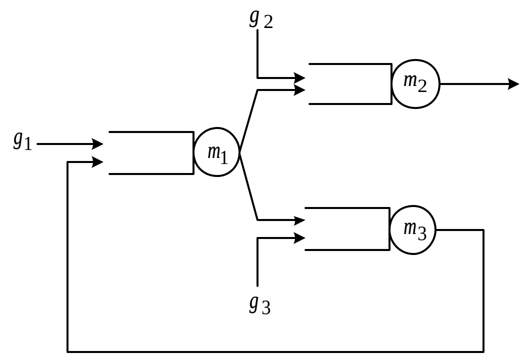
Queueing Networks

Systems modeled by queueing networks can roughly be grouped into four categories

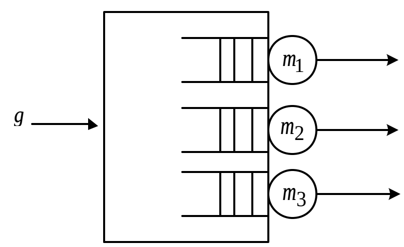
1. **Open networks** – Customers arrive from outside the system are served and then depart.

   Example: Packet switched data network.



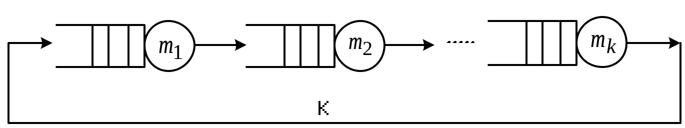
2. **Networks with population constraints** – Customers arrive from outside the system if there is room in the queues. They enter, served and then depart.

   Example: queues sharing a common buffer pool.



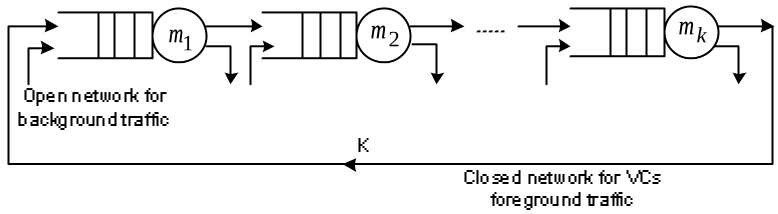
3. **Closed networks** – Fixed number of customers (K) are trapped in the system and circulate among the queues.

   Example: CPU job scheduling problem



4. **Mixed network** – Any combination of the types above.

   Example: simple model of virtual circuit that is window flow controlled.



Several features can occur in queueing networks that do not occur in single queue.

1. **Jocking** – Customers moving among parallel queues.
2. **Blocking** – Customer waiting depart a server and join next queue is unable to due to limited waiting space, and therefore stays in server (blocking it.)
3. **Forking** – Customer leaving a queue clones into multiple customers going along different routes.
4. **Joining** – Multiple streams of customers are combined into a single stream.

Forking and joining are used in models of parallel processing systems.

i. Open networks

The simplest type of network is an open network with the following assumptions (called Jackson Network)

- Assume arbitrary network of M queues
- Service time of queue i is exponentially distributed with rate μ_i.
- Arrivals from outside the network to queue i are a Poisson process with mean rate γ_i.

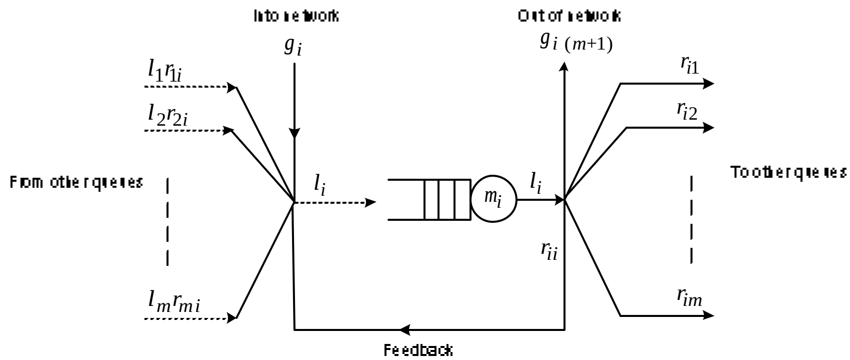
Let

$r_{i,j}$ – routing probability that a customer completing service at queue i goes to queue j.

$r_{i,m+1}$ – routing probability that a customer completing service at queue i leaves the network.

Note $\sum_{j=1}^{m+1} r_{i,j} = 1$

Let λ_i be the total mean customer arrival rate to queue i.



and $\rho_i = \dfrac{\lambda_i}{\mu_i}$

We can see that at each queue i

$$\lambda_i = \gamma_i + \sum_{j=1}^{m+1} r_{ji}\, \lambda_j$$

Note that the flow conservation equation holds in any open network regardless of arrival and service distributions. This equation can easily be solved for λ_i.

Let $\lambda = [\lambda_i, \lambda_i, \ldots, \lambda_m]$ $\qquad \gamma = [\gamma_i, \gamma_i, \ldots, \gamma_m]$

$R = [r_{ij}] \quad 1 \le i \le m \quad 1 \le j \le m$ $\qquad \Leftarrow$ R does not include $r_{i\,m+1}$

The flow conservation equation can be written in matrix vector form as

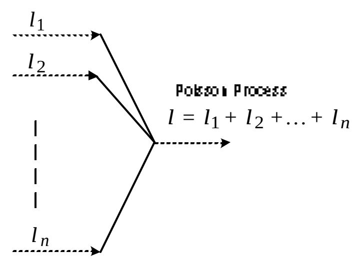
$$\lambda = \gamma + \lambda R$$

sloving $\lambda(I - R) = \gamma$

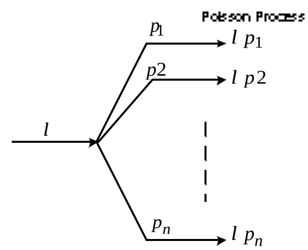
$$\lambda = \gamma (I - R)^{-1}$$

Now consider queue i in the Jackson network, from the analysis of the single M/M/1 queue we know

1. Merging of independent Poisson processes is Poisson with rate equal to the sum of the individual rates. That is Poisson process.



2. The departure process of an M/M/1 queue is Poisson with rate equal to input rate of queue.
3. Probabilistic splitting of a Poisson process results in a Poisson process.



Combining these results, we can see that the input and output processes of each queue i in the network is Poisson process.

Let $\tilde{n}_i(t)$ be the number of customers in the system at queue i at the time t.

The state of the network is defined by the vector $\left(\tilde{n}_1(t), \tilde{n}_2(t), \ldots, \tilde{n}_m(t)\right)$, under the assumptions above the process $\left\{\left(\tilde{n}_1(t), \tilde{n}_2(t), \ldots, \tilde{n}_m(t)\right), t \geq 0\right\}$ is a m dimensional Markov process. Steady state can be determined by letting $P(n)$ denote steady state probability.

Let $\quad P(n) \quad = \quad \lim_{t \to \infty} P\left\{\tilde{n}_1(t) = n_1, \tilde{n}_2(t) = n_2, \ldots, \tilde{n}_m(t) = n_m\right\}$

and $\quad P(n - 1_i) \quad = \quad \lim_{t \to \infty} P\left\{\tilde{n}_1(t) = n_1, \tilde{n}_2(t) = n_2, \ldots, \tilde{n}_i(t) = n_i - 1, \tilde{n}_m(t) = n_m\right\}$

$\Rightarrow$ decrease by 1 in the ith queue

Similarly,

$$P(n + 1_i) \quad = \quad \lim_{t \to \infty} P\left\{\tilde{n}_1(t) = n_1, \tilde{n}_2(t) = n_2, \ldots, \tilde{n}_i(t) = n_i + 1, \tilde{n}_m(t) = n_m\right\}$$

$\Rightarrow$ increase by 1 in the ith queue

Writing the steady state flow balance equation rate in to state n = rate out of state n

$$\sum_{i=1}^{m} \gamma_i \cdot P(n-1_i) + \sum_{i=1}^{m} \mu_i \cdot r_{i\,m+1} \cdot P(n+1_i) + \sum_{i=1}^{m}\sum_{j=1}^{m} r_{ji} \cdot \mu_j \cdot P(n+1_j-1_i) = \left[\sum_{i=1}^{m} \lambda_i + \sum_{i=1}^{m} \mu_i\right] \cdot P(n)$$

First term on the left hand side is transition from *n-1* to *n* caused by external arrival.

Second term on left hand side is transition from *n+1* to *n* by departure of customer from network.

Third term on left hand side is transition from *n+1* to *n* by departure from the *i*th queue with *n+1* to the *i*th queue with *n-1*.

Right hand side all ways can leave state *n* with arrival or departure.

The flow balance equation above can easily be verified by constructing the state transition diagram for the two queues case (i.e. M=2).

The solution to the steady state flow balance equation is the well known **Product Form**.

$$P(n) = C\rho_1^{n_1}\rho_2^{n_2}\cdots\rho_m^{n_m} = C\prod_{i=1}^{m}\rho_i^{n_i}$$

where $\rho_i = \dfrac{\lambda_i}{\mu_i}$ and C is a constant.

Prove by substituting $P(n) = C\prod_{i=1}^{m}\rho_i^{n_i}$ into the steady state flow balance equation and show that it is a solution.

C is determined by $\displaystyle\sum_{n_m=0}^{\infty}\sum_{n_{m-1}=0}^{\infty}\cdots\sum_{n_2=0}^{\infty}\sum_{n_1=0}^{\infty} C\rho_1^{n_1}\rho_2^{n_2}\cdots\rho_m^{n_m} = 1$

results in $C = \prod_{i=1}^{m}(1-\rho_i)$

Hence,

$$P(n) = \prod_{i=1}^{m}(1-\rho_i)\rho_i^{n_i} \qquad \text{for stability } \rho_i < 1 \quad ; \forall i$$

$\Rightarrow$ essentially the product form of M independent M/M/1 queues steady state probabilities, $p_n = (1-\rho)\rho^n$ (M/M/1 steady state)

$$P(n) = \prod_{i=1}^{m} \pi_{n_i}$$

From the state probabilities, one can easily determine performance measures.

Know that each queue i is a M/M/1 queue with ρ_i

$\Rightarrow$ $L_i = \dfrac{\rho_i}{1-\rho_i}$, $\quad W_i = \dfrac{1}{\mu_i - \lambda_i}$, $\quad \pi_{n_i} = (1-\rho_i)\rho_i^{n_i}$, etc…

(all M/M/1 measures apply) $W_{q_i} = \dfrac{\rho_i}{\mu_i - \lambda_i}$

For the network as a whole let

LN – Average number of customers in network.

$$LN = \sum_{i=1}^{m} L_i = \sum_{i=1}^{m} \frac{\rho_i}{1-\rho_i}$$

γN – total average load on network.

$$\gamma N = \sum_{i=1}^{m} \gamma_i$$

WN – Average delay through network.

$$WN = \frac{LN}{\gamma N} = \frac{1}{\gamma N}\sum_{i=1}^{m} \frac{\rho_i}{1-\rho_i} = \sum_{i=1}^{m} \frac{\lambda_i}{\gamma N} W_i$$

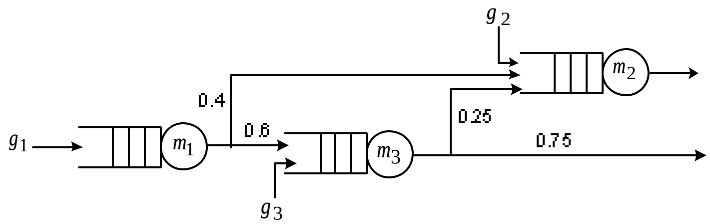
Note that in applying this solution to packet switched networks

$\rho_i = \dfrac{\lambda_i}{\mu C_i}$ where μ–average packet length, C_i–capacity of link i

can also add a deterministic delay d_{ij} corresponding to the time it takes a customer to move from the ith queue to the jth queue (propagation delay) and still get Jackson network as above, only WN changes.

$$WN = \sum_{i=1}^{m} \frac{\lambda_i}{\gamma N}\left[W_i + \sum_{j=1}^{m} r_{ij} \cdot d_{ij} \right]$$

Example: Three node network shown below, assuming Poisson external arrivals and exponential service at each queue.



Given

$$\gamma_1 = 0.5\ ,\ \gamma_2 = 0.25\ ,\ \gamma_3 = 0.25\ \Rightarrow \gamma N = \sum_{i=1}^{3} \gamma_i = 1$$

$$\mu_1 = 1\ ,\ \mu_2 = 1\ ,\ \mu_3 = 1$$

From the diagram $r_{12} = 0.4$, $r_{13} = 0.6$, $r_{32} = 0.25$, $r_{24} = 1.0$, $r_{34} = 0.75$

Solving the flow conservation equation for λ_i

$$\gamma = [0.5,\quad 0.25,\quad 0.25] \qquad R = \begin{bmatrix} 0 & 0.4 & 0.6 \\ 0 & 0 & 0 \\ 0 & 0.25 & 0 \end{bmatrix}$$

$$\lambda = \gamma (I - R)^{-1}$$

using Matlab to solve results in

$$\lambda = [0.5,\quad 0.5875,\quad 0.55]$$

$$\Rightarrow \rho_i = \frac{\lambda_i}{\mu_i}\ ;\quad \rho_1 = 0.5\ ,\ \rho_2 = 0.5875\ ,\ \rho_3 = 0.55$$

$$\rho_i < 1 \quad ; \forall i \Rightarrow \text{stable system}$$

$$WN = \frac{1}{\gamma N} \sum_{i=1}^{3} \frac{\rho_i}{1 - \rho_i} = 3.646$$

Many slight generalizations to open Jackson networks exist. The most widely known are BCMP networks. BCMP networks also have a product form :

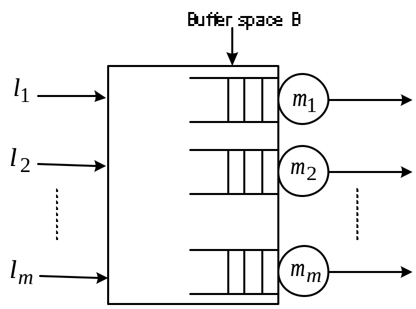
$$P(n) = C \prod_{i=1}^{m} \rho_i^{n_i}$$

However, form of C depends on the system model. Some of the additional features that can be modeled include: multiple classes of jobs, state dependent exponential servers, coxian service distribution with ∞ number of servers, etc. A good discussion can be found in Jain's textbook.

ii. Networks with population constraints

Consider M queue system – results only for simple cases. Customers arrive from outside the network according to a Poisson process with rate λ_i to queue i and exponential service distribution with rate μ_i at queue i.

Simple example: M output queues at an output buffer of a packet switch.



As in open network case study $\left(\tilde{n}_1(t), \tilde{n}_2(t), \ldots, \tilde{n}_m(t)\right)$.

This process is a finite state space M dimensional Markov process with state space

$$S = \left\{ (n_1, n_2, \ldots, n_m): 0 \le n_i \le B \;\; \forall i; \; \sum_{i=1}^{m} n_i \le B \right\}$$

The steady state probability

$$P(n) = \lim_{t \to \infty} P\left\{ \tilde{n}_1(t) = n_1, \tilde{n}_2(t) = n_2, \ldots, \tilde{n}_m(t) = n_m \right\}$$

again has a product form

$$P(n) = \frac{1}{G} \prod_{i=1}^{m} \rho_i^{n_i}$$ where G is the normalization constant found by

$$\sum_{n \in S} P(n) = 1 \quad \Rightarrow \quad G = \sum_{n \in S} \prod_{i=1}^{m} \rho_i^{n_i}$$

In general, G must be determined numerically. This becomes difficult when S is large.

From $P(n)$ one can determine various mean performance measures.

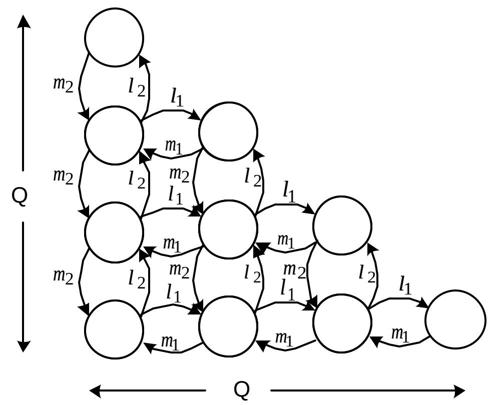
L_i – Average number of customers in queue i.

$$L_i = \sum_{j=0}^{B} j \left(\sum_{n_i = j;\, n \in S} P(n) \right)$$

$$LN = \sum_{i=1}^{m} L_i$$

Example: $B = 3$, $M = 2$

State diagram (n_1, n_2) : S = { (n_1, n_2) ; $0 \le n_1 \le 3$, $0 \le n_2 \le 3$, $n_1 + n_2 < 3$ }



For this example,

$$P(n_1, n_2) = \frac{1}{G} \prod_{i=1}^{m} \rho_i^{n_i} = \frac{1}{G} \rho_1^{n_1} \rho_2^{n_2}$$

We can write out G

$$G = 1 + \rho_1 + \rho_1^2 + \rho_1^3 + \rho_2 + \rho_1 \rho_2 + \rho_1^2 \rho_2 + \rho_2^2 + \rho_1 \rho_2^2 + \rho_2^3$$

Let $\lambda_1 = 0.5$, $\lambda_2 = 1$, $\mu_1 = 1$, $\mu_2 = 1 \Rightarrow \rho_1 = 0.5$, $\rho_2 = 1$

G = 6.125

$$P(n_1, n_2) = \frac{1}{6.125} (0.5)^{n_1} (1)^{n_2}$$

P(2, 1) = 0.0408

$$L_1 = \sum_{n_1=0}^{3} n_1 \left(\sum_{j = n_1;\, n \in S} P(j, n_2) \right) = 1\big(P(1,0) + P(1,1) + P(1,2)\big) + 1\big(P(2,0) + P(2,1)\big) + 3P(3,0)$$

$L_1 = 0.4694$

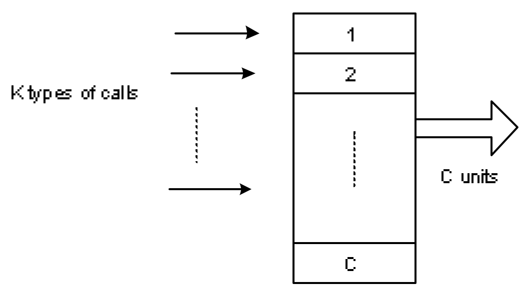
Similarly $L_2 = 1.2653 \Rightarrow LN = 1.7347$

For a good discussion of additional generalizations see S. Lam and J. Wong, “Queueing Network Models of Packet Switching Networks, Part 2: Network with Population Size Constraints,” Performance Evaluation, vol.2, 1982, pp.161-180.
Note that the computation of G is difficult for large S.

The other main application of networks with population constraints is in circuit switched networks where a switch can be modeled as a multirate loss system.

Consider a single link in a modern circuit switched network like ISDN. Various services are offered and each service has different characteristics (call arrival rate, holding time, bandwidth.) Assume K types of connections, each type i arrives according to a Poisson process rate λ_i , each type i connection holding time exponentially distributed with rate μ_i (results hold for general holding time.)

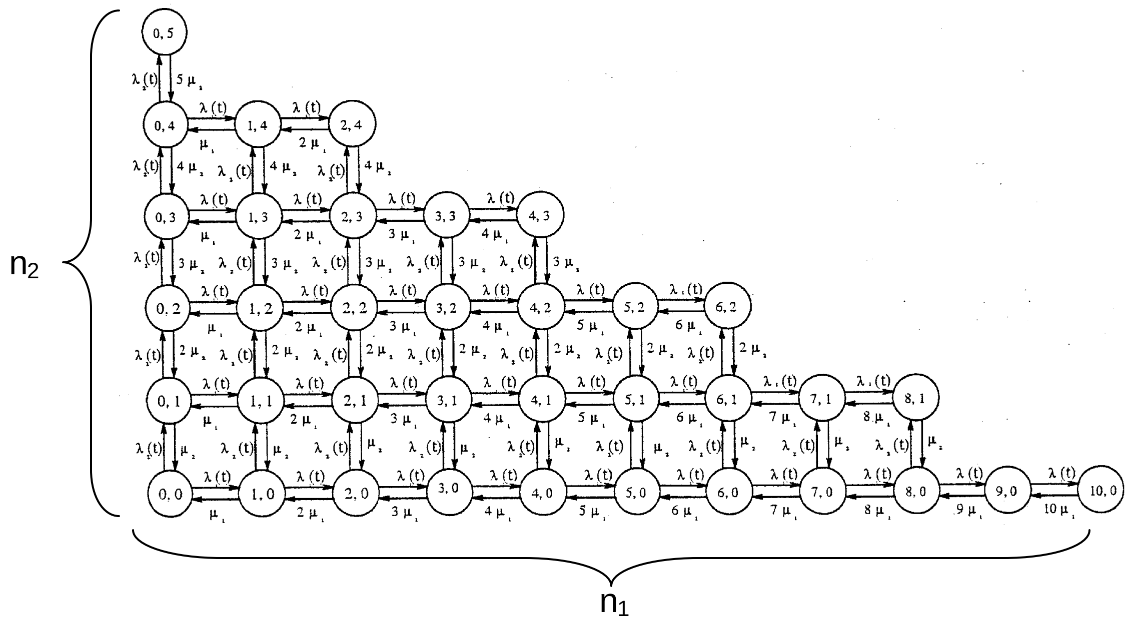
Each type i connection requires m_i basic units of bandwidth. The total bandwidth available is C units.



Let $\tilde{n}_i(t)$ = number of type i connection in system at time t.

The K tuple $\left(\tilde{n}_1(t), \tilde{n}_2(t), \ldots, \tilde{n}_m(t)\right)$ defines a K dimensional Markov process with finite state space S where $0 \le n_i \le \lfloor C/m_i \rfloor$ and $\sum_{i=1}^{K} n_i m_i \le C$

The example of the state transition diagram for the case of C = 10, K = 2, m_1 = 1, m_2 = 2 is shown below



The steady state probabilities

$$P(n_1, n_2, \ldots, n_K) \quad = \quad \lim_{t \to \infty} P\left\{ \tilde{n}_1(t) = n_1, \tilde{n}_2(t) = n_2, \ldots, \tilde{n}_K(t) = n_K \right\}$$

have the product form as before when $\rho_i = \dfrac{\lambda_i m_i}{\mu_i}$

$$P(n_1, n_2, \ldots, n_K) = \frac{1}{G(k)} \prod_{i=1}^{K} \frac{\rho_i^{n_i}}{n_i!}$$ $\Leftarrow$ when K=1, get Erlang B model M/G/C/C

and

$$G(k) = \sum_{n \in S} \prod_{i=1}^{K} \frac{\rho_i^{n_i}}{n_i!}$$

Mainly interested in connection blocking rates PB_i

$$PB_i \quad = \quad \sum_{n \in S} P(n_1, n_2, \ldots, n_K)$$

where type i blocked $\Leftarrow$ sum over states where $C - m_i < \sum_{j=1}^{K} n_j m_j$

For state transition example

$$PB_2 = P(0,5) + P(1,4) + P(2,4) + P(3,3) + P(4,3) + P(5,2) + P(6,2) + P(7,1) + P(8,1) + P(9,0) + P(10,0)$$

$PB_1 = P(0,5) + P(2,4) + P(4,3) + P(6,2) + P(8,1) + P(10,0)$

Numerical example:

$C = 48$, K=2, k=1 voice 64 Kbps $\Rightarrow m_1 = 1$

k=2 H_2 ISDN video 384 Kbps $\Rightarrow m_2 = 6$

$\lambda_1 = 15$, $\lambda_2 = 0.125$, $\mu_1 = 1$, $\mu_2 = 0.5$

offered load $\sum_{i=1}^{K} \frac{\lambda_i m_i}{\mu_i} = 30$

$PB_1 = 0.0248$, $PB_2 = 0.086$

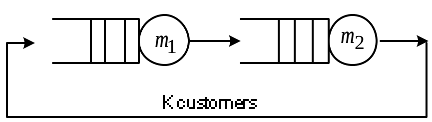
iii. Closed Queueing Networks

Simplest case K customers circulating among M queues. Each queue i has exponentially distributed service time μ_i. The routing probability for a customer completing service at queue i to go to queue j is r_{ij} and $\sum_{j=1}^{m} r_{ij} = 1$

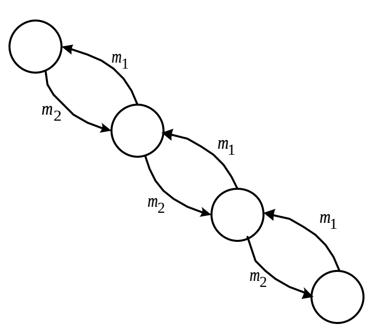
As before state of network defined by $\left(\tilde{n}_1(t), \tilde{n}_2(t), \ldots, \tilde{n}_m(t)\right)$ which is M dimensional Markov process. The state space S is determined by

$$S = \left\{ (n_1, n_2, \ldots, n_m) : 0 \le n_i \le K \ \forall i; \ \sum_{i=1}^{m} n_i \le K \right\}$$

For example, M = 2, K = 3 in network below



(n_1, n_2) state diagram



Again consider steady state probabilities

$$P(n) \quad = \quad \lim_{t \to \infty} P\left\{ \tilde{n}_1(t) = n_1, \tilde{n}_2(t) = n_2, \ldots, \tilde{n}_m(t) = n_m \right\}$$

Flow balance equation in steady state

rate in = rate out

$$\sum_{i=1}^{m} \sum_{j=1}^{m} r_{ji} \cdot \mu_j \cdot P(n + 1_j - 1_i) = \left(\sum_{i=1}^{m} \mu_i \right) \cdot P(n)$$

The left hand side is the same open network except removing terms for external arrivals and departures.

Notice that λ_i does not appear in the equation or in the state diagram of the example above. The solution of the flow balance equation is once again a product form with

$$P(n) = \frac{1}{G(K,M)} \prod_{i=1}^{M} \rho_i^{n_i} \qquad \text{where } \rho_i = \frac{\lambda_i}{\mu_i}$$

and $G(K,M)$ is a normalization constant so that $\sum_{n \in S} \rho_i = 1$ is given by

$$G(K,M) = \sum_{n \in S} \prod_{i=1}^{M} \rho_i^{n_i}$$

In order to determine $G(K,M)$ and $P(n)$ need $\lambda_i ;\forall i$

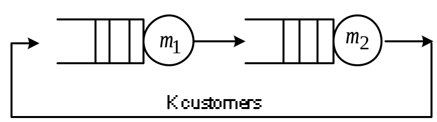
Flow conservation equation is

$$\lambda_i \quad = \quad \sum_{j=1}^{m+1} r_{ji} \lambda_j \Leftarrow \text{same as open network case without external arrivals or departures.}$$

Procedure is to arbitrarily set $\lambda_1 = 1$ or $\lambda_1 = \mu_1$ and determine $\lambda_i ; i > 1$ relative to λ_1 value. Changing λ_1 value will result in change of $G(K,M)$. $P(n)$ will remain constant.

For example, consider the tandem queue model with K=3.

Customer with $\mu_1 = 1$, $\mu_2 = 2$



From the diagram $r_{12} = r_{21} = 1 \Rightarrow \lambda_1 = \lambda_2$

State space $S = \{\ (0,3),\ (1,2),\ (2,1),\ (3,0)\ \}$

$$G(K,M) = G(3,2) = \sum_{n \in S} \prod_{i=1}^{M} \rho_i^{n_i} = \rho_2^3 + \rho_1 \rho_2^2 + \rho_1^2 \rho_2 + \rho_1^3$$

choosing $\lambda_1 = 1 \Rightarrow \lambda_2 = 1 \quad \Rightarrow \rho_1 = 1,\ \ \rho_2 = 0.5$

$G(3,2) = 1.875$ and $P(n) = \dfrac{1}{G(K,M)} \prod_{i=1}^{M} \rho_i^{n_i}$

results in

$P(0,3) = \rho_2^3 / G(3,2) = 0.0667 \quad , \quad P(1,2) = \rho_1 \rho_2^2 / G(3,2) = 0.1333$

$P(2,1) = \rho_1^2 \rho_2 / G(3,2) = 0.2667 \quad , \quad P(3,0) = \rho_1^3 / G(3,2) = 0.5333$

To illustrate the arbitrary value for λ_1 let $\lambda_1 = 0.5 \Rightarrow \lambda_2 = 0.5$

$\rho_1 = 0.5,\ \ \rho_2 = 0.25 \Rightarrow G(3,2) = 0.5333$ as before when $\lambda_1 = 1$

From $P(n)$, one can compute the standard mean performance measures

$$L_i = \sum_{j=0}^{K} j \left(\sum_{n_i = j;\, n \in S} P(n) \right) \qquad ;\ \text{note} \quad \sum_{i=1}^{M} L_i = K$$

From the example above,

$L_1 = 1P(1,2) + 2P(2,1) + 3P(3,0) = 2.2667$

$L_2 = 1P(2,1) + 2P(1,2) + 3P(0,3) = 0.7333$

Note that to find W_i, one needs to find the effective arrival rate

$$e_i = \mu_i \left(1 - \sum_{n_i = 0;\, n \in S} P(n) \right)$$

The effective server utilization $\rho_{e_i} = \dfrac{e_i}{\mu_i} = \left(1 - \displaystyle\sum_{n_i = 0;\, n \in S} P(n) \right)$

For the two queues example above

$e_1 = \mu_1 \big(1 - P(0,3)\big) = 0.9333 \qquad \rho_{e_1} = 0.9333$

$e_2 = \mu_2 \big(1 - P(3,0)\big) = 0.9333 \qquad \rho_{e_2} = 0.4667$

Note that one can also find e_i by first determining e_1 then using flow conservation equation.

$$W_1 = L_1 / e_1 = 2.4286$$

$$W_2 = L_2 / e_2 = 0.7857$$

As in the case of networks with population constraints, the computation of $G(K,M)$ is difficult when the state space become large. Can show that for a closed network of M queues with K customers the number of states is given by

$$\text{Number of states} = \binom{K+M-1}{M-1}$$

For even small networks, this is large. For example $K = 9$, $M = 2 \Rightarrow$ 3,628,800 states

Several algorithms have been proposed for computational efficiency in determining $G(K,M)$. One popular technique is Buzen's algorithm (also called convolution algorithm.) Buzen developed simple algorithm by noting

$$G(K,M) \quad = \quad G(K,M-1) + r_m G(K-1,M)$$

Proof is shown in Jain's text book.

This can be computed recursively by noting

$$G(0,m) = 1 \qquad m = 1,2,\ldots,M$$

$$G(k,1) = r_1^k \qquad k = 1,2,\ldots,K$$

This can be computed in a simple tabular form

$$\begin{array}{cc}
 & \begin{array}{ccccc} r_1 & r_2 & r_3 & & r_M \\ 1 & 2 & 3 & \ldots & M \end{array} \\
\begin{array}{c} 0 \\ 1 \\ 2 \\ \vdots \\ K \end{array} & \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ r_1 & r_1 + r_2 & r_1 + r_2 + r_3 & \ldots & \\ r_1^2 & r_1^2 + r_2(r_1 + r_2) & \ldots & \ldots & \\ \vdots & \vdots & \vdots & \vdots & \\ r_1^K & \ldots & \ldots & \ldots & \end{bmatrix}
\end{array}$$

The $i\ j$ element in the table is computed by taking the $i,(j-1)$ element adding $r_i \cdot (i-1,\ j\ element)$

For the two queue example previously discussed.

$\lambda_1 = 0.5$, $\lambda_2 = 0.5$, $\rho_1 = 0.5$, $\rho_2 = 0.25$

	ρ_1	ρ_2
	1	2
0	1	1
1	0.5	0.75
2	0.25	0.4315
3	0.125	0.2344

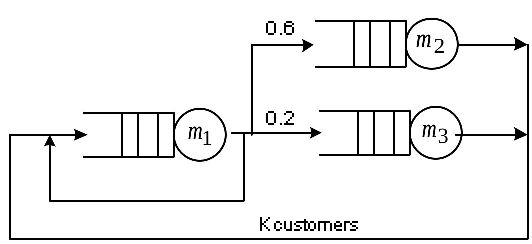
One of the advantages of this technique is that the performance measures can be written in terms of $G(K,M)$

$$L_i = \frac{1}{G(K,M)} \sum_{k=1}^{K} \rho_i^k G(K-k,M)$$

$$e_i = \lambda_i \frac{G(K-1,M)}{G(K,M)}$$

$$P(n_i \geq k) = \rho_i^k \frac{G(K-k,M)}{G(K,M)}$$

Example 2: Consider the simple model of a computer system shown below, queue 1– the CPU, queue 2–disk drive, and queue 3–I/O.



From the diagram $r_{11} = 0.2$, $r_{12} = 0.6$, $r_{13} = 0.2$, $r_{21} = r_{31} = 1$

$\mu_1 = 1$, $\mu_2 = 1$, $\mu_3 = 1$

Choosing $\lambda_1 = 10 \Rightarrow \lambda_2 = 6$, $\lambda_3 = 2$, and $\rho_1 = 1$, $\rho_2 = 1.2$, $\rho_3 = 2$

Computing $G(4,3)$

$\rho_1 = 1$ $\quad\quad$ $\rho_2 = 1.2$ $\quad\quad$ $\rho_3 = 2$

	1	2	3
0	1	1	1
1	1	2.2	4.2
2	1	3.64	12.04
3	1	5.368	29.448
4	1	7.4416	66.3376

$$e_1 = \lambda_1 \frac{G(3,3)}{G(4,3)} = 10 \times \frac{29.448}{66.3376} = 4.4391$$

Similarly,

$e_2 = 2.6635$ and $e_3 = 0.8878$

$$L_1 = \frac{1}{G(4,3)} \sum_{k=1}^{4} \rho_1^k G(4-k,3) = \frac{1}{G(4,3)} \left[\rho_1 G(3,3) + \rho_1^2 G(2,3) + \rho_1^3 G(1,3) + \rho_1^3 G(0,3) \right]$$

$L_1 = 0.7038$

Similary,

$L_2 = 0.9347$ and $L_3 = 2.3615$

$W_1 = L_1 / e_1 = 0.1585$, $W_2 = 0.3509$, $W_3 = 2.6599$

Let's look at two rather typical approximation approaches. The first is widely used for packet networks and the second for circuit switched networks.

1. Whitt's method for open's network of G/G/1 queue (QNA)

The basic idea is to use the KLB G/G/1 two moment approximation at each queue *i* in the network. The model of queue *i* is similar to the arbitrary queue studied in Jackson networks.

Assume arbitrary network of M queues, define

λ_i – Total mean customer arrival rate to queue *i*.

γ_i – Mean arrival rate from outside of network to queue *i*.

$r_{i,j}$ – Routing probability customer leaving queue *i* goes to queue *j*.

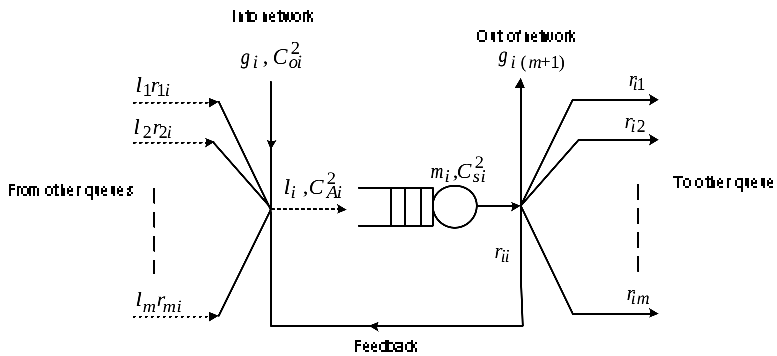
$r_{i,m+1}$ – Probability customer leaving queue *i* exits the network.

μ_i – Mean service rate at queue *i*.

Co_i^2 – Squared coefficient of variation of outside arrivals to *i*.

Cs_i^2 – Squared coefficient of variation of service process at *i*.

C_{Ai}^2 – Squared coefficient of variation of arrival process at *i*.



As in the Open Jackson network case, find mean arrival rate at each queue *i* by the flow conservation equation

$$\lambda_i = \gamma_i + \sum_{j=1}^{m} r_{ji}\,\lambda_j$$

$\Rightarrow \lambda = \gamma (I - R)^{-1}$ Matrix vector solution

To apply KLB equation need C_{Ai}^2 at each queue. This requires the application of three approximations (parallel to Jackson network approach) for

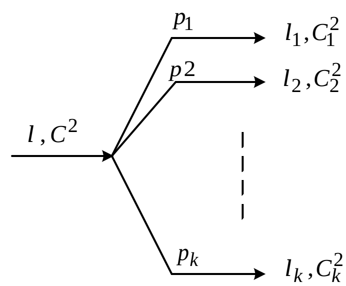
1. Departure process approximation

Mean departure rate = λ

C_d^2 = squared coefficient of variation of departure process.

$C_d^2 \approx \rho^2 C_s^2 + (1 - \rho^2) C_A^2$ $\Leftarrow$ based on renewal process approximation

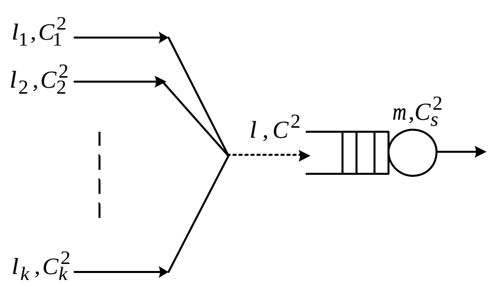
2. Splitting: If a process with mean λ and C^2 is probabilistically split into K stream with probabilities p_i ($\sum_{i=1}^{K} p_i = 1$). We can approximate C_i^2 as below



Where $\lambda_i = p_i \lambda$

$$C_i^2 \approx p_i C^2 + (1 - p_i)$$

3. Merging: The C^2 of a merger of K streams is approximated by $\lambda = \sum_{i=1}^{K} \lambda_i$



$$C_{Ai}^2 = 1 - W_i + W_i \left[\frac{\gamma_i}{\lambda} C_{oi}^2 + \sum_{j=1}^{M} \frac{\lambda_i \rho_{ji}}{\lambda} \left[\rho_{ji} \left(\rho_j C_{sj}^2 + (1 - \rho_j^2) C_{Aj}^2 \right) + (1 - \rho_{ji}) \right] \right]$$

yields a system of linear equations to solve for C_{Ai}^2

where

$$W_i = \left[1 + 4(1 - \rho_i)^2 \left(\sum_{j=1}^{M} \frac{(\lambda_i \rho_{ji})^2}{\lambda_i^2} - 1 \right) \right]^{-1}$$

This approximation tends to do pretty well on network-wide measures LN, WN, etc…, but not so well for individual queues.