Repeated stochastic bargaining game to share queuing network resources

T. B. Ravaliminoarimalalason, H. Z. Andriamanohisoa, F. Randimbindrainibe

Abstract— **Resources in a queuing network are very rare and expensive. Sharing them to the customers of the network is a hard problem, and it 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, especially the repeated stochastic bargaining game. Our model, which we call "myopic model", is tougher even in an instability phenomenon.**

Index Terms—**Bargaining game, Queue, Repeated game, Stochastic game.**

I. Introduction

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

The contribution we provide in this paper is how to share the server resources between those customers. Our method is based on the game theory, especially a repeated stochastic bargaining game. In this way, we propose a model of myopic player which optimizes only his current gain.

II. GAME THEORY ELEMENTS

A. A bargaining game

Let's briefly review our model using a bargaining game. It is a non-cooperative game, among strategic game, which opposed *n* players sharing a resource *C*. Assuming that each player *i* receive a utility $u_i(c_i)$ for an allocated resource c_i.

For all *i*, we suppose that the utility function $u_i(c_i)$ is concave, strictly increasing and continuous for $c_i > 0$ and that the derivative $u_i'(c_i)$ is finite.

A solution to that game is the maximization of the social utility on the players set [1].

$$\text{Argmax} \sum_{i=1}^{n} u_i(c_i) \text{ such as } \sum_{i=1}^{n} c_i \le C,\ c_i \ge 0,\ i = 1,\ldots \tag{1}$$

When the function $u_i(c_i)$ is concave, the optimal solution exists because the bargaining set is a convex set.

Players bid the resource by proposing payments $(p_1,\ldots,p_n)$ where p_i is the payment proposed by player *i*. In [2], we can see that allocation proportional to player's proposal is an optimal solution to such game. In spite of this, a problem is found when a player has proposed a very high payment. To overcome this inconvenient, we propose a gain which is function of the utility and the payment proposal to have more equilibrium between players.

We assume that the system ask an elementary price μ to each player. Given the payments $(p_1,\ldots,p_n)$ proposed by the players, and the allocated resources $(c_1,\ldots,c_n)$, we must have for all *i* :

$$c_i = \frac{p_i}{\mu} \tag{2}$$

If the queue server allocated all available resource, it is easy to find that the elementary price is given by (3).

$$\mu = \frac{\sum p_i}{C} \tag{3}$$

By sharing all available resource, and assuming that players are identical (the server asks the same price μ to each of them), a single elementary price μ exists for each proposed payment $(p_1,\ldots,p_n)$ [3].

In our model, given a price μ , each player has to maximize his gain function g_i on the possible proposal p_i set.

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

This gain function is defined from the utility compared to resource allocated to the player in return for the payment he proposed. We will use this function to limit abuses in term of higher price proposal for some players. At a higher price will decrease this gain.

The equilibrium can be defined as the price proposal $(p_1^*,\ldots,p_n^*)$ where players will maximize their gains, and the system will grant the elementary price μ defined in (5).

$$g_i(p_i^*,\mu) \ge g_i(p_i,\mu),\ p_i \ge 0,\ i = 1,\ldots,n,\ \mu = \frac{\sum p_i^*}{C} \tag{5}$$

In that case, the solution $c_i^* = p_i^*/\mu$ is an optimal solution satisfying (1).

To prove it, we can use the lagrangian method to establish that the conditions in (5) are identical to the conditions in (1) with $c_i^* = p_i^*/\mu$.

T. B. Ravaliminoarimalalason, PhD School in Sciences and Technical Engineering and Innovation, University of Antananarivo, Antananarivo, Madagascar, + 261340016433.

H. Z. Andriamanohisoa, PhD School in Sciences and Technical Engineering and Innovation, University of Antananarivo, Antananarivo, Madagascar, + 261341033943.

F. Randimbindrainibe, PhD School in Sciences and Technical Engineering and Innovation, University of Antananarivo, Antananarivo, Madagascar, + 261340646690.

B. *A repeated game*

With repeated game we can model situations where players interact repetitively playing the same game [4]-[6]. They choose their actions simultaneously without knowing the choices of other players. So, a basic game is played in each period of a discrete time $t = 1,\ 2,\ \ldots$, and at the end of each period, the players observe the performed actions.

We are given a game in normal form $J = \left(N, \left(S_i\right)_{i\in N}, \left(g_i\right)_{i\in N}\right)$ where N is the players set, S_i the set of possible strategies for player *i*, and g_i his associated gain.

At the step $t = 1$, each player *i* chooses an action $s_i^1 \in S_i$ independently of other players. Let's denote $s^1 = \left(s_i^1\right)_{i\in N}$ the vector joint played actions in step 1. At the end of this step, s^1 is revealed to all players.

At step t ($t \geq 2$), knowing the history $h^t = (s^1, s^2, \ldots, s^{t-1})$ of the played actions during the past, each player *i* chooses an action $s_i^t \in S_i$ independently of other players. Let's denote H^t the set of possible history at step *t*.

It remains to define the gain function. For this, we should determine how the players evaluate the result of an infinite length of history. Indeed, if $(s^1, s^2, \ldots) \in H^\infty$, the player *i* receive a gain $g_i(s^t)$ at step *t*. In a model of discounted game, the player gives more weight to a unit of gain received today compared to a unit of gain received tomorrow. For this, we will use a discount factor $\delta \in \left]0,1\right[$. Thus, one unit of gain owned at step 2 is only δ unit of gain in step 1, and one unit of gain owned at step *t* is δ^{t-1} unit of gain in step 1.

In this context, the gain owned by player *i* during a game play $h = (s^1, s^2, \ldots)$ and evaluated at time *t* = 1 is expressed by (6) [6].

$$g_i^\delta(h) = (1-\delta)\sum_{t=1}^{\infty}\delta^{t-1}g_i(s^t) \qquad (6)$$

In (6), the factor $(1-\delta)$ is a normalization factor to take the gain back to the same unit at any step.

C. *A stochastic game*

Stochastic games are extension of Markov process in case of several agents called players in a common environment [7]-[9]. These players play a joint action which defines the owned gains and the new state of the environment.

A stochastic game can be defined by a quintuplet $\left(N, E, \left(S_i\right)_{i\in N}, \left(g_i\right)_{i\in N}, T\right)$ where:

- N is the set of players who act to the game
- E is the finite set of states of the game
- S_i is the set of possible strategies (actions) for player *i*
- g_i is the gain function, which is a function of the state of the game and the strategies played by all players : $g_i : E \times S_1 \times \ldots \times S_N \to \mathbf{R}$
- T is the transition model between states, which depends to joint strategies : $T : E \times S_1 \times \ldots \times S_N \times E \to \left[0,1\right]$

At each step of the game, given the current state $e \in E$, players choose strategies $s = (s_1, \ldots, s_N)$ to play. Each player *i* own a gain $g_i(e,s)$, and then, the system goes from state *e* to state e' according to the transition model *T* who satisfies (7).

$$\sum_{e\in E} T(e,s,e') = 1 \qquad (7)$$

We call a policy $\pi_i : E \to \left[0,1\right]^{\mathrm{Card}\, S_i}$ the vector whose elements define a probability distribution over the strategies of player *i*, specific to a game in normal form defined by the state *e*. for player *i*, the policy defines a local strategy in each state within the meaning of game theory. The expected utility refers to the expected gain on the strategies of opposing players. For a joint policy $\pi = (\pi_1, \ldots, \pi_N)$, we define the expected utility of player *i* for each state *e* as expressed in (8).

$$u_i(\pi, e) = \mathrm{E}_{s\in S}\left[g_i(e,s)\right] \qquad (8)$$

where $S = S_1 \times \ldots \times S_N$, and E denotes the expectation function.

And therefore, we can also define the utility $U_i(\pi, e)$ of states for player *i*, associated to a joint policy π, as the expected utility for player *i* from the state *e* if all players follow this joint policy.

$$\begin{aligned} U_i(\pi,e) &= \mathrm{E}_{s\in S}\left[\sum_{t=1}^{\infty}\delta^{t-1}u_i(\pi,e) \mid e^0 = e\right] \\ &= u_i(\pi,e) + \delta\sum_{s\in S}\sum_{e\in E}T(e,s,e')\pi(e,s)U_i(\pi,e') \end{aligned} \qquad (9)$$

where $\pi(e,s)$ designates the probability of the joint strategies *s* on the state *e* according to the joint policy π, and δ is the discount factor.

In a stochastic game, a Nash equilibrium is a vector strategy $\pi^* = (\pi_1^*, \ldots, \pi_N^*)$ as for all state $e \in E$ and for all player *i* [9] :

$$U_i\left((\pi_i^*, \pi_{-i}^*), e\right) \geq U_i\left((\pi_i, \pi_{-i}^*), e\right),\ \ \forall \pi_i \in \Pi_i \qquad (10)$$

where Π_i is the set of policies offered to player *i*. The notation (π_i^*, π_{-i}^*) means the vector of policies π^* where π_i^* is the policy of player *i* and π_{-i}^* the joint policy $\left(\pi_j^*\right)_{j\neq i}$ of players other than *i*.

III. 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 the 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 :

$$b_i^t = b_i^{t-1} - c_i^{t-1} \qquad (11)$$

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

$$dec(b_i^t) = \begin{cases} stay & \text{if } b_i^t > 0 \\ move & \text{if } b_i^t \le 0 \end{cases} \qquad (12)$$

This principle is illustrated on Fig. 1.

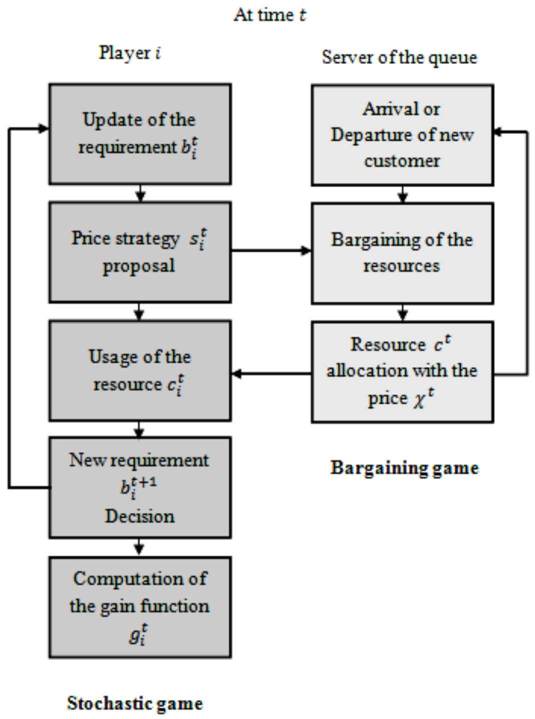
At time t
Player i
Server of the queue
Update of the requirement b_i^t
Arrival or Departure of new customer
Price strategy s_i^t proposal
Bargaining of the resources
Usage of the resource c_i^t
Resource c^t allocation with the price χ^t
New requirement b_i^{t+1} Decision
Bargaining game
Computation of the gain function g_i^t
Stochastic game

Fig. 1. Games at time t

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 $S_i = \left\{ s_i^t \right\}$, is also the set of possible strategies for that player. This strategy is developed on the next paragraph.

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

The transition model between local states is defined by the function $T_i^t : B_i \times S_1 \times \cdots \times S_{N^t} \times B_i \to [0,1]$ as expressed by (13) :

$$\sum_{b_i^{t+1} \in B_i} T_i(b_i^t, s^t, b_i^{t+1}) = 1 \qquad (13)$$

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 $(N^t, (B_i), (S_i), (g_i), (T_i))$.

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 (14) obtained from (4)

$$g_i^t(p_i^t, \mu^t) = u_i\left(\frac{p_i^t}{\mu^t}\right) - p_i^t \qquad (14)$$

On (14), the function $u_i\left(c_i^t\right)$ 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 a concave function 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 (15) and (16).

$$u_i\left(c_i^t, b_i^t\right) = A\frac{\log(c_i^t)}{\log(b_i^t)} \qquad (15)$$

$$u_i\left(c_i^t, b_i^t\right) = A - B\left(\frac{c_i^t}{b_i^t}\right)^2 \qquad (16)$$

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 $p^t = (p_1^t, \ldots, p_{N^t}^t) \in \mathbf{R}^{N^t}$. Once received by the server, it computes the resource to allocate $c^t = (c_1^t, \cdots, c_{N^t}^t)$.

On (14), the price μ^t is not yet known beforehand, so the players are not able to compute the optimal proposal p_i^t. We suppose that price anticipation described by (3) is used. The function gain that he must maximize is defined by (17) assuming that $\sum_{j \ne i} p_j^t$ is constant.

$$g_i^t(p_i^t)=u_i\left(\frac{p_i^t}{\sum_j p_j^t}C\right)-p_i^t \tag{17}$$

By cancelling the derivative with respect to p_i^t :

$$\left(g_i^t\right)'=u_i'\left(\frac{p_i^t}{\sum_j p_j^t}C\right).\left(1-\frac{p_i^t}{\sum_j p_j^t}\right).C-1=0 \tag{18}$$

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

$$u_i'\left(d_i^t\right).\left(1-\frac{d_i^t}{C}\right)=\frac{1}{C} \tag{19}$$

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 (19) independently of the other players. So, once the joint strategy $s^t=(d_1^t,\mathrm{L}\ ,d_{N^t}^t)$ 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 the stochastic game of the player i*

1) *Model of the state transition*

As we suggested that the 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 $b_i^{t+1}-b_i^t$:

$$b_i^{t+1}=b_i^t-c_i^t \tag{20}$$

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 $c_i^t=b_i^t-b_i^{t+1}$. The model $T_i^t:B_i\times S_1\times\cdots\times S_{N^t}\times B_i\to[0,1]$ of the state transition is expressed by (21).

$$T_i(b_i^t,s^t,b_i^{t+1})=\begin{cases}1 & \text{if } b_i^t-b_i^{t+1}=C\dfrac{d_i^t}{\sum_j d_j^t}\\ 0 & \text{otherwise}\end{cases} \tag{21}$$

where the joint strategy of all player is $s^t=(d_1^t,\mathrm{L}\ ,d_{N^t}^t)$, 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 (22).

$$g_i^t(d_i^t)=u_i(d_i^t)-\mu^t d_i^t \tag{22}$$

2) *Impact of the game history and its future*

As we are faced to a repeated game, players can use the history of the game to bargain the resource of the server. Let $h^t=\left\{b^1,d^1,\mu^1,\mathrm{L}\ ,b^{t-1},d^{t-1},\mu^{t-1},b^t\right\}\in H^t$ 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 the 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. It differs from a simple strategy d_i^t by using the observations up the time *t*.

$$\begin{aligned}\pi_i^t:O_i^t&\to A_i\\ o_i^t\ \mathrm{a}\ \ a_i&=\pi_i^t(o_i^t)\end{aligned} \tag{23}$$

In (23), A_i indicates the set of possible proposals. Equation (23) mentions that policies π_i^t are function of observation of player *i* at time *t*. So, the player *i* have to determine a policy which can ensure a best response. The better is to find a time independent policy that we call a stationary policy π_i, so as to be usable at any time by simply basing to observations. We can get this policy because the current requirement b_i^t depends only on the previous requirement b_i^{t-1} and the allocated resource at this time that is function of the proposal d_i^{t-1} as given by (21). In that case, the policy is markovian.

Let denote $\pi=\left(\pi_i,\pi_{-i}\right)$ 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 joint stationary policy π is denoted as $G_i^t(b_i^t,\pi)$. This gain is expressed by the recurrent relation (24).

$$G_i^t(b_i^t,\pi)=g_i^t(b_i^t,d_i^t)+\delta\sum_{b_i^{t+1}\in B_i}T(b_i^{t+1}\mid b_i^t).G_i^{t+1}(b_i^{t+1},\pi) \tag{24}$$

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

$$\pi_i^*(\pi_{-i})=\operatorname*{Arg\,max}_{\pi_i}G_i^t\left(b_i^t,\left(\pi_{\ddot{i}},\pi_{-\ddot{i}}\right)\right) \tag{25}$$

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

Equation (25) 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 (26).

$$G_i^t(b_i^t,\pi^m) = g_i^t(b_i^t,d_i^t) \tag{26}$$

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

If the 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 (27).

$$t_i^f = \min_t \left\{ b_i^t \leq 0, t > t_i^i \right\} \tag{27}$$

The sojourn time for player *i* is expected by (28).

$$t_i^S = t_i^f - t_i^i \tag{28}$$

The total gain for player *i* is expected by (29).

$$\begin{aligned} G_i^T &= \sum_{t=t_i^i}^{t_i^f} \delta^{t_i-t} G_i^t(b_i^t,\pi^m) \\ &= \sum_{t=t_i^i}^{t_i^f} \delta^{t_i-t} g_i^t(b_i^t,\pi^m) \\ &= \sum_{t=t_i^i}^{t_i^f} \delta^{t_i-t} \left(u_i(\pi^m) - \mu^t \pi^m \right) \end{aligned} \tag{29}$$

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 (40).

$$\begin{aligned} \overline{G_i^T} &= \frac{G_i^T}{t_i^f - t_i^i} \\ &= \frac{\sum_{t=t_i^i}^{t_i^f} \delta^{t_i-t} \left(u_i(\pi^m) - \mu^t \pi^m \right)}{t_i^f - t_i^i} \end{aligned} \tag{30}$$

IV. 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 [10]. We compared the established model to other models to know his performance.

As described in Fig. 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) [11],
- Our myopic players model.

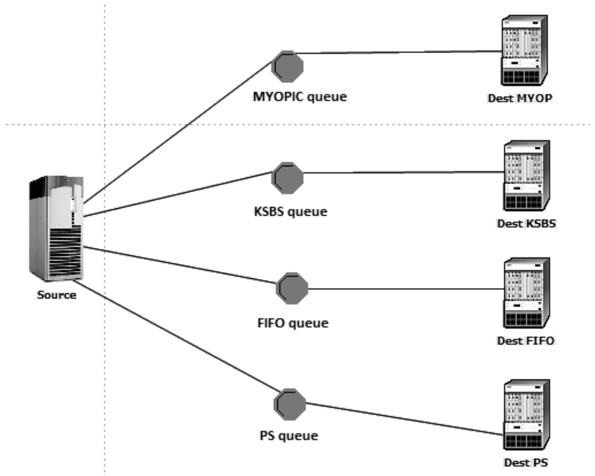

Fig. 2. Simulation description

"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 λ = 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.

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.

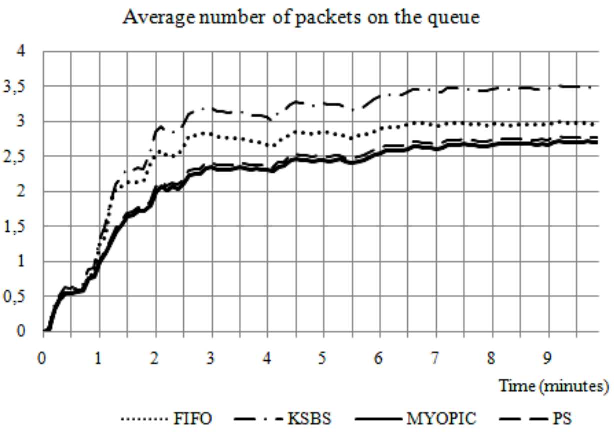

Fig. 3. Evolution of the average number of packets on each queue on a stable system

We find on Fig. 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.

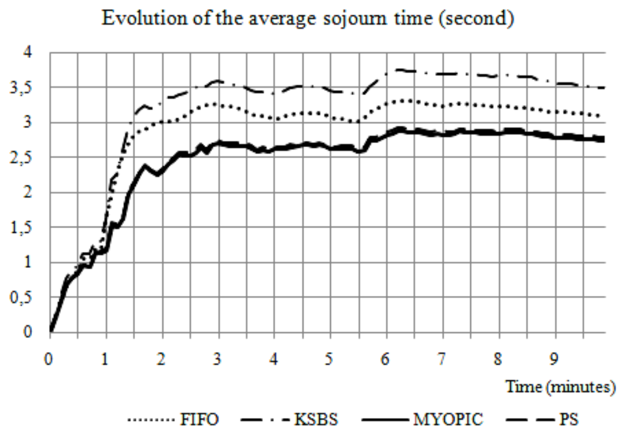

Fig. 4. Evolution of the average sojourn time on each queue on a stable system

As in Fig. 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.

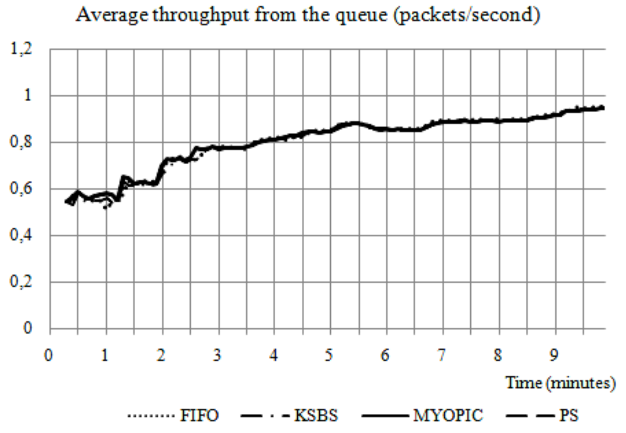

Fig. 5. Evolution of the throughput from each queue on a stable system

In Fig. 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.

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 Fig. 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 Fig. 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.

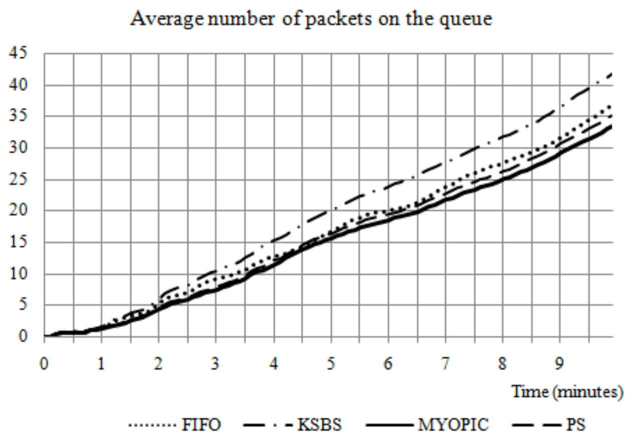

Fig. 6. Evolution of the average number of packets on each queue on an unstable system

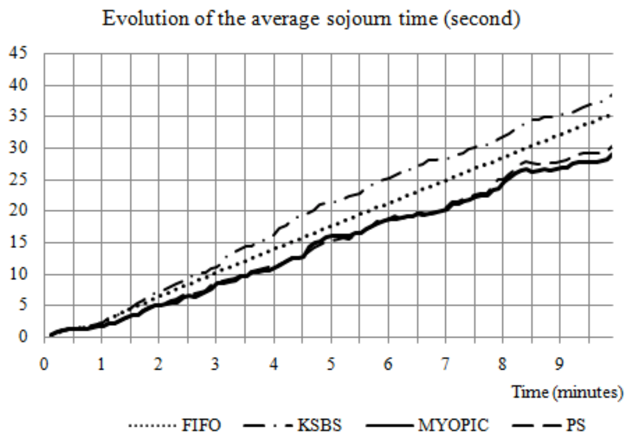

Fig. 7. Evolution of the average sojourn time on each queue on an unstable system

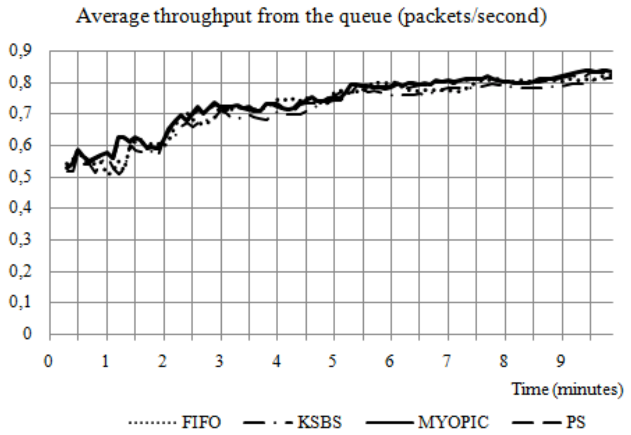

Fig. 8. Evolution of the average throughput from each queue on an unstable system

Table I 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 Fig. 8.

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

Node Name	[Total]
Dest FIFO	495
Dest KSBS	488
Dest MYOPIC	501
Dest PS	496

Source	607

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).

V. 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*, Centre for Communication Systems Research, University of Surrey - Guildford - UK, 2002

[11] E. Kalai, M. Smorodinsky, “Other Solutions to Nash’s Bargaining Problem”, *Econometrica*, vol. 43, 1975, pp. 513-518

T. B. Ravaliminoarimalalason, PhD School in Sciences and Technical Engineering and Innovation, University of Antananarivo, Antananarivo, Madagascar, + 261340016433.

H. Z. Andriamanohisoa, PhD School in Sciences and Technical Engineering and Innovation, University of Antananarivo, Antananarivo, Madagascar, + 261341033943.

F. Randimbindrainibe, PhD School in Sciences and Technical Engineering and Innovation, University of Antananarivo, Antananarivo, Madagascar, + 261340646690.