Analysis of Closed Networks

using

The Convolution and Mean Value Algorithms

Analysis of a Closed Queueing Network with
Exponentially Distributed Service Times

Convolution Algorithm

Recursively calculate the normalization constant *G(M)* for a network with *M* jobs

Mean Value Analysis Algorithm (MVA)

Directly calculate the performance measures for each queue using a recursive algorithm

Consider network of K queues with routing probabilities $\{p_{ij}\}$ $i,j=1, \ldots, K$ and M jobs circulating in the network

Convolution Algorithm

• Finding the normalization constant *G(M)* is important as it is required for obtaining the joint state probability distribution of the queueing network.

Other performance measures may then be found from the state distribution.

• For non-trivial values of *K* and/or *M*, direct computation of *G(M)* would not be feasible.

• The *Convolution Algorithm* provides an easy, numerical approach to finding *G(M)* recursively.

The algorithm finds G(1), G(2) ……G(M-1), G(M) in sequence

Knowing the intermediate values of G(.) help, as one can directly use them to compute performance measures, especially for networks of single server queues

 3

Some Useful Results for Closed Network of Single Sever Queues

For the *K* queues, *j=1,…….,K*

$$P\{n_j \geq n\} = u_j^n \frac{G(M-n)}{G(M)} \qquad (5.26)$$

Actual Utilization

$$r_j = u_j \frac{G(M-1)}{G(M)} \qquad (5.27)$$

Actual Throughput

$$l_j = m_j r_j = m_j u_j \frac{G(M-1)}{G(M)} \qquad (5.28)$$

Mean Number in queue Q_j

$$E\{n_j\} = \sum_{m=1}^{M} u_j^m \frac{G(M-m)}{G(M)} \qquad (5.30)$$

 4

Convolution Algorithm for Network of Single Server Queues

g(n,k)= Normalization Constant when there are *n* jobs circulating in a network with *k* queues ($Q_1 \ldots\ldots Q_k$) where the individual queues have the same relative utilization as in the original network

$$g(n,k) = \sum_{n_1+\ldots+n_K=n} \left(\prod_{i=1}^{k} u_i^{n_i} \right) \qquad (5.31)$$

Convolution Algorithm for Network of Single Server Queues
(continued)

- **Initialization**

$$g(0,k)=1 \qquad g(n,1)=u_1^n \qquad k=1,\ldots,K \quad n=1,\ldots\ldots,M$$

- **Recursion**

$$g(n,k) = g(n,k-1) + u_k\, g(n-1,k) \qquad (5.35)$$

- **Termination**
  when g(M,K) has been calculated *G(M)=g(M,K)*

For *L=1,……,(M -1)*, the intermediate steps of the recursion give *G(L)=g(L,K)*

Convolution Algorithm for Network of Multi-Server Queues or Queues with State-Dependent Service Rates

State Dependent Service Rate at Q_i :

$m_i(j)$ = Service Rate at Q_i when it has j jobs

Multi-server Queue at Q_i (s_i servers): $m_i(j) = \min\{jm, s_i m\}$

Define

$$u_i = \frac{l_i^*}{m_i(1)} \qquad i=1,\ldots\ldots,K \qquad (5.37)$$

$$A_i(n) = \prod_{j=1}^{n} \frac{m_i(j)}{m_i(1)} \qquad i=1,\ldots\ldots,K \qquad (5.38)$$

Convolution Algorithm for Network of Multi-Server Queues or Queues with State-Dependent Service Rates
(*continued*)

- **Initialization**

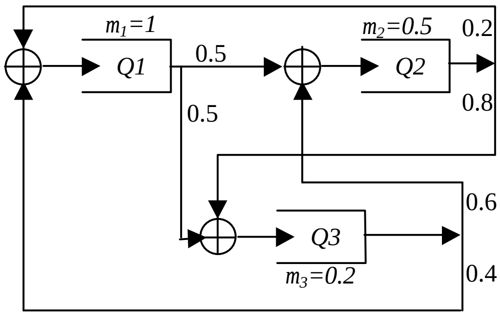
$$g(0,k) = 1 \qquad g(n,1) = \frac{u_1^{\,n}}{A_1(n)} \qquad k=1,\ \ldots.,K \quad n=1,\ldots\ldots,M$$

- **Recursion**

$$g(n,k) = \sum_{j=0}^{n} \frac{(u_k)^j}{A_k(j)}\, g(n-j, k-1) \qquad (5.39)$$

- **Termination**
  when g(M,K) has been calculated $\qquad G(M)=g(M,K)$

Example



- *Closed Network of Single Server, Infinite Buffer Queues*
- *Exponential Service Time Distribution at each queue*
- *Number of Circulating Jobs = M = 4*

Flow-Balance Equations

$$l_1^* = 0.2\, l_2^* + 0.4\, l_3^*$$
$$l_2^* = 0.5\, l_1^* + 0.6\, l_3^*$$

Choosing Q_1 as the reference queue, solve flow balance for -

Relative Throughputs	$l_1^*=1,\ l_2^*=1.5385,\ l_3^*=1.7308$
Relative Utilizations	$u_1=1,\ u_2=3.077,\ u_3=8.654$
Visit Ratios	$V_1=1,\ V_2=1.5385,\ V_3=1.7308$

Using the Convolution Algorithm, we get -

Table 5.1. Values of g(n,k) for n=0,1,2,3,4 and k=1,2,3

k / n	1	2	3
0	1	1	1
1	1	4.077	12.731
2	1	13.545	123.72
3	1	42.678	1113.35
4	1	132.32	9767.26

Normalization Constant *G(4) = g(4,3) = 9767.26*

State Probability Distribution

$$P(n_1, n_2, n_3) = \frac{1}{9767.26}(3.077)^{n_2}(8.654)^{n_3}$$

for n_1, n_2, n_3 ³*0* , and $n_1+n_2+n_3=4$

Actual Throughputs

$$l_1 = 0.114, \quad l_2 = 0.1754, \quad l_3 = 0.1973$$

Mean Number in Queue

$$N_1 = 0.1281, \quad N_2 = 0.5178, \quad N_3 = 3.3541$$

MVA: Mean Value Analysis Algorithm

• Directly computes performance measures of the network without computing the normalization constant or the state probability distribution

• Based on *Mean Value Theorem* (Reiser and Lavenberg)

> *“A customer arriving to a queue in a Product-Form Network sees the same average number in the queue as an outside observer will see if the network had one less customer”*

MVA: Mean Value Analysis Algorithm

• For a network of *K* queues with *M* jobs, MVA works recursively - starting with zero jobs in the system, incrementally adding jobs until *M* jobs have been added to the network

• May be applied to networks of queues following any of the following service disciplines

> *FCFS, LCFS, PS (Processor Sharing) or IS (Infinite Number of Servers)*

• Service times must be exponentially distributed

Notation

$N_j(n)$	Mean number of jobs in Q_j when there are a total of n in the network (this includes the job currently being served at Q_j)
$W_j(n)$	Mean time spent by a job in the queue Q_j when there are n in the network (this is the total delay at Q_j including the service time of the job)
$1/m_j$	Mean service time for a job at Q_j. Note that this will be the same regardless of the number of jobs in Q_j
V_j	Visit Ratio of Q_j. This is as defined earlier.

With Q_1 as the reference queue

$$V_i = \frac{l_i^*}{l_1^*} \qquad i = 1, \ldots, K \qquad (5.18)$$

MVA Algorithm for Single Server Queues with Single Traffic Class

- **Initialization** Set $N_k(0)=0$ for each queue, i.e. $k=1,\ldots, K$
- **Recursion** Do steps (1), (2) and (3) successively for $m=1,2,\ldots,M$

Step 1: Calculate for $k=1,2,\ldots, K$

$$\begin{aligned} W_k(m) &= \frac{1}{m_k} && \textit{for IS queues} \\ &= \frac{N_k(m-1)+1}{m_k} && \textit{for FCFS, LCFS, PS queues} \end{aligned} \qquad (5.42)$$

Step 2: Applying Little's Result

Overall Throughput $l = \dfrac{m}{\sum_{k=1}^{K} W_k(m) V_k}$ (5.43)

Step 2: Update $N_k(m)$ for $k=1,\ldots\ldots,K$ as

$$N_k(m) = V_k \, l \, W_k(m) \qquad (5.44)$$

• **Termination** Terminate recursion when $m=M$ is reached

• Total average time spent by a job in the k^{th} queue (i.e. Q_k) is given directly by W_k for $k=1,\ldots\ldots,K$ using (5.42) for $m=M$

(Note that this is the time spent per visit to Q_k)

• Queueing Delay for a job at Q_k for $k=1,\ldots\ldots,K$ given by

$$\begin{aligned} W_{qk} &= 0 && \textit{for IS queues} \\ &= W_k - \frac{1}{m_k} && \textit{for FCFS, LCFS, PS queues} \end{aligned} \qquad (5.45)$$

• Number in Q_k given by $N_k=N_k(M)$ for $k=1,\ldots..,K$

• Throughput of the Network = l using (5.43) for $m=M$

• Actual Throughput of Q_k given by $l_k = l \; V_k$ for $k=1,\ldots..,K$

• Number waiting in queue in Q_k given by $N_{qk}= l_k W_{qk}$ for $k=1,\ldots..,K$

MVA Algorithm for Multi-Server Queues with Single Traffic Class

(c_k servers at Q_k, where each server works with service rate m_k, exponential service, FCFS or LCFS)

- **Initialization** Set $N_k(0)=0$, $p_k(0,0)=1$, and $p_k(j,0)=0$ for $j=1,\ldots, (c_k-1)$ and $k=1,\ldots, K$
- **Recursion** Do steps (1), (2), (3) and (4) successively for $m=1,2,\ldots,M$

Step 1: Calculate for k=1,2,……, K

$$W_k(m) = \frac{N_k(m-1)+1+S_k}{m_k} \quad \textit{for FCFS, LCFS queues} \qquad (5.47)$$

with $$S_k = \sum_{j=1}^{c_k-1}(c_k - j)\,p_k(j-1, m-1) \qquad (5.48)$$

($p_k(x,y)$ updated in Step 4)

Step 2: Applying Little's Result

Overall Throughput $$l = \frac{m}{\sum_{k=1}^{K} W_k(m) V_k} \qquad (5.49)$$

Step 3: Update $N_k(m)$ for k=1,……,K as

$$N_k(m) = V_k\, l\, W_k(m) \qquad (5.50)$$

Step 4: Update $p_k(j,m)$ for k=1,……,K as

$$\begin{aligned} p_k(j,m) &= 1 - \sum_{i=1}^{K} p_k(i,m) && \textit{for } j=0 \\ &= \frac{l\,p_k(j-1, m-1)}{m_k} && \textit{for } j=1,\ldots\ldots, M \end{aligned} \qquad (5.51)$$

- **Termination** Terminate recursion when $m=M$ is reached

• When recursion terminates, N_k and W_k will be directly available as $N_k=N_k(M)$ and $W_k=W_k(M)$ for $k=1,\ldots, K$

• Actual network throughput l will also be available for $m=M$

• Actual throughputs of the individual queues may be found using $l_k= l\ V_k$ for $k=1,\ldots,K$

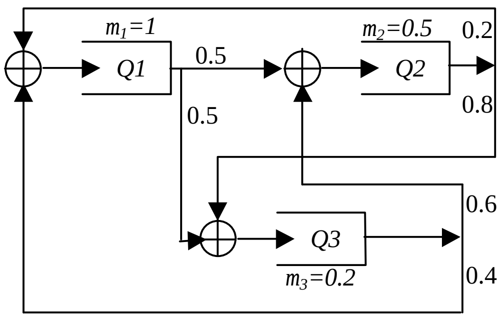
• Average Queueing Delay at Q_k may be found as $W_{qk} = W_k - \dfrac{1}{m_k}$ for $k=1,\ldots, K$

• Number waiting in queue in Q_k given by $N_{qk}= l_k W_{qk}$ for $k=1,\ldots,K$

MVA Algorithm may also be stated for Open Networks of queues with state-dependent exponential service where there are multiple job classes. See Section 5.7.3

 21

Example



• *Closed Network of Single Server, Infinite Buffer Queues*

• *Exponential Service Time Distribution at each queue*

• *Number of Circulating Jobs = M = 4*

 22

Flow-Balance Equations

$$l_1^* = 0.2\, l_2^* + 0.4\, l_3^*$$
$$l_2^* = 0.5\, l_1^* + 0.6\, l_3^*$$

Choosing Q_1 as the reference queue, solve flow balance for -

Relative Throughputs $l_1^*=1,\ l_2^*=1.5385,\ l_3^*=1.7308$

Relative Utilizations $u_1=1,\ u_2=3.077,\ u_3=8.654$

Visit Ratios $V_1=1,\ V_2=1.5385,\ V_3=1.7308$

•Initialize $N_1(0)= N_2(0)= N_3(0)=0$

• Do the MVA recursion for $m = 1, 2, 3$ and 4. (The following values were obtained at each step.)

m=1

$$W_1(1)=1, W_2(1)=2, W_3(1)=5$$
$$l = 0.07855$$
$$N_1(1)=0.07855,\ N_2(1)=0.2417,\ N_3(1)=0.6798$$

m=2

$$W_1(2)=1.07855, W_2(2)=2.4834, W_3(2)=8.399$$
$$l = 0.1029$$
$$N_1(2)=0.11098,\ N_2(2)=0.3932,\ N_3(2)=1.49586$$

m=3

$$W_1(3) = 1.11098, W_2(3) = 2.7864, W_3(3) = 12.4793$$
$$l = 0.1111$$
$$N_1(3) = 0.12346, N_2(3) = 0.4764, N_3(3) = 2.4002$$

m=M=4

$$W_1(4) = 1.12346, W_2(4) = 2.9528, W_3(4) = 17.001$$
$$l = 0.114$$
$$N_1(4) = 0.12806, N_2(4) = 0.51783, N_3(4) = 3.35411$$

Throughput of the Network = 0.114

Actual Throughputs of Q_1,Q_2, Q_3 and Q_4 = *(0.114, 0.175, 0.197)*

Actual Utilizations of Q_1,Q_2, Q_3 and Q_4 = *(0.114, 0.350, 0.985)*