Load Balancing Control for Parallel Systems

Jean-Claude Hennet
LAAS-CNRS, 7 avenue du Colonel Roche, 31077 Toulouse, France
E-mail hennet@laas.fr, Phone: (33) 61 33 63 13, Fax: (33) 61 55 35 77

Abstract    The addressed problem is to construct and to implement a load balancing policy in a Discrete Event System with multiple routing choices. The construction scheme determines optimal steady-state routing ratios, which are next transformed into valuations on the arcs of a sub-Petri net used to generate the real-time loading decisions. Such an implementation scheme performs as the optimal static policy in the deterministic context, and it allows flexibility and reactivity in a perturbed or in a fully random context.

Keywords    Discrete Event Systems, Queueing Networks, Petri Nets, Load Balancing, Bernoulli splitting

1 INTRODUCTION

Control of Discrete Event Systems is a challenging domain both for theory and applications. One major issue in the design of control policies is to conciliate efficiency, robustness and flexibility. In loading and scheduling problems, the deterministic approach to optimal control problems often fail because of the curse of dimensionality in combinatorial problems. The stochastic approach allows the construction of simpler models, but under some drastic assumptions, mainly used to fit into the framework of Markov Decision Processes or Queueing Networks theory. Furthermore, very few results are available for the design of optimal dynamic control policies.

The problem of optimally routing customers is basic for load balancing problems arising in many Discrete Event Systems such as computer systems, manufacturing plants and telecommunication networks.

Dynamic load balancing policies use the information of the current state of the system at each customer arrival date. In particular, the optimal routing problem for parallel M/M/1 queues has been shown to admit an optimal solution of the switch-over type [11] for discounted cost criteria and, under some ergodicity conditions, for average cost criteria. But the determination of the switching surfaces, by algorithms such as policy iteration, is difficult, and even generally untractable in the case of infinite-dimensional state-spaces.

On the contrary, static load balancing policies are often much easier to compute. And they can be very simply implemented, because they do not use any real-time observation of the system state. They can be divided into two groups: the deterministic ones, which assign customers to queues in a pre-determined periodical order, and the stochastic ones, which route customers to queues according to a Bernoulli splitting random process with pre-determined probabilities.

Theoretical and practical comparisons [7] of these routing schemes in simple cases have confirmed the superiority of dynamic load balancing policies over static load balancing policies and of deterministic static load balancing policies over Bernoulli load balancing policies.

However, it is important to note that, on the average, decisions taken by static and dynamic optimal policies are the same: over a long time horizon they send the same average percentages of customers on each route.

Even in the simple case of n parallel M/M/1 queues with different mean service rates, the optimal dynamic loading policy may be extremely difficult to compute, and the dependence of the solution with respect to system parameters has not yet been explicited.

The approach proposed in this paper consists of three stages:

- Resolution of a static optimization problem to determine the optimal average routing ratios. Several algorithms have been constructed for solving this problem in different frameworks and under different bodies of assumptions on release dates and service times [10], [4], [9]. This paper gives explicit expressions of the optimal steady-state loading parameters for a system of parallel queues with different service rates. Average routing parameters are computed under the assumptions of a Poisson input, of exponentially distributed service times, and under a Bernoulli splitting of the input flow. The first step of the method determines which servers should be used and which ones should not be used by the policy. Then, the optimal routing parameters are obtained as the optimal solution of an unconstrained
problem.
- Construction of a cyclic scheme achieving a deterministic implementation of the optimal routing ratios.
  The proposed method is an extension of an original method proposed by H. Ohl et al. [8]. In this respect, the contribution of this paper is mainly in the technique for computing the arc valuations of the control subnetwork implementing the optimal routing ratios.
- Introduction of sequencing flexibility by an increase of the number of tokens in the control subnetwork [8]. If the number of extra tokens is large enough, the modified control scheme can compensate for a temporary unavailability of some servers. The firing of some extra transitions in the cases of failures and of local overloads is a complementary means for short-circuiting unavailable routes.

2 A STEADY-STATE FLOW DISPATCHING POLICY

Consider an open queueing network which admits an exact, generalized, or approximate product form. Its random input flow is supposed to be Poisson with mean arrival rate \( \lambda > 0 \). This open network is of the Jackson type [1]. The steady state probability distribution of customers in the system is given by:

\[
P(y_1, ..., y_n) = p_1(y_1) \times \cdots \times p_n(y_n)
\]

with, for \( i = 1, ..., n, \)

\[
p_i(y_i) = (1 - \rho_i) \rho_i^{y_i} \text{ and } \rho_i = \frac{\alpha_i \lambda}{\mu_i}.
\]

If \( \rho_i < 1 \) for \( i = 1, ..., n, \) the average value of the time spent in the system (or mean response time) is:

\[
E(T) = \sum_{i=1}^{n} \frac{\alpha_i}{\mu_i - \alpha_i \lambda}.
\]

By Little’s formula [6], minimization of this performance index is equivalent to minimizing the total average number of jobs in the network. Under a uniform weighting of the inventory costs, such a minimization also corresponds to the minimal average total inventory cost.

Without loss of generality, the \( n \) servers can be ordered in the decreasing order of their mean service rate:

\[
\mu_1 \geq \cdots \geq \mu_n.
\]

The constraints of the problem are related to the following requirements:

1) The control policy has to be feasible. This condition is satisfied under the following constraints:

\[
0 \leq \alpha_i \leq 1 \quad \forall i \in (1, .., n)
\]

\[
\sum_{i=1}^{n} \alpha_i = 1
\]

2) Stability of the network requires restrictions:

\[
\lambda < \sum_{i=1}^{n} \mu_i
\]

\[
\frac{\alpha_i \lambda}{\mu_i} < 1 \text{ for } i = 1, ..., n.
\]

Condition (7) also plays a crucial role in the study of dynamic loading policies. In that context, it guarantees the existence of control policies for which the system is ergodic and the minimal average cost is finite [11].
2.2 Optimal Routing Parameters

Assume that condition (7) is satisfied by the problem data. The optimization problem, denoted (CP), for criterion \( E(T) \) takes the form:

\[
\min_{\alpha_1,\ldots,\alpha_n} \sum_{i=1}^{n} \frac{\alpha_i}{(\mu_i - \alpha_i \lambda)} \quad (9)
\]

subject to:

\[
0 \leq \alpha_i \leq \min(1, \frac{\mu_i}{\lambda}) \quad \text{for} \quad i = 1, \ldots, n-1, \quad (10)
\]

and:

\[
0 \leq 1 - \sum_{j=1}^{n-1} \alpha_j \leq \min(1, \frac{\mu_n}{\lambda}) \quad (11)
\]

Constraints are linear and for \( i = 1, \ldots, n, \)

\[
\frac{\partial E(T)}{\partial \alpha_i} = \frac{\mu_i}{(\mu_i - \alpha_i \lambda)^2} > 0 \quad (12)
\]

Therefore, within its set of constraints, this problem is convex.

2.2.1 The unconstrained optimality conditions

Consider first the case when the optimum of the constrained problem is also the optimum of the unconstrained problem, namely, when the global minimum of (9) satisfies (10) and (11). In this case, the optimal solution is simply obtained by solving the first order optimality conditions of the unconstrained problem, that is, for \( i = 1, \ldots, n - 1: \)

\[
dE(T) \left( \frac{d}{d\alpha_i} \right) = \frac{\mu_i}{(\mu_i - \alpha_i \lambda)^2} - \frac{\mu_n}{(\mu_n - (1 - \sum_{j=1}^{n-1} \alpha_j) \lambda)^2} = 0 \quad (13)
\]

If constraints (10) and (11) are satisfied at the unconstrained optimum of (9), the first order optimality conditions yield, for \( i = 1, \ldots, n - 1: \)

\[
\frac{\mu_i - \alpha_i \lambda}{\mu_n - (1 - \sum_{j=1}^{n-1} \alpha_j) \lambda} = \sqrt{\frac{\mu_i}{\mu_n}} \quad (14)
\]

Then, for any \( i \in \{1, \ldots, n\} \) and for any \( j \in \{1, \ldots, n\}, \)

\[
(\mu_j - \alpha_j \lambda) = \sqrt{\frac{\mu_j}{\mu_i}} (\mu_i - \alpha_i \lambda) \sqrt{\mu_j} \quad (15)
\]

Summing over \( j \) the 2 terms of this equation yields:

\[
\sum_{j=1}^{n} \mu_j - \lambda = \left( \frac{\mu_i - \alpha_i \lambda}{\sqrt{\mu_i}} \right) \sum_{j=1}^{n} \sqrt{\mu_j} \quad (16)
\]

And therefore, for \( i \in \{1, \ldots, n\}, \)

\[
\alpha_i = \frac{1}{\lambda} (\mu_i - \tau_n \sqrt{\mu_i}), \quad \text{with} \quad \tau_n = \frac{(\sum_{j=1}^{n} \mu_j - \lambda)}{\sum_{j=1}^{n} \sqrt{\mu_j}} \quad (17)
\]

By construction, this set of parameters satisfies:

\[
\sum_{i=1}^{n} \alpha_i = \frac{1}{\lambda} \left( \sum_{i=1}^{n} \mu_i - \tau_n \sum_{i=1}^{n} \sqrt{\mu_i} \right) = 1. \quad (18)
\]

The global minimum of (9) is feasible (and therefore optimal) for the constrained problem if and only if:

\[
0 \leq \mu_i - \tau_n \sqrt{\mu_i} \leq \min(\mu_i, \lambda) \quad \text{for} \quad i = 1, \ldots, n \quad (19)
\]

Stability condition (7) implies that \( \tau_n \) is positive. Then, condition (18) can be replaced by:

\[
0 \leq \mu_i - \tau_n \sqrt{\mu_i} \leq \lambda \quad \text{for} \quad i = 1, \ldots, n \quad (19)
\]

and the left part of these inequalities becomes equivalent to:

\[
\mu_i \geq \tau_n^2 \quad \text{for} \quad i = 1, \ldots, n. \quad (20)
\]

2.2.2 Solving the constrained problem

If the left part of inequality (19) is violated for some \( i_0, 0 < i_0 \leq n \), it is also violated for any \( i : i_0 \leq i \leq n \). Then, a restricted choice problem can be formulated, for which \( \alpha_i = 0 \) for \( i = i_0, \ldots, n \). To show the relevance of the problem, define the optimality parameter associated to routings restricted to the first routes:

\[
\tau_{m} = \frac{(\sum_{i=1}^{m} \mu_i - \lambda)}{\sum_{i=1}^{m} \sqrt{\mu_i}}. \quad (21)
\]

Under the convention \( \mu_{n+1} = 0 \), \( \tau_{m} \) can be defined for \( m = 1, \ldots, n+1 \). Its evolution complies with the following lemmas:

**Lemma 1**

*Evolution of \( \tau_{m} \) for \( m = 1, \ldots, n-1 \), follows this rule:*

If \( \mu_{m+1} < \tau_{m} \sqrt{\mu_{m+1}} \) then, \( \tau_{m+1} < \tau_{m} \).

If \( \mu_{m+1} = \tau_{m} \sqrt{\mu_{m+1}} \) then, \( \tau_{m+1} = \tau_{m} \).

If \( \mu_{m+1} > \tau_{m} \sqrt{\mu_{m+1}} \) then, \( \tau_{m+1} > \tau_{m} \).

**Proof**

By definition of \( \tau_{m} \) for \( m = 1, \ldots, n-1 \) (21),

\[
\tau_{m} \sum_{i=1}^{m+1} \sqrt{\mu_i} - \tau_{m} \sqrt{\mu_{m+1}} = \sum_{i=1}^{m+1} \mu_i - \mu_{m+1} - \lambda.
\]

And the rule presented in Lemma 1 simply expresses the following relation:

\[
\mu_{m+1} - \tau_{m} \sqrt{\mu_{m+1}} = (\tau_{m+1} - \tau_{m}) \sum_{i=1}^{m+1} \sqrt{\mu_i}. \quad (22)
\]
Lemma 2
The evolution of the optimality parameter, $\tau_m$, is increasing with $m$ for $1 \leq m \leq m^*$. It takes its maximal value for $m^*$ ($1 \leq m^* \leq n$), which is the smallest index such that:

$$
\{ \sum_{i=1}^{m^*} \mu_i > \lambda, \mu_{m^*+1} \leq \tau_{m^*}^2 \}.
$$

(23)

Then, the value of $\tau_m$ monotonously decreases with $m$ for $m^* < m \leq n + 1$.

Proof
• Non-emptiness of the set of indices defined by (23): For any set of rate parameters $(\lambda, \mu_1, ..., \mu_n)$, stability condition (7) implies $\tau_n > 0$. Then, under convention $\mu_{m+1} = 0$, relations (23) are satisfied for $m^* = n$. The set of indices satisfying (23) is not empty and finite. It has a minimal element: $m^*$.

• Evolution of $\tau_m$:
  The first assertion of Lemma 2 is inoperative if $m^* = 1$.

  If $m^* \geq 2$, consider the values of index $m$ in the range $(1, m^* - 1)$.

  - If $\sum_{i=1}^{m} \mu_i \leq \lambda$, then, $\tau_m = 0$. And $\mu_{m+1} > 0$ implies $\mu_{m+1} > \tau_m \sqrt{\mu_{m+1}}$. Then, from the third assertion of Lemma 1, $\tau_{m+1} > \tau_m$.

  - If $\sum_{i=1}^{m} \mu_i > \lambda$, then, $\tau_m > 0$. For $m < m^*$, non-satisfaction of (23) requires $\mu_{m+1} > \tau_m \sqrt{\mu_{m+1}}$ implying $\tau_{m+1} > \tau_m$.

  - In particular, $\tau_{m^*-1} < \tau_m^*$. And, from the definition of $\tau_m$ (21) at $m^* - 1$ and $m^*$,

    $$
    \mu_{m^*} - \tau_m \sqrt{\mu_{m^*}} = (\tau_m - \tau_{m^*-1}) \sum_{i=1}^{m^*-1} \sqrt{\mu_i}.
    $$(24)

  Therefore,

    $$
    \mu_{m^*} - \tau_m \sqrt{\mu_{m^*}} > 0
    $$(25)

  Now, for $m^* < m \leq n + 1$, $\sum_{i=1}^{m} \mu_i \geq \sum_{i=1}^{m^*} \mu_i > \lambda$. Then, $\tau_m > 0$. From (23), $\mu_{m+1} < \tau_m \sqrt{\mu_{m+1}}$.

  Hence, from Lemma 1,

    $$
    0 < \tau_{m+1} < \tau_m.
    $$

  Replace now $m^*$ by $m^* + 1$ in relation (24) to obtain:

    $$
    \mu_{m^*+1} < \tau_{m^*+1} \sqrt{\mu_{m^*+1}}.
    $$

And thus for $m^* < n$,

    $$
    \mu_{m^*+2} \leq \mu_{m^*+1} < \tau_{m^*+1}^2.
    $$

From Lemma 1, it implies $0 < \tau_{m^*+2} < \tau_{m^*+1}$. Repetition of this argument to successive values of $m$ up to $n + 1$ shows that the value of $\tau_m$ monotonously decreases with $m$ for $m > m^* \geq n + 1$, and finally that the maximal value of $\tau_m$ is obtained for $m^*$.

$\square$

The constrained problem can now be solved using the following proposition:

Proposition 1
Under the feasibility condition $\lambda < \sum_{i=1}^{n} \mu_i$, consider the smallest index, $m^*$, with $1 \leq m^* \leq n$ satisfying conditions (23). Then, the optimal choice of routing parameters satisfies:

$$
\{ \alpha_j = 0 \text{ for } m^* < j \leq n, \\
\alpha_i = \frac{1}{\lambda} (\mu_i - \tau_m \sqrt{\mu_i}) \text{ for } i = 1, ..., m^*.
$$

(26)

Proof
• Feasibility of the proposed policy:
  The set of routing parameters defined in Proposition 1 satisfies:

    $$
    \sum_{i=1}^{n} \alpha_i = \sum_{i=1}^{m^*} \alpha_i = \frac{1}{\lambda} (\sum_{i=1}^{m^*} \mu_i - \tau_m \sum_{i=1}^{m^*} \sqrt{\mu_i}) = 1.
    $$

  If $m^* = 1$ satisfies (23), relations (26) define the set of feasible routing parameters : $(\alpha_1 = 1, \alpha_i = 0 \text{ for } i = 2, ..., n)$.

  If $m^* \geq 2$, since $\mu_i \geq \mu_{m^*}$, for $i = 1, ..., m^*$, relation (25) with $\tau_{m^*} > 0$, implies:

    $$
    \mu_i - \tau_{m^*} \sqrt{\mu_i} > 0 \text{ for } i = 1, ..., m^*.
    $$

  Then, from (26), $\alpha_i \geq 0$ for $i = 1, ..., m^*$.

  Furthermore, the right part of inequality (11) $(\alpha_i \leq \frac{\mu_i}{\lambda} \leq 1)$ is satisfied for $i = 1, ..., m^*$, from $\tau_{m^*} > 0$.

  Therefore, the parameters of Proposition 1 (26), define a feasible solution to optimization problem (CP).

• Optimality among policies satisfying (23) and (26):
  Suppose the existence of an index $k$: $m^* < k \leq n$ also satisfying conditions (23). The selected routing parameters associated to the $k$ routes problem is:

    $$
    \alpha_j = 0 \text{ for } k < j \leq n
    $$

    $$
    \alpha_i = \frac{1}{\lambda} (\mu_i - \tau_m \sqrt{\mu_i}), \text{ for } i = 1, ..., k
    $$

  Denote $J_k = \sum_{i=1}^{n} \alpha_i$ with $\alpha_i$ defined by (28), (29). Replace $\alpha_i$ by its expression to obtain:

    $$
    J_k = \frac{\sum_{i=1}^{k} \mu_i - \lambda}{\lambda \tau_k^2} - \frac{k}{\lambda}
    $$

If both $m^*$ and $k$ satisfy relation (23), use Lemma 2 to obtain $\tau_k \leq \tau_{m^*}$ and the majoration:

$$J_k \geq J_{m^*} + \frac{(k - m^*)\mu_k}{\lambda \tau_{m^*}^2} = \frac{(k - m^*)}{\lambda}$$

Then, $\mu_k \leq \mu_{m^*+1} \leq \tau_{m^*}^2$ implies:

$$J_k \geq J_{m^*}$$

The choice of the smallest value of $m$ for which relations (23) are satisfied is therefore optimal in the class of policies considered in this paragraph.

- Optimality relatively to any other policy

By construction, the policy is optimal if $m^* = n$.

If $m^* < n$, the proposed solution is the best among those satisfying

$$\alpha_j = 0 \text{ for } m^* < j \leq n.$$ 

Under the convexity property of the problem, it now suffices to show that the criterion cannot decrease under any infinitesimal admissible move with $\delta \alpha_j > 0$ for any $j$; $m^* < j \leq n$, starting from solution (26).

$$\delta E(T) = \frac{1}{\mu_j} \delta \alpha_j + \sum_{i=1}^{m^*} \frac{\mu_i}{\mu_i - \lambda} \delta \alpha_i$$  \hspace{1cm} (30)

under the admissibility constraint:

$$\delta \alpha_j + \sum_{i=1}^{m^*} \delta \alpha_i = 0$$  \hspace{1cm} (31)

Using relation $\frac{\mu_i}{\mu_i - \lambda} = \frac{1}{\tau_{m^*}^2}$, relation (30) takes the form:

$$\delta E(T) = \frac{1}{\mu_j} \delta \alpha_j + \frac{1}{\tau_{m^*}^2} \sum_{i=1}^{m^*} \delta \alpha_i$$  \hspace{1cm} (32)

Conditions (23) imply:

$$\mu_j \leq \tau_{m^*}^2 \text{ for } j = m^* + 1, \ldots, n$$  \hspace{1cm} (33)

Hence,

$$\delta E(T) = \left( \frac{1}{\mu_j} - \frac{1}{\tau_{m^*}^2} \right) \delta \alpha_j > 0$$  \hspace{1cm} (34)

□

Note that in particular, if $\mu_1 > \lambda$, $\tau_1 = \frac{\mu_1 - \lambda}{\sqrt{\mu_1}}$. Then, the steady state policy characterized by:

$$\alpha_1 = 1, \quad \alpha_i = 0 \text{ for } i = 2, ..., n.$$  \hspace{1cm} (35)

is optimal if and only if $\mu_2 \leq \tau_1^2$.

3 TOWARDS A REAL-TIME IMPLEMENTATION OF ROUTING PARAMETERS

Once optimal average routing ratios have been computed, they can be implemented either directly through a stochastic or a deterministic state-independent load-balancing policy, or can be used as reference values for building a dynamic load-balancing policy.

Implementation only uses the results of the static optimization scheme. And thus, at this stage, the technique used for computing the average routing ratios does not actually matter. So, in particular, the technique presented in this section can be applied to systems much more complex than the simple one analyzed in section 2.

In terms of mean response time when there is no failure, superiority of cyclical dynamic load-balancing policy versus static load-balancing policy has been shown by several authors [7]. In the particular case of Fig.1 with all service rates equal, superiority of the RoundRobin rule over the Bernoulli splitting technique was theoretically established in Ephremides et al. [3]. Furthermore, cyclical sequencing is a good intermediate stage in the design of a flexible loading device in which sequencing flexibility and routing flexibility can be introduced [8].

3.1 Construction of a cyclical routing device

Routing alternatives are now represented as a free-choice place, labelled 0, in a Petri net representing the system. On Fig.2, a firing of transition $i$ corresponds to a release on route $i$.

![Figure 2: A free-choice Petri net representation](image)

A cyclical firing sequence for routes $1, ..., n$ is obtained by introducing $n$ places for decision tokens and valuated arcs between these places and the route input transitions. To evaluate the average steady state routing
ratios, an infinite feeding of tasks at place 0 can be assumed. Then, place 0 can be suppressed, and only the distribution of tokens determines the system evolution. The logics of such a cyclical loading device are represented on Fig. 3.

![Petri net](image)

**Figure 3:** A Petri net representation of the loading mechanism

The cyclical Petri net characterizing the loading device is represented by the following incidence matrix:

$$
C = \begin{bmatrix}
-p_1 & 0 & \ldots & p_n \\
p_1 & -p_2 & 0 & \ldots \\
\vdots & \vdots & \ddots & \vdots \\
0 & \ldots & \ldots & -p_n
\end{bmatrix}
$$

(36)

The invariant T-semiflot: \( u \in \mathbb{R}^n \) such that \( Cu = 0 \) is given by:

$$
u_i = \frac{\prod_{j \neq i} p_j}{\sum_{k=1}^{n} \prod_{j \neq k} p_j}
$$

(37)

If the Petri net of Fig. 3 is live, the invariant T-semiflot defined by the firing sequence \( u \) characterizes the cyclical stationary behaviour of the Petri net [2]. To obtain the desired loading ratios \( \alpha_1, \ldots, \alpha_n \), it suffices to determine valuations \( p_1, \ldots, p_n \) which satisfy the set of equations:

$$
u_i = \alpha_i \quad \text{for } i=1, \ldots, n
$$

or equivalently, for \( i=1, \ldots, n

$$
\prod_{j \neq i} p_j = \alpha_i P \quad \text{with } P = \prod_{k=1}^{n} \prod_{j \neq k} p_j \text{ and } p_i > 0.
$$

(38)

From expression (37), it is clear that parameters \( p_i \) can be multiplied by any positive constant without changing the routing ratios. Furthermore, for the set of parameters \( (p_1, \ldots, p_n) \) to be used as a set of valuations on the arcs of the control Petri net, it has to be approximated by a set of positive integers. Thus, if system (38) can be solved for a set of positive real parameters \( (p_1, \ldots, p_n) \), the order of magnitude of parameters \( p_i \) has to be selected so as to achieve a trade-off between:

- a sufficient precision in spite of the rounding to the nearest integers
- a "reasonable" number of tokens in the Petri net.

Such a trade-off may depend on the considered application. If precision is not essential, the smallest valuation \( (p_1 \text{ under a decreasing ordering of the } \alpha_i) \) can be set equal to 1. If routing ratios have to be more precisely met, the value of \( p_1 \) should be increased, as in the Example of the following section, where the value of \( p_1 \) is set equal to 10.

To solve the system of equations (38) in terms of real positive parameters, take the natural logarithm of each term of the equations to obtain the following set of linear equalities (with unconstrained real variables and \( P \) arbitrarily set to 1):

$$
Ay = b
$$

(39)

with

$$
A = \begin{bmatrix}
0 & 1 & \ldots & 1 \\
1 & 0 & 1 & 1 \\
\vdots & \vdots & \ddots & \vdots \\
1 & 1 & \ldots & 1
\end{bmatrix}
$$

$$
y^T = [y_1, \ldots, y_n] \text{ with } y_i = \log(p_i)$$

$$
b^T = [y_1, \ldots, y_n] \text{ with } b_i = \log(\alpha_i).
$$

Solution of (39) is \( y = A^{-1}b \), and the corresponding vector of real valuations, \( p = [p_1, \ldots, p_n]^T \) has positive components. It is obtained by:

$$
A^{-1} = \frac{1}{n-1} \begin{bmatrix}
-(n-2) & 1 & \ldots & 1 \\
1 & -(n-2) & \ldots & 1 \\
\vdots & \vdots & \ddots & \vdots \\
1 & 1 & \ldots & -(n-2)
\end{bmatrix}
$$

(40)

For a normalized value of \( p_1 \) equal to \( \pi_1 \) (\( \pi_1 = 1 \) or \( \pi_1 = 10 \) for instance), the vector of integer valuations \( \pi = [\pi_1, \ldots, \pi_n] \) is obtained as:

$$
\pi = \text{round}\left( \frac{\pi_i}{p_1} \right).
$$

(41)

where round(\(_\)) rounds a positive real number to the closest non-negative integer.

Now, the condition to be satisfied to obtain the desired routing ratios is liveness of the Petri net. Assuming that
there is no blocking on any of the \( n \) routes, a necessary and sufficient condition for liveness can be obtained as an extension to a result by Ohl et al. [8] to the case of \( n \) routes. Vector \( x = [1, \ldots, 1]^T \) is an invariant P-semiflot of the control Petri net: \( x^TC = 0 \). Therefore, if \( m_i \) denotes the number of tokens in place \( i \), the total number of tokens in the control Petri net,

\[
N = \sum_{i=1}^{n} m_i
\]

is a constant along the system evolution.

- if \( N \leq \sum_{i=1}^{n} \pi_i - n \), all the transitions may be simultaneously disabled, for some particular markings,
- if \( N > \sum_{i=1}^{n} \pi_i - n \), there is always at least one transition enabled, and since the control Petri net is an elementary cycle, all the transitions are live.

Therefore, the liveness condition is as follows:

*The control Petri net is live if the number of tokens in the control Petri net, \( N \), satisfies:*

\[
N \geq \sum_{i=1}^{n} \pi_i - n + 1. \tag{42}
\]

Even for \( N = \sum_{i=1}^{n} \pi_i - n + 1 \), if \( n > 2 \), several transitions may be simultaneously enabled, and some decisions remain to be taken. And release flexibility increases with \( N \), for \( N > \sum_{i=1}^{n} \pi_i - n + 1 \). This feature is basic for avoiding immediate general blocking when a failure occurs.

### 3.2 Introducing routing flexibility

Under condition (42), an increase of the number of tokens in the control graph allows new state transitions. For some markings, several transitions are simultaneously enabled, and different sequences can be generated. Conflicts can then be solved using priority rules which may be static or may depend on the state of the network. However, under any feasible conflict resolution strategy, the same average routing ratios are satisfied, since the invariant T-semiflot remains the same.

In the queueing network of section 2, the capacity of queues was supposed infinite and the servers were always available. A basic extension of this model consists of introducing resource availability conditions on the firing of transitions \( 1, \ldots, n \), as presented on Fig. 4 by the presence of a token in place \( 1', \ldots, n' \). Then, if a failure occurs, causing a blocking on route \( i \), the availability condition cannot be satisfied and control tokens tend to accumulate at place \( i \). After some time, all the transitions \( j \neq i \) get disabled by a lack of tokens. Such an accumulation of items can be used to detect the failure, and the repair action can start. The process of routing optimization and implementation may also be recomputed. However, it can also be interesting to rapidly react to such a failure by short-circuiting the unavailable route. This can be done by adding new transitions \( \tau_i \) from node \( i \) to node \( i+1 \), with valuated arcs \( q_i \), as on Fig. 4, so that \( \tau_i \) gets enabled if the number of tokens in place \( i \) gets bigger than \( q_i \). If the total number of tokens, \( N > \sum_{i=1}^{n} \pi_i - n + 1 \), is selected large enough to always have at least 2 control places with \( m_i \geq p_i \) (or at least one control place with \( m_i \geq 2p_i \)),

\[
N \geq \sum_{i=1}^{n} \pi_i - n + \pi_n + \pi_{n-1},
\]

a possible choice of \( q_i \) is obtained considering the smallest marking of place \( i \) for which all the routes \( j \neq i \) are possibly disabled by a lack of tokens. It is:

\[
q_i = N - \sum_{j \neq i} \pi_j + n - 1.
\]

![Figure 4: Control device with availability conditions and Supervision](image)

Non liveness of the supervisory control cycle is normal since its transitions should be enabled only in perturbed conditions and on a non permanent basis. Several different techniques can be thought of to increase routing flexibility. The supervisory scheme presented here is just a tentative adaptive version of the proposed control structure. Its interest is mainly to set into evidence the potential flexibility of the valuated cyclical routing technique.

### 4 Example

Consider a system of 5 parallel queues, with exponentially distributed arrival times and service times with mean rates: \( \lambda = 1, \mu_1 = 0.8, \mu_2 = 0.6, \mu_3 = 0.4 \),
\( \mu_4 = 0.2, \mu_5 = 0.1 \).

Computation of the optimality parameter \( \tau_m \) for \( m = 1, \ldots, 5 \) gives: \( \tau_1 = -0.2236, \tau_2 = 0.2397, \tau_3 = 0.3476, \tau_4 = 0.3638, \tau_5 = 0.3589 \).

Then, from Proposition 1, the optimal steady-state Bernoulli splitting follows the dispatching probabilities: \( \alpha_1 = 0.4746, \alpha_2 = 0.3182, \alpha_3 = 0.1699, \alpha_4 = 0.0373, \alpha_5 = 0 \).

In steady state conditions and in the mean, server 5 should not be used. This property remains valid for values of \( \lambda \) less than \( \lambda_5 \simeq 1.131 \). But server 5 should be used with a positive probability for \( \lambda > \lambda_5 \).

For \( \lambda = 1 \), under the Bernoulli splitting policy with routing probabilities \( \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5 \) given below, the mean system response time is given by

\[
E_B(T) = \frac{1}{{\text{N}}} \sum_{i=1}^{4} \frac{\alpha_i}{\mu_i - \alpha_i \lambda} \simeq 3.555
\]

From these values, it is possible to build a deterministic static load balancing policy by constructing a control Petri net as in section 2. Only the first 4 routes are considered. The vector of valuations \( \pi \) is computed by relation (41), after solving system (39). The selected value of \( \pi_1 \) is 10. It gives:

\[
\pi = [10 \ 15 \ 28 \ 127]^T.
\]

Implementation of this control Petri net gives the following vector of average a posteriori routing parameters:

\[
\alpha' = [0.4756 \ 0.3171 \ 0.1699 \ 0.0374]^T.
\]

The condition for liveness of the control Petri net is that the sum of the tokens in places 1, 2, 3, 4, denoted \( N \), satisfies: \( N \geq 177 \).

Constraints on machine availability and the supervisory control structure of Fig.4, presented in section 3.2 can be integrated with, for instance, \( N = \pi_1 + \pi_2 + 2\pi_3 + 2\pi_4 - 4 = 422 \), and the vector of valuations:

\[
q = [255 \ 260 \ 273 \ 372]^T.
\]

5 CONCLUSION

Static load balancing policies are generally much simpler to compute than dynamic ones. However, the concerns for flexibility and reactivity of routing control devices urges to the use of dynamic load balancing policies. In this paper, explicit expressions of the optimal steady-state routing probabilities have been given for the case of parallel \( M/M/1 \) queues with different mean service rates. In particular, a simple test based on the computation of the optimality parameters, \( \tau_m \) gives a threshold value \( (\tau_m^2) \) for the minimal mean service rate of the servers to be used by the policy. Instead of a direct implementation of the optimal steady-state routing parameters as Bernoulli splitting parameters applying to the input flow, it is proposed to implement a cyclical control Petri net achieving optimal steady-state performance. Some simple techniques can then be introduced to increase the natural flexibility of the control device and its adaptivity in case of perturbations.

References


