TELE302 Lecture 8
Queueing Analysis: II

Jeremiah Deng

University of Otago

28 July 2015

Lecture Outline

1. Quick review
2. Variation I: $M/M/1/K$
3. Variation II: $M/M/c/c$
4. Variation III: to infinity

Overview

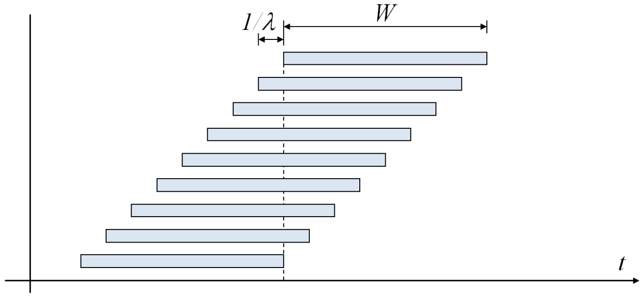
- Last lectures:
  - Queueing is everywhere
  - Markovian modeling of arrivals and departures
  - M/M/1
- Today: Let's play some *Queueing Variations*
  - What about limited buffer for queues?
  - What about multiple servers?

Kendall Notation $(A/B/c/K/m/Z)$

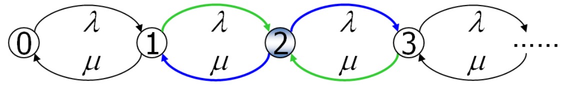
- A: interarrival time distribution
- B: service time distribution
  - M for exponential
  - G for general
- c: number of servers
- Optional:
  - K: maximum number of allowed customers
  - m: size of the customer population
  - Z: queueing discipline, typically FIFO

Little's Theorem

- **Number of customers in the system at time t is** $N = \lambda W$
  - λ: avg. arrival rate, $1/\lambda$: average interarrival time
  - W: Avg. time in system
- Little's Theorem actually holds for *every* queueing system.



M/M/1 Queuing



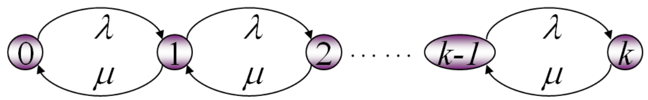
- Interarrival time: exponentially distributed, mean=$1/\lambda$
- Job processing time: exponentially distributed, mean=$1/\mu$
- Steady state requires $\lambda < \mu$
- FIFO
- One server
- Chances the server is busy: $P[N \geq 1] = 1 - p_0 = \rho$.
- Expected number in system: $L = E\{N\} = \sum_n np_n = \frac{\rho}{1-\rho}$

A Little Variation to $M/M/1$

- In M/M/1, there is no limit on the queue length.
- In reality, services usually don't support unlimited queueing (memory, ports etc.)
- If a customer finds no available position in a limited queue, it is supposed to disappear!

$M/M/1/K$ Analysis

- Transition diagram



- Steady state solutions:
  - $\sum_{n=0}^{K} p_n = 1$
  - $p_n = (\frac{\lambda}{\mu})^n p_0, 0 \leq n \leq K$ (K was ∞ in $M/M/1$).

Expected Customer Numbers

- Solution to the probabilities:

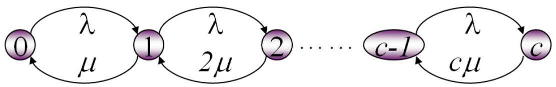
$$p_n = \rho^n (1-\rho)/(1-\rho^{K+1}).$$

- Expected customer number in system:
  - $L = E\{N\} = \sum_{n=0}^{K} n p_n = \frac{\rho}{1-\rho} - \frac{(K+1)\rho^{K+1}}{1-\rho^{K+1}}$
  - Smaller than that of $M/M/1$ (why?).

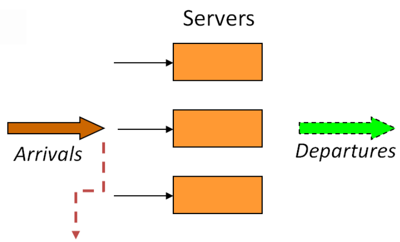
$M/M/1/K$ Rejections

- Probability that an arriving customer is rejected is (simply) p_K.
- Rejection rate is therefore $p_K \lambda$.
- **Actual** arrival rate into the system is
  - $\lambda' = (1-p_K)\lambda$.
- Server utilization is $\lambda'/\mu = (1-p_K)\lambda/\mu$.
  - Server less occupied because of rejections.

$M/M/c/c$



- There are c servers.
- If an arrival finds all servers busy, it is dropped.
- Models
  - Telephone exchange
  - Virtual circuits



$M/M/c/c$ Results

- $p_n = p_0 \left(\frac{\lambda}{\mu}\right)^n \frac{1}{n!}$.
- Probability of a lost arrival (or, loss ratio):

$$p_c = \frac{(\lambda/\mu)^c/c!}{\sum_{n=0}^{c}(\lambda/\mu)^n/n!}.$$

  - *aka* the **Erlang B formula**.
- Holds also for an $M/G/c/c$ (i.e., with arbitrary service time).

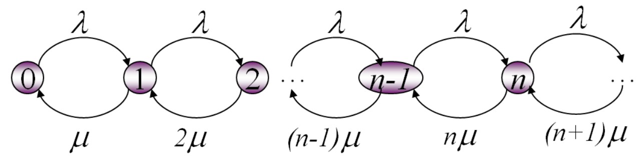
The Reverse Erlang Problem

- Given $\rho = \lambda/\mu$, and desired loss probability p, what is c?
- Recursive implementation of Erlang B (Copper, 1982):

$$B(\rho, 0) = 1$$
$$B(\rho, j) = \frac{\rho B(\rho, j-1)}{\rho B(\rho, j-1) + j},$$

  with $j = 1, 2, ..., c$, and $p = B(\rho, c)$.
- Note $B(\rho, j)$ is monotonously decreasing versus c (Zeng, 2003).
- This recursive algorithm allows us to get to the right c value that gives the p.

Another Variation: $M/M/\infty$



- ∞: infinite number of servers!
- Transition diagram
- Number of customers in system (Poisson distribution):
  $p_n = \left(\frac{\lambda}{\mu}\right)^n \frac{e^{-\lambda/\mu}}{n!}$.

$M/M/\infty$ Results

- Trivial -
  - Zero queueing length: $L_q = 0$
  - Zero queueing time: $W_q = 0$
- Expected number of customers in system

$$L = \lambda/\mu$$

- Expected time in system: $W = 1/\mu$.

An Example (bonus Q.)

- A scientific satellite communicates with an earth station through an antenna.
- Antenna connected to a multiplexor with attached queue that feeds information to 2000 attached disk drives.
- Each disk drive writes at an average rate of 106 bits per second. Message are 104 bits in length on average, and arrive at an average rate of 105 messages per second.
- Each message is written as a unit to a single, arbitrary disk drive, or it goes in the queue if no disk drive is available.
- Q: How long does it take for a message to be processed?
- $\Rightarrow M/M/\infty$ or $M/M/2000$

References

- Harchol-Balter, Chapter 14
- Next:
  - More variations: M/M/c, M/G/1, priority queueing, ...
  - Networked Queueing
  - Lab this week: Queueing Tutorial and Simulations