BCMP networks

There are N nodes and R classes (or types) of job. For each class, we must specify routing probabilities through the network (these can be class dependent). A class can either be open (jobs enter from outside and eventually leave) or closed (jobs never leave). As discussed in class, the nodes are allowed to be one of four types:

1. **FCFS** Here, jobs are served in a first come, first served order. Multiple classes may visit a node, but in this case the service time distributions must be the same (and exponentially distributed) for all classes. The service rates may be load-dependent.

2. **PS** Here jobs are served using processor sharing, with each waiting job getting an equal share of capacity. Jobs of different classes may have different service requirements and the service rates (for each class) may depend on the queue length at the node. The service distributions must be so-called Coxian type (essentially a combination of exponential distributions), but only the expected value needs to be determined.

3. **IS** or **delay** Here an infinite number of servers is available, or equivalently, each job is served by "their own" server. Jobs of different classes may have different service requirements and the service rates (for each class) may depend on the queue length at the node. The service distributions must be so-called Coxian type (essentially a combination of exponential distributions), but only the expected value needs to be determined.

4. **LCFS-PR** Here jobs are served on a last come first serve basis, with preemption (also, work done on preempted jobs is not lost). Further restrictions are the same as in the previous two cases.

At this point, let me indicate that the results below depend only on the mean service times. This is why I have not discussed Coxian distributions. It will suffice at this point to note that we can approximate most distributions by a Coxian distribution and thus when the nodes are of the last 3 types, there is no practical limitation on the service time distributions.

Upon leaving node i, a job of class r goes to node j and becomes a job of class s with probability $P_{i,r;j,s}$. A job will leave the network with probability $P_{i,r;0}$. Of course, there can only be arrivals from outside of the system for classes that are open. In this case, there are two possibilities which are allowed.

The first possibility is that there is a single Poisson process with rate $\lambda(k)$ where k is the total population in the network. Upon arrival to the system, a job goes to node i as a class r job with probability $P_{0;i,r}$.

The second possibility is that each routing chain has its own arrival stream, with a rate that depends only on the population of that chain (which we will give by $\lambda_c(k_c)$, with $c \in C$, where C is the set of routing chains and k_c is

REVLO BETA

Simpler, Smarter Loyalty

Check out Revlo!

A free loyalty and rewards platform
for Twitch fans and streamers!

www.revlo.co

the population in routing chain c. For each stream, with probability $P_{0;i,c}$ an arrival joins node i.

For each routing chain c, we want to write an equation for the net arrival rate to node i of class r jobs. This can be written as

$$\lambda_{i,r} = \lambda^*_{i,r} + \sum_{(j,s)} \lambda_{j,s} P_{j,s;i,r}.$$

Here, $\lambda^*_{i,r}$ is the arrival rate of jobs from outside of the system. For closed networks it is 0; for open networks it equals $\lambda r_{0;i,c}$ (one arrival process) or $\lambda_r r_{0;i,r}$ (arrivals per chain/class). This equation has a very simple intuitive explanation. The left side is the arrival rate to (i,r), the first term on the right hand side is the arrival rate to (i,r) from outside, and the final term is the sum of the arrival rates to (i,r) from all other (node,class) pairs in the network. Using this equation, assuming the system is stable, one can calculate the throughputs for open chains $\lambda_{i,r}$, and visit ratios for closed chains $V_{i,r}$. Of course, it is not uncommon that for closed systems, the visit ratios are given directly (think of what we have done in class and that you have done in assignments).

The main result is now stated (the proof is really beyond the scope of the course, but is not particularly difficult... if anybody is interested, just ask). We need a couple of definitions, to define what the state of the queueing network is. Let $\bar{N}_i$ be the vector $(N_{i,1}, N_{i,2}, \ldots, N_{i,R})$ denote the state of node i, where $N_{i,r}$ gives the number of class r jobs at node i. The state of the system is the vector $\bar{N} = (\bar{N}_1, \bar{N}_2, \ldots, \bar{N}_N)$ and the total number of jobs in the system is K.

BCMP Theorem The steady-state probability distribution in a BCMP network has the following product form:

$$\pi_{\bar{n}} = \frac{1}{G} A(\bar{n}) \prod_{i=1}^{N} \pi_i(\bar{n}_i),$$

where G is a normalizing constant (it assures that the probabilities sum to one), $A(\bar{n})$ is a function of the external arrival processes only, and the functions $\pi_i(\bar{n}_i)$ are the "per-node" steady-state distributions.

The important point of this result is that there are explicit expressions for the p functions. They are as follows (note that n_i is $\sum_{r=1}^{R} n_{i,r}$)

When node i is of type FCFS, we have in the load-independent case

$$\pi_i(\bar{n}_i) = n_i! \left(\prod_{r=1}^{R} \frac{1}{n_{i,r}!} V_{i,r}^{n_{i,r}} \right) \left(\frac{1}{\mu_i} \right)^{n_i},$$

and in the load-dependent case

$$\pi_i(\bar{n}_i) = n_i! \left(\prod_{r=1}^{R} \frac{1}{n_{i,r}!} V_{i,r}^{n_{i,r}} \right) \prod_{j=1}^{n_i} \frac{1}{\mu_i(j)}$$

When node i is of type PS or LCFS-PR, we have in the load-independent case

$$\pi_i(\bar{n}_i) = n_i! \prod_{r=1}^{R} \frac{1}{n_{i,r}!} \left(\frac{V_{i,r}}{\mu_{i,r}}\right)^{n_{i,r}},$$

and in the load-dependent case

$$\pi_i(\bar{n}_i) = n_i! \prod_{r=1}^{R} \frac{1}{n_{i,r}!} V_{i,r}^{n_{i,r}} \prod_{j=1}^{n_i} \frac{1}{\mu_{i,r}(j)}.$$

When node i is of type IS, we have in the load-independent case

$$\pi_i(\bar{n}_i) = \prod_{r=1}^{R} \frac{1}{n_{i,r}!} \left(\frac{V_{i,r}}{\mu_{i,r}}\right)^{n_{i,r}},$$

and in the load-dependent case

$$\pi_i(\bar{n}_i) = \prod_{r=1}^{R} \frac{1}{n_{i,r}!} V_{i,r}^{n_{i,r}} \prod_{j=1}^{n_i} \frac{1}{\mu_{i,r}(j)}.$$

Finally, the term $A(\bar{n})$ is determined by the arrival processes in the following manner. If all chains are closed, then $A(\bar{n}) = 1$. If the arrivals depend on the total system population, then it is equal to $A(\bar{n}) = \prod_{j=0}^{k-1} \lambda(j)$, where k is the network population. If the arrivals are per chain, then $A(\bar{n}) = \prod_{c=1}^{N_C} \prod_{j=0}^{k_c-1} \lambda_c(j)$, where N_C is the number of routing chains and k_c is the population in routing chain c.

At this point, all of this notation may seem a bit much, so there will be two examples given at this point which are special cases of the BCMP theorem which are of great practical interest. After that, an example will be given that we will spend some time on.

Example. Single-class, load-independent open networks. Here, the arrival process is Poisson of constant rate λ (there is no load dependence for the arrivals). Also, the service rates are fixed. If the node is FCFS, PS or LCFSPR, there is only one server. Then

$$\pi_{\bar{n}} = \prod_{i=1}^{N} p_i(n_i),$$

where

$$\pi_i(n_i) = \begin{cases} (1-\rho_i)\rho_i^{n_i}, & \text{FCFS, PS, LCFSPR type,} \\ e^{-\rho_i}\frac{\rho_i^{n_i}}{n_i!}, & \text{IS type,} \end{cases}$$

where ρ_i is defined as

$$\rho_i = \begin{cases} \sum_{r\in R_i}, \frac{\lambda V_{i,r}}{\mu_i} & \text{FCFS type,} \\ \sum_{r\in R_i} \frac{\lambda V_{i,r}}{\mu_{i,r}}, & \text{IS, PS, LCFSPR type,} \end{cases}$$

where R_i is the set of classes that require service at node i. You should be able to verify this result yourself, it is a decent exercise to get used to all of the notation. Note that $A(\bar{n})$ has been absorbed into the definition of ρ_i. This result should be somewhat intuitive. It says that the the system decomposes into M/M/1 (or M/M/∞) queues with the appropriate arrival rates.

Example. Closed, multi-class, load-independent BCMP networks. A lot of computer systems examples have load-independent servers, multiple customer classes (but no class changes) and fixed populations per class. Here,

$$\pi_{\bar{n}} = \frac{1}{G} \prod_{i=1}^{N} p_i(\bar{n}_i),$$

with

$$\pi_i(\bar{n}_i) = \begin{cases} n_i! \left(\frac{1}{\mu_i}\right)^{n_i} \prod_{r=1}^{R} \frac{1}{n_{i,r}!} V_{i,r}, & \text{FCFS type,} \\ n_i! \prod_{r=1}^{R} \frac{1}{n_{i,r}!} \left(\frac{V_{i,r}}{\mu_{i,r}}\right)^{n_{i,r}}, & \text{PS, LCFSPR type,} \\ \prod_{r=1}^{R} \frac{1}{n_{i,r}!} \left(\frac{V_{i,r}}{\mu_{i,r}}\right)^{n_{i,r}}, & \text{IS type.} \end{cases}$$

Note that $n_i = \sum_{r=1}^{R} n_{i,r}$.