Processor sharing networks

Samuli Aalto
Department of Communications and Networking
Aalto University

May 7, 2022

1 Processor sharing networks

A *processor sharing network* [4, 9] is a flow-level model of a data network loaded with *elastic traffic* [5, 6] consisting of *flows* (such as file transfers using TCP). Elasticity refers to the property that the transmission rate of a flow is not fixed but it is adjusted according to the congestion state of the system.

A processor sharing network consists of a set of links $j \in \{1, \ldots, J\}$ with capacities C_j (in bits/sec). These link capacities are shared by all flows in the network. Each flow is associated with a class $k \in \{1, \ldots, K\}$. New class-k flows arrive according to a Poisson process with intensity λ_k (in 1/sec). Let B_k denote the size of a class-k flow (in bits). We assume that it is exponentially distributed with mean $E[B_k] = 1/\beta_k$. Let

$$\sigma_k := \lambda_k/\beta_k \tag{1}$$

denote the load of class k (in bits/sec). We assume that all flows in class k follow the same route. Let $A = (a_{k,j})$ denote the route matrix, i.e.,

$$a_{k,j} = \begin{cases} 1, & \text{if link } j \text{ belongs to the route of class } k; \\ 0, & \text{otherwise.} \end{cases}$$

The state of the whole network is described by vector

$$N(t) = (N_1(t), \ldots, N_K(t)),$$

where $N_k(t)$ refers to the number of customers in class k at time t. The state space is clearly

$$\mathcal{S} = \{n = (n_1, \ldots, n_K); n_k \in \{0, 1, \ldots\}\}.$$

In addition, let e_k denote the unit vector to direction k in this space, $e_k = (n_1, \ldots, n_K)$ with $n_k = 1$ and $n_{k'} = 0$ for $k' \neq k$.

In the dynamic setting with a variable number of flows, the first question is whether the whole system is stable or not. Necessary stability conditions are that for all $j \in \{1, \ldots, J\}$,

$$\sum_{k=1}^{K} \sigma_k a_{k,j} < C_j. \tag{2}$$

Whether these necessary conditions are also sufficient depends on how the link capacities are allocated to the flows. There have been many proposals how to allocate the resources to the flows in a fair way. The classical fairness concept is *max-min fairness* [1]. *Proportional fairness* was introduced in [2, 3] and *potential delay minimization* in [8]. All these allocation schemes belong to the family of α-*fair* allocations [7]. It has been shown that these conditions are also sufficient for α-fair allocations [5]. In the following sections we introduce still another fairness concept, called *balanced fairness* [9, 10, 11, 12, 13], for which the necessary stability conditions are sufficient, as well.

Let $\phi_k(n)$ denote the *inter-class allocations* (in bits/sec), i.e., how the link capacities are shared among the flow classes, which may depend on the state $n \in \mathcal{S}$ of the system. If $n \in S$ is such that $n_k = 0$, we assume that $\phi_k(n_k) = 0$. The inter-class allocations are *feasible* if, for all links j,

$$\sum_{k=1}^{K} \phi_k(n) a_{k,j} \leq C_j. \tag{3}$$

The family of all feasible allocations is called the *capacity set*, and it is denoted by $\mathcal{C}$,

$$\mathcal{C} := \{c = (c_1, \ldots, c_K) \geq 0 : \sum_{k=1}^{K} c_k a_{k,j} \leq C_j \ \forall j \in \{1, \ldots, J\}\}. \qquad (4)$$

The *intra-class* allocations determine how the inter-class allocations $\phi_k(n)$ are shared among the flows within the same class. In a processor sharing network, the intra-class allocations are assumed to be *fair* in the sense that each flow in class k gets an equal share denoted by

$$\psi_k(n) := \frac{\phi_k(n)}{n_k}, \quad n \in \mathcal{S},\ k \in \{1, \ldots, K\}. \qquad (5)$$

Due to the exponential assumptions made above, $N(t)$ is an irreducible Markov process with the following (positive) state transition rates for any $n \in \mathcal{S}$ and $k \in \{1, \ldots, K\}$:

$$\begin{aligned} q(n, n + e_k) &= \lambda_k, \\ q(n + e_k, n) &= \beta_k \phi_k(n + e_k). \end{aligned}$$

2 Whittle networks

A processor sharing network is a *Whittle network* if

(i) the inter-class allocations $\phi_k(n)$ are feasible for any state $n \in \mathcal{S}$ and

(ii) there is a positive function $\Phi(n)$ defined on $\mathcal{S}$ such that $\Phi(0) = 1$ and the inter-class allocations satisfy

$$\phi_k(n + e_k) = \frac{\Phi(n)}{\Phi(n + e_k)}, \quad n \in \mathcal{S},\ k \in \{1, \ldots, K\}. \qquad (6)$$

Function $\Phi(n)$ is called the corresponding *balance function*.

Proposition 1

For a Whittle network the inter-class allocations are balanced in the following sense: for any $n \in \mathcal{S}$ and $k, k' \in \{1, \ldots, K\}$

$$\frac{\phi_k(n + e_k + e_{k'})}{\phi_{k'}(n + e_k + e_{k'})} = \frac{\phi_k(n + e_k)}{\phi_{k'}(n + e_{k'})}. \tag{7}$$

Below we show that the steady-state distribution of a stable Whittle network is of *product-form* [4, 9].

Theorem 1

Consider a stable Whittle network, and let $\Phi(n)$ denote the corresponding balance function. The steady-state distribution of process $N(t)$ is given by

$$P\{N = n\} = \frac{\Phi(n)}{G} \prod_{k=1}^{K} \sigma_k^{n_k}, \quad n \in \mathcal{S},$$

where the normalization constant G is defined by

$$G := \sum_{n' \in \mathcal{S}} \Phi(n') \prod_{k=1}^{K} \sigma_k^{n'_k}.$$

Proof Let $n \in \mathcal{S}$ and denote

$$\pi(n) := \frac{\Phi(n)}{G} \prod_{k=1}^{K} \sigma_k^{n_k},$$

where G is the normalization constant defined above. Since the system is stable, we know that $\pi(n)$ is a proper distribution with $G < \infty$. In other words, the normalization condition (N) is satisfied. It remains to prove that the global balance equations (GBE) are also satisfied for any $n \in \mathcal{S}$:

$$\sum_{n' \neq n} \pi(n) q(n, n') = \sum_{n' \neq n} \pi(n') q(n', n). \tag{8}$$

Let $n \in \mathcal{S}$. For any $k \in \{1, \ldots, K\}$, we have recursion

$$\pi(n + e_k) = \pi(n) \frac{\Phi(n + e_k)}{\Phi(n)} \sigma_k = \pi(n) \frac{\lambda_k}{\beta_k \phi_k(n + e_k)}, \tag{9}$$

where the last equality follows from (1) and (6). Thus, by (9),

$$\pi(n)\lambda_k = \pi(n + e_k)\beta_k\phi_k(n + e_k),$$

which is equivalent with

$$\pi(n)q(n, n + e_k) = \pi(n + e_k)q(n + e_k, n). \tag{10}$$

From this we observe that the *detailed balance equations* (DBE) are satisfied for any $n \in \mathcal{S}$ and $k \in \{1, \dots, K\}$. The global balance equations (8) follow from these detailed balance equations in a straightforward way by summing up the related DBE's. $\square$

As in a single-server M/G/1-PS queue, it can be shown that the steady-state distribution of a stable Whittle network is *insensitive* to the flow size distributions (as long as the mean flow sizes $1/\beta_k$ remain the same for all classes k) [9, 10].

3 Balanced fairness

Consider a Whittle network. Let $n \in \mathcal{S}$. It follows from (6) that for any k such that $n - e_k \in \mathcal{S}$ we have

$$\phi_k(n) = \frac{\Phi(n - e_k)}{\Phi(n)},$$

where $\Phi(n)$ refers to the corresponding balance function. Thus, to get feasible balanced allocations, we have to require that

$$\left(\frac{\Phi(n - e_1)}{\Phi(n)}, \dots, \frac{\Phi(n - e_K)}{\Phi(n)}\right) \in \mathcal{C}.$$

Balanced fairness refers to the case where these feasible balanced allocations are maximized.

Balanced allocations are *balanced fair* (BF) if the corresponding balance function $\Phi(n)$ is constructed recursively as follows: let $\Phi(n) = 0$ for any $n \notin \mathcal{S}$, $\Phi(0) = 1$, and for any $n \in \mathcal{S} \setminus \{0\}$

$$\Phi(n) = \min\left\{\alpha > 0 : \left(\frac{\Phi(n - e_1)}{\alpha}, \ldots, \frac{\Phi(n - e_K)}{\alpha}\right) \in \mathcal{C}\right\}. \qquad (11)$$

It is easy to see that these balanced fair allocations are unique for the considered Whittle network. In addition, the necessary stability conditions are also sufficient for balanced fair allocations as shown in [10].

Proposition 2

Consider a Whittle network with balanced fair allocations. The system is stable if and only if the stability conditions given in (2) are satisfied that for all links $j \in \{1, \ldots, J\}$.

For a stable Whittle network with balanced fair allocations, the steady-state distribution is of product-form, see Theorem 1. As the final result, we give an alternative recursion formula for the balance function $\Phi(n)$ of the balanced fair allocations.

Proposition 3

Consider a Whittle network with balanced fair allocations. The balance function $\Phi(n)$ satisfies the following recursion for any $n \in \mathcal{S} \setminus \{0\}$:

$$\Phi(n) = \max_{j \in \{1,\ldots,J\}} \frac{1}{C_j} \sum_{k=1}^{K} \Phi(n - e_k) a_{k,j}. \qquad (12)$$

Proof Let $n \in \mathcal{S} \setminus \{0\}$ and $j \in \{1, \ldots, J\}$. Since, we have, for any $k \in \{1, \ldots, K\}$,

$$\phi_k(n) = \frac{\Phi(n - e_k)}{\Phi(n)},$$

it follows from the capacity constraints (3) that

$$\sum_{k=1}^{K} \frac{\Phi(n-e_k)}{\Phi(n)} a_{k,j} \leq C_j.$$

Thus,

$$\Phi(n) \geq \frac{1}{C_j} \sum_{k=1}^{K} \Phi(n-e_k) a_{k,j},$$

which shows that

$$\Phi(n) \geq \max_{j \in \{1,\ldots,J\}} \frac{1}{C_j} \sum_{k=1}^{K} \Phi(n-e_k) a_{k,j}.$$

On the other hand, by (11), we are looking for the smallest possible value satisfying this inequality, which justifies our claim (12). □

4 Performance

Consider a stable Whittle network with balanced fair allocations. The traffic consists of elastic flows, such as file transfers using TCP, with each flow characterized by its size, i.e., the total amount of bits to be transferred. An important performance measure for such elastic flows is the total time needed for transferring all the bits, which is called *flow-level delay*, or just briefly, delay. Below we show how to calculate the mean steady-state delay. In addition, we consider the average bit rate of a flow, which is defined to be the mean flow size divided by the mean delay.

Let T_k denote the steady-state delay for class k. It follows from Little's formula that

$$E[T_k] = \frac{E[N_k]}{\lambda_k},$$

where N_k refers to the steady-state number of flows in class k. Let us define vector

$$\sigma := (\sigma_1, \ldots, \sigma_K).$$

In addition, we consider the normalization constant G as a function of this vector σ,

$$G(\sigma) := \sum_{n \in \mathcal{S}} \Phi(n) \prod_{k=1}^{K} \sigma_k^{n_k}.$$

From Theorem 1, we get

$$E[N_k] = \frac{\sigma_k}{G(\sigma)} \sum_{n \in \mathcal{S}} (n_k + 1) \Phi(n + e_k) \prod_{i=1}^{K} \sigma_i^{n_i}.$$

It follows that

$$E[N_k] = \frac{\sigma_k}{G(\sigma)} \frac{\partial G(\sigma)}{\partial \sigma_k}, \tag{13}$$

and, by Little's formula,

$$E[T_k] = \frac{E[B_k]}{G(\sigma)} \frac{\partial G(\sigma)}{\partial \sigma_k}, \tag{14}$$

In addition, the *average bit rate* of a class-k flow (in bits/sec) is given by

$$\gamma_k := \frac{E[B_k]}{E[T_k]} = \frac{G(\sigma)}{\frac{\partial G(\sigma)}{\partial \sigma_k}}. \tag{15}$$

Finally, let $X := N_1 + \ldots + N_K$ denote the steady-state total number of flows and T the steady-state flow-level delay for an arbitrary flow. Clearly, we have

$$E[X] = \sum_{k=1}^{K} \frac{\sigma_k}{\gamma_k}, \quad E[T] = \frac{1}{\lambda} \sum_{k=1}^{K} \frac{\sigma_k}{\gamma_k}, \tag{16}$$

where $\lambda := \lambda_1 + \ldots + \lambda_K$.

References

[1] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed., Prentice Hall, 1992

[2] F. Kelly, Charging and rate control for elastic traffic, *European Transactions on Telecommunications* **8**, 33–37, 1997

[3] F. Kelly, A. Maulloo, and D. Tan, Rate control for communication networks: Shadow prices, proportional fairness and stability, *Journal of the Operational Research Society* **49**, 237–252, 1998.

[4] R. Serfozo, *Introduction to Stochastic Networks*, Springer, 1999

[5] L. Massoulié and J. Roberts, Bandwidth sharing and admission control for elastic traffic, *Telecommunication Systems* **15**, 185–201, 2000

[6] S. Ben Fredj, T. Bonald, A. Proutière, G. Régnié and J. Roberts, Statistical bandwidth sharing: A study of congestion at flow level, in *Proc. of ACM SIGCOMM*, pp. 111-122, 2001

[7] J. Mo and J. Walrand, Fair end-to-end window-based congestion control, *IEEE/ACM Transactions on Networking* **8**, 556–567, 2000

[8] L. Massoulié and J. Roberts, Bandwidth sharing: Objectives and algorithms, *IEEE/ACM Transactions on Networking* **10**, 320–328, 2002

[9] T. Bonald and A. Proutière, Insensitivity in processor-sharing networks, *Performance Evaluation* **49**, 193–209, 2002

[10] T. Bonald and A. Proutière, Insensitive bandwidth sharing in data networks, *Queueing Systems* **44**, 69–100, 2003

[11] T. Bonald, A. Proutière, J. Roberts and J. Virtamo, Computational aspects of balanced fairness, in *Proc. of ITC 18*, pp. 801–810, 2003

[12] T. Bonald and A. Proutière, On performance bounds for balanced fairness, *Performance Evaluation* **55**, 25–50, 2004

[13] T. Bonald, L. Massoulié, A. Proutière and J. Virtamo, A queueing analysis of max-min fairness, proportional fairness and balanced fairness, *Queueing Systems* **53**, 65–84, 2006