Mobile Cloud Offloading for Malware Detections with Learning

Yanda Li, Jinliang Liu, Qiangda Li, Liang Xiao*

* Dept. of Communication Engineering, Xiamen Univ., 361005 China.
Email: lxiao@xmu.edu.cn

***Abstract*—Accurate malware detections on mobile devices such as smartphones require fast processing of a large number of data and thus cloud offloading can be used to improve the security performance of mobile devices with limited resources. The performance of malware detection with cloud offloading depends on the computation speed of the cloud, the population sharing the cloud resources and the bandwidth of the radio access. In the paper, we investigate the offloading rates of smartphones connecting to the same security server in a cloud under dynamic network bandwidths and formulate their interactions as a non-cooperative mobile cloud offloading game. The Nash equilibrium of the mobile cloud offloading game and the existence condition are presented. An offloading algorithm based on Q-learning is proposed for smartphones to determine their offloading rates for malware detection with unknown parameters such as transmission costs. Simulation results show that the proposed offloading strategy can achieve the optimal rate and improve the user's utility under dynamic network bandwidths.**

***Keywords*: Mobile cloud, offloading, malware detection, Q-learning**

I. Introduction

A large number of mobile malware including virus, worms, Trojans and spy tools have been found in circulation to attack mobile devices such as smartphones and tablets, resulting in privacy leakage, economic loss, power depletion and network performance degradation. Dynamic malware detection strategies have been proposed to analyze the runtime behavior of thousands of applications on smartphones [1]–[3]. However, due to the limited battery life, computation resources and radio network bandwidth, mobile devices are more vulnerable to malware attacks than PC platforms. For example, the latency to download the latest signature files for malware detection from security database can lead to zero-day attacks. In addition, accurate malware inspection has to investigate the application behavior on kernel layer, indicating the process of a huge number of log data generated by operating systems with application execution, e.g., 117,903 lines of log data have to be evaluated for a single application called Norton mobile security on an Android smartphone [3].

Therefore, cloud-based malware detection has been proposed in [4] to improve security of mobile devices, in which a security server in a cloud hosts a synchronized replica of a smartphone with a certain granularity in virtual environment to detect attacks simultaneously without overburdening the phone. Another cloud-based intrusion detection framework has been proposed for smartphones in [5]. The cloud-based malware detection can address the zero-day attacks and has the advantage of scalability, elasticity and inexpensive. However, most existing works on cloud-based malware detection focus on the implementation on a single smartphone and ignore the limited and dynamic cloud computational resources and network bandwidth, which can degrade the security performance especially in the presence of a large amount of log data involved in the continuous operation of malware detection on a smartphone.

Being critical to perform fast, energy-efficient and accurate malware detection on smartphones, cloud offloading has attracted extensive research attention for general purposes. For example, offloading of multimedia, gaming, and calculator applications has been analyzed via experiments in [6]. Offloading in mobile clouds provides a tradeoff between the operating time saving and extend battery life in [7]. The performance of offloading has been found to depend on the network bandwidth and device state in [8]. The connection to WiFi, 2G and 3G networks corresponds to significantly different bandwidth and energy costs of a smartphone and impacts on its optimal offloading strategy in [9]. A Bayesian-based offloading strategy was proposed in [10], which chooses to offload only if the estimated network bandwidth is large enough. Fine-grained offloading of mobile code was developed in [11], to save energy for applications such as face recognition, games and language translations on mobile devices in. The competition among mobile users that offload request to the data center is formulated as an Bayesian game in [12].

This paper investigates the malware detection of a number of smartphones that offload to the same security cloud via base stations or access points based on game theory that has shown its strength to improve network security [13]–[16]. More specifically, the interactions among the mobile users that compete for the limited computational resources of a security server in a cloud to detect malware are formulated as a mobile cloud offloading game. The Nash equilibrium (NE) of the mobile cloud offloading game and the existence condition of a unique NE are presented.

In addition, repeated interactions among smartphones in their offloading to the same cloud for malware detection with

The work of L. Xiao was partly supported by NSFC (61271242,61001072).

978-1-4673-7131-5/15/$31.00 ©2015 IEEE

unknown parameters are formulated as a dynamic mobile cloud offloading game with time variant network conditions, in which bandwidths change randomly over time. A larger bandwidth motivates users to offload more data, while too much offloading decreases the utility of a smartphone due to the limited processing capacity of the server and a large transmission cost and sometimes results in congestion of the network. Since system parameters such as the transmission costs are unknown, smartphones determine their offload ratios by repeated try-and-error trials. A Q-learning algorithm is proposed for the mobile cloud offloading game to update its offloading rate based on the each smartphone in channel conditions and offloading rates of the other smartphones. The proposed offloading strategy can achieve the optimal offloading rate and increase the utility of the smartphone in the malware detection.

M	Number of users
C_i	Data for detection of user i
b_i^k	Transmission bandwidth of user i at time k
x_i	Offloading rate of user i
$\mathbf{A}$	Offloading rate set
R	Computation processing speed of server
p_i	Transmission power of user i
$\mathbf{s}$	States of user i
u_i	Utility of user i
x_i^*	The best strategy of user i
$\boldsymbol{x}_{-i}^*$	Best strategy set of users excluding user i
Q_i	Q function of user i
$V_i(\mathbf{s})$	Value of state $\mathbf{s}$

TABLE I

SUMMARY OF SYMBOLS AND NOTATION.

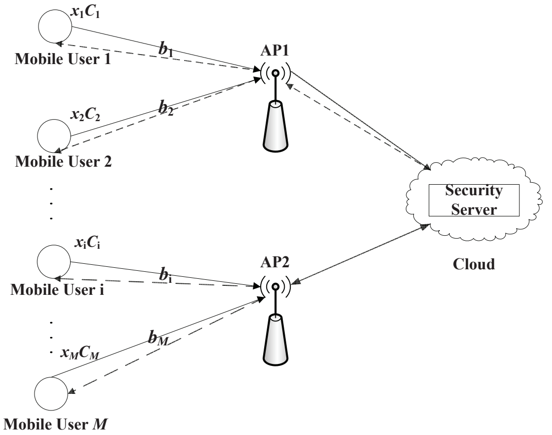


Fig. 1. A mobile cloud offloading system for M mobile users that share a cloud with high performance resources such as CPUs, disks and memory to detect malware via different access points/base stations.

The remainder of the paper is organized as follows. We formulate the mobile cloud offloading game for malware detection of smartphones in Section II. We present the NE of the static mobile cloud offloading game in Section III and propose an offloading strategy for smartphones based on Q-learning in Section IV. Simulation results are provided in Section V and conclusion is drawn in Section VI.

II. Mobile Cloud Offloading Game for Malware Detection

A. System Model

In a mobile cloud offloading system as shown in Fig. 1, M mobile users such as smartphones or tablets offload security-related data to a security server in a cloud via servered access points (APs) or base stations (BSs), in order to improve their malware detection speeds by utilizing the computational resources of the server, such as its CPUs, disks and memories.

Without loss of generality, user i chooses x_i^k of its total amount of malware detection data denoted by C_i at time k, with $1 \leq i \leq M, 0 \leq x_i^k \leq 1$ and $C_i \geq 0$. The transmission delay of these $x_i^k C_i$ data is mostly determined by the radio transmission bandwidth, denoted by b_i^k, with $b_i^k \in [B_i^n]_{1 \leq n \leq N}$, and $B_i^n \geq 0$, $\forall 1 \leq n \leq N$, where N is the number of bandwidth quantization levels. The time index, k, in the superscript can be omitted, if no confusion incurs.

Each mobile user i aiming to maximize its offloading gain from the cloud and minimize its transmission cost is rational and selfish. User i determines the offloading rate denoted by x_i with $0 \leq x_i \leq 1$, and the amount of the offloading data $x_i C_i$ to achieve its target, which means user i chooses a portion of the total data for detection, i.e., $x_i C_i$ of the program to be processed in the cloud.

B. Game Model

As showed in Fig. 1, the offloading interactions among M mobile users connecting to the same security server via APs are formulated as an non-cooperative mobile cloud offloading game, in which mobile users as the players choose their offloading rates $[x_i]_{1 \leq i \leq M}$, from their action set denoted by $\mathbf{A} = [A_n]_{1 \leq n \leq N_x}$, where N_x denotes the number of possible offloading rates, e.g., $\mathbf{A} = [0, 0.1, 0.2, \cdots, 1]$, with $N_x = 11$. For example, if $x_i = 0$, user i doesn't offload and completely detects the data locally. If $x_i = 1$, user i offloads all the data to the cloud.

The computational capacity of the server in the cloud, denoted by R, is used to process the offloading data to detect malware and virus in the smartphones. User i offloads an amount of data $x_i C_i$ to the server, which allocates to user i $x_i C_i / (\sum_{m=1}^{M} x_m C_m)$ of the capacity. The transmission cost of user i depends on transmission bandwidth b_i. Let p_i be the transmission power of unit data for user i. Based on the processing speed in the malware detection, the instant utility of user i is defined as

$$u_i = \frac{x_i C_i}{\sum\limits_{m=1}^{M} x_m C_m} R - p_i \frac{x_i C_i}{b_i}, \quad i = 1,\ 2,\ \cdots,\ M. \qquad (1)$$

In summary, the static mobile cloud offloading game is denoted by $\mathbf{G} = \langle i, x_i, u_i \rangle_{1 \leq i \leq M}$, in which user i chooses its offloading rate x_i from the action set $\mathbf{A}$ to maximize its individual utility u_i.

III. Nash Equilibrium of the Static Offloading Game

In a one-shot offloading game **G**, M smartphones choose their offloading rates simultaneously. The NE strategy of the static game **G**, denoted by $[x_i^*]_{1 \le i \le M}$, is the best response in the game. According to definition [17], NE strategy is the best response of players, with $u_i(x_i^*, \boldsymbol{x}_{-i}^*) \ge u_i(x_i, \boldsymbol{x}_{-i}^*)$, for $\boldsymbol{x}_{-i}^* = [x_m^*]_{m \ne i, 1 \le m \le M}$ and $0 \le x_i^* \le 1, \forall 1 \le i \le M$.

Lemma 1. *In the static mobile cloud offloading game* **G**, *the optimal offloading rate of user* i $(1 \le i \le M)$, *is given by*

$$x_i^* = \begin{cases} 0, & \Pi_1 \\ 1, & \Pi_2 \\ \sqrt{Rb_i \frac{\sum\limits_{m \ne i}^{M} x_m C_m}{C_i p_i}} - \frac{\sum\limits_{m \ne i}^{M} x_m C_m}{C_i}, & o.w. \end{cases}, \tag{2}$$

where

$\Pi_1 : \frac{Rb_i C_i}{p_i} < \sum\limits_{m \ne i}^{M} x_m C_m,$

$\Pi_2 : (\sum\limits_{k \ne i}^{M} x_m C_m)^2 + (2 - \frac{Rb_i}{p_i}) C_i \sum\limits_{m \ne i}^{M} x_m C_m < {C_i}^2.$

Proof: By Eq. (1), we have

$$\frac{\partial u_i}{\partial x_i} = -R \frac{x_i {C_i}^2}{\left(\sum\limits_{m=1}^{M} x_m C_m\right)^2} + R \frac{C_i}{\sum\limits_{m=1}^{M} x_m C_m} - \frac{p_i}{b_i} C_i. \tag{3}$$

The second derivative is

$$\frac{\partial^2 u_i}{\partial {x_i}^2} = -\frac{2R{C_i}^2 \sum\limits_{m \ne i}^{M} x_m C_m}{\left(\sum\limits_{m=1}^{M} x_m C_m\right)^3}. \tag{4}$$

It is clear that $\frac{\partial^2 u_i}{\partial {x_i}^2} < 0$, indicating that the utility u_i is strictly concave. By $\frac{\partial u_i}{\partial x_i = 0}$, we have

$$-R \frac{x_i {C_i}^2}{\left(\sum\limits_{m=1}^{M} x_m C_m\right)^2} + R \frac{C_i}{\sum\limits_{m=1}^{M} x_m C_m} - \frac{p_i}{b_i} C_i = 0, \tag{5}$$

and thus

$$\widetilde{x_i} = \sqrt{Rb_i \frac{\sum^{M} x_m C_m}{C_i p_i}} - \frac{\sum^{M} x_m C_m}{C_i}. \tag{6}$$

If $0 \le \widetilde{x_i} \le 1$, we have $x_i^* = \widetilde{x_i}$. If $\sqrt{Rb_i \frac{\sum\limits_{m \ne i}^{M} x_m C_m}{C_i p_i}} - \frac{\sum\limits_{m \ne i}^{M} x_m C_m}{C_i} < 0$, i.e., $\frac{Rb_i C_i}{p_i} < \sum\limits_{m \ne i}^{M} x_m C_m$, we have $x_i^* = 0$. Otherwise, if $\sqrt{Rb_i \frac{\sum\limits_{m \ne i}^{M} x_m C_m}{C_i p_i}} - \frac{\sum\limits_{m \ne i}^{M} x_m C_m}{C_i} > 1$, i.e., $(\sum\limits_{k \ne i}^{M} x_m C_m)^2 + (2 - \frac{Rb_i}{p_i}) C_i \sum\limits_{m \ne i}^{M} x_m C_m < {C_i}^2$, we have $x_i^* = 1$. ■

Remark: The amount of resources of the security server and the transmission costs determine the NE of the static offloading game.

IV. Offloading Strategy based on Q-Learning

In a dynamic mobile cloud offloading game, with time variant network environment, it is challenging for each smartphone to accurately estimate its radio bandwidth and the actions of the other mobile users that offload to the same security server. Hence, reinforcement learning methods, such as Q-learning, can be employed by smartphones to achieve their optimal offloading strategies ultimately.

In this dynamic mobile cloud offloading game, the users decide their offloading rates in sequence. As shown in Algorithm 1, user i chooses its offloading rate at time k, x_i^k, based on the offloading rates of previous users, denoted by $\boldsymbol{x}_{-i}$. The system state of user i, denoted by $\mathbf{s}_i^k$, consists of its bandwidth and the offloading rates of the previous $i-1$ users at time k, i.e., $\mathbf{s}_i^k = (\boldsymbol{x}_{-i}, b_i^k)$.

Let $Q_i(\mathbf{s}_i^k, x_i^k)$ denote the Q function of state $\mathbf{s}$ of user i at time k. The Q function and value function denoted by V_i are updated respectively by

$$Q_i(\mathbf{s}_i^k, x_i^k) \leftarrow (1-\alpha) Q_i(\mathbf{s}_i^k, x_i^k) + \alpha (u_i(\mathbf{s}_i^k, x_i^k) + \delta V_i(\mathbf{s}_i^{k+1})), \tag{7}$$

$$V_i(\mathbf{s}_i^k) = \max_{x_i \in \mathbf{A}} Q_i(\mathbf{s}_i^k, x), \tag{8}$$

where $\alpha \in (0,1]$ is the learning rate, $\delta \in [0,1]$ is the discount factor indicating the greedy strategy of the user, V_i is the value function representing the highest value of the state, and u_i is the instantaneous utility that user i receives at time k.

Each user applies the ε-greedy policy in each time slot to choose its action, in which the action with the highest expected utility is taken with a high possibility, and the rest actions are selected randomly with a small probability, $1-\varepsilon$ for user i. Thus the probability for user i to choose offload Θ rate is given by

$$\Pr(x_i = \Theta) = \begin{cases} 1 - \varepsilon_i, & \Theta = \Theta^* \\ \frac{\varepsilon_i}{N_x}, & \Theta \ne \Theta^*, \Theta \in \mathbf{A} \end{cases}, \tag{9}$$

where the optimal offloading strategy denoted by Θ^*, is given by

$$\Theta^* = \arg\max_{x \in \mathbf{A}} Q_i(\mathbf{s}^{k+1}, x). \tag{10}$$

Next, user $i+1$ chooses its action x_{i+1}^k similarly, with $i \le M-1$. The process repeats for the next time slot after all the M users make decisions.

V. Simulation Results

Simulations have been performed to evaluate the performance of the mobile cloud offloading game, in which $M = 2$, $C_1 = C_2 = 1$, $p_1 = p_2 = 1$ and $R = 8$, unless specified otherwise.

In the first simulation, both users have constant transmission bandwidths over time, with $1/6 \le b_1 \le 1/2$ and $b_2 = 1/4$. The performance of the NE of a static mobile cloud offloading

Algorithm 1 Q-learning process of user i.

```
Set $\alpha = 0.7$, $\delta = 0.8$
Initialize $M$, $A$, $b_i$, $p_i$, $\forall 1 \leq i \leq M$
Set $Q_i(\mathbf{s}^1, x)$=0, $V_i(\mathbf{s})$=0, $\forall \mathbf{s}, x$
Repeat (for each episode)
    Set $\mathbf{s}^k = (\boldsymbol{x}_{-i}, b_i^k)$ of user $i$;
    For $k = 1, 2, 3, ...$
        Select $x_i^k$ via Eq. (9);
        Observe $\mathbf{s}^{k+1}$ and $u_i$;
        Update $Q_i(\mathbf{s}^k, x_i^k)$ via Eq. (7);
        Update $V_i(\mathbf{s}^k)$ via Eq. (8);
    End for
End for
```

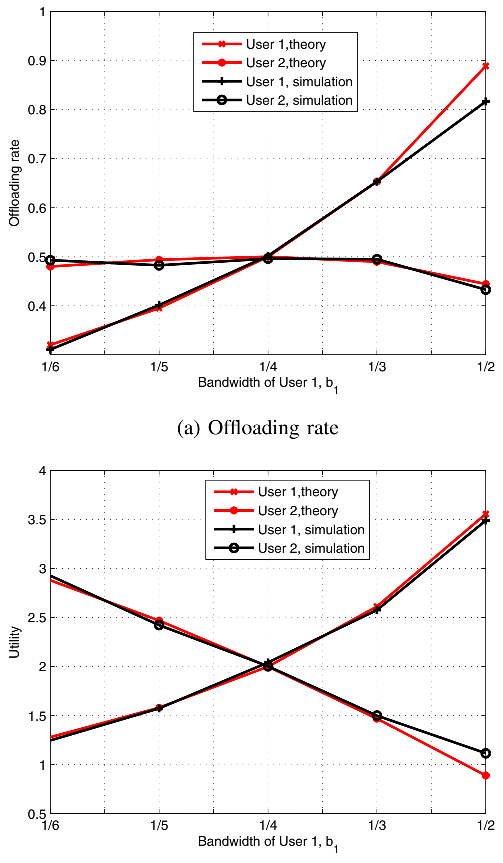
game, as theoretic results, is similar to the stable performance of a dynamic game, as shown in Fig. 2. In addition, a user increases its offloading rate and thus obtains a higher utility, under a higher transmission bandwidth. On the other hand, the other mobile user has a performance degradation if competing a smartphone with a higher bandwidth.

In the second simulation, we have evaluated the performance of a dynamic mobile cloud offloading game, averaged over all realizations of random and time variant bandwidths, with $b_1^k \in \{1/6, 1/5, 1/4\}$ and $b_2^k \in \{1/8, 1/7, 1/6\}$, $k = 1, 2, 3, \cdots$, i.e., user 1 usually has no worse transmission condition. As shown in Fig. 3, both mobile users quickly achieve their optimal offloading strategies. For instance, the average utility of user 1 increases from 1.5 to more than 2.5 after 1000 time slots from the start of the game. In addition, user 1 with a generally higher bandwidth relies more on the cloud and achieves a higher utility.

Finally, the instantaneous utilities of two users in the dynamic offloading game are shown in Fig. 4, in which both b_1^k and b_2^k randomly change every 100 time slots, with $b_1^k \in \{1/6, 1/5, 1/4\}$ and $b_2^k \in \{1/8, 1/7, 1/6\}$, $\forall k$. It is shown that both users achieve their optimal offloading strategies with stable and high utilities.

VI. Conclusion

In this paper, a mobile cloud offloading game has been formulated to analyze how smartphones choose their offloading rates to detect malware in a security server according to the transmission bandwidths and the amounts of security data to be processed of the smartphones. The NE of the static offloading game has been provided for the case with all the system parameters known by all the smartphones connecting to this security server. In addition, for the case that the system parameters are unknown to the smartphones, a Q-learning based offloading strategy has been proposed for a dynamic offloading game with time-variant network bandwidths and amounts of data to be processed for malware detection. Simulation results show that the proposed offloading strategy can fast reach a desirable performance and improve the smartphone's utility in dynamic network environments. For example, the utility of a smartphone increases from 1.5 to 2.5 after 1000 time slots, in a dynamic offloading game with two smartphones.



(a) Offloading rate

(b) Utility

Fig. 2. Performance of the mobile cloud offloading game with two smartphones under constant bandwidth b_1 and $b_2 = 1/4$.

References

[1] L. Xie, X. Zhang, J. P. Seifert, and S. Zhu, "pbMDS: A behavior-based malware detection system for cellphone devices," in *Proc. ACM Conf. Wireless network security*, 2010, pp. 37 – 48.

[2] B. Amos, H. Turner, and J. White, "Applying machine learning classifiers to dynamic android malware detection at scale," in *Proc. IEEE Int'l Wireless Communications and Mobile Computing Conference*, 2013, pp. 1666 – 1671.

[3] T. Isohara, K. Takemori, and A. Kubota, "Kernel-based behavior analysis for android malware detection," in *Proc. IEEE Int'l Conf. Computational Intelligence and Security (CIS)*, 2011, pp. 1011 – 1015.

[4] G. Portokalidis, P. Homburg, K. Anagnostakis, and H. Bos, "Paranoid android: Versatile protection for smartphones," in *Proc. IEEE Annual Computer Security Applications Conference*, 2010, pp. 347 – 356.

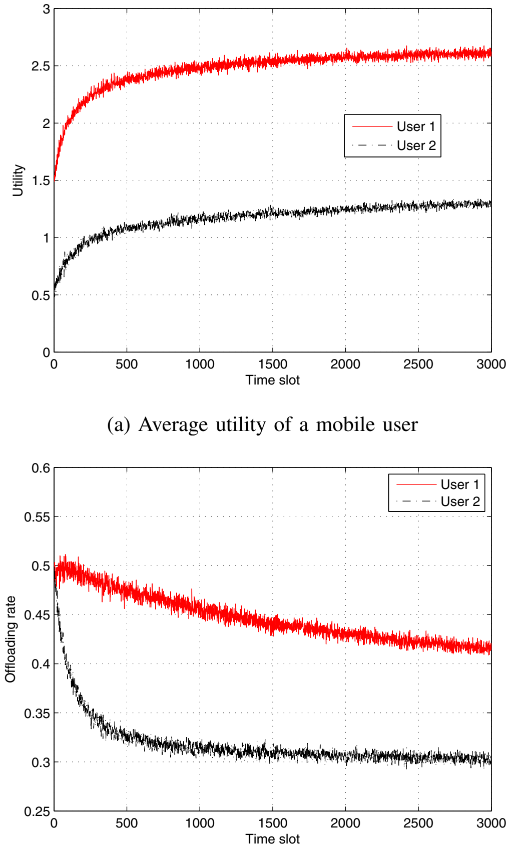
[5] A. Houmansadr, S. A. Zonouz, and R. Berthier, "A cloud-based intrusion detection and response system for mobile phones," in *Proc. IEEE Int'l Conf. Dependable Systems and Networks Workshops*, 2011, pp. 31 – 32.

[6] K. Kumar, J. Liu, Y. H. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile Networks and Applications*, vol. 18, no. 1, pp. 129 – 140, 2013.

[7] H. Wu, Q. Wang, and K. Wolter, "Tradeoff between performance improvement and energy saving in mobile cloud offloading systems," in *Proc. IEEE Int'l Conf. Communications Workshops*, 2013, pp. 728 – 732.

[8] K. Kumar and Y. H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?," *Computer*, vol. 43, no. 4, pp. 51 – 56, 2010.

[9] M. V. Barbera, S. Kosta, A. Mei, and J. Stefa, "To offload or not to offload? The bandwidth and energy costs of mobile cloud computing,"



(a) Average utility of a mobile user

(b) Average offloading rate

Fig. 3. Performance of a dynamic mobile cloud offloading game with two smartphones averaged over all the realizations of time-variant bandwidths, $b1 \in \{1/6, 1/5, 1/4\}$ and $b2 \in \{1/8, 1/7, 1/6\}$.

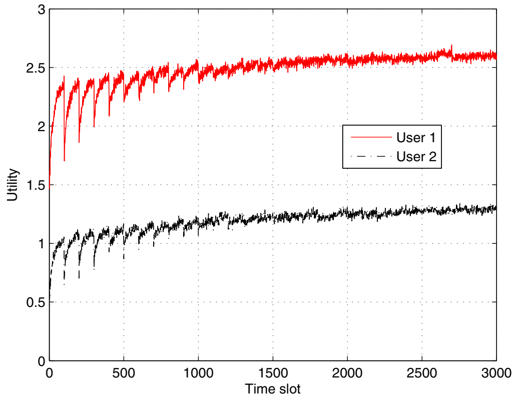


Fig. 4. Performance of a dynamic mobile cloud offloading with two smartphones with bandwidths changing randomly every 100 time slots, $b_1^k \in \{1/6, 1/5, 1/4\}$ and $b_2^k \in \{1/8, 1/7, 1/6\}, \forall k$.

in *Proc. IEEE Int'l Conf. Computer Commun. (INFOCOM)*, 2013, pp. 1285 – 1293.

[10] R. Wolski, S. Gurun, C. Krintz, and D. Nurmi, "Using bandwidth data to make computation offloading decisions," in *Proc. IEEE Int'l Symp. Parallel and Distributed Processing*, 2008, pp. 1 – 8.

[11] E. Cuervo, A. Balasubramanian, and et al., "MAUI: Making smartphones last longer with code offload," in *Proc. ACM Int'l Conf. Mobile systems, applications, and services*, 2010, pp. 49 – 62.

[12] Y. Wang, X. Lin, and M. Pedram, "A bayesian game formulation of power dissipation and response time minimization in a mobile cloud computing system," in *Proc. IEEE Int'l Conf. Mobile Services*, 2013, pp. 7 – 14.

[13] L. Xiao, W.S Lin, Y. Chen, and K.J.R Liu, "Indirect reciprocity security game for large-scale mobile wireless networks," *IEEE Trans. Information Forensics Security*, vol. 7, no. 4, pp. 1368 – 1380, Aug. 2012.

[14] L. Xiao, J. Liu, Y. Li, N. B. Mandayam, and H. V. Poor, "Prospect theoretic analysis of anti-jamming communications in cognitive radio networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Austin, Dec. 2014.

[15] L. Xiao, N. B. Mandayam, and H. V. Poor, "Prospect theoretic analysis of energy exchange among microgrids," *IEEE Trans. Smart Grids*, vol. 6, no. 1, pp. 63 – 72, Jan. 2015.

[16] L. Xiao, J. Liu, Q. Li, and H. V. Poor, "Secure mobile crowdsensing game," in *Proc. IEEE Int'l Conf. on Commun. (ICC)*, Jun. 2015, to appear.

[17] Martin M. J. Osborne and A. Rubinstein, *A course in game theory*, MIT press, 1994.