

19.Closed Model

Visits in closed models are similar to open models but we need the reference station = 1 (In the P matrix the reference station column will have all 0) So I is a vector with all 0 but the reference position where there is 1

Express performance matrix for closed models

in open models we used a trick: $A_k = N_k$. Since $R_k = (1 + A_k)D_k = (1 + \lambda R_k)D_k \Rightarrow R_k = \frac{D_k}{1 - U_k}$
But in closed models this is no longer true as the number of jobs that arrive isn't equal to the jobs that left the system.

The number of jobs at arrival in closed models can be expressed through the "Arrival Theorem":

$$A_k(N) = N_k(N - 1)$$

We can represent the Residence time of the system (With N jobs in the system) as:

$$R_k(N) = (1 + A_k(N))D_k = (1 + N_k(N - 1))D_k$$

Suppose we know $N_k(N - 1)$ we can compute the response time:

$$R(N) = \sum R_k(N), \quad X(N) = \frac{N}{R_k(N)}$$

Using Little's law, the queue length of each station can finally be determined.

$$N_k N = X N R_k(N)$$

The algorithm can then increase the population to N+1 jobs, since for the arrival theorem we have that $A_k(N + 1) = N_k(N)$, and the process can be repeated.

$$A_k N + 1 = N_k N$$

$$R_k(N + 1) = (1 + A_k(N + 1))D_k = 1 + N_k N D_k$$

Response time law

In time-sharing systems, the "think time" is usually not considered in the system response time.

$$R_{tot} = R_{sys} + Z \quad N = X R_{tot} \quad N = X(R_{sys} + Z)$$

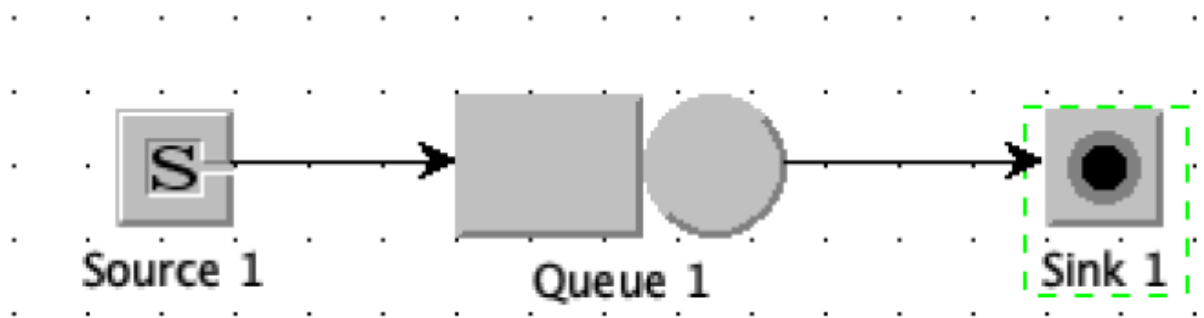
In particular, Little's law becomes the so-called "Response Time Law" which explicitly excludes the think time from the total response time of the system computed in the usual way.

The Response Time Law is:

$$R = \frac{N}{X} - Z$$

Separable Models

As queuing policies we consider LCFS, PS and FCFS



We consider for both service and arrival process:

- Exponential (c.v. = 1)
 - Erlang, with c.v. = 0.5
 - Hyper-Exponential, with c.v. = 2
- Kingsman Formula considers FCFS

With the considered arrival rate and demand, analytical results for the M/M/1 model would be:

$$R = \frac{D}{1 - \lambda \cdot D} = \frac{1}{1 - 0.8 \cdot 1} = 5s$$

Note that, beside the M/M/1 case (orange background), such results are also obtained for the PS queuing policy with exponential inter-arrival time, and for M/M/1/LCFS (yellow background).

Arrival (Average $\lambda = 0.8$ j/s)	Service (Average D = 1s)			
	LCFS			
		Exp	Erl	Hyper
	Exp	4.9845	3.5012	10.9925
	Erlang	3.2733	1.8351	9.2016
	Hyper-exp	10.7959	9.1543	17.2105
	PS			
		Exp	Erl	Hyper
	Exp	4.9813	5.031	4.9152
	Erlang	3.369	2.366	4.1592
	Hyper-exp	10.8972	13.8811	7.8524
	FCFS			
		Exp	Erl	Hyper
	Exp	5.0473	3.5016	11.0676
	Erlang	3.2566	1.8468	9.1904
	Hyper-exp	10.8286	9.0071	17.2209

Also no difference in both FCFS and LCFS have same Average Response Time

It has been discovered that the analytical results such the one presented for Open and Closed networks, are valid for a large variety of systems.

In particular, the techniques e can be applied if a system is a separable queuing network.

A queueing network is separable if it fulfills five main properties.

In the following, we will present the properties directly from the book Quantitative System Performance by E. D. Lazoswka.

First assumptions: service center flow balance

$$A_k(T) = C_k(T)$$

This is always true in stable system when there aren't losses. System with losses cannot be separate.

Second assumption

Two jobs cannot change the state at the same time in the same time. Real systems almost certainly display one step behaviour (even for milliseconds)

Third assumption: Routing homogeneity

If there is not dependency on the routing on the left of the queue the system can have routing homogeneity and can be separable

Of the routing policies we have seen, the probabilistic routing fully satisfies the previous assumption.

Other routing policies, such as Join the Shortest Queue (JSQ) clearly violate this constraint since the decision of the next station depends on the whole state of the system.

The previous assumption has a very big implication: as long as nodes have the same demands, two queuing networks have the same performance, independently of their topology.

Fourth assumption:

Device homogeneity. Tricky and important assumption

Fifth assumption:

Homogeneous external arrivals

These assumptions are strict but almost always valid for computer systems.