CDA6530: Performance Models of Computers and Networks

Chapter 5: Basic Queuing Networks
Open Queuing Network

- Jobs arrive from external sources, circulate, and eventually depart
Closed Queuing Network

- Fixed population of $K$ jobs circulate continuously and never leave
  - Previous machine-repairman problem
Feed-Forward QNs

- Consider two queue tandem system

![State Transition Diagram]

- Q: how to model?
  - System is a continuous-time Markov chain (CTMC)
  - State \((N_1(t), N_2(t))\), assume to be stable
  - \(\pi(i,j) = P(N_1=i, N_2=j)\)
  - Draw the state transition diagram
    - But what is the arrival process to the second queue?
Poisson in ⇒ Poisson out

- **Burke’s Theorem:** Departure process of $M/M/1$ queue is Poisson with rate $\lambda$ independent of arrival process.

- **Poisson process addition, thinning**
  - Two *independent* Poisson arrival processes adding together is still a Poisson ($\lambda = \lambda_1 + \lambda_2$) *Why?*
  - For a Poisson arrival process, if each customer leaves with prob. $p$, the remaining arrival process is still a Poisson ($\lambda = \lambda_1 \cdot p$)
State transition diagram: \((N_1, N_2), N_i=0,1,2,\ldots\)

\[
\pi(i, j) = (1 - \rho_1)\rho_1^i (1 - \rho_2)\rho_2^j \quad i, j \geq 0
\]

\[\rho_i = \frac{\lambda}{\mu_i}\]
- For a $k$ queue tandem system with Poisson arrival and exponential service time

- Jackson’s theorem:

$$P(N_1 = n_1, N_2 = n_2, \ldots, N_k = n_k) = \prod_{i=1}^{k} (1 - \rho_i) \rho_i^{n_i},$$

- Above formula is true when there are feedbacks among different queues
  - Each queue behaves as M/M/1 queue in isolation
Example

- $\lambda_i$: arrival rate at queue $i$

\[
\lambda_1 = 4 + \frac{\lambda_2}{4} \\
\lambda_2 = 5 + \frac{\lambda_1}{2}
\]

$\Rightarrow \lambda_1 = 6, \; \lambda_2 = 8$

\[
\pi(n_1, n_2) = \frac{1}{4} \left(\frac{3}{4}\right)^{n_1} \frac{1}{5} \left(\frac{4}{5}\right)^{n_2}
\]

In M/M/1:

\[
E[N] = \frac{\rho}{1 - \rho} = \frac{\lambda}{\mu - \lambda}
\]

\[
E[T] = E[N]/(r_1 + r_2) = 7/9 \text{ time units}
\]

Why?

UCF Stands For Opportunity


- **$T^{(i)}$:** response time for a job enters queue $i$

$$E[T^{(1)}] = \frac{1}{\mu_1 - \lambda_1} + \frac{E[T^{(2)}]}{2}$$

$$E[T^{(2)}] = \frac{1}{\mu_2 - \lambda_2} + \frac{E[T^{(1)}]}{4}$$

Why?

In M/M/1:

$$E[T] = \frac{1}{\mu - \lambda}$$
Extension

- results hold when nodes are multiple server nodes \((M/M/c)\), infinite server nodes finite buffer nodes \((M/M/c/K)\) (careful about interpretation of results), PS (process sharing) single server with arbitrary service time distr.
Closed QNs

- Fixed population of $N$ jobs circulating among $M$ queues.
  - single server at each queue, exponential service times, mean $1/\mu_i$ for queue $i$
  - routing probabilities $p_{i,j}$, $1 \leq i, j \leq M$
  - visit ratios, \{v_i\}. If $v_1 = 1$, then $v_i$ is mean number of visits to queue $i$ between visits to queue 1

$$v_i = \sum_{j=1}^{M} v_j p_{j,i} \quad i = 2, \ldots M$$

- $\gamma_i$: throughput of queue $i$,

$$\frac{\gamma_i}{\gamma_j} = \frac{v_i}{v_j}, \quad 1 \leq i, j \leq M$$
- Open QN has infinite no. of states
- Closed QN is simpler

- How to define states?
  - No. of jobs in each queue
Steady State Solution

- **Theorem (Gordon and Newell)**

  \[ \pi(\vec{n}) = \frac{1}{G(N)} \prod_{i=1}^{M} \left( \frac{v_i}{\mu_i} \right)^{n_i} \quad \vec{n} \geq \vec{0}; \sum_{i=1}^{M} n_i = N \]

  where \( \vec{n} = (n_1, \ldots, n_M) \), and \( G(N) \) is a constant chosen so that \( \sum \pi(\vec{n}) = 1 \).

- For previous example, \( v_i \)?

  \[ v_1 = 1, \quad v_2 = \frac{3}{4}, \quad v_3 = \frac{1}{4} \]
Mean Value Analysis (MVA) Algorithm

- **Key idea:** a job that moves from one queue to another, at time of arrival to queue sees a system with the same statistics as system with *one less customer*.
  - We only consider single server nodes
MVA Algorithm

- System with population of n jobs
  - $\bar{N}_i(n)$ - average number of jobs at node $i$
  - $\bar{T}_i(n)$ - average response time at node $i$
  - $\gamma_i(n)$ - throughput of node $i$

0. $\bar{N}_i(0) = 0, \ 1 \leq i \leq M$ \hspace{1cm} \textit{initialization}

for $n = 1$ to $N$ do

1. $\bar{T}_i(n) = [1 + \bar{N}_i(n - 1)]/\mu_i, \hspace{2cm} \text{Why?}$

2. $\gamma(n) = n/(\sum_{i=1}^{M} v_i \bar{T}_i(n))$

3. $\gamma_i(n) = v_i \gamma(n), \hspace{1cm} 1 \leq i \leq M$

   $\bar{N}_i(n) = \gamma_i(n) \bar{T}_i(n), \hspace{1cm} 1 \leq i \leq M$ \hspace{1cm} \text{Why?}
Example: File Server

- Each workstation requests file server’s CPU and I/O service
  - Workstation = job
  - What is $v_i$?