

## Research Article

# Evolutionary Game-Theoretic Solution for Virtual Routers with Padding Misbehavior in Cloud Computing

**Xia-an Bi, Xiaohui Wang, and Hong Liu**

*College of Mathematics and Computer Science, Hunan Normal University, Changsha, Hunan 410081, China*

Correspondence should be addressed to Xia-an Bi; [bixiaan@hnu.edu.cn](mailto:bixiaan@hnu.edu.cn)

Received 12 May 2015; Accepted 21 July 2015

Academic Editor: Son Nguyen

Copyright © 2015 Xia-an Bi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the development of cloud computing and virtualization, a physical router can be multiplexed as a large number of virtual routers. TCP-based interactive applications have an incentive to improve their performance by padding “junk packets” into the network among real communication packets. This padding misbehavior will upgrade short TCP flows from “mice” to “elephants” and consequently lead to network congestion and breakdown. This paper presents a detailed solution and analysis for describing the normal behavior and padding misbehavior of virtual routers. In particular, a system model for analyzing behavior of virtual routers is based on evolutionary game model, and, through analyzing the stability of the equilibrium points, the stable point is the solution to the problem. The clear evolutionary path of network applications with the normal behavior and padding misbehavior is analyzed by the corresponding graph. Then this paper gives the behavior control suggestions to effectively restrain the padding misbehavior and maintain stable high-throughputs of the router. The simulation results demonstrate that our solution can effectively restrain the padding misbehavior and maintain stable high-throughputs of the router simultaneously compared with the classical queue management.

## 1. Introduction

During the past decade, cloud computing has been significantly developed, which provides a large number of cloud services. Lots of cloud computing service providers appear, such as Amazon and Google. The basis of cloud computing is to build virtual networks to meet the needs of the cloud services, and the deployment of virtual network requires virtual routers [1]. An ideal goal of virtualization is that large numbers of virtual routers can simultaneously run on the same physical platform. Scaling to virtual routers poses serious challenges to the performance of routers, such as low memory access, low false positive probability, flexibility of update, and high scalability. With the rapid development of processor architecture and virtual networks, the operational mode has changed dramatically. First, the physical routing platform has been transforming from single-core processors to multicore processors. Second, a physical router needs to be multiplexed as a large number of virtual routers, as Figure 1 shows. Therefore, it is significant to study the efficient

algorithm of a physical router to support scalable virtual routers [2].

However, with the increasing demands of network bandwidth, network applications are also facing challenges. The challenges are to solve the problem of bandwidth competition in data transmission [3]. It is a more serious problem that a cloud computing service provider synchronously provides services for large numbers of different applications. In traditional network, it is well known that short TCP flows may experience significant performance degradations when they multiplex with long-lived TCP flows and UDP flows. The reason is that they tend to occupy more bandwidth resources for a long time and usually seize bandwidth in an unfriendly way [4]. Consequently, a new phenomenon appears. In order to obtain enough bandwidth resources, TCP-based interactive applications, such as gaming, telnet, or persistent HTTP, have an incentive to improve their performance. In particular, they send “junk packets” into the network among real communication packets, which is called padding misbehavior. This kind of misbehavior may

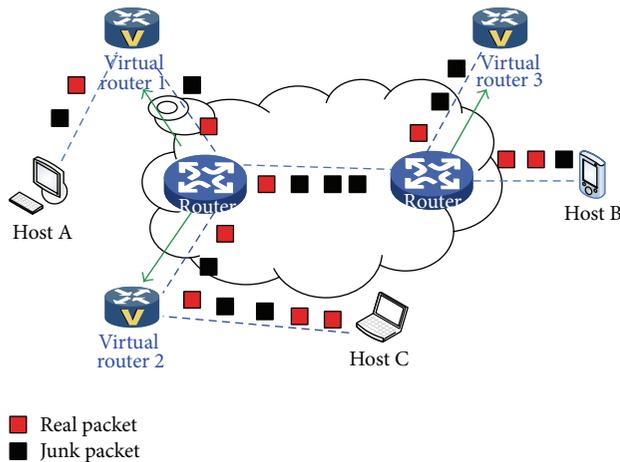


FIGURE 1: Padding misbehavior of network applications.

guarantee the bandwidth benefits of short TCP flows, but it is a tragedy to the cloud network, because the misbehavior will upgrade short TCP flows from “mice” to “elephants,” leading to network congestion eventually [5].

Unfortunately, whether at the transportation or the application levels, it is easy to upgrade an interactive application to a fully backlogged flow. The TCP-based interactive applications realize the padding misbehavior by utilizing virtual routers on one physical router. As shown in Figure 1, in a card game application, there are three terminals, Host A, Host B, and Host C, and they play cards together. The card game applications activate three virtual routers in the cloud computing network for the hosts playing the game. The red blocks are the real packets for their card game, while the black blocks are the junk packets. The three hosts apply this kind of padding misbehavior in order to continuously occupy the network bandwidth and improve their performance. On account of consuming huge bandwidth resources of the network, such misbehavior easily lead to network congestion [6].

In recent years, the studies of the above problems focus on the queue management schemes which are generally based on two packet processing types, Drop-tail and RED [7], for instance, ARED [8], BLUE [9], PI [10], and PAQM [11] algorithms. However, both network-based and endpoint-based mechanisms that check for TCP-friendliness are incapable of detecting any violation, simply because all flows are TCP friendly. In these methods several packets would be dropped without considering the padding misbehavior. So the exiting schemes could not effectively avoid such misbehavior and the terminals could not apply appropriate communication behavior.

Game theory has been a mature and hot topic with many applications in various research fields. In order to solve the above mentioned problems, the evolutionary game theory which extends traditional game analysis is utilized in this paper. Evolutionary game theory which was first developed by Fisher to explain the approximate equality of the sex ratio in mammals in 1930 has been used to explain a number of aspects of human behavior [12]. In the evolutionary game theory, the decision-making behavior

is based on the interaction among agents and helps the system to realize utility maximization. Affected by internal and external factors, participants’ strategies and game’s results will be presented in different states [13]. These features enable the game theory to well model the interaction of short TCP flows and obtain the solution of the padding misbehavior of virtual routers.

The main contributions of this paper can be summarized as follows:

- (1) We built a novel game model which is based on the evolutionary game for describing the normal behavior and padding misbehavior of virtual routers.
- (2) This paper analyses the stability of the equilibrium points which depends on the value of  $\Delta J \cdot \text{tr} J$ , and the stable point is the solution to the problem.
- (3) This paper analyses the corresponding evolutionary path of network applications with the normal behavior and padding misbehavior and gives the behavior control suggestions of padding misbehavior.

The rest of the paper is organized as follows. In Section 2, we build a game model of virtual routers for describing behavior. Section 3 gives the stability analysis of our evolutionary game model. Section 4 is further discussions and control suggestions. Section 5 is Simulation Experiment and Result Analysis. Finally, Section 6 concludes the paper.

## 2. System Model and Problem Analysis

*2.1. Motivation.* This paper, respectively, proposes a model and an algorithm on the basis of the evolutionary game theory. Comparing with the other game theories, the evolutionary game theory has four characteristics which can be described by a dynamic process for virtual routers to select normal behavior and padding misbehavior in our problem. In the stable equilibrium of the evolutionary game, none of the virtual routers is able to obtain the global utility information in our problem, so it is an appropriate choice to model our problem based on the evolutionary game.

- (1) Dynamics selection: in evolutionary games, players can observe other players’ behavior, learn from observation, and make the best decisions according to their knowledge. Furthermore, dynamic equations can be utilized to analyze how players adjust their behavior to achieve the desired solution.
- (2) Bounded rationality: in the evolutionary game, players slowly change their strategies in order to achieve ideal solution comparing with the players in a classical single-play and noncooperative game; this is due to the lack of global information, which is leading the player’s inadequate rationality.
- (3) Efficient solution: in the evolutionary game theory, evolutionary equilibrium provides such an efficient solution, and it ensures the stability which means all the players will not change their chosen strategies over time.

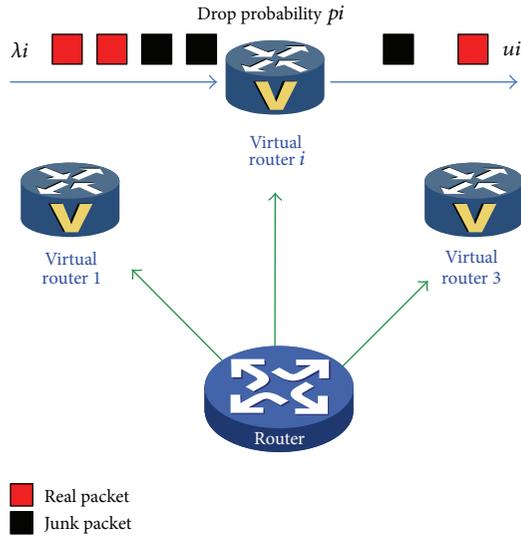


FIGURE 2: The obtained throughput of virtual routers.

- (4) **Stability:** in Nash equilibrium, no player will change his or her strategy. Because they could not get benefits by changing his or her chosen strategy while other players maintain theirs unchanged [14].

### 2.2. System Model for Describing Behavior of Virtual Routers.

This paper considers an evolutionary game model for virtual routers as Figure 2 shows, and there are a set of virtual routers  $VR_i, i \in \{1, 2, \dots, n\}$ . The virtual routers can be activated or inactivated by different TCP-based interactive applications. Usually, the applications adopt normal behavior and a single packet loss can force a short-lived TCP flow to experience long retransmission, so some interactive applications adopt padding misbehavior by virtual routers for data transmission. The problems we want to solve are how to restrain the padding misbehavior and what the condition of the selection convergence is. So we model the problem by using a dynamic evolutionary game.

The evolutionary game model can be described by the elements as follows.

- (1) **Players:** virtual routers  $VR_i, i \in \{1, 2, \dots, n\}$  are the players of the evolutionary game.
- (2) **Strategy:** the strategies of the players are the two kinds of behavior: normal behavior and padding misbehavior.
- (3) **Input packet rate:** it is assumed that the average packet arrival rate of the virtual routers obey the Poisson distribution. As shown in Figure 2,  $\lambda_i$  is the input rate of  $VR_i$ .
- (4) **Drop packet probability:** the algorithm is designed to use drop probability to manage the packet flows which pass through the ports of the physical router. When network congestion occurs, the algorithm will drop some packets with drop probability. Different behavior has different drop probabilities. It is assumed that the drop probability of virtual routers which

adopt normal behavior is  $p$ , and the drop probability of virtual routers which adopt padding behavior is  $q$ . Obviously, the values' relationship of the two behavior is  $q > p$ .

- (5) **Output packet rate:** it is the throughput of the virtual routers and is also the player's departure rate of passing through the physical router port. As shown in Figure 2,  $u_i$  is the input rate of  $VR_i$ .
- (6) **Payoff:** the payoff is the obtained throughput of a flow when it departs from the virtual router. It is also the characteristic value of the noncooperative TCP-based interactive applications, and the Nash equilibrium of the whole system could be reached by these values.

In cloud computing network, there are lots of applications. They belong to two kinds: one kind is UDP-based applications and the other kind is TCP-based interactive applications. An application choosing normal behavior concentrates on mutual benefits, which is willing to share the stability networking in the data transmission process. An application choosing padding misbehavior tends to obtain more and more network bandwidth and this will cause network congestion [15]. During data's transmission, we assume that any application in the network has the opportunity to adopt padding misbehavior, which is used to achieve the goal of maximizing the benefits of the application.

Let  $X$  denote the proportion of applications which choose normal behavior in the TCP-based interactive applications; then  $1 - X$  denote the proportion of applications which choose padding misbehavior in the TCP-based interactive applications. Let  $Y$  denote the proportion of applications which choose normal behavior in the UDP-based applications; then  $1 - Y$  denote the proportion of applications which choose padding misbehavior in the UDP-based applications.

We assume the following. (1) Whether it is TCP-based interactive applications or UDP-based applications, as long as both game participants adopt normal behavior, then they both gain  $u_i = \lambda_i(1 - p)$ . The gain obeys the linear pricing model, and the reason is that, with the drop package probability increasing, more packages will be dropped, and the number of nondropped packages will decrease. The linear pricing model is beneficial to avoid congestion and improve the system performance. (2) Whether it is TCP-based interactive applications or UDP-based applications, as long as both game participants adopt padding misbehavior, then they both gain  $u_i = 0$ . In this case, the whole network will go into congestion state and breakdown. (3) Whether it is TCP-based interactive applications or UDP-based applications, as long as one chooses padding misbehavior and the other chooses normal behavior; then the participants who choose normal behavior gain  $u_i = \lambda_i(1 - p) + C$ , where  $C$  denotes the benefit because of the penalty of padding misbehavior. The participants who choose padding misbehavior gain  $u_i = (1 - K_i)\lambda_i(1 - q)$ , where  $K_i$  is the padding factor which equals the percentage of padding packets in the whole packets, and  $q > p$ , which means that this misbehavior will obtain punishment. So we can get the payoff matrix of TCP-based interactive applications or UDP-based applications as shown in Table 1.

TABLE 1: The payoff matrix of TCP-based interactive applications or UDP-based applications.

	TCP-based interactive applications	
	Normal behavior ( $X$ )	Padding misbehavior ( $1 - X$ )
UDP-based applications		
Normal behavior ( $Y$ )	$(\lambda i(1 - p), \lambda j(1 - p))$	$(\lambda i(1 - p) + C, (1 - K_j)\lambda j(1 - q))$
Padding misbehavior ( $1 - Y$ )	$((1 - K_i)\lambda i(1 - q), \lambda j(1 - p) + C)$	$(0, 0)$

### 3. Solution and Stability Analysis of the Model

It is assumed that an evolutionary game has  $m$  strategies. At time  $t$ , the number of individuals that choose strategy  $i$  ( $i = 1, 2, \dots, m$ ) is denoted by  $n_i(t)$ , so the total number of individuals is denoted by  $N = \sum_{i=1}^m n_i(t)$ . Then the state of each strategy is given by  $X(t) = \{x_1(t), x_2(t), \dots, x_k(t)\}$ , where  $x_i = n_i/N$  is the proportion of strategy  $i$ . In our game model, the replicator dynamics can be defined as follows [16]:

$$\frac{d(x_i(t))}{dt} = x_i(t) * [U_i(t) - \bar{U}(t)], \quad i \in \{1, 2, \dots, I\}. \quad (1)$$

By the utility  $U_i(t)$ , the average utility  $\bar{U}(t)$ , and  $d(x_i(t))/dt = 0$ , we can obtain the fixed points of the differential equations. The stable fixed points form a proportion distribution which is the evolutionary equilibrium, and the dynamic evolutionary game will converge to the equilibrium.

Based on the replicator equation (1), we can get the dynamic equations for the evolution of the padding misbehavior and normal behavior in the network

$$\begin{aligned} \frac{dX_R}{dt} &= X_R \cdot (E_R - \bar{E}) \\ &= X_R \cdot [E_R - X_R \cdot E_R - (1 - X_R) \cdot E_O] \\ &= X_R \cdot (1 - X_R) \cdot (E_R - E_O), \end{aligned} \quad (2)$$

$$\begin{aligned} J &= \begin{bmatrix} \frac{\partial(dX/dt)}{\partial X} & \frac{\partial(dX/dt)}{\partial Y} \\ \frac{\partial(dY/dt)}{\partial X} & \frac{\partial(dY/dt)}{\partial Y} \end{bmatrix} \\ &= \begin{bmatrix} (1 - 2X) [(1 - Y)(1 - K_i)\lambda i(1 - q) - YC] & -X(1 - X) [(1 - K_i)\lambda i(1 - q) + C] \\ -Y(1 - Y) ((1 - K_j)\lambda j(1 - q) + C) & (1 - 2Y) [(1 - X)(1 - K_j)\lambda j(1 - q) - XC] \end{bmatrix}. \end{aligned} \quad (4)$$

So we can get

$$\begin{aligned} \Delta J &= (1 - 2X) [(1 - Y)(1 - K_i)\lambda i(1 - q) - YC] \\ &\quad \cdot (1 - 2Y) [(1 - X)(1 - K_j)\lambda j(1 - q) - XC] \\ &\quad - Y(1 - Y) ((1 - K_j)\lambda j(1 - q) + C) \end{aligned}$$

where  $X_R$  denotes the proportion of applications which choose normal behavior,  $0 \leq X_R \leq 1$ ,  $1 - X_R$  denotes the proportion of applications which choose padding misbehavior,  $E_R$  denotes the expected revenue of the applications which choose normal behavior,  $E_O$  denotes the expected revenue of the applications which choose padding misbehavior, and  $\bar{E}$  denotes the average expected revenue of all applications.

Based on the payoff matrix in Table 1, we can get

$$\begin{aligned} \frac{dX}{dt} &= X(1 - X) [Y\lambda i(1 - p) \\ &\quad + (1 - Y)(1 - K_i)\lambda i(1 - q) - Y\lambda i(1 - p) - YC] \\ &= X(1 - X) [(1 - Y)(1 - K_i)\lambda i(1 - q) - YC], \end{aligned} \quad (3)$$

$$\begin{aligned} \frac{dY}{dt} &= Y(1 - Y) [X\lambda j(1 - p) \\ &\quad + (1 - X)(1 - K_j)\lambda j(1 - q) - X\lambda j(1 - p) - XC] \\ &= Y(1 - Y) [(1 - X)(1 - K_j)\lambda j(1 - q) - XC]. \end{aligned}$$

Let  $dX/dt = 0$ ,  $dY/dt = 0$ , and we can get five equilibrium points of the model:  $O(0, 0)$ ,  $A(0, 1)$ ,  $B(1, 0)$ ,  $C(1, 1)$ , and  $D(1 - C/((1 - K_j)\lambda j(1 - q) + C), 1 - C/((1 - K_i)\lambda i(1 - q) + C))$ .

The stability of the equilibrium point is obtained by local stability analysis of the Jacobian matrix of the system [17]. When  $\Delta J \cdot \text{tr}J < 0$ , the equilibrium point is stable; when  $\Delta J \cdot \text{tr}J > 0$ , the equilibrium point is not stable; when  $\Delta J \cdot \text{tr}J = 0$ , the equilibrium point is a saddle point. Based on formulas (3), we can get the Jacobian  $J$ :

$$\cdot X(1 - X) [(1 - K_i)\lambda i(1 - q) + C],$$

$$\begin{aligned} \text{tr}J &= (1 - 2X) [(1 - Y)((1 - K_i)\lambda i(1 - q) - YC) \\ &\quad + (1 - 2Y) [(1 - X)(1 - K_j)\lambda j(1 - q) - XC]]. \end{aligned} \quad (5)$$

Then we can get the stability of the five equilibrium points.

- (1) To the first equilibrium point  $O(0, 0)$ :  $\Delta J > 0$ ,  $\text{tr}J > 0$ , so  $\Delta J \cdot \text{tr}J > 0$ ; the equilibrium point is not stable. When the initial state is at point  $O(0, 0)$ , the whole network will go into another stable state.
- (2) To the second equilibrium point  $A(0, 1)$ :  $\Delta J > 0$ ,  $\text{tr}J < 0$ , so  $\Delta J \cdot \text{tr}J < 0$ ; the equilibrium point is stable. When the initial state is at point  $A(0, 1)$ , the evolutionary game will converge.
- (3) To the third equilibrium point  $B(1, 0)$ :  $\Delta J > 0$ ,  $\text{tr}J < 0$ , so  $\Delta J \cdot \text{tr}J < 0$ ; the equilibrium point is stable. When the initial state is at point  $B(1, 0)$ , the evolutionary game will converge.
- (4) To the fourth equilibrium point  $C(1, 1)$ :  $\Delta J > 0$ ,  $\text{tr}J > 0$ , so  $\Delta J \cdot \text{tr}J > 0$ ; the equilibrium point is not stable. When the initial state is at point  $C(1, 1)$ , the whole network will go into another stable state.
- (5) To the fifth equilibrium point  $D(1 - C/((1 - Kj)\lambda j(1 - q) + C), 1 - C/((1 - Ki)\lambda i(1 - q) + C))$ :  $\Delta J < 0$ ,  $\text{tr}J = 0$ , so  $\Delta J \cdot \text{tr}J = 0$ ; the equilibrium point is not stable, and it is a saddle point.

#### 4. Further Analysis and Behavior Control Suggestions

The active queue management schemes, for instance, RED [7], ARED [8], BLUE [9], PI [10], and PAQM [11], are generally based on two packet processing types, Drop-tail and RED. But in these methods several packets would be dropped without considering the padding misbehavior. Therefore, network congestion could not be effectively avoided and the terminals could not apply appropriate communication behavior. So it is necessary to design a new scheme to avoid congestion caused by padding misbehavior and maintain stable high-throughputs of the router system.

From the above analysis, we can learn that penalty cost is a good way to avoid padding misbehavior. The larger the penalty cost, the more the willingness of the applications to choose normal behavior. On the other hand, there are also other factors that will affect the padding misbehavior in the cloud computing network. Figure 3 shows the corresponding evolutionary path of network applications with the normal behavior and padding misbehavior. The four fixed points  $A, B, C$ , and  $D$  are the vertices of a square. To the saddle point  $D(1 - C/((1 - Kj)\lambda j(1 - q) + C), 1 - C/((1 - Ki)\lambda i(1 - q) + C))$ , it is clear that there are four curves, and they divide the square into four parts: I, II, III, and IV. And the position of the point  $D$  depend on the parameters  $C, Kj, Ki, \lambda i$ , and  $q$ . It is clear that the different initial states of the game can lead to different results. When the initial state is in the areas I and II, the game will converge to the stable equilibrium point  $B(1, 0)$ . When the initial state is in the areas III and IV, the game will converge to the stable equilibrium point  $A(0, 1)$ .

To avoid network congestion, the queue management algorithms could restrain padding misbehavior effectively.

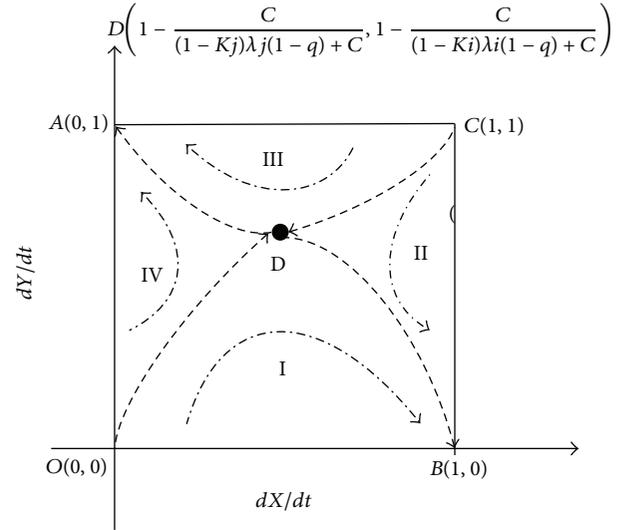


FIGURE 3: Evolutionary path of network applications.

Therefore, we provide the suggestions aiming at each application to adopt normal behavior in the real network. For TCP-based interactive applications and UDP-based applications, if they both adopt normal behavior, the whole network will run effectively by the network congestion control mechanism; if one of them adopts padding misbehavior, the input packet rate  $\lambda i$  or  $\lambda j$  is bigger than the normal case, and in order to make the state close to the stable point  $A(0, 1)$  or  $B(1, 0)$ , the queue management algorithms should drop all the input packets of the applications with padding misbehavior. This way is different from the traditional queue management algorithms dropping packets randomly.

#### 5. Simulation Experiment and Result Analysis

In this section, we present the software simulation experiments on NS-2 (Network Simulator version 2) to test the performance of our solution [18]. NS2 is a discrete event simulator targeted at networking research. It provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. It consists of the following modules: flow generation module, event queue manager module, statistics collection and query module, and so on. The simulation topology is shown in Figure 1, and multiple TCP-based interactive applications and UDP-based applications share links between two physical routers. We compare the performance and effectiveness of our solution with RED solution introduced in Section 1.

In the simulation, it is assumed that the throughputs of Host A, Host B, and Host C are, respectively, set to 100 Kbp, 200 Kbps, and 300 Kbps. The link capacity is 700 Kbps, and each host has the intention to pad “junk packets” in its TCP connection which means everyone applies the padding misbehavior in order to seize more bandwidth. The hosts realized the padding misbehavior by activating the corresponding virtual routers, and the behavior of virtual routers may dynamically transfer from normal behavior to padding

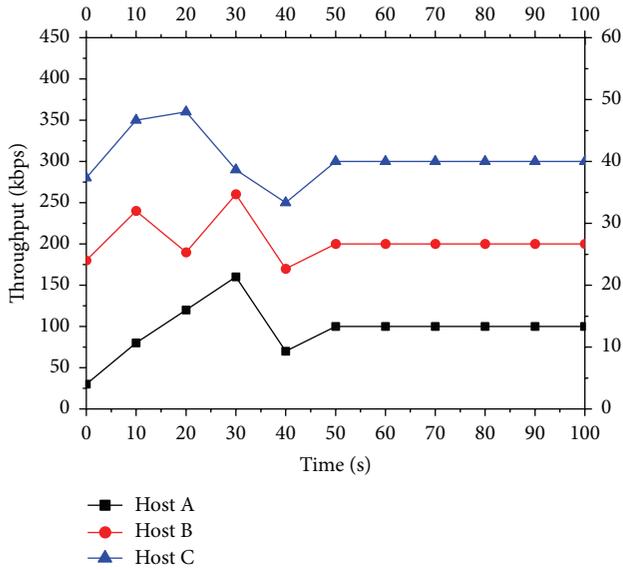


FIGURE 4: The hosts' throughputs of our solution.

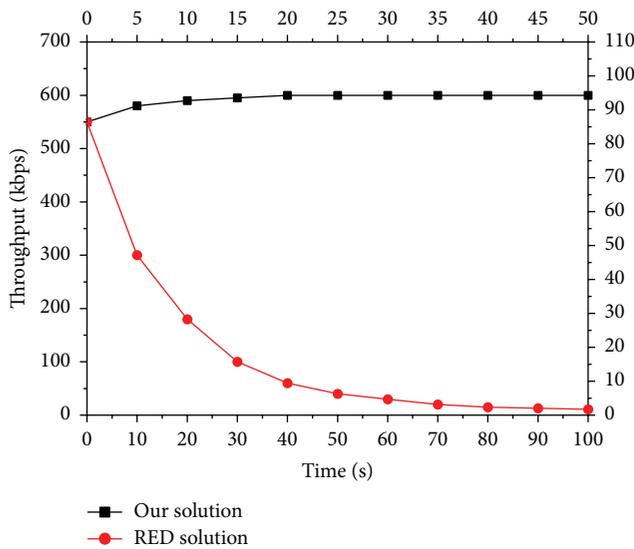


FIGURE 5: Comparison of the router throughput.

misbehavior. In order to verify that our solution is able to handle this complex scenario, the experimental results show the four hosts' changes of behavior and the whole throughput of the router.

Figures 4 and 5, respectively, show the three hosts' throughputs and the total throughput of RED solution and our solution. As shown in Figure 4, each host's throughput is not stable from 0 s to 50 s, and every host wants to seize bandwidth through padding "junk packets." However, any host which used the padding misbehavior will be punished, which means that the corresponding drop probability will increase. Finally, the hosts' throughputs will remain unchanged. As shown in Figure 5, in our solution the total throughput is stable and the congestion is successfully avoided. By contrast, the total throughput of RED solution obviously decreases as

the time increases which means that congestion occurs, and RED solution could not restrain the padding misbehavior.

The experiment results show that our solution is able to effectively restrain the padding misbehavior and maintain stable high-throughputs of the router simultaneously compared with the classical queue management.

## 6. Conclusion

The core issue of network virtualization is to run multiple virtual router instances on a common physical router hardware platform, and this poses serious challenges to the performance of routers. In this paper, we built a novel game model which is based on the evolutionary game for describing the normal behavior and padding misbehavior of virtual routers and discusses the stability of the equilibrium points which depends on the value of  $\Delta J \cdot \text{tr}J$ , and the stable point is the solution to the problem. Then we analyze the corresponding evolutionary path of network applications with the normal behavior and padding misbehavior and give the control suggestions of padding misbehavior. The study shows that the behavior control suggestions are able to effectively restrain the padding misbehavior and maintain stable high-throughputs of the router.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This work is supported by the Hunan Provincial Education Department Scientific Research Fund of China (no. 15C0825); Hunan Provincial Science and Technology Program Project of China (no. 2015JC3066); Scientific Research Foundation for Ph.D. Hunan Normal University (no. Math 120641); and Youth Scientific Research Fund of Hunan Normal University (no. 11301).

## References

- [1] J. Shamsi and M. Brockmeyer, "QoSMap: QoS aware mapping of virtual networks for resiliency and efficiency," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '07)*, November 2007.
- [2] N. Feamster, L. Gao, and J. Rexford, "How to lease the internet in your spare time," *SIGCOMM Computer Communication Review*, vol. 37, no. 1, pp. 61–64, 2007.
- [3] Y. Sun, F. Zhang, and J. Qi, "Cognitive networks congestion prediction method based on bayesian networks," *International Journal of Advancements in Computing Technology*, vol. 4, no. 7, pp. 329–338, 2012.
- [4] A. Mondal and A. Kuzmanovic, "When TCP friendliness becomes harmful," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 152–160, Anchorage, Alaska, USA, May 2007.

- [5] A. Mondal and A. Kuzmanovic, "Upgrading mice to elephants: effects and end-point solutions," *IEEE/ACM Transactions on Networking*, vol. 18, no. 2, pp. 367–378, 2010.
- [6] J. S. Miller, A. Mondal, R. Potharaju, P. A. Dinda, and A. Kuzmanovic, "Understanding end-user perception of network problems," in *Proceedings of the 1st ACM SIGCOMM Workshop on Measurements Up the Stack (SIGCOMM '11)*, pp. 43–48, ACM, Toronto, Canada, August 2011.
- [7] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, 1993.
- [8] W.-C. Feng, D. D. Kandlur, D. Saha, and K. G. Shin, "A self-configuring RED gateway," in *Proceedings of the 18th International Conference on Computer Communications (INFOCOM '99)*, pp. 1320–1328, New York, NY, USA, March 1999.
- [9] J. Wu, J. Liu, and Z. Guo, "A class of active queue management algorithm based on BP neural network," in *Proceedings of the 21th Chinese Control and Decision Conference (CCDC '09)*, pp. 1580–1583, Guilin, China, June 2009.
- [10] S. Mascolo, "Congestion control in high-speed communication networks using the Smith principle," *Automatica*, vol. 35, no. 12, pp. 1921–1935, 1999.
- [11] V. Firoiu and M. Borden, "Study of active queue management for congestion control," in *Proceedings of the 19th IEEE International Conference on Computer Communications (INFOCOM '00)*, pp. 1435–1445, Tel-Aviv, Israel, March 2000.
- [12] R. A. Fisher, "The distribution of gene ratios for rare mutation," *Proceedings of the Royal Society of Edinburgh*, vol. 50, pp. 205–220, 1930.
- [13] J. M. Smith, *Evolution and the Theory of Games*, Cambridge University Press, Cambridge, UK, 1982.
- [14] X.-A. Bi, D.-F. Zhang, W.-W. Li, and S.-M. He, "A population evolution algorithm for supporting scalable virtual routers," *Journal of Convergence Information Technology*, vol. 7, no. 4, pp. 255–263, 2012.
- [15] D. Crisan, R. Birke, K. Barabash, R. Cohen, and M. Gusat, "Datacenter applications in virtualized networks: a cross-layer performance study," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 1, pp. 77–87, 2014.
- [16] P. D. Taylor and L. B. Jonker, "Evolutionarily stable strategies and game dynamics," *Mathematical Biosciences*, vol. 40, no. 1-2, pp. 145–156, 1978.
- [17] D. Friedman, "Evolutionary games in economics," *Econometrica. Journal of the Econometric Society*, vol. 59, no. 3, pp. 637–666, 1991.
- [18] D. Kumar, S. Ahmed, and Y. Ryu, "Quality of Service (QoS) of voice over MAC protocol 802.11 using NS-2," *Journal of Convergence Information Technology*, vol. 3, no. 4, pp. 76–83, 2008.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

