

# A packet dropping mechanism for efficient operation of M/M/1 queues with selfish users<sup>☆</sup>



Yi Gai<sup>a,\*</sup>, Hua Liu<sup>b</sup>, Bhaskar Krishnamachari<sup>b</sup>

<sup>a</sup> Intel Labs, Hillsboro, OR 97124, USA

<sup>b</sup> Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089, USA

## ARTICLE INFO

### Article history:

Received 13 December 2014

Revised 22 September 2015

Accepted 23 December 2015

Available online 28 January 2016

### Keywords:

M/M/1 queue

Game theory

Price of Anarchy

Nash equilibrium

## ABSTRACT

We consider a fundamental game theoretic problem concerning selfish users contributing packets to an M/M/1 queue. In this game, each user controls its own input rate so as to optimize a desired tradeoff between throughput and delay. We first show that the original game has an inefficient Nash Equilibrium (NE), with a Price of Anarchy (PoA) that scales linearly or worse in the number of users. In order to improve the outcome efficiency, we propose an easily implementable mechanism design whereby the server randomly drops packets with a probability that is a function of the total arrival rate. We show that this results in a modified M/M/1 queueing game that is an ordinal potential game with at least one NE. In particular, for a linear packet dropping function, which is similar to the Random Early Detection (RED) algorithm used in Internet Congestion Control, we prove that there is a unique NE. We also show that the simple best response dynamic converges to this unique equilibrium. Finally, for this scheme, we prove that the social welfare (expressed either as the summation of utilities of all players, or as the summation of the logarithm of utilities of all players) at the equilibrium point can be arbitrarily close to the social welfare at the global optimal point, i.e. the PoA can be made arbitrarily close to 1. We also study the impact of arrival rate estimation error on the PoA through simulations.

© 2015 Elsevier B.V. All rights reserved.

## 1. Introduction

In the past 20 years, the usage of the Internet has transitioned from being primarily academic/research-oriented to one that is primarily commercial in nature. In the current Internet environment, each commercial entity is inherently interested only in its own profit. Developing

network mechanisms that are designed to handle selfish behavior has therefore gained increasing attention in recent years. The game theoretic approach, which was originally designed to model and guide decisions in economic markets, provides a valuable set of tools for dealing with selfish behavior [2–7].

In this work, we consider the network congestion problem at a single intermediate store-and-forwarding spot in the network. Several users send their packets to a single server with Poisson arrival rate. The server processes the packets on a first come first serve (FCFS) basis with an exponentially distributed service time. This is an M/M/1 queueing model [8]. There exists a trade-off in this M/M/1 queueing model between throughput (representing the benefit from service), and delay (representing the waiting cost in the queue). In the gateway

<sup>☆</sup> This research was sponsored in part by the U.S. Army Research Laboratory under the Network Science Collaborative Technology Alliance, Agreement number W911NF-09-2-0053, and by the U.S. National Science Foundation under CNS-0831545. This work is an extended version of the conference paper [1].

\* Corresponding author. Tel.: +1 503 712 5198.

E-mail addresses: [yigaiee@gmail.com](mailto:yigaiee@gmail.com), [yi.gai@intel.com](mailto:yi.gai@intel.com) (Y. Gai), [hual@usc.edu](mailto:hual@usc.edu) (H. Liu), [bkrishna@usc.edu](mailto:bkrishna@usc.edu) (B. Krishnamachari).

congestion control context [9], a measure that is widely used to describe this trade-off is called “Power”, which is defined as the weighted ratio of the throughput to the delay. When the users are selfish, we can formulate a basic M/M/1 queueing game. In this game, we assume that the users are selfish, and each control their own input arrival rate to the server. Each user’s utility is modeled to be the power ratio for that user’s packets.

This classic M/M/1 queueing game has been formulated and studied in [10–14]. The results from these prior works and our own results in this work are in agreement that the basic M/M/1 queueing game has an inefficient Nash Equilibrium. We are therefore motivated to design an incentive mechanism to force the users to operate at an equilibrium that is globally efficient. In particular, we focus on the design of a packet dropping scheme implemented at the server for this purpose. Our objective is that the dropping scheme should be as simple as possible, and it should minimize the Price of Anarchy (PoA, the ratio of the social optimum welfare to the welfare of the worst Nash equilibrium) to be as close to 1 as possible.

A key contribution of this work is the formulation of a modified M/M/1 queueing game with a randomized packet dropping policy at the server. We consider a simple and low overhead policy in our formulation, wherein the server need only monitor the sum of the rates of all users in the system. We show that this modified game with a packet dropping scheme is an *ordinal potential game* [15], which implies the existence of at least one pure Nash Equilibrium.

We show first that utilizing a step-function for packet dropping whereby the server drops all the packets when the sum-rate is greater than a threshold (and none when the sum-rate is below the threshold), results in infinite number of undesired Nash Equilibria which harms the PoA.

This raises the question whether a more sophisticated approach can do better. We show that indeed this is possible. In particular, we develop an incentive mechanism with a linear packet dropping that can improve the system efficiency to be arbitrarily close to the global optimal point (i.e., a PoA arbitrarily close to 1). This mechanism is similar to the Random Early Detection (RED) used for congestion avoidance on the Internet [16]. We prove the uniqueness of NE of the game with this mechanism. We also show that best response dynamics will converge to the unique NE.

Our paper is organized as follows. Section 2 summarizes the related work. We present the model of an M/M/1 queue game in Section 3. The social welfare and Price of Anarchy are described in Section 4 to investigate the efficiency of the NE. Then, in Section 5, we propose to design an incentive packet dropping scheme implemented at the server to improve the efficiency. Section 6 proves that the game defined with packet dropping policy is an ordinal potential game by giving the potential function. Section 7 shows the best response function. In Section 8, we show the behavior when utilizing a simple step-function for packet dropping. In Section 9 we propose the RED-like linear packet dropping incentive scheme. We show that with this scheme, it is possible to make the Price of Anarchy arbitrarily close to the optimal point. The uniqueness of NE of such a game is proved in Section 10. In Section 11, we

show that the best response dynamics will converge to the unique Nash Equilibrium. In Section 12, we undertake simulations to see how the process of statistically estimating the input arrival rates in a real system would impact the PoA. We conclude the work in Section 13.

## 2. Related work

Throughput-delay tradeoffs in M/M/1 queues with selfish users have been previously studied in [10–14]. A utility function for each user is defined as the corresponding application’s power and each user is treated as a player in such a game and adjusts its arrival rate to handle the trade-off between throughput and delay. Every user is assumed to be selfish and only wants to maximize its own utility function in a distributed manner.

Bharath-Kumar and Jaffe [10] wrote one of the earliest papers on the formulation of throughput-delay tradeoffs in M/M/1 queues with selfish users. The paper discusses the properties of power as a network performance objective function. A class of greedy algorithms where each user updates its sending rate synchronously to the best response of all other users’ rates to maximize the power is proposed. Convergence of the best response to an equilibrium point is shown in this paper.

Douligeris and Mazumdar [11] extended Bharath-Kumar and Jaffe’s work to the case with different weighting factors defined in the power function for different users and provided analytical results describing the Nash Equilibrium. They showed that the equilibrium point that the greedy best response dynamic algorithm converged to was a unique Nash Equilibrium.

The work by Zhang and Douligeris [12] proved the convergence of the best response dynamics for this basic M/M/1 queueing game under the multiple users case. Thus all these prior works [10–12] studied only variants of the basic game. Their work, along with ours, shows that this basic game results in an inefficient outcome. Our work is the first to develop a mechanism design for this problem that addresses this shortcoming by showing how to achieve near-optimal performance using a packet-dropping policy.

Dutta et al. [13] studied a related problem involving a server that employs an oblivious active queue management scheme, i.e. drops packets depending on the total queue occupancy with the same probability regardless of which flow they come from. They also consider an M/M/1 setting with users offering Poisson traffic to a server with exponential service time. The users’ actions are the input rates and the utilities the goodput/output rates. The existence and the quality of symmetric Nash equilibria are studied for different packet dropping policies. Although our work also explores oblivious packet dropping schemes, it is different from and somewhat more challenging to analyze than [13], because our utility function reflects the tradeoff between goodput and delay.

In another, more recent work, [14], Su and van der Schaar have discussed linearly coupled communication games in which users’ utilities are linearly impacted by their competitors’ actions. An M/M/1 FCFS queueing game with the power as the utility function is one illustrative

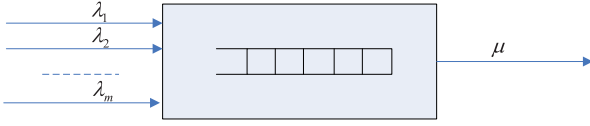


Fig. 1. An M/M/1 queue.

example of linearly coupled communication games. They also quantify the Price of Anarchy in this case, and investigate an alternative solution concept called Conjectural Equilibrium, which requires users to maintain and operate upon additional beliefs about competitors.

There have been also several other papers related to queueing games, albeit with different formulations. Haviv and Roughgarden [17] considered a system with multiple servers with heterogeneous service rates. Arrivals from customers are routed to one of the servers, and the routing decisions are analyzed based on NE or social optimization schemes. PoA is shown to be upper bounded by the number of servers for the social optimum. Wu and Starobinski [18] analyzed the PoA of  $N$  parallel links where the delays of links are characterized using unbounded delay functions such as M/M/1 or M/G/1 queueing functions. Economides and Silvester [19] studied a multiserver two-class queueing game and developed the routing policy.

For more general surveys on game theoretic formulations of networking problems, we refer the reader to [4,20].

### 3. Problem formulation

We consider a M/M/1 FCFS queue game as shown in Fig. 1. There are  $m$  users with independent Poisson arrivals and the arrival rates are  $\lambda_1, \lambda_2, \dots, \lambda_m$ . There is a single server and the service time is exponentially distributed with mean  $\frac{1}{\mu}$ .

We consider each user as a player for this game and the users are selfish. Each player wants to maximize its own utility function by adjusting its rate sending to the queue.

Note that there is a tradeoff between the throughput and delay for each user, i.e., given the rates of all other users, if the input rate increases, the delay increases too. In this paper, we consider the measurement of this tradeoff between the throughput and delay of the each user, and it is known as the “power”, which is widely used in the gateway congestion control context [9]. We consider the power as the utility function of each user to measure its throughput-delay tradeoff.<sup>1</sup> For a given user  $i$ , the power is defined as:

$$\text{Power} = \frac{\text{Throughput}^{\alpha_i}}{\text{Delay}} \quad (1)$$

where  $\alpha_i$  is a parameter chosen based on the relative emphasis placed on throughput versus delay.  $\alpha_i > 1$  when throughput is more important, while  $0 < \alpha_i < 1$  when we want to emphasis delay more, and  $\alpha_i = 1$  when the throughput and delay are emphasized equally.

<sup>1</sup> We note that other utility functions can be used, and we focus in this research on the throughput-delay tradeoff.

For M/M/1 queue, the throughput for user  $i$  is  $T_i = \lambda_i^e$  where  $\lambda_i^e$  is the effective rate served by the server. The delay for user  $i$  is calculated as:  $D = \frac{1}{\mu - \sum_{i=1}^m \lambda_i^e}$ .

So the power for user  $i$  can be expressed as:

$$P_i = \frac{T_i^{\alpha_i}}{D} = (\lambda_i^e)^{\alpha_i} \left( \mu - \sum_{i=1}^m \lambda_i^e \right) \quad (2)$$

In this M/M/1 game with  $m$  players, each player is selfish and wants to adjust its arrival rate  $\lambda_i$  to maximize its own utility function, given the arrival rates of all other players. Throughout the paper, we assume that the queue is stable and thus  $0 \leq \sum_{i=1}^m \lambda_i^e < \mu$ .

When there is no dropping policy implemented at the server,  $\lambda_i^e = \lambda_i$ , and the optimization problem for each player  $i$  is:

$$\begin{aligned} \max \quad & U_i(\lambda_i, \lambda_{-i}) = \lambda_i^{\alpha_i} \left( \mu - \sum_{i=1}^m \lambda_i \right) \\ \text{s.t.} \quad & \sum \lambda_i < \mu \\ & \lambda_i \geq 0 \quad \forall i = 1, 2, \dots, m \end{aligned} \quad (3)$$

### 4. Social welfare and price of anarchy

In [11] the above M/M/1 queue game is studied and a unique pure NE is proved to be:

$$\lambda_i^{NE} = \frac{\mu \alpha_i}{\sum_{k=1}^m \alpha_k + 1}, \forall i \quad (4)$$

When  $\alpha_i = \alpha, \forall i$ , this unique NE is expressed as

$$\lambda_i^{NE} = \frac{\mu \alpha}{\alpha m + 1}, \forall i. \quad (5)$$

Now suppose all users cooperate to achieve the maximal system utility. We consider two ways to define the social optimal function: the sum of the utilities of all the users and the sum of the logarithm of the utilities of all the users. Defining the social optimal function as the sum of the utilities of all the users is a common way for evaluating the system efficiency and we present the analysis results under this definition first. However, the fairness among the users should also be considered and it is not revealed under this definition; so we also consider a log-sum-utility social welfare function which provides for utility fairness.

We can measure the efficiency of the system using two well known measures called the Price of Anarchy (PoA) and Price of Stability (PoS), that respectively compare the performance of selfish users in the worst and best case Nash Equilibrium with the global optimum achievable with non-selfish users. The definition of PoA and PoS of a game  $G$  is:

$$\begin{aligned} \text{PoA}(G) &\triangleq \max_{a \in \mathcal{E}(G)} \frac{U(a^{OPT})}{U(a)} \\ \text{PoS}(G) &\triangleq \min_{a \in \mathcal{E}(G)} \frac{U(a^{OPT})}{U(a)} \end{aligned}$$

where  $\mathcal{E}$  is the set of all the Nash Equilibriums in game  $G$ .

#### 4.1. Sum-utility

The optimization problem is defined as:

$$\begin{aligned} \max \quad & \sum \lambda_i^\alpha (\mu - \sum \lambda_i) \\ \text{s.t.} \quad & 0 \leq \sum \lambda_i < \mu \\ & \lambda_i \geq 0 \quad \forall i = 1, 2, \dots, m \end{aligned} \quad (6)$$

Here we consider two cases:

(1)  $\alpha > 1$

We first calculate  $U^{OPT}$ .

$$\begin{aligned} U^{OPT} &= \max_{\lambda_i} \sum \lambda_i^\alpha (\mu - \sum \lambda_i) \\ &\leq \max_{\lambda_i} (\sum \lambda_i)^\alpha (\mu - \sum \lambda_i) \end{aligned} \quad (7)$$

$$= \max_{\lambda} \lambda^\alpha (\mu - \lambda) \quad (8)$$

(8) shows the upper bound of  $U^{OPT}$  and we can get  $\lambda^* = \frac{\mu\alpha}{\alpha+1}$  for (8) to achieve its maximum value. We note that equality holds in Eq. (7) when  $\lambda_i = \lambda^*$  for some  $i$ , and  $\lambda_j = 0, \forall j \neq i$ . Hence this is also the solution for  $P_{sys}^{OPT}$ . So when  $\lambda_i = \lambda^* = \frac{\mu\alpha}{\alpha+1}$  for some  $i$ , and  $\lambda_j = 0, \forall j \neq i$ , we get the optimal solution  $P_{sys}^{OPT}$  as follows:

$$U^{OPT} = \frac{\alpha^\alpha \mu^{\alpha+1}}{(\alpha+1)^{\alpha+1}}$$

Then we calculate  $U^{NE}$  when players are selfish. When  $\lambda_i^{NE} = \frac{\mu\alpha}{\alpha m + 1}, \forall i$  as shown in Eq. (5):

$$U^{NE} = \frac{m\alpha^\alpha \mu^{\alpha+1}}{(\alpha m + 1)^{\alpha+1}} \quad (9)$$

Note that there is only one NE in the game, so PoA and PoS are the same and they are derived as below:

$$PoA(G) = PoS(G) = \frac{U^{OPT}}{U^{NE}} = \frac{(\alpha m + 1)^{\alpha+1}}{m(\alpha + 1)^{\alpha+1}} \quad (10)$$

In this case we find that the PoA and PoS are proportional to  $m^\alpha$ .

(2)  $\alpha < 1$

The calculation is similar as above, and details are omitted.

$$\begin{aligned} U^{OPT} &= \max_{\lambda_i} \sum \lambda_i^\alpha (\mu - \sum \lambda_i) \\ &\leq m \max_{\lambda_i} \left( \frac{\sum \lambda_i}{m} \right)^\alpha (\mu - \sum \lambda_i) \\ &= \max_{\lambda} m^{1-\alpha} \lambda^\alpha (\mu - \lambda) \end{aligned}$$

We also get  $\lambda^* = \frac{\mu\alpha}{\alpha+1}$ .

$$U^{OPT} = \frac{m^{1-\alpha} \alpha^\alpha \mu^{\alpha+1}}{(\alpha+1)^{\alpha+1}}$$

PoA and PoS are:

$$PoA(G) = PoS(G) = \frac{U^{OPT}}{U^{NE}} = \frac{(\alpha m + 1)^{\alpha+1}}{m^\alpha (\alpha + 1)^{\alpha+1}} \quad (11)$$

In this case we find that the PoA and PoS are proportional to  $m$ .

Thus in both cases, we find that the PoA and PoS degrade linearly or worse with the number of users.

#### 4.2. Sum-log-utility

Now let's consider the sum of the logarithm of the utilities of all the users. The reason we consider the logarithm function in the social welfare is because when all users cooperate to achieve the optimum, fairness among the users should also be considered, and a logarithmic function would ensure this [21]. The social welfare optimization problem is:

$$\max \sum_{i=1}^m \log \left[ \lambda_i^\alpha \left( \mu - \sum_{i=1}^m \lambda_i \right) \right] \quad (12)$$

$$\begin{aligned} \text{s.t.} \quad & 0 \leq \sum_{i=1}^m \lambda_i < \mu \\ & \lambda_i \geq 0 \quad \forall i = 1, 2, \dots, m \end{aligned}$$

Note that for each player, maximizing the logarithm of its utility function is equivalent to maximizing the utility function itself. Therefore the NE remains the same as before.

Denote  $\lambda = \sum_{i=1}^m \lambda_i$ . We have the following theorem for finding out the social optimum:

**Theorem 1.** The solution for the social welfare optimization problem is:  $\lambda_i^* = \frac{\mu\alpha}{m(\alpha+1)}$ .

**Proof.** see Appendix A.  $\square$

Note that  $\lambda_i^{NE}$  is shown in (4) and by substituting it into (3), we get the power for user  $i$  as:

$$U_i^{NE} = \frac{\alpha^\alpha \mu^{\alpha+1}}{(\alpha m + 1)^{\alpha+1}}$$

In general, the log-utility terms can be negative. To ensure that both the numerator and denominator terms in the PoA and PoS are non-negative in this case, we use a monotonic exponential mapping. Note that there is only one NE in the game, so PoA and PoS are the same, and they are derived as below:

$$\begin{aligned} PoA(G) = PoS(G) &= \frac{e^{U^{OPT}}}{e^{U^{NE}}} \\ &= \frac{\left( \frac{\alpha^\alpha \mu^{\alpha+1}}{m^\alpha (\alpha+1)^{\alpha+1}} \right)^m}{\left( \frac{\alpha^\alpha \mu^{\alpha+1}}{(\alpha m + 1)^{\alpha+1}} \right)^m} = \left( \frac{(\alpha m + 1)^{\alpha+1}}{m^\alpha (\alpha + 1)^{\alpha+1}} \right)^m > 1 \end{aligned} \quad (13)$$

From (13) we can see that PoA increases monotonically as  $m$  increases and goes to infinity as  $m$  goes to infinity. So we want to implement an incentive mechanism to improve the PoA.

#### 5. An incentive packet dropping scheme

Note that  $\lambda_i^{NE} = \frac{\mu\alpha}{\alpha m + 1} > \frac{\mu\alpha}{m(\alpha+1)} = \lambda_i^*$ . This inspires us to find an incentive packet dropping mechanism implemented at the server and we wish this packet dropping mechanism to be as simple as possible. So we consider the case where the server need only monitor the sum of the rates of all users in the system and implement the packet dropping policy based only on this information. Then the packet dropping function could be expressed as  $P_d(\sum \lambda_i)$ .

So the optimization problem for each user  $i$  with a dropping policy in the system is:

$$\begin{aligned} \max \quad & U_i(\lambda_i, \lambda_{-i}) \\ & = \left( \lambda_i \left( 1 - P_d \left( \sum \lambda_i \right) \right) \right)^{\alpha_i} \\ & \quad \times \left( \mu - \sum \left( \lambda_i \left( 1 - P_d \left( \sum \lambda_i \right) \right) \right) \right) \\ \text{s.t.} \quad & \sum \lambda_i \left( 1 - P_d \left( \sum \lambda_i \right) \right) < \mu \\ & \lambda_i \geq 0 \quad \forall i = 1, 2, \dots, m \end{aligned}$$

To facilitate the derivation, denote  $P(\sum \lambda_i) = 1 - P_d(\sum \lambda_i)$  and thus  $P(\cdot)$  is the probability of keeping packets in the system. Then the optimization problem for each player  $i$  is:

$$\begin{aligned} \max \quad & U_i(\lambda_i, \lambda_{-i}) \\ & = \left( \lambda_i P \left( \sum \lambda_i \right) \right)^{\alpha_i} \left( \mu - \sum \left( \lambda_i P \left( \sum \lambda_i \right) \right) \right) \\ \text{s.t.} \quad & \sum \lambda_i P \left( \sum \lambda_i \right) < \mu \\ & \lambda_i \geq 0 \quad \forall i = 1, 2, \dots, m \end{aligned} \quad (14)$$

We denote the above game as  $G_p = (N, \{A_i\}, \{U_i\})$ .

## 6. Potential game

In this section, we prove that when the dropping function is a function which only depends on the sum of total incoming rates, the game is a potential game and thus there exists at least one pure NE.

**Definition 1.** a game  $G = (N, \{A_i\}, \{U_i\})$  is called an *ordinal potential game* if there exists a global function  $\phi : \mathcal{A} \rightarrow \mathbb{R}$  such that for every player  $i \in N$ , for every  $a_{-i} \in \mathcal{A}_{-i}$  and for every  $a'_i, a''_i \in A_i$ ,

$$\begin{aligned} \text{sgn}(U_i(a'_i, a_{-i}) - U_i(a''_i, a_{-i})) \\ = \text{sgn}(\phi(a'_i, a_{-i}) - \phi(a''_i, a_{-i})) \end{aligned} \quad (16)$$

where  $\text{sgn}(x)$  is the sign function that takes on the value  $-1$  when  $x < 0$ ,  $0$  when  $x = 0$ , and  $1$  when  $x > 0$ . Also, the following [Theorem 2](#) holds for the existence of NE in a potential game:

**Theorem 2** (Monderer–Shapley, [15]). *Every potential game with finite-players, continuous utilities, and compact strategy sets possesses at least one pure-strategy equilibrium.*

Now we will prove that the M/M/1 queueing game with a packet dropping function  $P_d(\sum \lambda_i)$  is a potential game.

**Theorem 3.**  $G_p$  is a ordinal potential game with potential function

$$\begin{aligned} \phi(\lambda_1, \lambda_2, \dots, \lambda_m) \\ = \left( \mu - P \left( \sum \lambda_i \right) \sum_{i=1}^m \lambda_i \right) \left( \prod_{i=1}^m \left( \lambda_i P \left( \sum \lambda_i \right) \right)^{\alpha_i} \right) \end{aligned}$$

**Proof.**

$$\begin{aligned} & \phi(\lambda'_i, \lambda_{-i}) - \phi(\lambda''_i, \lambda_{-i}) \\ & = \left( \mu - P(\lambda'_i + \lambda_{-i}) \left( \sum_{j \neq i} \lambda_j + \lambda'_i \right) \right) \\ & \quad \times \left( \prod_{j \neq i} \left( \lambda_j P \left( \sum \lambda_j \right) \right)^{\alpha_j} \right) \lambda_i'^{\alpha_i} P(\lambda'_i + \lambda_{-i})^{\alpha_i} \\ & \quad - \left( \mu - P(\lambda''_i + \lambda_{-i}) \left( \sum_{j \neq i} \lambda_j + \lambda''_i \right) \right) \\ & \quad \times \left( \prod_{j \neq i} \left( \lambda_j P \left( \sum \lambda_j \right) \right)^{\alpha_j} \right) \lambda_i''^{\alpha_i} P(\lambda''_i + \lambda_{-i})^{\alpha_i} \\ & = \left( \prod_{j \neq i} \left( \lambda_j P \left( \sum \lambda_j \right) \right)^{\alpha_j} \right) \\ & \quad \times \left[ \left( \mu - P(\lambda'_i + \lambda_{-i}) \left( \sum_{j \neq i} \lambda_j + \lambda'_i \right) \right) (\lambda_i' P(\lambda'_i + \lambda_{-i}))^{\alpha_i} \right. \\ & \quad \left. - \left( \mu - P(\lambda''_i + \lambda_{-i}) \left( \sum_{j \neq i} \lambda_j + \lambda''_i \right) \right) (\lambda_i'' P(\lambda''_i + \lambda_{-i}))^{\alpha_i} \right] \\ & = \left( \prod_{j \neq i} \left( \lambda_j P \left( \sum \lambda_j \right) \right)^{\alpha_j} \right) (U_i(\lambda'_i, \lambda_{-i}) - U_i(\lambda''_i, \lambda_{-i})) \end{aligned}$$

□

Note that  $G_p$  has a finite number of players and continuous utilities. However its strategy sets are not compact in (14) so we could not directly apply [Theorem 2](#) to claim there exists at least one NE in  $G_p$ . But we modify the  $G_p$  to be the equivalent game as follows:

$$\begin{aligned} \max \quad & U_i(\lambda_i, \lambda_{-i}) = \left( \lambda_i P \left( \sum \lambda_i \right) \right)^{\alpha_i} \\ & \quad \times \left( \mu - \sum \left( \lambda_i P \left( \sum \lambda_i \right) \right) \right) \\ \text{s.t.} \quad & \sum \lambda_i P \left( \sum \lambda_i \right) \leq \mu \\ & \lambda_i \geq 0 \quad \forall i = 1, 2, \dots, m \end{aligned} \quad (17)$$

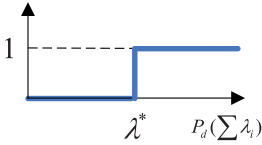
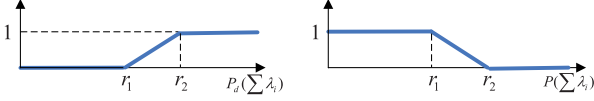
since any solution to the maximization problem in  $G_p$  will not satisfy  $\sum \lambda_i P(\sum \lambda_i) = \mu$ . Now strategy sets of  $G_p$  are compact and thus there exists at least one NE in  $G_p$ .

Note that in the following when we describe PoA and PoS for the packet dropping game  $G_p$ , we respectively compare the worst and best NE obtained for this game with respect to the social welfare (global optimum) that can be obtained through cooperation without packet dropping.

## 7. Best response function

From now, for tractability, we consider the case  $\alpha_i = \alpha, \forall i$  for our proposed incentive packet dropping scheme.



Fig. 2. Step dropping function  $P_d(\Sigma\lambda_i)$ .Fig. 3. Illustration of  $P_d(\Sigma\lambda_i)$  and  $P(\Sigma\lambda_i)$ .

$\forall i$ , let

$$\frac{\partial U_i(\lambda_i, \lambda'_{-i})}{\partial \lambda_i} = 0$$

If  $P(\Sigma\lambda_i)$  is differentiable with respect to  $\lambda_i$ , we will have

$$\begin{aligned} & \alpha P\mu - (\alpha + 1)PP'\lambda_i\lambda_{-i} \\ & - (\alpha + 1)PP'\lambda_i^2 - \alpha P^2\lambda_{-i} - (\alpha + 1)P^2\lambda_i + \alpha P'\lambda_i\mu = 0 \end{aligned} \quad (18)$$

where  $P'$  is the derivative of  $P(\Sigma\lambda_i)$  with respect to  $\lambda_i$ .

The above defines an implicit best response function  $\mathcal{F}(\lambda_i, \lambda_{-i}) = 0$  which shows the relationship between  $\lambda_i$  and  $\lambda_{-i}$ .

## 8. Step dropping function

An intuitive dropping policy that first comes to mind is a step function as shown in Fig. 2.

The expression of  $P_d(\Sigma\lambda_i)$  is:

$$P_d(\Sigma\lambda_i) = \begin{cases} 0 & : \sum \lambda_i \leq \lambda^* \\ 1 & : \sum \lambda_i > \lambda^* \end{cases} \quad (19)$$

We have the following result for the corresponding packet dropping game.

**Theorem 4.**  $\lambda'$  is a N.E. if and only if  $\sum \lambda'_i = \lambda^*$ .

**Proof.** see Appendix B.  $\square$

Based on Theorem 4, the NEs of the game with the step dropping function are not unique. PoS = 1 since there exists a NE with  $\lambda_i = \lambda'_i, \forall i$ . However, in the sum-utility case, PoA =  $m^{\alpha-1}$  when  $\alpha > 1$ , and PoA =  $m^{1-\alpha}$  when  $\alpha < 1$ . Moreover, PoA is infinite in the sum-log-utility case since there exists a NE which has one user  $i$  with  $\lambda_i = 0$ . Hence this is not a desirable result for improving the efficiency. We therefore next consider a slightly more sophisticated dropping function that has a linear profile.

## 9. Linear dropping function

We consider the game with the following linear function of  $P(\Sigma\lambda_i)$  (and thus the packet dropping function

$P_d(\Sigma\lambda_i) = 1 - P(\Sigma\lambda_i)$  is also a linear function.) Fig. 3 illustrates  $P_d(\Sigma\lambda_i)$  and  $P(\Sigma\lambda_i)$ .

$$P(\Sigma\lambda_i) = \begin{cases} 1 & : 0 \leq \sum \lambda_i \leq r_1 \\ A\left(\sum \lambda_i\right) + D & : r_1 \leq \sum \lambda_i \leq r_2 \\ 0 & : \sum \lambda_i \geq r_2 \end{cases} \quad (20)$$

where

$$\begin{cases} A = \frac{1}{r_1 - r_2} \\ D = -Ar_2 \end{cases}$$

$$P' = \frac{\partial P}{\partial \lambda_i} = A$$

For linear dropping scheme, (18) becomes:

$$\begin{aligned} & \alpha P\mu - (\alpha + 1)PA\lambda_i\lambda_{-i} - (\alpha + 1)PA\lambda_i^2 \\ & - \alpha P^2\lambda_{-i} - (\alpha + 1)P^2\lambda_i + \alpha A\lambda_i\mu = 0 \end{aligned} \quad (21)$$

The above also defines an implicit function  $\mathcal{F}(\lambda_i, \lambda_{-i}) = 0$ .

Denote  $\lambda_i^e = P(\Sigma\lambda_i)\lambda_i$  and  $\lambda_{-i}^e = P(\Sigma\lambda_i)\lambda_{-i}$ . First we want find out if we could design a dropping policy in this linear form such that the system could have a NE that is the same as the social optimum. If not, we will then explore how much efficiency it could achieve.

**Theorem 5.** There does not exist a linear packet dropping policy such that PoA = 1.

**Proof.** Assume the above Theorem does not hold, when  $P \neq 0$ , substituting  $\lambda_i$  and  $\lambda_{-i}$  with  $\lambda_i^e/P$  and  $\lambda_{-i}^e/P$  we have,

$$\begin{aligned} & \alpha P\mu - \frac{1}{P}(\alpha + 1)A\lambda_i^e\lambda_{-i}^e - \frac{1}{P}(\alpha + 1)A(\lambda_i^e)^2 \\ & - \alpha P\lambda_{-i}^e - (\alpha + 1)P\lambda_i^e + \frac{1}{P}\alpha A\lambda_i^e\mu = 0 \\ \Rightarrow & P(\alpha\mu - \alpha\lambda_{-i}^e - (\alpha + 1)\lambda_i^e) \\ & = \frac{1}{P}[(\alpha + 1)A\lambda_i^e\lambda_{-i}^e + (\alpha + 1)A(\lambda_i^e)^2 - \alpha A\lambda_i^e\mu] \\ & \text{Since } \lambda^* = \frac{\mu\alpha}{\alpha+1} \text{ implies } \alpha\mu = (\alpha + 1)\lambda^*, \text{ we have} \\ & P[(\alpha + 1)\lambda^* - \alpha\lambda_{-i}^e - (\alpha + 1)\lambda_i^e] \\ & = \frac{1}{P}A\lambda_i^e[(\alpha + 1)(\lambda_{-i}^e + \lambda_i^e) - (\alpha + 1)\lambda^*] \\ \Rightarrow & P[(\alpha + 1)(\lambda^* - \lambda_{-i}^e - \lambda_i^e) + \lambda_{-i}^e] \\ & = \frac{1}{P}A\lambda_i^e(\alpha + 1)(\lambda_{-i}^e + \lambda_i^e - \lambda^*) \end{aligned} \quad (22)$$

Note that  $P\lambda_i + P\lambda_{-i} = \lambda^*$  implies that  $\lambda_i^e + \lambda_{-i}^e = \lambda^*$ . So the right-hand side of the equality is 0. While the left-hand side of the equality is  $P\lambda_{-i}^e$ . Since  $\lambda_{-i}^e \neq 0$ , so  $P\lambda_{-i}^e \neq 0$ . Thus the left-hand side of the equality is not 0 and this leads to a contradiction. Therefore Theorem 5 holds.  $\square$

Theorem 5 shows that we could not design a linear packet dropping policy with PoA = 1. The following theorem shows that we could design an incentive packet dropping policy such that PoA could be arbitrarily close to 1.

**Theorem 6.** Given any  $\epsilon$ , there exists a linear packet dropping policy such that  $1 < PoA \leq 1 + \epsilon$ .

**Proof.** Note (22) in the proof of Theorem 5 implies:

$$P^2[(\alpha + 1)(\lambda^* - \lambda_{-i}^e - \lambda_i^e) + \lambda_{-i}^e] = A\lambda_i^e(\alpha + 1)(\lambda_{-i}^e + \lambda_i^e - \lambda^*) \quad (23)$$

The right-hand side of (23) is greater than 0 only when  $\lambda_{-i}^e + \lambda_i^e < \lambda^*$  (note that  $A < 0$ ). Then given  $A$  and  $\lambda_i^e, \lambda_{-i}^e$  such that  $\lambda_{-i}^e + \lambda_i^e < \lambda^*$ , we will have a solution for  $P^2$  and thus we could get the value of  $D$ .

This means that we can design a packet dropping scheme such that it has a NE that satisfies  $p\lambda_i + p\lambda_{-i} \rightarrow \lambda^*$  from the left side (left approximation). If we can further prove that the NE is unique in this game (see Theorem 8), then give any  $\epsilon > 0$ , we could find a linear packet dropping policy at the server such that  $1 < PoA \leq 1 + \epsilon$ .  $\square$

We propose Algorithm 1 to show how to design the parameters  $r_1$  and  $r_2$  in our proposed incentive packet dropping policy to achieve a desired PoA such that  $1 < PoA \leq 1 + \epsilon$  given any  $\epsilon$ . We denote  $\lambda_e = \sum \lambda_i^e$ . Line 1 ensures that the sum of the rates will be less than  $\mu$  before the server starts to drop packets.  $\tilde{p}$  is the value of  $P(\Sigma\lambda)$  at the desired NE which is derived from the desired PoA.  $\tilde{p} = 1 - Pr\{\text{the packet dropping probability at desired NE}\}$ . Line 2 is the calculation of desired NE. The choice of  $\tilde{\lambda}$  is based on the desired value of PoA, i.e., given  $\epsilon > 0$ , we could accordingly derive the value of a desired sum rate such that  $1 < PoA \leq 1 + \epsilon$ . Since  $(\tilde{\lambda}_e, \tilde{p})$  is a solution of  $P(\Sigma\lambda)$ , line 3 shows how to therefore get the expression of  $A$  and  $D$ . Then at line 4, we could solve Eq. (21) given all the values above and get the value of  $r_2$ . Based on the result of  $r_2$ , the value of  $r_1$  is calculated.

Note that (21) is a quadratic equation for the parameters  $A$  and  $D$  given the values of all the other variables. But

**Algorithm 1** Parameter calculation for incentive packet dropping scheme.

**Input:** PoA bound parameter  $\epsilon$

**Output:**  $r_1$  and  $r_2$  of our proposed incentive packetdropping policy in (20) such that  $1 < PoA \leq 1 + \epsilon$ .

- 1: Pick any  $\tilde{p}$  such that  $\frac{a}{a+1} < \tilde{p} < 1$ .
- 2: Calculate a desired sum rate  $\tilde{\lambda}$ , of which

$$\tilde{\lambda} = \left\{ \frac{\tilde{\lambda}_e}{m\tilde{p}}, \frac{(m-1)\tilde{\lambda}_e}{m\tilde{p}} \right\} \quad (24)$$

is the desired NE such that  $1 < PoA \leq 1 + \epsilon$ . Note that  $\lambda_e = \tilde{p}\lambda$ .

- 3: Suppose  $P(\Sigma\lambda_i) = A\Sigma\lambda_i + D$  pass through the point  $(\tilde{\lambda}, \tilde{p})$ . Then we have

$$A = \frac{\tilde{p}}{\tilde{\lambda} - r_2}, D = -\frac{\tilde{p}r_2}{\tilde{\lambda} - r_2} \quad (25)$$

- 4: Insert the above values of the variables into (21) and get the value of  $r_2$ .
- 5: Insert the value of  $r_2$  into (25) and get the value of  $A$ . Then  $r_1 = \frac{1}{A} + r_2$ .

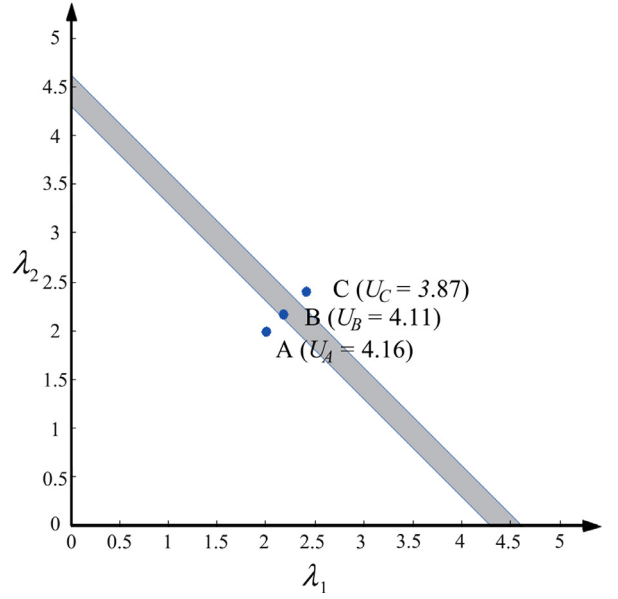


Fig. 4. An example of our incentive packet dropping scheme.

with Algorithm 1, we could always find a unique solution as stated in Theorem 7.

**Theorem 7.** Algorithm 1 yields a unique linear packet dropping scheme, i.e., unique values for  $r_1$  and  $r_2$  for any PoA bound.

**Proof.** After inserting the value  $\tilde{p}$ , and  $\tilde{\lambda} = \left\{ \frac{\tilde{\lambda}_e}{m\tilde{p}}, \frac{(m-1)\tilde{\lambda}_e}{m\tilde{p}} \right\}$  into (21) at line 4, we have the equality

$$\tilde{p}^2 \left[ (\alpha + 1)(\lambda^* - \tilde{\lambda}_e) + \frac{(m-1)\tilde{\lambda}_e}{m\tilde{p}} \right] = \frac{\tilde{p}}{\tilde{\lambda}_e/\tilde{p} - r_2} \frac{\tilde{\lambda}_e}{m\tilde{p}} (\alpha + 1)(\tilde{\lambda}_e - \lambda^*) \quad (26)$$

It is obvious that the above is a linear equation of the variable  $r_2$ . And thus we could get a unique solution of  $r_2$ . Therefore, there is always a unique solution of  $r_1$  and  $r_2$  provided by Algorithm 1.  $\square$

Our proposed packet dropping scheme is similar to the Random Early Detection (RED) algorithm. It is simple and easy to be implemented with low overhead at the server. Fig. 4 shows an example of our linear packet dropping policy with  $\mu = 6$ ,  $m = 2$  and  $\alpha = 2$ . (12) and (13) are used to calculate the utility and PoA. Point A represents  $\lambda^*$ , which is then calculated to be 4 ( $\lambda_1 = \lambda_2 = 2$ ). We assume  $\tilde{p} = 0.9$ , PoA bound parameter  $\epsilon = 0.05$ . Implementing Algorithm 1 with Matlab, we pick  $\tilde{\lambda}_e = 3.9$ , we then get  $r_1 = 4.3012$ ,  $r_2 = 4.6222$ ,  $A = -3.1154$ ,  $D = 14.4000$ . Point B represents the Nash Equilibrium with our proposed packets dropping policy. Point C represents the Nash Equilibrium of the original game without packet dropping policy. The shaded area shows the cases where packet dropping happens. For the comparison, the utilities are shown in the figure and we can see that PoA is improved from 1.3396 to 1.0455.

## 10. Uniqueness of NE

If we use the packet dropping scheme in Algorithm 1 we are guaranteed that the game  $G_p$  always has a NE with the desired PoA bound. Now our question is whether the scheme yields a unique NE. This is important not only for finding out whether our proposed scheme is efficient but also for the convergence issues. As surveyed in [22], there are not many general results on equilibrium uniqueness. We were unable to find any existing theorem that we could use directly to prove the uniqueness of NE in our M/M/1 queueing game. This makes the analysis of this incentive design problem more challenging.

**Theorem 8.** *There is a unique NE for the M/M/1 Game with the linear packet dropping scheme described in Algorithm 1.*

To prove 8, we first prove the following three lemmas.

**Lemma 9.**  $|A|$  increases monotonically as  $(\lambda^* - \tilde{\lambda}_e)$  decreases where  $\tilde{\lambda}_e$  is the total rate of all users at desired NE (as in Algorithm 1).

**Proof.** Note that (26) is equivalent to:

$$\begin{aligned} \tilde{p}^2 \left[ (\alpha + 1)(\lambda^* - \tilde{\lambda}_e) + \frac{(m-1)\tilde{\lambda}_e}{m\tilde{p}} \right] \\ = A \frac{\tilde{\lambda}_e}{m\tilde{p}} (\alpha + 1)(\tilde{\lambda}_e - \lambda^*) \end{aligned}$$

This means

$$\frac{(m-1)\tilde{\lambda}_e}{m\tilde{p}(\lambda^* - \tilde{\lambda}_e)} = (\alpha + 1) \left( |A| \frac{\tilde{\lambda}_e}{m\tilde{p}} - \tilde{p}^2 \right) \quad (27)$$

Note that as  $(\lambda^* - \tilde{\lambda}_e)$  decreases,  $\tilde{\lambda}_e$  increases and  $\frac{1}{\lambda^* - \tilde{\lambda}_e}$  increases, so the left-hand side of (27) increase. This implies  $|A|$  increases, and thus Lemma 9.  $\square$

**Lemma 10.**  $\forall \lambda_{-i} < r_1$ ,  $\frac{\partial U_i(\lambda_i, \lambda_{-i})}{\partial \lambda_i} > 0$  at  $\lambda_i = (r_1 - \lambda_{-i})^+$ .

**Proof.** Note that  $P = 1$  at  $r_1$ , then

$$\begin{aligned} \frac{\partial U_i(\lambda_i, \lambda_{-i})}{\partial \lambda_i} \\ = \alpha\mu - (\alpha + 1)A\lambda_i\lambda_{-i} - (\alpha + 1)A\lambda_i^2 - \alpha\lambda_{-i} \\ - (\alpha + 1)\lambda_i + \alpha A\lambda_i\mu \\ = \alpha\mu - \alpha\lambda_{-i} - (\alpha + 1)\lambda_i \\ - A\lambda_i[(\alpha + 1)(\lambda_{-i} + \lambda_i) - \alpha\mu] \\ = \alpha\mu - (\alpha + 1)(\lambda_{-i} + \lambda_i) + \lambda_{-i} \\ - A\lambda_i[(\alpha + 1)(\lambda_{-i} + \lambda_i) - \alpha\mu] \\ = [\alpha\mu - (\alpha + 1)(\lambda_{-i} + \lambda_i)](1 + A\lambda_i) + \lambda_{-i} \\ = [\alpha\mu - (\alpha + 1)r_1](1 + A\lambda_i) + \lambda_{-i} \\ = [(\alpha + 1)\lambda^* - (\alpha + 1)r_1](1 + A\lambda_i) + \lambda_{-i} \\ = (\alpha + 1)(\lambda^* - r_1)(1 + A(r_1 - \lambda_{-i})) + \lambda_{-i} \end{aligned}$$

Denote (28) as  $g(\lambda_{-i})$ . Then,

$$\frac{\partial g}{\partial \lambda_{-i}} = -(\alpha + 1)(\lambda^* - r_1)A + 1. \quad (28)$$

When  $|A|$  is large enough such that  $r_1 > \lambda^*$  and  $|A| > \frac{1}{(\alpha+1)(r_1-\lambda^*)}$ ,  $\frac{\partial g}{\partial \lambda_{-i}} < 0$ .

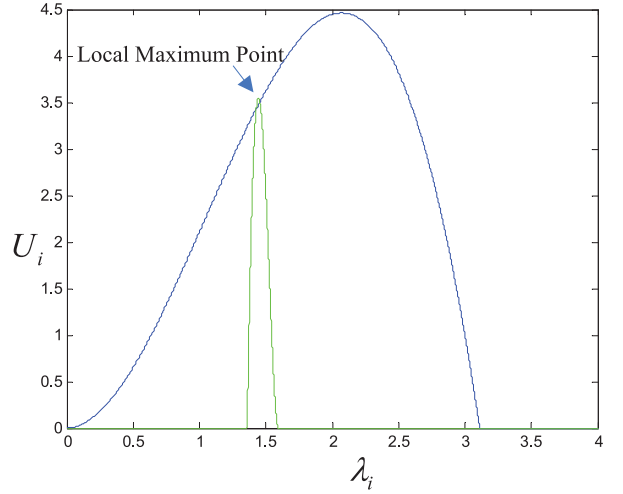


Fig. 5. Illustration of local maximum point of  $U_i(\lambda_i, \lambda_{-i})$ .

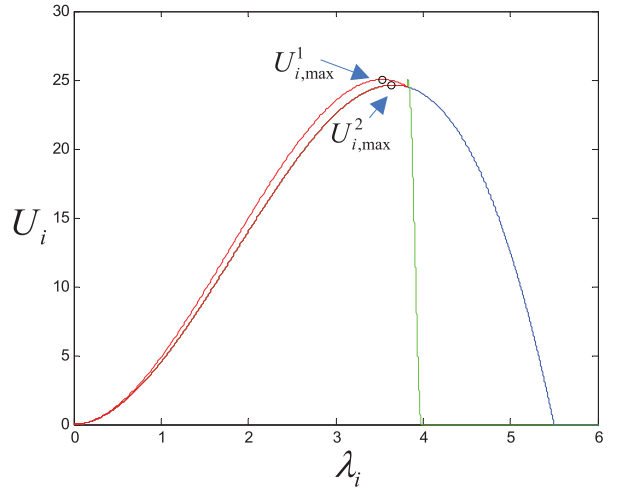


Fig. 6. Illustration of local maximum point of  $U_{i,max}^1$  and  $U_{i,max}^2$ .

From Lemma 9 we know that as  $\tilde{\lambda}_e$  gets closer to  $\lambda^*$ ,  $|A|$  increases. This means that when we design a dropping policy with PoA approaching to 1,  $|A|$  could be large enough such that  $r_1 > \lambda^*$  and  $|A| > \frac{1}{(\alpha+1)(r_1-\lambda^*)}$ .

So  $g(\lambda_i)$  achieves the minimum value when  $\lambda_{-i} = r_1$ . Then

$$\begin{aligned} \frac{\partial U_i(\lambda_i, \lambda_{-i})}{\partial \lambda_i} &> (\alpha + 1)(\lambda^* - r_1) + r_1 \\ &= (\alpha + 1)\lambda^* - \alpha r_1 \\ &= \alpha\mu - \alpha r_1 > 0. \end{aligned} \quad (29)$$

$\square$

Lemma 10 implies that given  $\lambda_{-i}$ ,  $U_i(\lambda_i, \lambda_{-i})$  which is a function of  $\lambda_i$ , has a maxima at  $r_1 - \lambda_{-i} < \lambda_i < r_2 - \lambda_{-i}$  as shown in Fig. 5.

Given  $\lambda_{-i}$ , if  $\lambda_{-i} + \frac{(\mu-\lambda_{-i})\alpha}{\alpha+1} < r_1$ , then  $U_i(\lambda_i, \lambda_{-i})$  will reach a local maximal point when  $\lambda_i = \frac{(\mu-\lambda_{-i})\alpha}{\alpha+1}$  as shown in Fig. 6. Note that  $P(\sum \lambda_i) = 1$  at this point. We



denote this local maximal value as  $U_{i,max}^1$ . Then  $U_i(\lambda_i, \lambda_{-i})$  will reach another local maximal point with  $\lambda_i^e = \frac{(\mu - \lambda_{-i})\alpha}{\alpha + 1}$ . Note that  $P(\Sigma \lambda_i) < 1$ . We denote this local maximal value as  $U_{i,max}^2$ .

**Lemma 11.** Given  $\lambda_{-i}$ , if  $\lambda_{-i} + \frac{(\mu - \lambda_{-i})\alpha}{\alpha + 1} < r_1$ ,  $U_{i,max}^1 < U_{i,max}^2$ .

**Proof.** Note that

$$U_{i,max}^1 = (\lambda_i^1)^\alpha (\mu - (\lambda_i^1) - \lambda_{-i}),$$

$$\text{where } \lambda_i^1 = \frac{(\mu - \lambda_{-i})\alpha}{\alpha + 1}.$$

$$U_{i,max}^2 = (P\lambda_i^2)^\alpha (\mu - (P\lambda_i^2) - P\lambda_{-i})$$

Denote  $\lambda_i^e = P\lambda_i^2$ . Note that  $\lambda_i^e$  ranges from  $r_1$  to 0 and we have  $r_1 > \lambda_i^1$ . Also note that

$$\max_{\lambda_i^e} (\lambda_i^e)^\alpha (\mu - \lambda_i^e - P\lambda_{-i}) > \max_{\lambda_i^e} (\lambda_i^e)^\alpha (\mu - \lambda_i^e - \lambda_{-i})$$

since  $P\lambda_{-i} < \lambda_{-i}$ . Thus,  $U_{i,max}^2 > U_{i,max}^1$ .  $\square$

We show in Lemma 11 that given  $\lambda_{-i}$ ,  $U_i(\lambda_i, \lambda_{-i})$  achieves the maximal point when  $P < 1$  under the condition that  $\lambda_{-i} + \frac{(\mu - \lambda_{-i})\alpha}{\alpha + 1} \leq r_1$ . When  $\lambda_{-i} + \frac{(\mu - \lambda_{-i})\alpha}{\alpha + 1} > r_1$ , there is only one maximal point for  $U_i(\lambda_i, \lambda_{-i})$ . When  $\lambda_{-i} + \frac{(\mu - \lambda_{-i})\alpha}{\alpha + 1} = r_1$ ,  $U_{i,max}^1$  and  $U_{i,max}^2$  will overlap, and since  $r_1 > \lambda^*$ , this case could not result in a NE and thus we do not consider this case.

Lemma 9, Lemma 10 and Lemma 11 show that given  $\lambda_{-i}$ ,  $U_i(\lambda_i, \lambda_{-i})$  achieves the maximal point when  $P < 1$ . Then we will prove the uniqueness of NE based on the expression  $P(\Sigma \lambda_i) = A(\lambda_i + \lambda_{-i}) + D$ .

**Proof of Theorem 8.** Suppose the  $G_p$  has more than one NE. Note that the game is symmetric, so there must exist one NE  $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_m\}$ , such that  $\exists i, j, \lambda_i \neq \lambda_j$ . This means there exists a NE  $\lambda = \{\lambda_i, \lambda_{-i}\}$  and a constant  $c$  such that  $\lambda = \{\lambda_i + c, \lambda_{-i} - c\}$  is also a NE.

We insert  $\lambda = \{\lambda_i, \lambda_{-i}\}$  into (23) and we get:

$$\begin{aligned} & [A(\lambda_i + \lambda_{-i}) + D]^2 [(\alpha + 1)(\lambda^* - \lambda_{-i}^e - \lambda_i^e) + \lambda_{-i}^e] \\ &= A\lambda_i^e (\alpha + 1)(\lambda_{-i}^e + \lambda_i^e - \lambda^*) \\ &\Rightarrow \frac{(\alpha + 1)(\lambda^* - \lambda_{-i}^e - \lambda_i^e) + \lambda_{-i}^e}{\lambda_i^e} \\ &= \frac{1}{[A(\lambda_i + \lambda_{-i}) + D]^2} A(\alpha + 1)(\lambda_{-i}^e + \lambda_i^e - \lambda^*) \end{aligned} \quad (30)$$

We also insert  $\lambda = \{\lambda_i + c, \lambda_{-i} - c\}$  into (23) and denote  $\tilde{c} = \frac{c}{A(\lambda_i + \tilde{c} + \lambda_{-i} - \tilde{c}) + D} = \frac{c}{A(\lambda_i + \lambda_{-i}) + D}$ . We have:

$$\begin{aligned} & [A(\lambda_i + \tilde{c} + \lambda_{-i} - \tilde{c}) + D]^2 \\ & [(\alpha + 1)(\lambda^* - \lambda_{-i}^e - \tilde{c} - \lambda_i^e + \tilde{c}) + \lambda_{-i}^e - \tilde{c}] \\ &= A(\lambda_i^e + \tilde{c})(\alpha + 1)(\lambda_{-i}^e - \tilde{c} + \lambda_i^e + \tilde{c} - \lambda^*) \end{aligned}$$

$$\begin{aligned} &\Rightarrow \frac{(\alpha + 1)(\lambda^* - \lambda_{-i}^e - \lambda_i^e) + \lambda_{-i}^e - \tilde{c}}{\lambda_i^e + \tilde{c}} \\ &= \frac{1}{[A(\lambda_i + \lambda_{-i}) + D]^2} A(\alpha + 1)(\lambda_{-i}^e + \lambda_i^e - \lambda^*) \end{aligned} \quad (31)$$

Note that the right-hand side of (30) and (31) are the same and thus the left-hand side of (30) and (31) are equal to each other. So,

$$\begin{aligned} & \frac{(\alpha + 1)(\lambda^* - \lambda_{-i}^e - \lambda_i^e) + \lambda_{-i}^e}{\lambda_i^e} \\ &= \frac{(\alpha + 1)(\lambda^* - \lambda_{-i}^e - \lambda_i^e) + \lambda_{-i}^e - \tilde{c}}{\lambda_i^e + \tilde{c}} \\ &\Rightarrow (\lambda_i^e + \tilde{c})(\alpha + 1)(\lambda^* - \lambda_{-i}^e - \lambda_i^e) + (\lambda_i^e + \tilde{c})\lambda_{-i}^e \\ &= \lambda_i^e (\alpha + 1)(\lambda^* - \lambda_{-i}^e - \lambda_i^e) + \lambda_i^e (\lambda_{-i}^e - \tilde{c}) \end{aligned}$$

$$\begin{aligned} &\Rightarrow \tilde{c}(\alpha + 1)(\lambda^* - \lambda_{-i}^e - \lambda_i^e) + \tilde{c}(\lambda_{-i}^e + \lambda_i^e) = 0 \\ &\Rightarrow \tilde{c}[(\alpha + 1)\lambda^* - \alpha(\lambda_{-i}^e + \lambda_i^e)] = 0 \\ &\Rightarrow \tilde{c}[\alpha\mu - \alpha(\lambda_{-i}^e + \lambda_i^e)] = 0 \\ &\Rightarrow \tilde{c}[\mu - (\lambda_{-i}^e + \lambda_i^e)] = 0 \\ &\Rightarrow \frac{c\alpha[\mu - (\lambda_{-i}^e + \lambda_i^e)]}{A(\lambda_i + \lambda_{-i}) + D} = 0. \end{aligned}$$

Note that  $\mu - (\lambda_{-i}^e + \lambda_i^e) > 0$ . This implies that  $c = 0$  and therefore Theorem 8 holds.  $\square$

## 11. Best response dynamics and convergence

In this section, we show that the best response dynamic [7], a simple learning mechanism, will lead the queuing game to converge to the pure Nash equilibrium.

Best response dynamic is a straightforward updating rule which proceeds as follows: whenever player  $i$  has an opportunity to revise her strategy, she will choose the best response to the actions of all the other players in the previous round. Mathematically, for a game  $G = (N, \{A_i\}, \{U_i\})$ , let  $a_i^t$  denotes the action of player  $i$  in iteration  $t$ ,

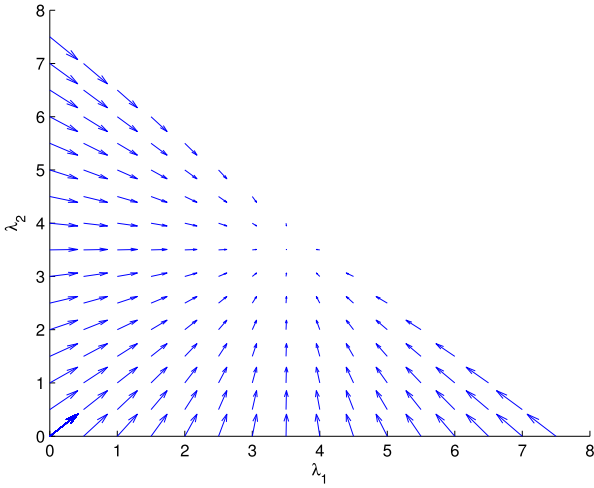
$$a_i^t = \arg \max_{a_i^t \in A_i} U_i(a_i^t, a_{-i}^{t-1}). \quad (32)$$

In general, the best response dynamic is not guaranteed to converge. However, if the process does converge, it is guaranteed to converge to a NE. Now, we want to investigate the convergence of our proposed M/M/1 Game with the packet dropping scheme, denoted as  $G_p$  in the previous section.

**Theorem 12.** Best response dynamic will converge to the unique NE for the M/M/1 Game with the proposed packet dropping scheme.

**Proof.** There is an important result about the convergence for ordinal potential game as shown in Theorem 21 in [7]: if  $G$  is an ordinal potential game with a compact action space and a continuous potential function, then the best response dynamic will (almost surely) either converge to a NE or every limit point of the sequence will be a NE.

We have showed in Theorem 3 that  $G_p$  is an ordinal potential game, and although the original definition of the



**Fig. 7.** Quiver plot for a two user example with  $\mu = 10$ ,  $\alpha = 2$ ,  $r_1 = 7.0321$  and  $r_2 = 7.8222$ . The vector length are scaled to  $\frac{1}{14}$  of the original length.

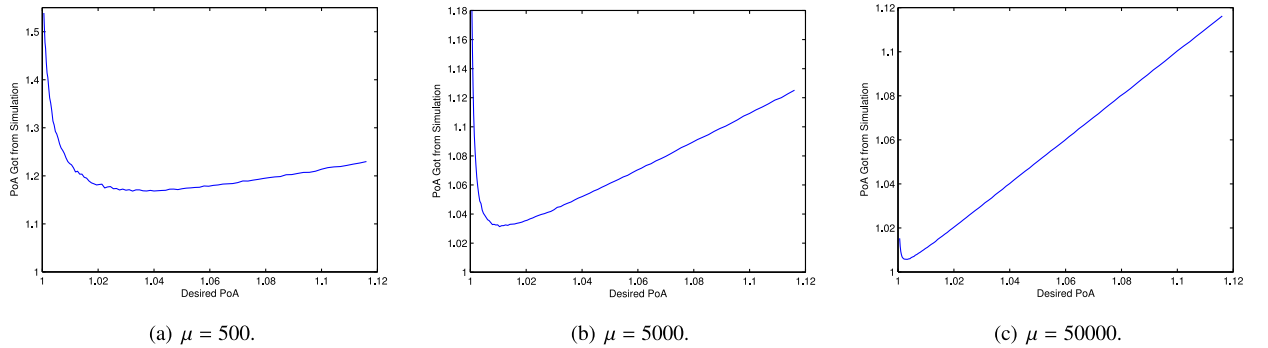
game does not have a compact action space, the equivalent modification as shown in (17) has a compact action space. We can also see that the potential function is continuous. We have also proved in Theorem 8 that there is a unique NE. Thus Theorem 12 holds.  $\square$

Fig. 7 is an illustration of Theorem 12 by the quiver plot. In Fig. 7, on the lower triangle of a grid (i.e., the

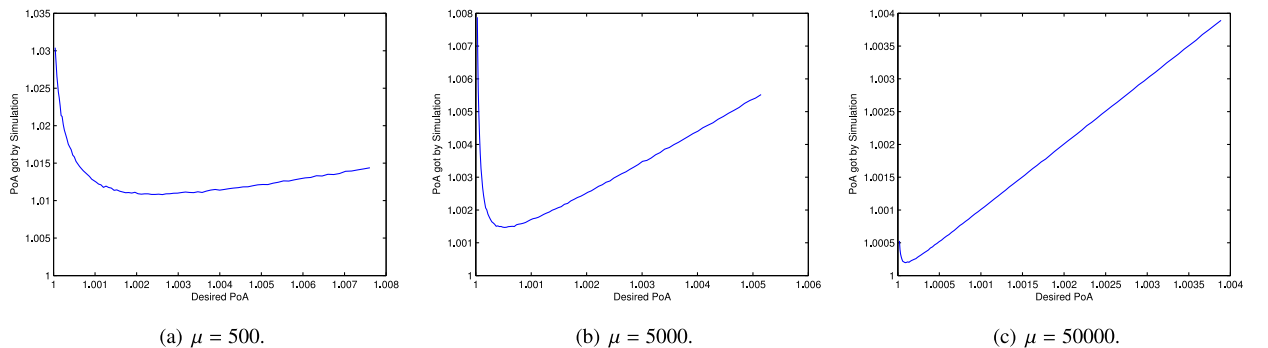
feasible operating domain), we plot vector summation for a two-user rate control queuing game with  $\mu = 10$ ,  $\alpha = 2$ ,  $r_1 = 7.0321$  and  $r_2 = 7.8222$ . At each point, the vectors' projections on  $\lambda_1$  and  $\lambda_2$  represent the best response for the corresponding users in next iteration. To make the plot neat, the length of each vector is scaled to  $\frac{1}{14}$  of the original length. The figure shows that at each point, the players in the best response dynamic move towards the equilibrium point. The length of the best response vectors are proportional to the distance from the equilibrium point. At the equilibrium point, the step size of the movement in the next iteration tends to zero, which implies the convergence of the best response dynamic.

## 12. Impact of arrival rate estimation

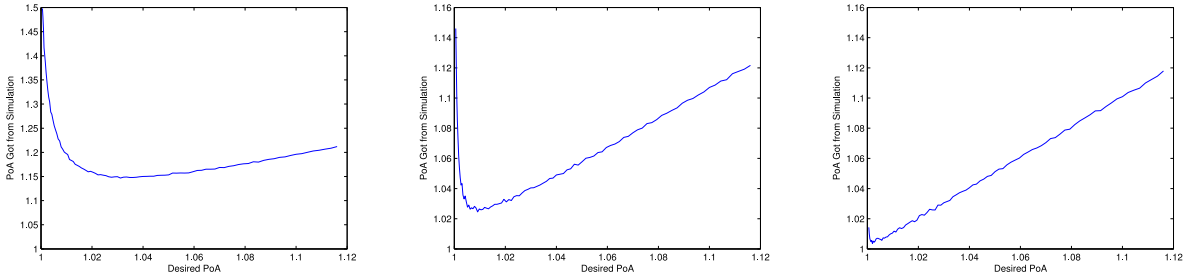
For a real system implementation, the server needs to estimate the total arrival rate from users in order to apply the incentive packet dropping scheme. The packets arrive randomly over time, so there will be a difference between the estimated total rate and the average total rate. This inaccuracy will cause a loss in the PoA. While applying the packets dropping scheme, we note that the closer to 1 the desired PoA is, the steeper (on the linear part) the packet dropping scheme is, and therefore, the greater the impact of estimation inaccuracy will be. So as the desired PoA approaches 1, on the one hand the PoA of the real system should increase due to the implementation of the incentive scheme, but on the other hand, the sensitivity to



**Fig. 8.** Simulation results of PoA under different service rates of a 3-user system with  $\alpha = 2$  (sum-utility definition).

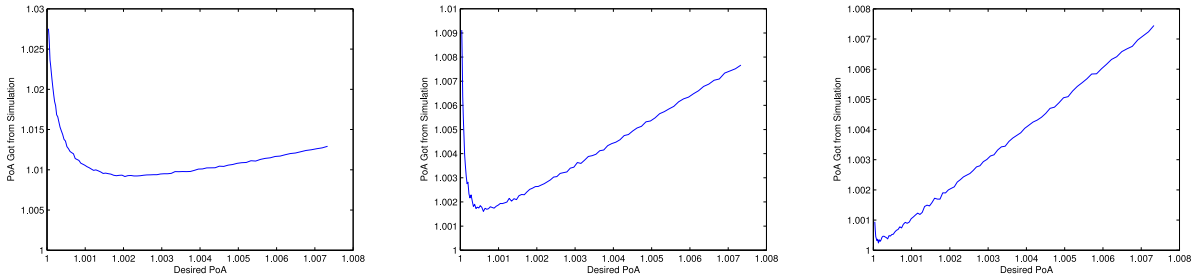


**Fig. 9.** Simulation results of PoA under different service rates of a 3-user system with  $\alpha = 2$  (sum-log-utility definition).



(a) The estimated total rate got based on the instantaneous arrival rate. (b) The estimated total rate got by averaging the continuous arrival rates in 10 slots. (c) The estimated total rate got by averaging the continuous arrival rates in 100 slots.

**Fig. 10.** Impact of estimation length on PoA of a 3-user system with  $\mu = 600$ ,  $\alpha = 2$  (sum-utility definition).



(a) The estimated total rate got based on the instantaneous arrival rate. (b) The estimated total rate got by averaging the continuous arrival rates in 10 slots. (c) The estimated total rate got by averaging the continuous arrival rates in 100 slots.

**Fig. 11.** Impact of estimation length on PoA of a 3-user system with  $\mu = 600$ ,  $\alpha = 2$  (sum-log-utility definition).

estimation error will reduce the gain in PoA. Therefore, the achieved PoA in practice may not be arbitrarily close to 1.

We show this fact in Figs. 8–11 where we show simulation results from a 3-user queue with  $\alpha = 2$ . In these simulations, we discretize time into slots. The server estimates the mean arrival rate from the previous time slot and applies the packet dropping function corresponding to a desired PoA to all packets in the current slot (the users contribute arrivals at a constant rate that corresponds to the equilibrium input for this desired PoA). The running time for all the simulations is  $10^5$  time slots. Figs. 8 and 9 show the simulation results of PoA under different service rates for both the sum-utility definition and sum-log-utility definition with the instantaneous arrival rate to the server as the estimated arrival rate. We can see that as the service rate varies from 500 to 5000 packets per time slot, the optimal point of PoA (i.e. the lowest achievable PoA) is getting closer to 1 because the estimation inaccuracy decreases. Also, the empirically achieved PoA is getting closer to the desired PoA.

Note that when the service rate is low, the achievable PoA we get could be very bad as shown in Figs. 8(a) and 9(a). In these cases, using more history data/longer estimation lengths will help to increase the estimation accuracy and improve PoA. The comparison results under different estimation lengths for  $\mu = 600$  packets per time slot are shown in Figs. 10 and 11. These simulation results illustrate a tradeoff between the optimal PoA and the overhead in computing and storage: while estimating with more history data will increase the estimation accuracy

and therefore increase PoA, it increases the overhead in terms of computing and storage.

### 13. Conclusion

In this paper, we have designed a novel incentive mechanism for M/M/1 queueing games with throughput-delay tradeoffs. Because the original game yields an inefficient Nash equilibrium, we propose to implement a linear packet dropping mechanism at the router. We show how the parameters of this mechanism can be optimized to ensure system efficiency that is arbitrarily close to the social welfare solution. Further, we prove that the proposed modification has a unique NE, and that the simple best response dynamics converges to this solution. Future work could consider extensions of this work to consider non-homogeneous users, other queuing models beyond the M/M/1 model, more complex arrangements of multiple routers in a network, as well as other system issues that may arise in practical implementations.

### Appendix A. Proof of Theorem 1

**Proof.**

$$\begin{aligned} & \sum_{i=1}^m \log \left[ \lambda_i^\alpha \left( \mu - \sum_{i=1}^m \lambda_i \right) \right] \\ &= \alpha \log \left( \prod_{i=1}^m \lambda_i \right) + m \log (\mu - \lambda) \end{aligned}$$

$$\begin{aligned} &\leq \alpha \log \left( \frac{\sum \lambda_i}{m} \right)^m + m \log(\mu - \lambda) \\ &= m \log \left( \frac{\lambda}{m} \right)^\alpha (\mu - \lambda) \end{aligned} \quad (\text{A.1})$$

Denote  $f(\lambda) = \lambda^\alpha (\mu - \lambda)$ . So maximize (12) is equivalent to maximize  $f(\lambda)$ . Take the derivative of  $f(\lambda)$  and let it equals 0. We get:

$$\frac{\partial f}{\partial \lambda} = 0 \Rightarrow \lambda^* = \frac{\mu \alpha}{\alpha + 1}$$

Note that equality holds in (A.1) only when  $\lambda_1 = \lambda_2 = \dots = \lambda_m$ . This implies  $\lambda_i^* = \frac{\lambda^*}{m} = \frac{\mu \alpha}{m(\alpha + 1)}$   $\square$

## Appendix B. Proof of Theorem 4

**Proof.** ( $\Leftarrow$ )

Suppose  $\sum \lambda'_i = \lambda^*$ .  $\forall i$ , let  $\frac{\partial U(\lambda_i, \lambda'_{-i})}{\partial \lambda_i} = 0$ . We get the optimal point  $\lambda_i^{**} = \frac{(\mu - \sum_{j \neq i} \lambda'_j) \alpha_i}{\alpha_i + 1}$ .

Note that

$$\begin{aligned} \lambda'_i &= \lambda^* - \sum_{j \neq i} \lambda'_j = \frac{\alpha \mu}{\alpha + 1} - \sum_{j \neq i} \lambda'_j \\ &= \frac{(\mu - \sum_{j \neq i} \lambda'_j) \alpha_i}{\alpha_i + 1} < \frac{(\mu - \sum_{j \neq i} \lambda'_j) \alpha_i}{\alpha_i + 1} = \lambda_i^{**} \end{aligned} \quad (\text{B.1})$$

Also,  $\forall \lambda_i < \lambda_i^{**}$ ,  $\frac{\partial U(\lambda_i, \lambda'_{-i})}{\partial \lambda_i} > 0$ , which means  $U(\lambda_i, \lambda'_{-i})$  increases monotonically with respect to  $0 \leq \lambda_i < \lambda_i^{**}$ , so  $U(\lambda'_i, \lambda'_{-i}) > U(\lambda_i, \lambda'_{-i})$ ,  $\forall \lambda_i \in [0, \lambda'_i]$ . Also note that  $U(\lambda_i, \lambda'_{-i}) = 0$ ,  $\forall \lambda_i \in (\lambda_i^*, \mu - \sum_{j \neq i} \lambda'_j)$ . Hence  $\lambda'_i \in B_i(\lambda'_{-i})$ .

Therefore,  $\lambda'$  is a N.E.

( $\Rightarrow$ )

Suppose  $\lambda'$  is a N.E.,  $\forall i$ ,  $\lambda'_i \in B_i(\lambda'_{-i})$ .

$\forall \lambda'_{-i}$ , consider the following two cases:

$$(1) \sum_{j \neq i} \lambda'_j \leq \lambda^*$$

Denote  $\lambda''_i = \lambda^* - \sum_{j \neq i} \lambda'_j$ . Then  $\langle \lambda''_i, \lambda'_{-i} \rangle$  is a N.E.,  $\lambda''_i \in B_i(\lambda'_{-i})$ .  $U(\lambda''_i, \lambda'_{-i}) = U(\lambda'_i, \lambda'_{-i}) > 0$ . So  $\lambda'_i \leq \lambda^* - \sum_{j \neq i} \lambda'_j = \lambda''_i$  (if not so,  $U(\lambda'_i, \lambda'_{-i}) = 0$ ).

Note that  $U(\lambda_i, \lambda'_{-i})$  increases monotonically with respect to  $0 \leq \lambda_i < \lambda'_i$ . Therefore,  $\lambda' = \lambda''$ .  $\sum \lambda'_i = \lambda^*$ .

$$(2) \sum_{j \neq i} \lambda'_j > \lambda^*$$

Under this case, since  $\sum \lambda'_i > \lambda^*$ , we have  $B_i(\lambda'_{-i}) = 0$ ,  $\forall i$ . Then  $\sum_{j \neq i} \lambda'_j \geq \lambda^*$  holds for all  $i$ .  $\forall i$ ,  $\lambda'_i < \mu - \sum_{j \neq i} \lambda'_j < \mu - \lambda^* = \mu - \frac{\mu \alpha}{\alpha + 1} = \frac{\mu}{\alpha + 1}$ . So

$$\sum_{i=1}^m \lambda'_i < \frac{\mu m}{\alpha + 1} < \mu \Rightarrow m < \alpha + 1 \quad (\text{B.2})$$

However, note that

$$\lambda^* = \frac{\mu \alpha}{\alpha + 1} < \sum_{j \neq i} \lambda'_j < \frac{\mu(m-1)}{\alpha + 1} \Rightarrow m > \alpha + 1 \quad (\text{B.3})$$

Since (B.2) and (B.3) contradict each other, there is no N.E.  $\lambda'$  such that  $\sum_{j \neq i} \lambda'_j > \lambda^*$ .  $\square$

## References

- [1] Y. Gai, H. Liu, B. Krishnamachari, A packet dropping-based incentive mechanism for m/m/1 queues with selfish users, in: Proceedings of IEEE International Conference on Computer Communications, INFOCOM, 2011, IEEE, 2011, pp. 2687–2695.
- [2] V. Srivastava, J.O. Neel, A.B. MacKenzie, R. Menon, L.A. DaSilva, J.E. Hicks, J.H. Reed, R.P. Gilles, Using game theory to analyze wireless ad hoc networks, IEEE Commun. Surv. Tutor. 7 (1–4) (2005) 46–56.
- [3] A. Ozdaglar, R. Srikant, Incentives and pricing in communication networks, Algorithmic Game Theory, Cambridge University Press, 2007, pp. 571–591.
- [4] E. Altman, T. Boulogne, R. El-Azouzi, T. Jiménez, L. Wynter, A survey on networking games in telecommunications, Comput. Oper. Res. 33 (2) (2006) 286–311.
- [5] W. Saad, Z. Han, M. Debbah, A. Hjørungnes, T. Basar, Coalitional game theory for communication networks, IEEE Signal Process. Mag. 26 (5) (2009) 77–97.
- [6] M.J. Osborne, A. Rubinstein, A Course in Game Theory, MIT Press, 1994.
- [7] A.B. MacKenzie, L.A. DaSilva, Game Theory for Wireless Engineers, Synthesis Lectures on Communications, vol. 1(1), Morgan & Claypool Publishers, 2006, pp. 1–86.
- [8] S.M. Ross, Introduction to Probability Models, Academic Press, 2014.
- [9] A. Mankin, K. Ramakrishnan, RFC1254-gateway congestion control survey [online].
- [10] K. Bharath-Kumar, J. Jaffe, A new approach to performance-oriented flow control, IEEE Trans. Commun. 29 (4) (1981) 427–435.
- [11] C. Douligeris, R. Mazumdar, A game theoretic perspective to flow control in telecommunication networks, J. Frankl. Inst. 329 (2) (1992) 383–402.
- [12] Z. Zhang, C. Douligeris, Convergence of synchronous and asynchronous greedy algorithms in a multiclass telecommunications environment, IEEE Trans. Commun. 40 (8) (1992) 1277–1281.
- [13] D. Dutta, A. Goel, J. Heidemann, Oblivious aqm and nash equilibria, in: Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications, INFOCOM 2003, vol. 1, IEEE, 2003, pp. 106–113.
- [14] Y. Su, M. Van Der Schaar, Linearly coupled communication games, IEEE Trans. Commun. 59 (9) (2011) 2543–2553.
- [15] D. Monderer, L.S. Shapley, Potential games, Games Econ. Behav. 14 (1) (1996) 124–143.
- [16] S. Floyd, V. Jacobson, Random early detection gateways for congestion avoidance, IEEE/ACM Trans. Netw. 1 (4) (1993) 397–413.
- [17] M. Haviv, T. Roughgarden, The price of anarchy in an exponential multi-server, Oper. Res. Lett. 35 (4) (2007) 421–426.
- [18] T. Wu, D. Starobinski, On the price of anarchy in unbounded delay networks, in: Proceeding of the 2006 Workshop on Game Theory for Communications and Networks, ACM, 2006, p. 13.
- [19] A.A. Economides, J.A. Silvester, Multi-objective routing in integrated services networks: a game theory approach, in: Proceedings of the Tenth Annual Joint Conference of the IEEE Computer and Communications Societies on Networking in the 90s, INFOCOM'91, IEEE, 1991, pp. 1220–1227.
- [20] H. Liu, B. Krishnamachari, S. Kapadia, Game theoretic tools applied to wireless networks, in: D. Agrawal (Ed.), Encyclopedia of Ad Hoc and Ubiquitous Computing, 2008.
- [21] D. Bertsimas, V.F. Farias, N. Trichakis, The price of fairness, Oper. Res. 59 (1) (2011) 17–31.
- [22] S. Lasaulce, M. Debbah, E. Altman, Methodologies for analyzing equilibria in wireless games, IEEE Signal Processing Magazine, Special issue on Game Theory for Signal Processing 26 (5) (2009) 41–52.



ad-hoc networks.

**Yi Gai** received her B.E. and M.S. degrees in Electrical Engineering at Tsinghua University, China, in 2005 and 2007, respectively. She received her Ph.D. degree in the Department of Electrical Engineering, University of Southern California (USC), where she was a recipient of an Annenberg Graduate Fellowship, WiSE Merit Fellowship and Oakley Fellowship. She received the USC center for applied mathematical sciences (CAMS) prize in 2011. She is currently with Intel Labs as a research scientist. Her research interests are in the areas of wireless networks, machine learning, mobile networks, and



**Hua Liu** received her B.S. in Computer Science at Fudan University, China in 2001. She received her Ph.D. degree in the Department of Computer Science, University of Southern California (USC) in 2010. Her research interests are in the areas of game theory, wireless networks, database, and mobile networks.



**Bhaskar Krishnamachari** received his B.E. in Electrical Engineering at The Cooper Union, New York, in 1998, and his M.S. and Ph.D. degrees from Cornell University in 1999 and 2002, respectively. He is currently an Associate Professor and a Ming Hsieh Faculty Fellow in the Department of Electrical Engineering at the University of Southern California's Viterbi School of Engineering. His primary research interest is in the design and analysis of algorithms and protocols for next-generation wireless networks.