Computing infrastructure

POLITECNICO DI MILANO



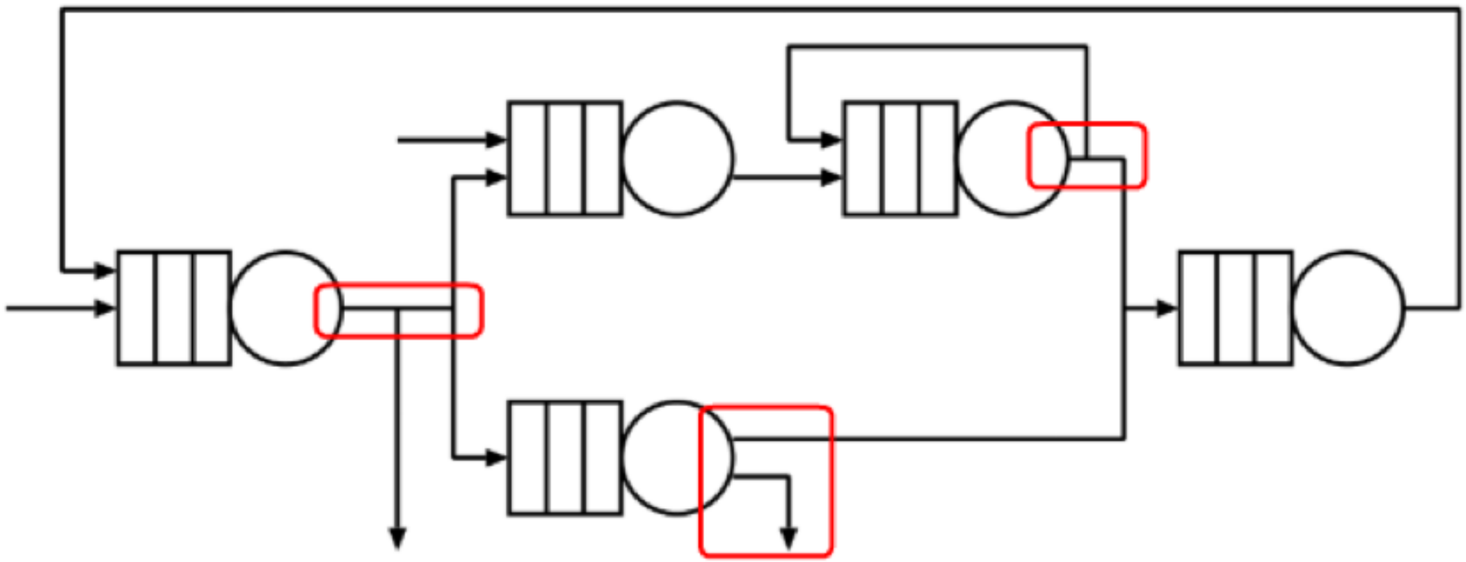


Computing visits

Routing probabilities

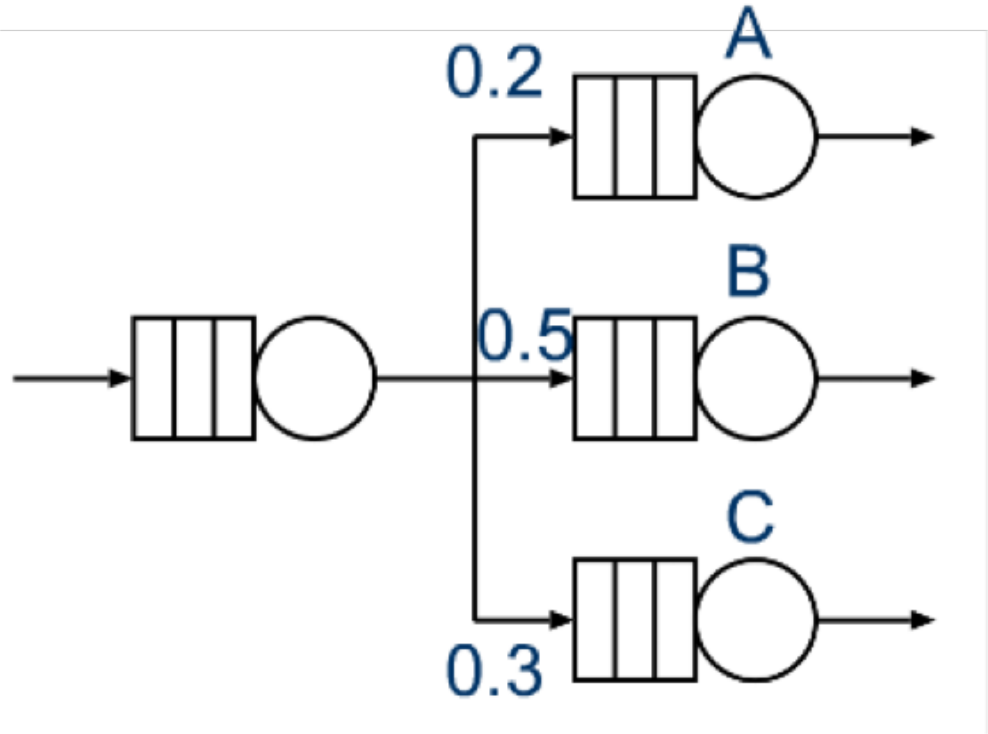
Whenever a job, after finishing service at a station has several possible alternative routes, an appropriate selection policy must be defined.

The policy that describes how the next destination is selected is called *routing*.



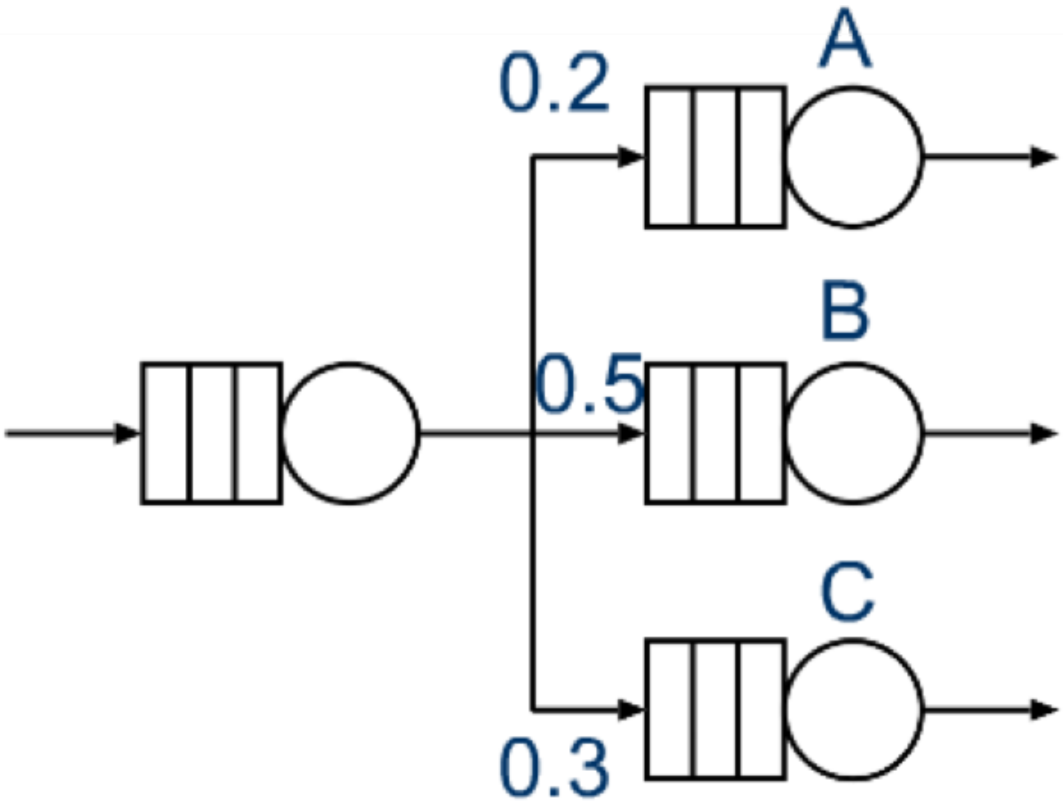
Probabilistic routing

We will focus on the so called *probabilistic routing*, in which each path has assigned a probability of being chosen by the job that left the upstream station.



Probabilistic routing

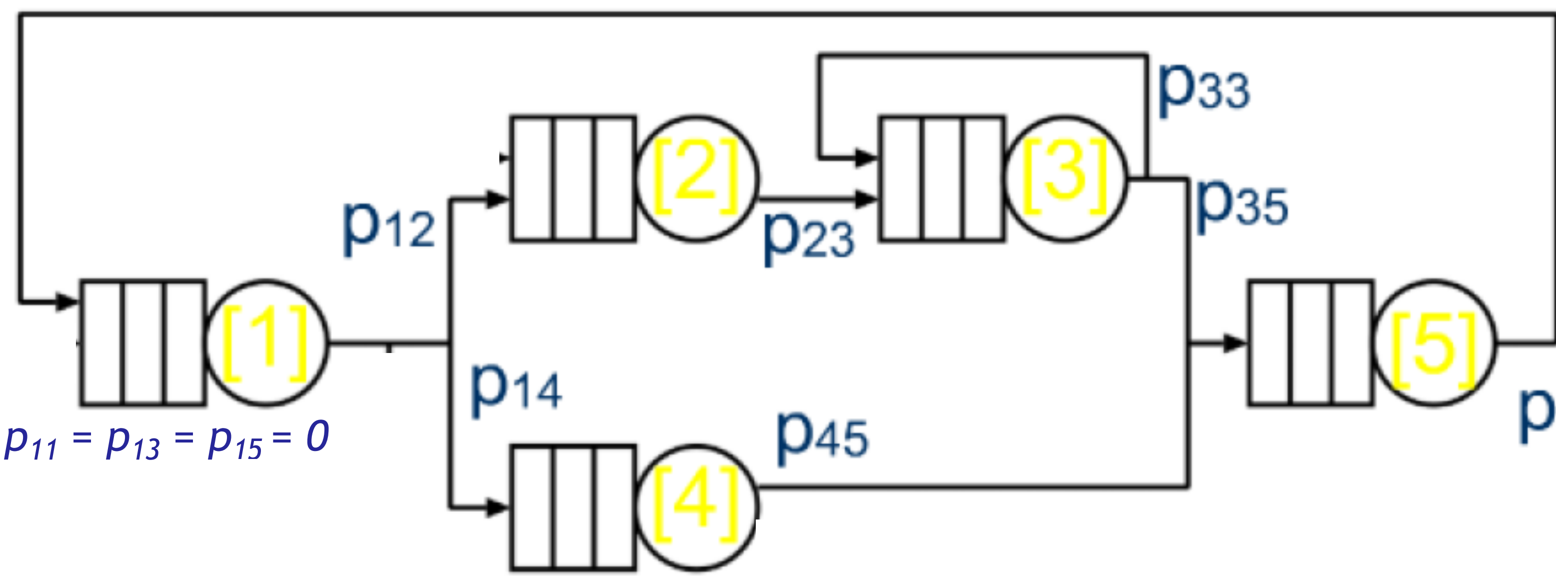
By appropriately assigning values to the probabilities associated to each possible downstream node, the modeler can match the flux of jobs in a real system.



Computing visits

Let us call p_{ij} the probability that a job, which finishes its service at node i, choses node j as its next destination.

If the considered route is not possible, we set $p_{ij} = 0$.



$p_{11} = p_{13} = p_{15} = 0$

Computing visits

We can then determine the visits v_k to each station k by solving the following linear system of equations:

$$\begin{cases} v_k = \sum_{i=1}^{K} v_i \cdot p_{ik} \\ \dots \end{cases}$$

The term v_k on the left hand side of the equations counts the jobs that visit station k.

In particular, this count is equal to the sum of number of fractions of jobs which are routed to the considered station k that arrives from from every other station i (p_{ik}).

Note that the summation includes also index k to allow the probability for performing self-loops.

Computing visits

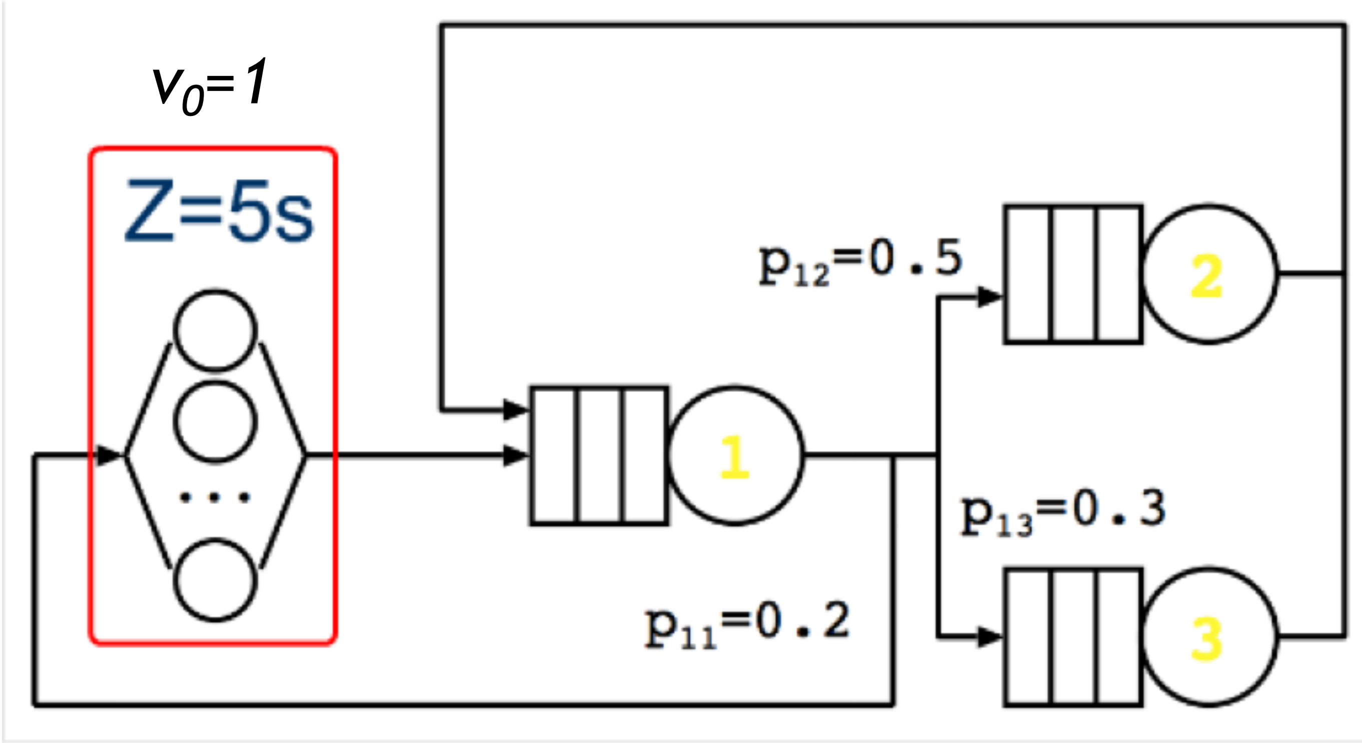
The way in which the system of equation is solved is different depending on whether we are considering open or closed models.

We start considering closed time-sharing models.

Computing visits

In time-sharing systems, each job passes exactly once infinite server that defines the think time of the jobs.

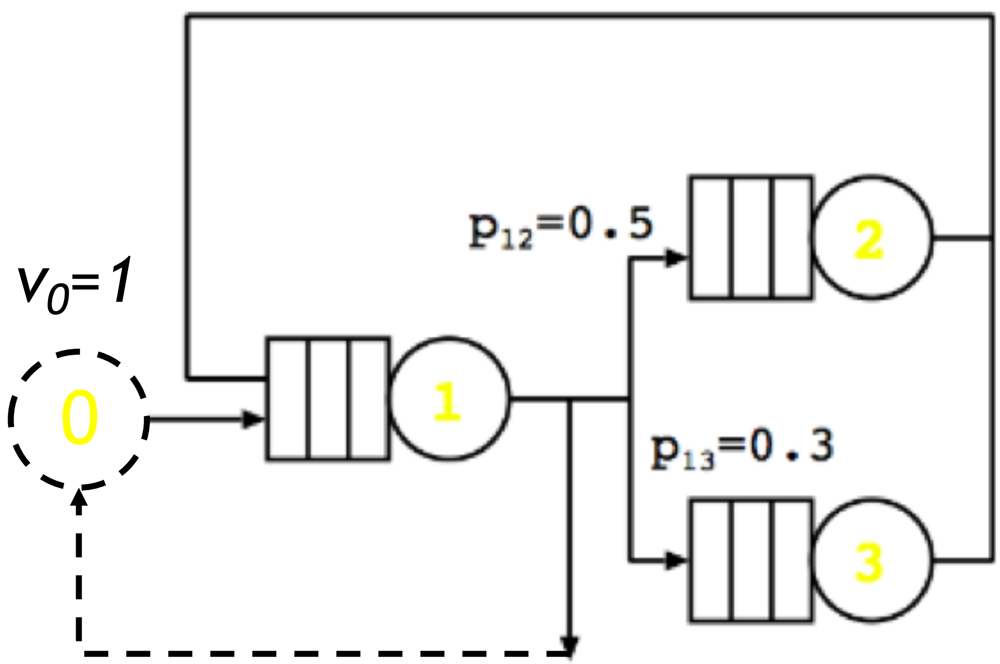
We call this station, *station number 0*, and we set its visits $v_0=1$



Computing visits

In open models, each job comes and leaves the system exactly once.

We add an external *virtual station*, again called *station number 0*, which represents the external environment, and we set its visits $v_0=1$



Computing visits in closed models

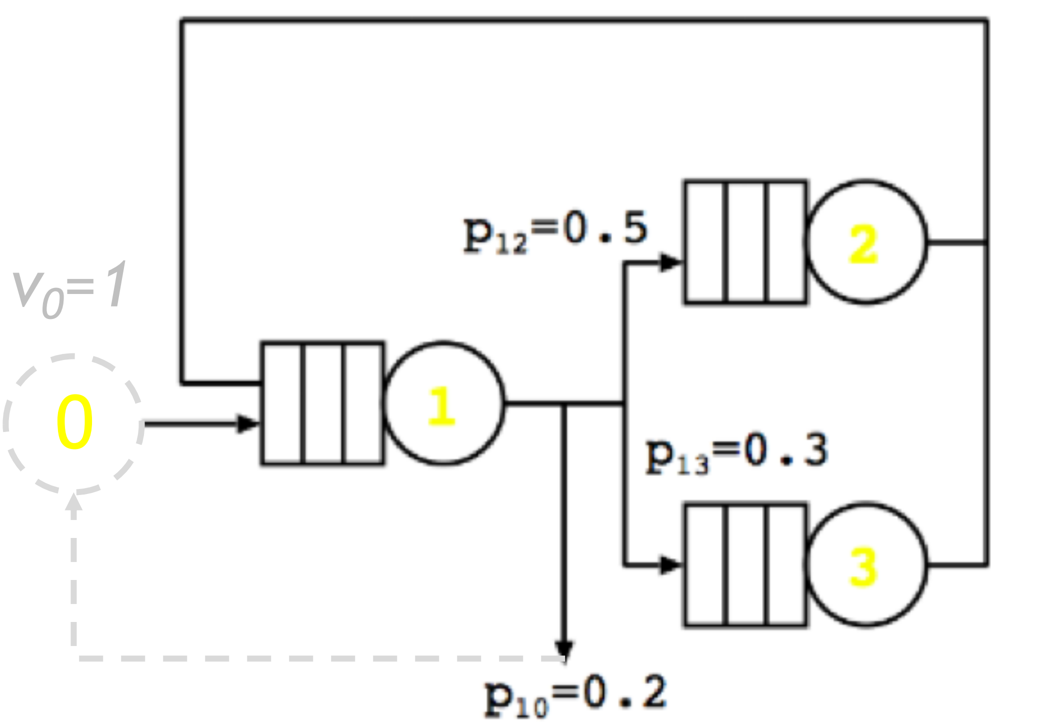
In both cases we add an extra equation that sets the visits to *station 0* equal to one.

$$\begin{cases} v_0 = 1 \\ v_k = \sum_{i=1}^{K} v_i \cdot p_{ik} \qquad \forall k > 0 \end{cases}$$

Computing visits in open models

Example

Compute the visits for the following open model:



$$\begin{cases} v_0 = 1 \\ v_1 = v_2 + v_3 + v_0 \\ v_2 = 0.5 \cdot v_1 \\ v_3 = 0.3 \cdot v_1 \end{cases}$$

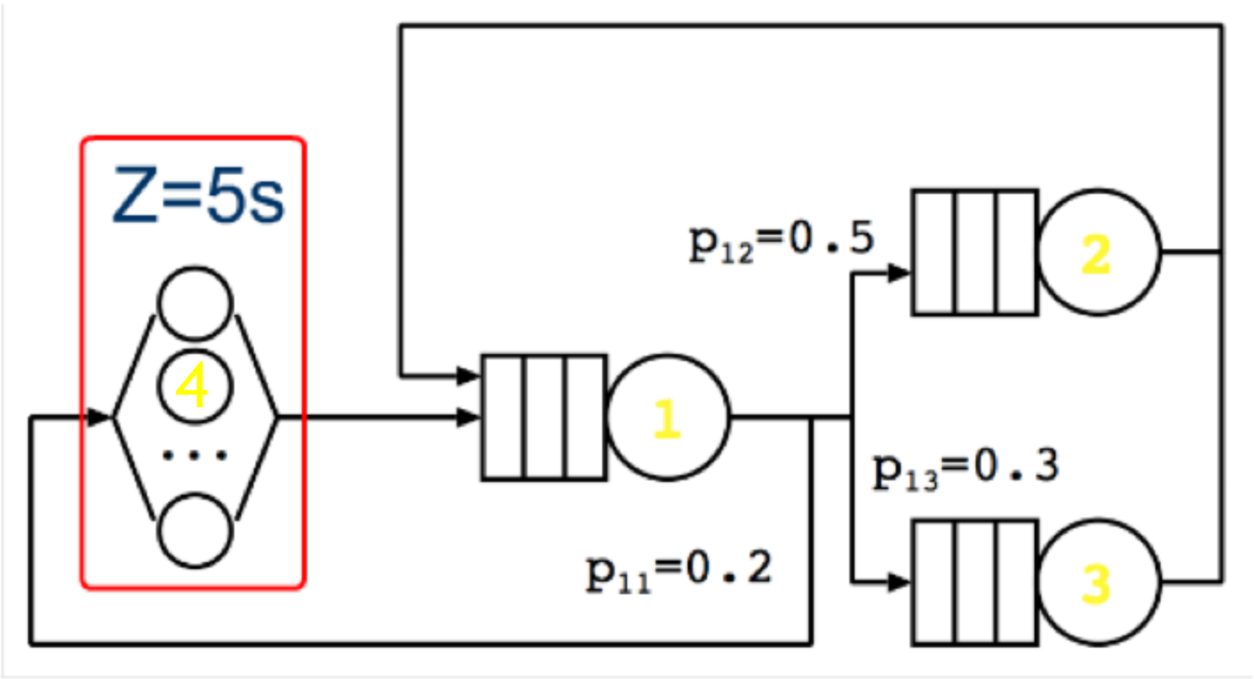
$$\begin{cases} v_0 = 1 \\ v_1 = 0.8 \cdot v_1 + 1 \\ v_2 = 0.5 \cdot v_1 \\ v_3 = 0.3 \cdot v_1 \end{cases}$$

$$\begin{cases} v_1 = 5 \\ v_2 = 2.5 \\ v_3 = 1.5 \end{cases}$$

Computing visits in closed models

Example

Compute the visits for the following closed time sharing model:



$$\begin{cases} v_0 = 1 \\ v_1 = v_2 + v_3 + v_4 \\ v_2 = 0.5 \cdot v_1 \\ v_3 = 0.3 \cdot v_1 \end{cases}$$

$$\begin{cases} v_0 = 1 \\ v_1 = 0.8 \cdot v_1 + 1 \\ v_2 = 0.5 \cdot v_1 \\ v_3 = 0.3 \cdot v_1 \end{cases}$$

$$\begin{cases} v_0 = 1 \\ v_1 = 5 \\ v_2 = 2.5 \\ v_3 = 1.5 \end{cases}$$