Article 16. Myopic model of resource sharing in a queuing network

T.B. RAVALIMINOARIMALALASON[1], M. RAKOTOMALALA[1]

[1]ED STII « Sciences cognitives et Applications » - Ecole Supérieure Polytechnique d'Antananarivo – Université d'Antananarivo

Mail : tokybaz@gmail.com

Abstract

Resources in a queuing network are often rare and expensive. Sharing them to customers of the network is a hard problem and impacts a lot the performances of the system. We present in this paper a new optimal and feasible method to share these resources based on the result of the game theory. We use of the theory of bargaining game and repeated stochastic game to find an optimal consensus for resources sharing based on customer aspirations. Our model, which we call "myopic model", proposes a system of myopic players who don't project future payoffs of the game, and which is a rather dynamic model depending on the evolution of customer aspirations over the time. Simulations of our hypothesis were performed by analyzing performances of our system compared to First-In First-Out queue, standard Processor Sharing model, and one using Kalai-Smorodinsky bargaining solution. The results have allowed us to appreciate ours, mostly in an instability phenomenon, and to propose a variety of applications, mainly in any system with shared resources, such as distributed systems, centralized systems as cloud computing, Search Engine Optimization algorithms, ...

Index terms : Game theory, Network, Queue, Resource, Sharing

I. Introduction

A processor sharing queue is a particular one where the word queue is a misnomer. All existing customers ahead the servers are immediately served with a part of its available resource. Each of them leaves the queue to move to another one after receiving service he has requested from server, whence concept of queuing network comes.

Contribution we provide in this paper is how to share server resources between those customers. Our method is based on the game theory, especially a model of myopic players which optimize only their current gain.

II. Model of customers and queuing network

A. Principle

On a given queue, customers arrive and others leave. Each customer needs a total of resource b_i to fulfill his requirement that he asked to server. In the following, we put that these resources are sampled and shareable in order to work in the discrete domain. At the beginning of each time interval t, each player i sends his strategy s_i^t for the bargaining of the resource C of the server. We restrict to a finite countable strategies set S_i . The server will evaluate these proposals to compute the resources c_i^t that he will assign to the players. Each resource c_i^t has a price p_i^t that the server sends with the resource to the player. Once received, each resource will be used by each player and they will calculate their gains. They also assess the remains of their respective requirement function of the consumed resources. We precise that the player requirement at the beginning of time t was assessed at the end of time $t-1$ after using the resource c_i^{t-1} allocated at this time.

As we consider the mobility of the players, each of them will decide from his requirement whether he will stay in his current queue, or move to the next queue. Initially, when the customer i arrives in a queue, the first requirement is noted b_i^0. Over time, depending on the allocated resources, that requirement becomes b_i^t as :

$$b_i^t = b_i^{t-1} - c_i^{t-1} \quad (1)$$

The decision of the customer is determined by (12).

$$dec(b_i^t) = \begin{cases} stay & if\ b_i^t > 0 \\ move & if\ b_i^t \leq 0 \end{cases} \quad (2)$$

This principle is illustrated on figure 1.

B. The game formulation

Let's model these actions and movements by a stochastic game. This is a stochastic game between N^t players. The number of players N^t varies over time as players arrive to or depart from the queue according to their requirements. At time t, for player i, the local state of the game is defined by the requirement b_i^t. By its finite cardinal, let's note B_i the set of possible requirements of player i. For player i, let's put s_i^t the strategy that he proposes to the server. The set of possible proposals, noted $S_i = \{s_i^t\}$, is also the set of possible strategies for that player.

Let $g_i^t : B_i \times S_1 \times \ldots \times S_{N^t} \to \mathbb{R}$ be a function gain for the player, function of the local state b_i^t and the joint strategies $s^t = (s_1^t, \ldots, s_{N^t}^t)$. We evaluate this gain from the allocated resources, which are themseleves based on price proposals s^t done by all players.

The transition model between local states is defined by the function $T_i^t : B_i \times S_1 \times \ldots S_{N^t} \times B_i \to [0,1]$ (3) :

$$\sum_{b_i^{t+1} \in B_i} T_i(b_i^t, s^t, b_i^{t+1}) = 1 \quad (3)$$

Since the requirements (local states) b_i^{t+1} and b_i^t are dependent, and are also function of the joint strategy s^t , we can say that this function can be well defined. We will further evaluate this transition function. From these data, it is possible to model the actions and movements of the players with a stochastic game defined by the quintuplet $(N^t, (B_i), (S_i), (g_i), (T_i))$.

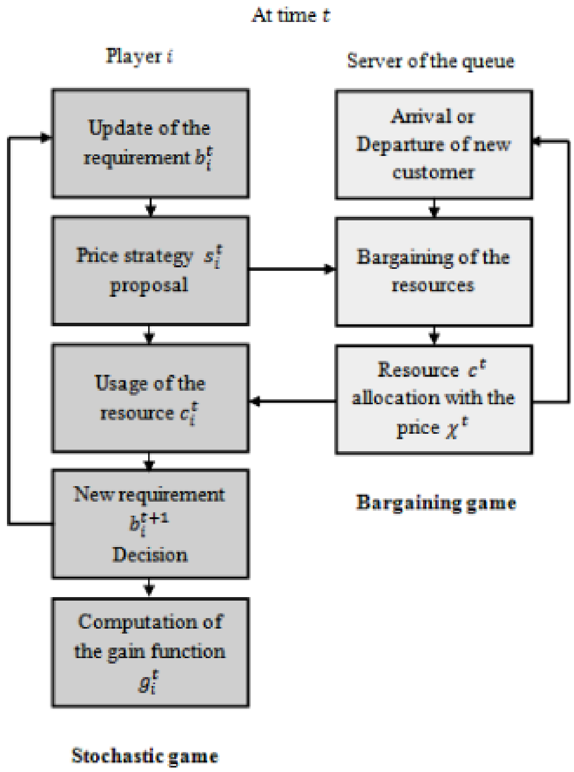

Figure 1. Games at time t

C. *Bargaining of the resource of the server*

Given a resource C of the server, it will be bargained through the customers of the queue, here called as players. At time t, each player must maximize his gain on the possible proposals set P_i as shown on (4).

$$g_i^t(p_i^t, \mu^t) = u_i\left(\frac{p_i^t}{\mu^t}\right) - p_i^t \quad (4)$$

On (4), the function $u_i(c_i^t)$ is the utility function of player i regarding the resource c_i^t that the server has allocated after the bargaining computation. This function must be concave as described on paragraph II. And to better assess the allocated resource, it is necessary that this utility function u_i is also function of the requirement b_i^t. We can use, for example, a logarithmic or quadratic valuation given in (5) and (6).

$$u_i(c_i^t, b_i^t) = A\frac{\log(c_i^t)}{\log(b_i^t)}, \qquad u_i(c_i^t, b_i^t) = A - B\left(\frac{c_i^t}{b_i^t}\right)^2 \quad (5), (6)$$

where A and B are arbitrary positive constants. The proof of the concavity of these functions comes from the negativity of their second derivatives.

Players send to server their proposals $p^t = (p_1^t, \dots, p_{N^t}^t) \in \mathbb{R}^{N^t}$. Once received by the server, it computes the resource to allocate $c^t = (c_1^t, \dots, c_{N^t}^t)$. On (4), the price μ^t is not yet known beforehand, so the players are not able to compute the optimal proposal p_i^t. The function gain that he must maximize is defined by (7) assuming that $\sum_{j \neq i} p_j^t$ is constant.

$$g_i^t(p_i^t) = u_i\left(\frac{p_i^t}{\sum_j p_j^t} C\right) - p_i^t \quad (7)$$

By cancelling the derivative with respect to p_i^t :

$$(g_i^t)' = u_i'\left(\frac{p_i^t}{\sum_j p_j^t} C\right) . \left(1 - \frac{p_i^t}{\sum_j p_j^t}\right) . C - 1 = 0 \quad (8)$$

Let's note $d_i^t = \frac{p_i^t}{\sum_j p_j^t} C$. So, we have :

$$u_i'(d_i^t).\left(1-\frac{d_i^t}{C}\right)=\frac{1}{C} \quad (9)$$

Let's call d_i^t the strategy of player i, that he sends to server at time t. This strategy will be assessed from (9) independently of the other players. So, once the joint strategy $s^t = (d_1^t, \dots, d_{N^t}^t)$ is received by the server, it computes the bargaining. As $d_i^t = \frac{p_i^t}{\mu^t}$, the price p_i^t to perform this computation can be derived (by the server) from the strategy d_i^t sent by the player i with some factor μ^t (the elementary price of resource) that he enforces.

D. Study of stochastic game of player i

1. Model of state transition

Player i plays a stochastic game defined by a quintuplet $(N^t, (B_i), (S_i), (g_i), (T_i))$. To model the dynamism of player i at time t, we defined his local state as his requirement b_i^t. The state change process is very clear after using the resource c_i^t allocated by the server. The transition from the state b_i^t to another state b_i^{t+1} means that the allocated resource at time t is equal to $b_i^{t+1} - b_i^t$:

$$b_i^{t+1} = b_i^t - c_i^t \quad (10)$$

The probability of transition from the state b_i^t to another state b_i^{t+1} can be assessed as the probability that the allocated resource at time t is $b_i^{t+1} - b_i^t = c_i^t$. The model $T_i^t : B_i \times S_1 \times \dots S_{N^t} \times B_i \to [0,1]$ of the state transition is expressed by (11).

$$T_i(b_i^t, s^t, b_i^{t+1}) = \begin{cases} 1 & if\ b_i^t - b_i^{t+1} = C\dfrac{d_i^t}{\sum_j d_j^t} \\ 0 & otherwise \end{cases} \quad (11)$$

Where joint strategy of all players is $s^t = (d_{1_t}, \dots, d_{N^t}^t)$, and $C\frac{d_i^t}{\sum_j d_j^t}$ means the allocated resource to player i at time t. The gain owned by that player at this time is expressed by (12).

$$g_i^t(d_i^t) = u_i(d_i^t) - \mu^t d_i^t \quad (12)$$

2. Impact of game history and its future

As we are faced to a repeated game, players can use history of the game to bargain the resource of the server. Let $h^t = \{b^1, d^1, \mu^1, \dots, b^{t-1}, d^{t-1}, \mu^{t-1}, b^t\} \in H^t$ where b^t indicates the requirements at time t, d^t the proposals sent to the server at this time, μ^t the elementary price enforced by the server, and H^t the set of possible history for this game. The game history that the player i can observe is called observation of player i that we denote o_i^t . This observation is limited because the player i is not able to observe some part of the history of his requirements, his proposals, and the elementary price sent by the server due to lack of memory. So, we have $o_i^t \subset h^t$. It is also required that player can observe his current state $b_i^t : b_i^t \in o_i^t$. Let's denote O_i^t the set of possible

observations up the time t for player i. Considering these observations, the player i can adjust his way of calculating his proposal at time t. Let's call it as policy of player i at time t, denoted π_i^t.differs from a simple strategy d_i^t by using the observations up the time t.

$$\begin{aligned} \pi_i^t &: O_i^t \to A_i \\ o_i^t &\mapsto a_i = \pi_i^t(o_i^t) \end{aligned} \quad (13)$$

In (13), A_i indicates the set of possible proposals. Equation (13) mentions that policies π_i^t are function of observation of player i at time t. So, the player i has to determine a policy which can ensure a best response. The better is to find a time-independent one that we call a stationary policy π_i, so as to be usable at any time by simply basing to observations. We can get it because the current requirement b_i^t depends only on the previous one b_i^{t-1} and the allocated resource at this time that is function of the proposal d_i^{t-1} as given by (11). In that case, the policy is markovian.

Let $\pi = (\pi_i, \pi_{-i})$ be the joint stationary policy. The gain $g_i^k(b_i^k, d_i^k)$ owned by the player i at step k is discounted by a factor δ^{k-t} at time t, and the total gain owned by this player at time t with the policy π is denoted as $G_i^t(b_i^t, \pi)$. This gain is expressed by the recurrent relation (14).

$$G_i^t(b_i^t, \pi) = g_i^t(b_i^t, d_i^t) + \delta \sum_{b_i^{t+1} \in B_i} T(b_i^{t+1} | b_i^t). G_i^{t+1}(b_i^{t+1}, \pi) \quad (14)$$

The policy π_i^* which ensures the best response for player i is given by (25).

$$\pi_i^*(\pi_{-i}) = \underset{\pi_i}{\text{Argmax}}\, G_i^t\left(b_i^t, (\pi_i, \pi_{-i})\right) \quad (15)$$

The problem is how player i can find this optimal policy π_i^*.

Equation (15) shows that the policy of player i ensuring the best response depends on other players policies, that is a function of other players states. Player i doesn't know other players states, so he doesn't able to compute his optimal policy. However, he can optimize only his immediate gain $g_i^t(b_i^t, d_i^t)$. The expected gain is therefore discounted by a factor $\delta = 0$. In that case, the proposal strategy d_i^t coincides with the optimal policy π_i^*. Let's call myopic policy this

policy which doesn't consider or ignore the impact of the future. The optimal policy is denoted π_i^m, which is a time independent function of a single variable, and depends only on the current state of player i.

3. *Performances of the model*

We can use two different measures to evaluate the performance of our model: the total gain (with discount) that each player must maximize, and the sojourn time that each player must minimize. The total gain owned at time t is given by (16).

$$G_i^t(b_i^t, \pi^m) = g_i^t(b_i^t, d_i^t) \tag{16}$$

Where the proposal d_i^t and the function $\pi^m(b_i^t)$ coincide.

If player i arrives to the queue at time t_i^i , the gain owned at a time t ($t \geq t_i^i$) is discounted by a factor $\delta^{t_i^i - t}$ during the calculation of the total gain. The player i will move from this queue at time t_i^f as (17).

$$t_i^f = \min_t \{ b_i^t \geq 0, t > t_i^i \} \tag{17}$$

The sojourn time for player i is expected by (18).

$$t_i^s = t_i^f - t_i^i \tag{18}$$

The total gain for player i is expressed by (19).

$$G_i^T = \sum_{t=t_i^i}^{t_i^f} \delta^{t_i - t} G_i^t(b_i^t, \pi^m) = \sum_{t=t_i^i}^{t_i^f} \delta^{t_i - t} g_i^t(b_i^t, \pi^m) = \sum_{t=t_i^i}^{t_i^f} \delta^{t_i - t} (u_i(\pi^m) - \mu^t \pi^m) \tag{19}$$

We can also use the expected total gain to penalize the player in term of time, by making him able to own more gain if he doesn't stay longer on the queue. The expected total gain is expressed by (20).

$$\overline{G_i^T} = \frac{G_i^T}{t_i^f - t_i^i} = \frac{\sum_{t=t_i^i}^{t_i^f} \delta^{t_i - t} (u_i(\pi^m) - \mu^t \pi^m)}{t_i^f - t_i^i} \tag{20}$$

III. Simulation evaluation and analysis

To evaluate our model, we tried to implement our model on queuing networks who convey packets simulated on OPNET Modeler software. We compared the established model to other models to know his performance.

As described in Figure 2, simulations consist of FIFO (First in First Out) queue, Classic PS (Processor Sharing) queue, Processor sharing queue using the KSBS (Kalai-Smorodinsky Bargaining Solution) and Our myopic players model.

"Source" sends same packets to "Dest MYOP", "Dest KSBS", "Dest FIFO" and "Dest PS" through the queues. The links between these entities feel no packet propagation delay or propagation error.

Simulations are based on the following parameters:

- The inter-arrival of packets T (time between successive generations of packet at "Source") has an exponential distribution parameter $\lambda = 1$ second
- The μ packet sizes generated by "Source" as an exponential distribution with parameter 1024 bits.
- The processing capacity C of the server of each queue is fixed. This capacity, expressed in bits per second (bps), is identical for all four queues of the system.

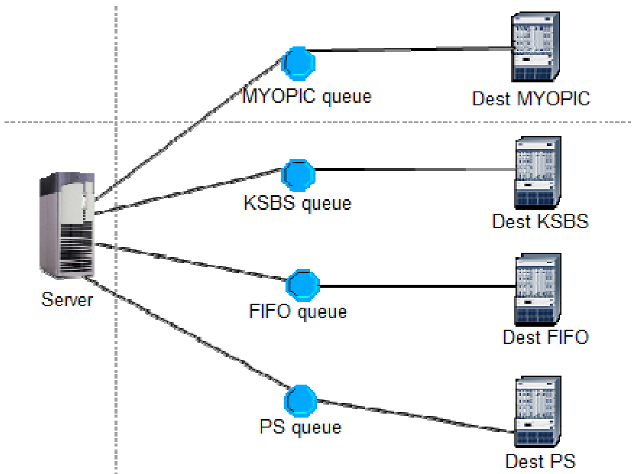

Figure 2. Simulation description

A. *Stable system*

Let's consider a stable system, where the capacity of server is greater than the load rate of the queue. We used $C = 1100$ bps for the simulation. During 10 minutes of simulation, we get the results below.

We find on Figure 3 similar properties of the PS queue and our MYOPIC queue. FIFO queue and the egalitarian solution have more packets queuing compared to PS and MYOPIC.

As in Figure 4, the average sojourn time is almost identical for PS queue and MYOPIC queue. There is a stable difference around 0.7 second between them, where the sojourn time for PS queue is greater than the one for MYOPIC queue.

In Figure 5, the system provides identical throughput. The number of packets per second which come out of each queue is almost the same after a long time of simulation.

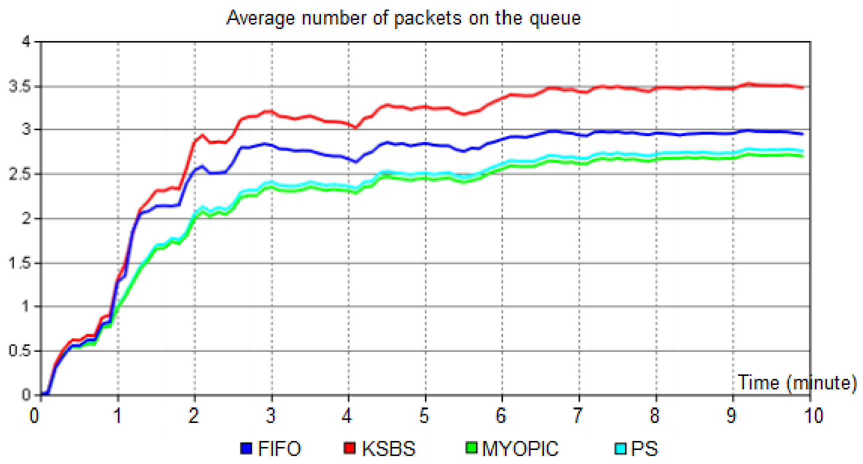

Figure 3. Evolution of the average number of packets on each queue on a stable system

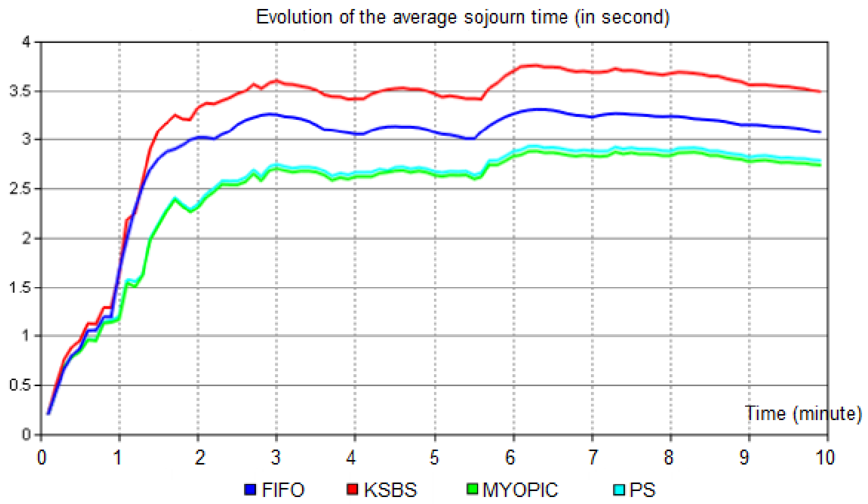

Figure 4. Evolution of the average sojourn time on each queue on a stable system

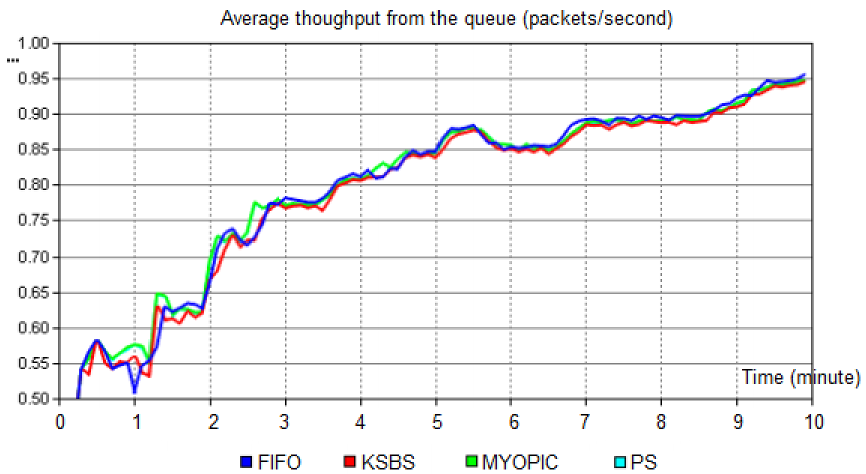

Figure 5. Evolution of the throughput from each queue on a stable system

B. Unstable system

Now, let's consider an unstable system, where the capacity of the server is lower than the load rate of the queue. For that, we used $C = 900$ bps. In Figure 6, at the 10th minute, we already find that the system is unstable; the number of packets on each queue is increasing but the lowest is shown by the MYOPIC queue. Contrary to a stable system, the difference between PS queue and MYOPIC queue performance is highlighted, the two curves diverge.

Till the 10th minute, we can read on Figure 6 that in average:

- 33.45 packets are found processed on the MYOPIC queue,
- 35.12 packets are found processed on the PS queue,
- 36.71 packets are found queuing on the FIFO queue,
- 41.64 packets are found queuing on the KSBS queue.

In term of average sojourn time, sojourn on a MYOPIC queue is lowest compared to the other scheduling method. We can interpret it as a low latency in practice.

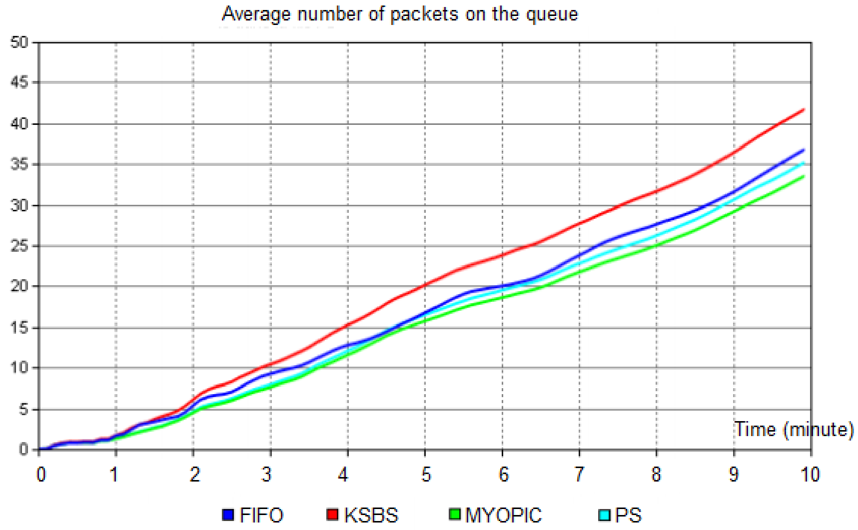

Figure 6. Evolution of the average number of packets on each queue on an unstable system

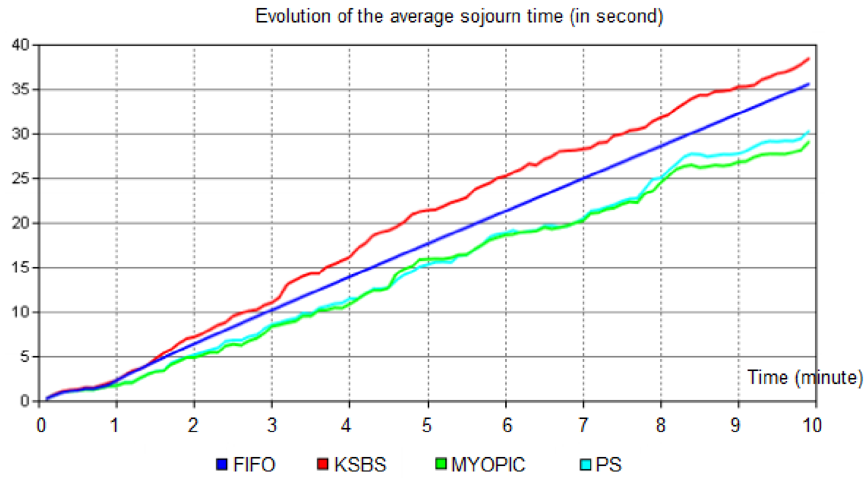

Figure 7. Evolution of the average sojourn time on each queue on an unstable system

Node name	Total
Dest FIFO	495
Dest KSBS	488
Dest MYOPIC	501
Dest PS	496
Source	507

Table 1. Number of packets at each destination at the end

Table 1 shows the number of packets received by each destination after the 10 minutes of simulation. It puts evidence the difference of the average throughput as in Figure 8.

Let's precise that the Source sends the same number of packets at the same time with a same distribution, but this big difference is due to the scheduling and processing on each queue.

We can say that our MYOPIC system can better manage the packets in case of instability compared to the classic PS queue (e.g: in case of temporary congestion).

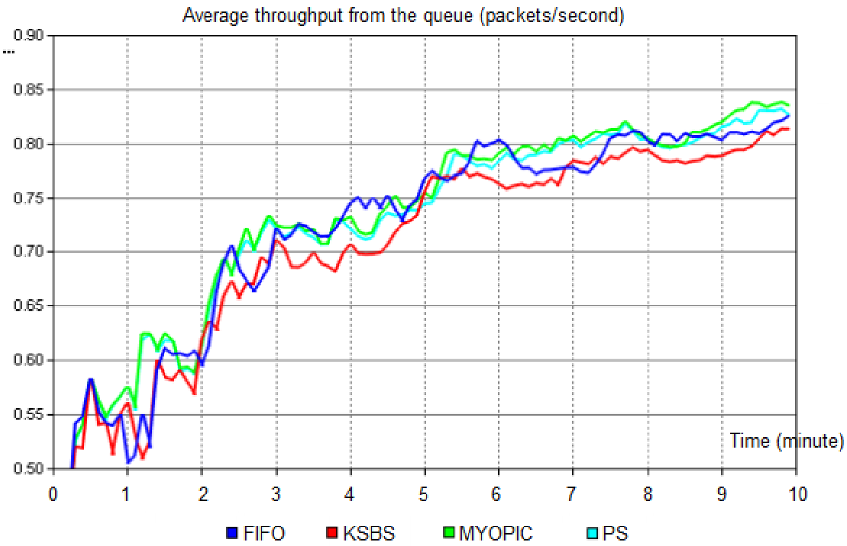

Figure 8. Evolution of the average throughput from each queue on an unstable system

IV. Conclusion

Our contribution consists of a new way to manage the resource of queue. Our methodology is based on a repeated stochastic bargaining game to share the resources of a queuing network. We introduced a new principle of a myopic player who doesn't optimize his future gain by the history of the game. The simulation shows the performance of our model which has a better scheduling during an instability period.

References

[1] J.F. Nash, "The Bargaining Problem", *Econometrica*, vol. 18, 1950, pp. 155-162

[2] T.B. Ravaliminoarimalalason, H.Z. Andriamanohisoa, F. Randimbindrainibe, "Modélisation de partage de ressources dans un réseau de files d'attente à processeur partagé par la théorie des jeux de marchandage. Résolution algorithmique.", *MADA-ETI*, vol. 2, 2014, pp 28-39,

[3] R. Johari, "The price of anarchy and the design of scalable resource allocation mechanisms", *Algorithmic game theory*, Cambridge University Press, 2007, pp. 543-568

[4] G.J. Mailath, L. Samuelson, *Repeated Games and Reputations : Long-run relationships*, Oxford University Press, 2006

[5] J.-F. Mertens, S. Sorin, S. Zamir, "Repeated games", *Core D.P.* - Université Catholique de Louvain, 1994, pp. 20-22

[6] D. Abreu, "Theory of Infinitely Repeated Games with Discounting", *Econometrica*, vol. 56, 1988, pp. 383-396

[7] L.S. Shapley, "Stochastic games", *Proceedings of the National Academy of Sciences of the U. S. A.*, vol. 39, 1953, pp. 1095-1100

[8] A. Neyman, S. Sorin, "Stochastic games and applications", *NATO Science Series*, Kluwer Academic Publishers, 2003

[9] A.M. Fink, "Equilibrium in a stochastic n-person game", *Journal of Sciences of Hiroshima University*, vol.28, 1964, pp. 89-93

[10] K.A. Chew, *Introduction to OPNET Modeler*, Center for Communication Systems Research, University of Surrey - Guildford - UK, 2002

[11] T.B. Ravaliminoarimalalason, *Application de la théorie des jeux au partage des ressources d'un réseau de files d'attente*, Thèse de Doctorat, Université d'Antananarivo, Dec. 2016

[12] E. Kalai, M. Smorodinsky, "Other Solutions to Nash's Bargaining Problem", *Econometrica*, vol. 43, 1975, pp. 513-518