_i(\mu + \lambda)}_{\text{Leave}} = \underbrace{\Pi_{i-1}\lambda + \Pi_{i+1}\mu}_{\text{Enter}} \quad \Longrightarrow \quad \text{C.T Balance Equation}$$

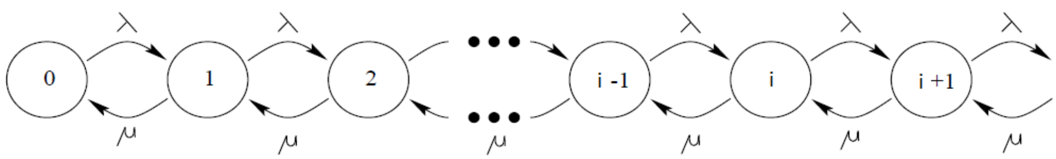


Figure 2: Continuous Time Markov Chain

Example: Write the balance equations for the system with λ and μ. Define: $\rho = \dfrac{\lambda}{\mu}$

$$\begin{aligned}
\Pi_0 \cdot \lambda = \Pi_1 \cdot \mu \qquad & \Rightarrow \qquad \Pi_1 = \Pi_0 \cdot \frac{\lambda}{\mu} \\
\Pi_1(\lambda + \mu) = \Pi_2 \cdot \mu + \Pi_0 \cdot \lambda \qquad & \Rightarrow \qquad \Pi_2 \cdot \mu = \Pi_0 \cdot \frac{\lambda}{\mu}(\lambda + \mu) - \Pi_0 \cdot \lambda \\
& \Rightarrow \qquad \Pi_2 \cdot \mu = \Pi_0[\frac{\lambda^2}{\mu} + \lambda - \lambda] \\
& \Rightarrow \qquad \Pi_2 = \Pi_0 \cdot \frac{\lambda^2}{\mu^2}
\end{aligned}$$

$$\boxed{\Pi_i = \Pi_0 \cdot \left(\frac{\lambda}{\mu}\right)^i = \Pi_0 \cdot \rho^i}$$

The total probability is 1. Therefore:

$$\begin{aligned}
\sum \Pi_i = 1 \quad & \Rightarrow \quad \Pi_0 \sum_{i=0}^{\infty} \rho^i = 1 \\
& \Rightarrow \quad \boxed{\Pi_0 = 1 - \rho} \\
& \Rightarrow \quad \boxed{\Pi_i = (1 - \rho)\rho^i}
\end{aligned}$$

Q: What is the probability that the system is not empty (serving some packet)?

A: $1 - \Pi_0 = \rho$

Q: What is the average # of packets getting service over time?
Take the system snapshot 100 times. In some of these snapshots there is 1 packet getting service and in some of them there is 0. Can it be 2?

A: $\dfrac{\text{\# of snapshots with 1 packet}}{\text{Total snapshots}}$ = Probability that the system is not empty

Expected # of packets in the system:

$$E[N_s] = \sum_{i=0}^{\infty} i \cdot \Pi_i = \frac{\rho}{1-\rho}$$

Expected # of packets in the queue:

$$E[NQ] = E[N_s] - E[N_{server}] = \frac{\rho}{1-\rho} - \rho$$

A4) What kind of queue is this?

This analysis is only correct for memoryless systems. You could have a queue where packet arrivals depend on the # of packets in the system some time ago. Then this analysis would not work. Our analysis is valid for Markovian processes.

Markovian processes have some useful properties:

- State transitions happen with exponential distribution.
- State transitions happen with arrivals and departures so our analyis is valid for some queue where interarrival times and service times (packet sizes) are exponentially distributed.

THIS WAS AN M/M/1 QUEUE!

$E[N_s]$, $E[NQ]$ are all valid for M/M/1 queues ONLY!

Exponential Distribution

A random variable is exponentially distributed with rate λ.

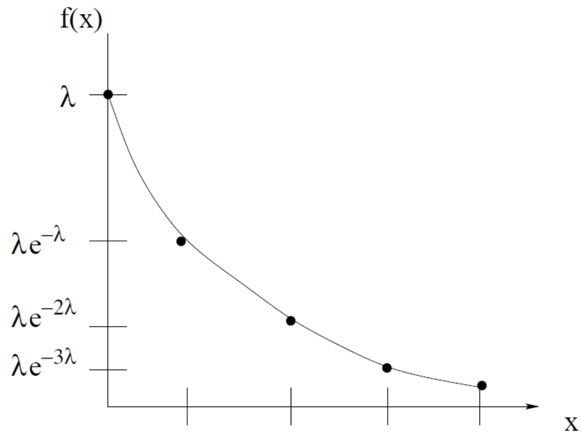
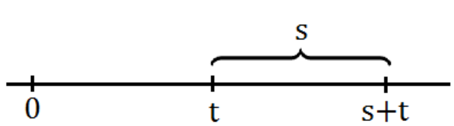


Figure 3: Exponential PDF

- $X \sim Exp(\lambda)$
- $f(x) = \begin{cases} \lambda e^{-\lambda x} & , \quad x \geq 0 \\ 0 & , \quad x < 0 \end{cases}$
- $F(x) = P(X \leq x) = \int_{-\infty}^{x} f(y)dy = 1 - e^{-\lambda x}$
- $E[x] = \int_{-\infty}^{\infty} x f(x)dx = \dfrac{1}{\lambda}$

Memoryless $\Longrightarrow X$ is $Exp(\lambda)$



$$P(X > s + t \mid X > t) = P(X > s)$$

Remark: Interarrivals are exponentially distributed $\Leftrightarrow$ Arrival is a Poisson process

Poisson Process:

- A sequence of events in time.
- $N(t)$: # of events that occur by time t.
- $N(t_1) - N(t_0)$ and $N(t_2) - N(t_1)$ are independent, which means $N(t)$ only depends on duration t(memoryless)

Then: Probability of k events on a time interval t is:

$$\frac{e^{-\lambda t}(\lambda t)^k}{k!} \quad , \quad \lambda\text{: rate of the Poisson process.}$$

Expected # of arrivals on an interval $t = \lambda t$

Properties:

1. If you merge n poisson streams with rates λ_i the resulting stream is also poisson with rate $\lambda = \sum_{i=1}^{n} \lambda_i$.
2. If you split a poisson process with rate λ into k substreams with probability p_i each resulting process is also poisson with rate $\lambda \cdot p_i$.
3. For a given single queue, if the arrivals are poisson with rate λ and the packet sizes are exponentially distributed, the departure is also a poisson process with rate λ.

Little's Law

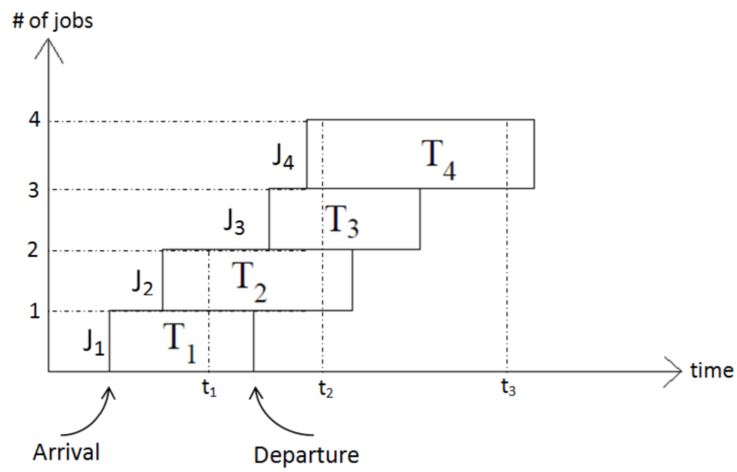


Figure 4: Arrival and departure of jobs (packets)

At any time t, the number of jobs in the system is: A(t)-C(t)

$$t_1 : 2 \text{ jobs}, \ J_1, J_2$$
$$t_2 : 3 \text{ jobs}, \ J_2, J_3, J_4$$
$$t_3 : 1 \text{ job}, \ J_4$$

As T$\rightarrow \infty$, Arrivals = Departures = N (What goes in goes out). The arrival rate is:

$$\lambda = \frac{N}{T}$$

Total amount of time spent in the system by all jobs is equal to the area between arrival curve and departure curve, which is J. Mean time in the system is:

$$E[T_s] = \frac{\text{Total time spent by all jobs}}{\text{\# of all jobs}} = \frac{J}{N}$$

The number of jobs in the system is:

$$E[N_s] = \frac{\text{Total time spent by all jobs}}{\text{Total time}} = \frac{J}{T}$$

$$\Rightarrow E[N_s] = \frac{J}{T} = \frac{J}{N} \cdot \frac{N}{T} = E[T_s] \cdot \lambda$$

Intuitively, rate of completion is equal to throughput $\implies X = \lambda$

I came, there are $E[N_s]$ things in the queue including myself. I wait for $\frac{1}{\lambda}$ to get out. Therefore:

$$E[T_s] = E[N_s] \cdot \frac{1}{\lambda}$$

Corollaries for Little's Law:

- $E[NQ] = \lambda \cdot E[TQ]$
- $E[N_{server}] = \lambda \cdot E[T_{server}]$

  $$\text{Utilization} = \lambda \cdot E[S] = \frac{\lambda}{\mu}$$
- $E[N_{redjobs}] = \lambda_{red} \cdot E[T_{red}]$

Finite Buffer Case:

Example: Single queue with G buffer space (including server). $\implies \quad \lambda \neq X$

Effective arrival rate is equal to the rate of jobs which an go through. Which is:

$$\lambda(1 - \text{P(G jobs in the system)}) \quad \implies \quad \text{Completion rate}$$

$$E[N_s] = \lambda(1 - \text{P(G jobs)})E[T_s]$$