An Optimal Distributed Malware Defense System for Mobile Networks with Heterogeneous Devices

Yong Li *, Pan Hui †, Depeng Jin *, Li Su *, Lieguang Zeng *
*State Key Laboratory on Microwave and Digital Communications
Tsinghua National Laboratory for Information Science and Technology
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
†Deutsche Telekom Laboratories/TU-Berlin, Ernst-Reuter-Platz 7, Berlin 10587, Germany
Email: jindp@mail.tsinghua.edu.cn

Abstract—**As malware attacks become more frequent in mobile networks, deploying an efficient defense system to protect against infection and to help the infected nodes to recover is important to contain serious spreading and outbreaks. The technical challenges are that mobile devices are heterogeneous in terms of operating systems, and the malware can infect the targeted system in any opportunistic fashion via local and global connectivity, while the to-be-deployed defense system on the other hand would be usually resource limited. In this paper, we investigate the problem of optimal distribution of content-based signatures of malware to minimize the number of infected nodes, which can help to detect the corresponding malware and to disable further propagation. We model the defense system with realistic assumptions addressing all the above challenges, which have not been addressed in previous analytical work. Based on the proposed framework of optimizing the system welfare utility through the signature allocation, we provide an encounter-based distributed algorithm based on Metropolis sampler. Through extensive simulations with both synthetic and real mobility traces, we show that the distributed algorithm achieves the optimal solution, and performs efficiently in realistic environments.**

I. Introduction

The target landscape for malware attacks (i.e., viruses, spam bots, worms and other malicious software) has moved considerably from the large-scale Internet to the growing popular mobile networks [1], with a total count of known mobile malware instances of more than 350 reported in early 2007 [2]. This is mainly because of two reasons. One is the emergence of powerful mobile devices, such as the iPhone, Blackberry, and Android devices, and increasingly diversified mobile applications, such as Multimedia Messaging Service (MMS), mobile games and peer-to-peer file sharing. The other reason is the introduction of mobile Internet, which indirectly induces the malware. Malware which traditionally resides in the wired Internet can now use mobile devices and networks to propagate. The potential effects of malware propagation on mobile users and service providers can be very serious, including deterioration of mobile device performance, excessive charges to mobile users due to excessive mobile data usage, and large scale network breakdowns caused by malware outbreak related issues [3]. Designing an efficient detection and defense system are necessary to prevent such large-scale outbreaks [4][5]; and it should be an urgent and high priority research agenda.

Currently, mobile malware can propagate by using two different dominant approaches. Via MMS, a malware can send a copy of itself to all devices whose numbers are found in the address book of the infected handset. This kind of malware propagates in the social graph formed by the phone address books, and can spread very quickly without geographical limitations. The other approach is to use short range wireless media such as Bluetooth to infect the devices in proximity as “proximity malware”. Recent work of [1] has investigated the proximity malware propagation features, and finds that it spreads slowly because of the human mobility, which offers ample opportunities to deploy a defense system. However, the approach for efficiently deploying such system is still an ongoing research issue. In this paper, we are the first to address the challenge of designing a defense system for both MMS and proximity malware. We introduce an optimal distributed solution to efficiently contain malware spreading and to help infected nodes to recover.

Consider a mobile network where a portion of the nodes are infected by malware. Our research problem is to deploy an efficient defense system to help the infected nodes to recover and prevent the healthy nodes from further infection. Typically, we should disseminate the content-based signatures of known malware to as many nodes as possible [6], [7]. The signature is obtained by using algorithms such as an MD5 hash over the malware content, and they are used by the mobile devices to detect various patterns in the malware and then to disable further propagation. Therefore, distributing these signatures into the whole network while avoiding unnecessary redundancy is our optimization goal. However, to address the above problem in a realistic mobile environment is challenging for several reasons.

First, typically we can not rely on centralized algorithms to distribute the signatures since the service infrastructure may not be always available. For example the existing centralized schemes such as social patching [5] and broadcast signature dissemination [6] have to rely on service provider networks and hence cannot be used without infrastructure support. Therefore, a sensible way for signature distribution is to use a distributed and cooperative way among users. Second, mobile devices in general have limited resources, i.e., CPU, storage, and battery power. Although their storage and CPU capacity have been increasing rapidly recently, it is still very resource-limited compared to desktops. Hence, in the to-be-deployed defense system, we should adequately consider the limitation of resources, especially the memory capacity to store the defense software and signatures. In this aspect, existing distributed malware coping schemes, such as signature flooding [6], which may exhaust the system resources and induce overhead cost, and CPMC [7], which does not take into account the resource limitation at all, are not practical for mobile networks. Finally, the mobile devices are heterogeneous in terms of operating systems (OS), and different malware target different systems. This heterogeneous feature as well as the propagation via both local and global connectivity should be taken into consideration in the design of the defense system for real use.

In this paper, we propose an optimal signature distribution scheme by considering the following realistic modelling assumptions, 1) the network contains heterogeneous devices as nodes, 2) different types of malware can only infect the targeted systems, and 3) the storage resource of each device for the defense system is limited. These assumptions are usually not addressed in previous analytical work for simplicity reasons [4], [5], [7]. Our contributions are summarized as follows:

- We formulate the optimal signature distribution problem with the consideration of the heterogeneity of mobile devices and malware, and the limited resources of the defence system.
- We give a centralized greedy algorithm for the signature distribution problem. We prove that the proposed greedy algorithm can obtain the optimal solution for the system, which provides the benchmark solution for our distributed algorithm design.
- We propose an encounter-based distributed algorithm to disseminate the malware signatures using Metropolis sampler [8]. Through extensive real and synthetic-trace driven simulations, we show that our distributed algorithm approaches the optimal system performance.

The rest of the paper is organized as follows. We describe the system models in Section II, and then formulate the associated optimization problems and give a centralized algorithm in Section III. In Section IV, we design a distributed algorithm. In Section V, we give the performance evaluation. Finally, we present the related work in Section VI and conclude the paper in Section VII.

II. System Description

In this section, we first give an overview of the signature distribution in the defense system, and then give the ordinary differential equation model for the studied system.

A. System Overview

Mobile malware that spreads in the mobile networks typically exploits both MMS and opportunistic contacts to propagate from one device to another device. In the network, there are different types of handsets and each malware only targets handsets with a specific OS. There is limitation in storage on each device for deploying the defense system. Although currently most smart phones have gigabytes of storage, users usually will not allocate all of it to defend from malware, and therefore this assumption is valid. Our goal is to minimize the infected nodes in the system by allocating the limited storage with consideration of different types of malware.

B. Notations and Malware Spreading Model

We consider a system of N heterogeneous wireless nodes belonging to K types, which can be infected by K types of malware, denoted by set $\mathbb{K}$. Since different types of malware will infect different classes of nodes, we let v_k denote the maximum number of nodes that malware k can infect, and let v_k^0 denote the number of infected nodes at the beginning time. In the defense system, we assume that there are S helpers denoted by set $\mathbb{S}$ to store the signatures to help other nodes to detect the malware. Let $x_{s,k}$ denote the indicator whether helper s has the signature to prevent malware k, A_s denote the maximum number of signatures that can be stored at helper s, and u_k denote the number of helpers for malware k.

We first consider the number of nodes infected by malware k in the system at time t, denoted by $\zeta_k(t)$. The dynamic system of malware spreading and defending can be described by the following ordinary differential equation model:

$$\frac{d\zeta_k\left(t\right)}{dt} = \lambda_{k1}\left(v_k - \zeta_k\left(t\right)\right)\zeta_k\left(t\right) - \lambda_{k2}u_k\zeta_k\left(t\right), \quad (1)$$

where λ_{k1} and λ_{k2} are the malware spreading rate by infected nodes and recovering rate by the helpers with signature of malware k, respectively. We obtain this ordinary differential equation by the epidemic spreading model that is widely used in the malware attack analysis [4]. On the other hand, we note that this equation is based on fluid model. The use of fluid approximation is a standard tool in modeling the information propagation of Delay Tolerant Network (DTN) [4], [9], [10].

III. Problem Formulation and Centralized Algorithm

Based on the malware spreading model, we first formulate the problem, and then give a greedy algorithm to achieve the optimal signature distribution.

A. Utility Function

In order to solve Equation (1), we define $z(t)$ as follows,

$$z(t) = (-\lambda_{k1})\zeta_k(t) + \frac{1}{2}(\lambda_{k1}v_k - \lambda_{k2}u_k). \quad (2)$$

Applying variable substitution and by Equation (1), we have

$$\frac{dz\left(t\right)}{dt} = z^2(t) - \frac{1}{4}(\lambda_{k1}v_k - \lambda_{k2}u_k)^2. \quad (3)$$

Obviously $z(0) \neq \pm\frac{1}{2}(\lambda_{k1}v_k - \lambda_{k2}u_k)$ is satisfied, then we have

$$\frac{dz}{z^2 - \frac{1}{4}(\lambda_{k1}v_k - \lambda_{k2}u_k)^2} = dt.$$

Integrating both sides of the above equation, we can get

$$z(t) = \frac{(\lambda_{k1}v_k - \lambda_{k2}u_k)(Ce^{(\lambda_{k1}v_k - \lambda_{k2}u_k)t} + 1)}{2(1 - Ce^{(\lambda_{k1}v_k - \lambda_{k2}u_k)t})}, \quad (4)$$

where the integral constant $C = \frac{z(0)-\frac{1}{2}(\lambda_{k1}v_k-\lambda_{k2}u_k)}{z(0)+\frac{1}{2}(\lambda_{k1}v_k-\lambda_{k2}u_k)}$.

Using Equation (2) and (4), we take the form and have

$$\zeta_k(t) = \frac{\lambda_{k1}v_k - \lambda_{k2}u_k}{\lambda_{k1}\left(1 - \frac{\lambda_{k1}v_k^0 - \lambda_{k1}v_k + \lambda_{k2}u_k}{\lambda_{k1}v_k^0} e^{-(\lambda_{k1}v_k - \lambda_{k2}u_k)t}\right)}.$$

Let h_k be the number of infected nodes by malware k given u_k at time T, which is defined as $h_k = H_k(u_k) = \zeta_k(T)$. We assume that for each malware there is an underlying utility function $G_k(h_k)$ that specifies the system utility of defending malware k given the number of infected nodes at time T. It is natural that $G_k(h_k)$ is a nonincreasing function of h_k. We define $F_k(u_k) = G_k(H_k(u_k))$. Then, we can obtain the following properties, which are proved in the Appendix.

Lemma 1: $H_k(u_k)$ is a decreasing, strictly convex function of the number of helpers in the defense system with the signature of malware k, u_k, when T is large.

Lemma 2: According to Lemma 1 and the condition that $G_k(h_k)$ is a non-increasing and concave function, $F_k(u_k)$ is an increasing and concave function of u_k.

B. Problem Definition

Based on the defined utility function, we use the sum of individual utilities as system welfare with different factor w_k, and specify the studied problem as the following optimization problem,

$$\begin{array}{ll} \text{maximize} & \sum_{k\in\mathbb{K}} w_k F_k\left(\sum_{s\in\mathbb{S}} x_{s,k}\right) \\ \text{over} & x_{s,k}\in\{0,1\}; \\ \text{subject to} & \sum_{k\in\mathbb{K}} x_{s,k} \le A_s, \end{array} \tag{5}$$

where $x_{s,k}=1$ means helper s has the signature of malware k; otherwise, $x_{s,k} = 0$, w_k is the weighting factor, and $F_k\left(\sum_{s\in\mathbb{S}} x_{s,k}\right)$ is the utility function of defending malware k defined in Section III-A. In the formulated problem, we note that the system utility is an increasing and concave function of u_k, and the constraint is convex. Therefore, we can derive the optimal solution by gradient descent algorithm if $x_{s,k}$ is allowed to take the real value. However, in the system, $x_{s,k}$ can either take 1 or 0. Therefore, we should design the corresponding algorithm to obtain the optimal system solution.

C. The Greedy Algorithm

Now, we give a greedy algorithm described in Algorithm 1 for the formulated problem. This kind of greedy approach is widely used in the algorithm design of mobile networks [10], [11]. The obtained result by Algorithm 1 is the optimal solution, which is proved by Theorem 1. The algorithm repeatedly chooses signatures to store for users: in each step, we try to select one signature that brings the maximum system utility for a helper that still has the storage. Therefore, our algorithm is likely to allocate more helpers to store the signatures of malware whose corresponding malware-defending utilities are larger than others, which is achieved by using the heterogenous features in terms of devices and malware.

Theorem 1: The optimal solution of problem defined by Equation (5) is obtained by Algorithm 1, whose computational complexity is $O(K + \sum A_s \cdot \log K)$.

Algorithm 1 The Greedy Algorithm to Maximize the System Welfare

```
Set $x_{s,k} = 0$, $u_k = 0$, $\triangle F_k = 0$ $(k \in \mathbb{K}, s \in \mathbb{S})$;
Initialize *set* $\Re = \{1, 2, \cdots, K\}$ and sum = 0;
**for** Every malware $k$ that $k \in \Re$ **do**
    $\triangle F_k \leftarrow w_k\left(F_k\left(u_k+1\right) - F_k\left(u_k\right)\right)$;
**end for**
**while** sum $\le \sum_{s\in\mathbb{S}} A_s$ and $\Re \ne \emptyset$ **do**
    Select $i$ = arg $\max_k\{\triangle F_k | k \in \mathbb{K}\}$;
    Select $l$ = arg $\max_s\{A_s - \sum_{k\in\mathbb{K}} x_{s,k} | x_{s,i} = 0,\ s \in \mathbb{S}\}$;
    Set $x_{l,i} = 1$;
    Update $u_i \leftarrow u_i + 1$, sum $\leftarrow$ sum + 1;
    Update $\triangle F_i \leftarrow w_i\left(F_i\left(u_i+1\right) - F_i\left(u_i\right)\right)$;
    **if** $u_i \ge S$ **then**
        $\Re \leftarrow \Re \backslash \{i\}$;
    **end if**
**end while**
```

Proof: Based on the problem definition, we can refine it as,

$$\begin{array}{ll} \text{maximize} & \sum_{k\in\mathbb{K}} w_k F_k\left(u_k\right) \\ \text{subject to} & \sum_{k\in\mathbb{K}} u_k \le \sum_{s\in\mathbb{S}} A_s. \end{array}$$

Based on this formula, we define

$$\Delta_k(i) = w_k(F_k(i+1) - F_k(i)), i \in \mathbb{N}, k \in \mathbb{K}. \tag{6}$$

Sort $\Delta_k(i)$ for all $i \in \mathbb{N}, k \in \mathbb{K}$, and denote the result as

$$\widehat{\Delta_1} \ge \widehat{\Delta_2} \ge \widehat{\Delta_3} \ge \ldots \ge \widehat{\Delta_n} \ge \ldots .$$

where $\widehat{\Delta_1}$ is the maximum of all $\Delta_k(i)$, $\widehat{\Delta_2}$ is the second and $\widehat{\Delta_j}$ is the jth largest one, respectively. At each step of the greedy algorithm to maximize the system welfare, we calculate

$$\Delta F_k = w_k(F_k(u_k+1) - F_k(u_k)) = \Delta_k(u_k), k \in \mathbb{K}, \tag{7}$$

choose $i = \text{argmax}_k\{\Delta F_k \mid k \in \mathbb{K}\}$, then update $u_i \leftarrow u_i + 1$ and add ΔF_i to the objective function; let

$$\widehat{\delta_j} = \max\{\Delta F_k \mid k \in \mathbb{K}, \text{at the } j\text{th step of the algorithm}\}.$$

For simplicity, we denote the corresponding $\Delta_k(i)$ of $\widehat{\Delta_j}$ and $\widehat{\delta_j}$ as follows:

$$\widehat{\Delta_j} = \Delta_{K_1(j)}(U_1(j)), \tag{8}$$

$$\widehat{\delta_j} = \Delta_{K_2(j)}(u_{K_2(j)}), \tag{9}$$

where $K_1(j)$, $U_1(j)$, and $K_2(j)$ are the corresponding indexes. Then we have Lemma 3, which is proved in the Appendix.

Lemma 3: In our greedy algorithm, it can be obtained that $\widehat{\delta_j} = \widehat{\Delta_j}$, i.e. the increment of the objective function at the $<j$th$>$ step is exactly the $<j$th$>$ largest element among all $\Delta_k(i)$, in which $i \in \mathbb{N}, k \in \mathbb{K}$.

Now we come back to the validity of our algorithm. From the definition of $\Delta_k(i)$, it can be derived that the objective function becomes

$$\sum_{k\in\mathbb{K}} w_k F_k(u_k) = \left(\sum_{k\in\mathbb{K}} \sum_{i=0}^{u_k-1} \Delta_k(i)\right) + \sum_{k\in\mathbb{K}} w_k F_k(0),$$
$$\text{subject to } \sum_{k\in\mathbb{K}} u_k \le C,$$

where C is a constant. Since $\sum_{k\in\mathbb{K}} w_k F_k(0)$ is a constant and $\left(\sum_{k\in\mathbb{K}} \sum\limits_{i=0}^{u_k-1} \Delta_k(i)\right)$ is a sum of at most C terms of $\Delta_k(i)$, it can be deduced that

$$\left(\sum\nolimits_{k\in\mathbb{K}} \sum_{i=0}^{u_k-1} \Delta_k(i)\right) \le \sum_{j=1}^{C} \widehat{\Delta_j} = \sum_{j=1}^{C} \widehat{\delta}_j, \qquad (10)$$

from the definition of $\widehat{\Delta_j}$ and Lemma 3. From (10) we have already proved that the objective function cannot exceed what we get from the greedy algorithm, thus our algorithm is valid. At the same time, it is obvious that the computational complexity of the algorithm is $O(K + \sum A_s \cdot \log K)$. ■

IV. Distributed Algorithm Using Metropolis Sampler

Now, we consider to design a distributed algorithm for the signature distribution problem. The algorithm is based on a simulated annealing technique called Metropolis sampler. In the following subsections, we first describe the basic notions and framework of the Metropolis sampler (details are available in [8]), then design the distributed algorithm based on simulated annealing with the Metropolis sampler, and finally demonstrate that the proposed algorithm converges to the optimal system performance.

A. The Metropolis Sampler

Consider a Gibbs distribution of $\pi(x)$, whose probability distribution is defined as,

$$\pi(x) = \frac{1}{Z}\exp\left(\frac{\xi(x)}{T}\right), \qquad (11)$$

where x is the configuration, $x\in\mathbb{X}$ being the set of all configurations, $T>0$ is a system parameter named *temperature*, $\xi(\bullet)$ is the *energy* function associating a real number $\xi(x)$ to each configuration $x\in\mathbb{X}$, and Z is the normalizing constant [8]. Since $\pi(x)$ takes its value in $[0,1]$, necessarily $-\infty\le\xi(x)<+\infty$. This distribution has an important property that it favors configuration of larger energy, especially when temperature T is small [8]. More specifically, if we change the configuration x according to certain probability, and after a large number of iterations, the probability distribution of the configuration x converges to the distribution of $\pi(x)$. Therefore, if we set $\xi(x)$ as our system welfare utility function defined in Section III-B as

$$\xi(x) = U(x) = \sum\nolimits_{k\in\mathbb{K}} w_k F_k\left(u_k\right),$$

where $u_k = \sum_{s\in\mathbb{S}} x_{s,k}$, $\pi(x)$ is very much concentrated on the large values of $U(x)$. In other words, we are looking for any configuration of $x_0 \in \mathbb{X}$ such that

$$U(x_0) \ge U(x) \text{ for all } x \in \mathbb{X}.$$

Such a configuration is called the global maximum of the object $U(x)$, and it depends on the global configuration set $\mathbb{X}$. If we choose a configuration according to the current configuration x from a subset $\mathbb{Y} \subset \mathbb{X}$, called the neighborhood of x, and then proceed iteratively as follows: we examine a tentative configuration $x' \in \mathbb{Y}$ chosen according to a rule specific to the algorithm of the Metropolis sampler defined afterwards by comparing $U(x')$ and $U(x)$. If $U(x') > U(x)$, the tentative configuration is accepted as the new configuration, and the system transfers according to an irreducible transition matrix. It iterates and eventually produces a solution, which is locally maximized. Generally speaking, the solution is not optimal since the possible existence of local maximum is not necessary the global maximum. However, by the simulated annealing based Metropolis sampler, the obtained local solution is globally optimal.

Now, we define the Metropolis sampler. Suppose the current configuration is x. At the step of n, a tentative configuration x' is selected according to the probability $\alpha_{x',x}(T_n)$ defined as,

$$\alpha_{x',x}(T_n) = \min\left(1, \frac{\pi(x')}{\pi(x)}\right) = \min\left(1, e^{\left(\frac{U(x')-U(x)}{T_n}\right)}\right), \qquad (12)$$

where T_n is the system temperature [8]. Otherwise x' is rejected. At the same time, let $Q = q_{i,j}$ be an irreducible transition matrix on the system states of $\mathbb{X}$. At step n, the current system configuration is denoted by $X_n(T_n)$, then the process $\{X_n(T_n)\}_{n\ge 0}$ is a homogeneous Markov chain with state space $\mathbb{X}$ and transition matrix $\mathbf{P}(T_n)$ with each element $p_{i,j} = q_{i,j}\alpha_{i,j}(T_n)$.

B. Encounter-based Distributed Algorithm

Based on the introduction of Gibbs distribution and Metropolis sampler, we now design a distributed algorithm for the signature dissemination. Recently, the Metropolis sampler is also used in the algorithm design of DTN [10], [12]. In this work, we use this mechanism to design a distributed algorithms for the mobile malware defense system. We consider every encounter between any two nodes as one step of configuration change of the algorithm. When two nodes i and j meet, each one adjusts its current configuration according to the configuration of the others. More specifically, one node, says i, randomly chooses a signature in its own buffer, and randomly chooses another one that is not in its buffer but in the buffer of node j to replace the chosen signature, which comes to a tentative configuration. After obtaining the replacement probability expressed by the current and tentative configuration, node i decides whether to replace it or not. We assume that the current configuration of the whole system is x, and of the node i is x_i, and node i chooses the signature c' from the buffer of node j to replace the c one. Consequently, the tentative new configuration $x_i' = x_i - 1_{i,c} + 1_{i,c'}$, where $1_{i,c}$ is a vector with the i th value equals 1 and others equal 0, and the system configuration changes to x'. According to these two configurations, we can derive the acceptance probability of the new configuration, as Lemma 4, which is proved in the Appendix. From Lemma 4, we can obtain that the acceptance probability is that if $U(x') \ge U(x)$, then c' is accepted, whereas if $U(x') < U(x)$, a chance, which diminishes as its deviation from c, is left to the signature c'.

This chance is necessary since the solution derived from the local information may deviate from the global optimization. Therefore, the intuition behind our distributed algorithm is that each helper keeps on selecting the signature that gives a higher contribution to the system utility according to the local information. When the nodes encounter more and more, the algorithm approaches to the global optimal solution.

Lemma 4: The acceptance probability $\alpha_{c',c}$ of tentative configuration x', which replaces the signature c with c', is $\alpha_{c',c} = \min(1, \beta)$, where β is expressed as follows,

$$\beta = \exp\left(\frac{w_c\left(\triangle F_c(u_c - 1) + w_{c'}\left(\triangle F_{c'}(u_{c'} + 1)\right)\right)}{T_n}\right),$$

where $\triangle F_c(u_c - 1) = F_c(u_c) - F_c(u_c - 1)$, $\triangle F_{c'}(u_{c'} + 1) = F_c{}'(u_{c'}) - F_{c'}(u_{c'} + 1)$ and T_n is defined in Algorithm 2.

Algorithm 2 The distributed algorithm for malware signature distribution for Node i to adjusts its configuration according to Node j, where T_0 is the initial temperature and n is the encounter counter set to 1 at the beginning

```
if $x_{i,k} == x_{j,k}$ for all $k \in \mathbb{K}$ then
    End the process;
end if
if $\exists k$: $x_{i,k} = 0$ and $x_{j,k} = 1$, which means there is at least one signature in node $j$, but not in node $i$ then
    Set $n \leftarrow n + 1$
    Select a signature $c$ from the buffer of user $i$ uniform randomly such that $x_{i,c} = 1$, and select a signature $c'$ from the buffer of user $j$ uniform randomly such that $x_{j,c'} = 1$ and $x_{i,c'} = 0$;
    Set the system temperature $T_n = \frac{T_0}{\log(n-1)}$
    Compute the acceptance probability $\alpha_{c',c}(T_n)$.
    Draw a random number $R$ uniform distributed in $(0, 1]$;
    if $R < \alpha_{c',c}(T_n)$ then
        User $i$ select signature of $c'$ and drop $c$ with probability of $\frac{1}{SK}\alpha_{c',c}(T_n)$
    end if
end if
```

We note that Algorithm 2 requires nodes to estimate u_c. In the distributed system, every node maintains values of local u_k, $k \in \mathbb{K}$, and updates through the exponential smoothing when two nodes meet by local information exchanging. For example, when nodes i encounters node j and their local information are u_k^i and u_k^j, for all signatures that nodes j carries, node i updates as $u_k^i \leftarrow \beta + (1 - \beta)u_k^i$, and for all other signatures $u_k^i \leftarrow (1 - \beta)u_k^i$ where β is the exponential decay rates. This mechanism is used widely, and its efficiency is verified by recent works in [10], [13].

V. Performance Evaluation

A. Centralized Greedy Algorithm

In this section, we present the numerical results with the goal to demonstrate that our greedy algorithm for the signature distribution, denoted OPT, can achieve the optimal solution and yield significant enhancement on the system welfare compared with prior heuristic algorithms. Related to the heuristic algorithms, we consider 1) Important First (IF), which uses as many as possible helpers to store the signature of the most popular malware, and 2) Uniform Random (UR), where each helper randomly selects the target signatures to store. In order to simulate a more realistic scenario, we model the malware in the system according to the market share of different handset OS of 2009, and set the malware spreading and recovering rates in our model by analyzing the real trace of *Cambridge* collected by the Haggle project [14]. Specifically, by analyzing the malware propagation via the contacts in the trace, we obtain the average malware spreading and recovering rates. In the simulation, we change these two values according to their average value and consider that a system with nodes can be infected by five different types of malware, which are RIM targeted malware 36%; Android targeted 28%; iPhone 21%; Windows Mobile 10% and others 5%. We set $N = 500$ and have 100 helpers to deploy the anti-malware software, and set $T = 10000\ s$. In the experiment setup, the number of initial infected nodes is 10% of all nodes.

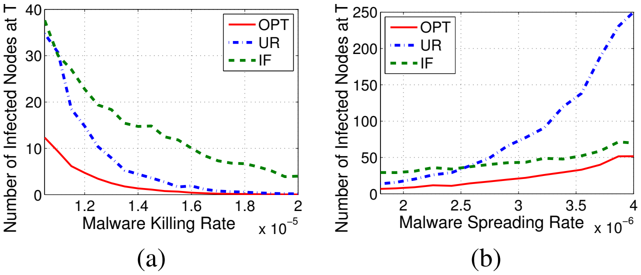
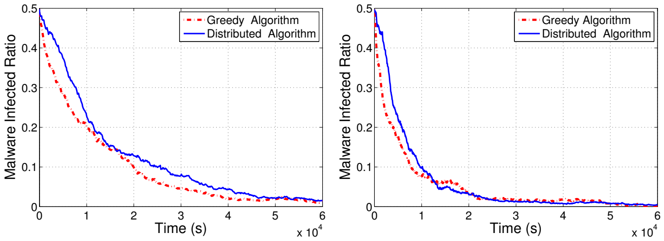


Fig. 1. Results of different defense system deploy algorithms with (a) variable malware recovering rate; and (b) variable malware spreading rate.

The simulation results is shown in Fig. 1. Fig. 1 (a) shows the number of infected nodes according to the malware recovering rates caused by the signature distribution of the greedy algorithm. We can observe that the number of infected nodes decreases with the increase of recovering rate. Among different algorithms, IF provides the worst performance. Fig. 1 (b) shows the number of infected nodes according to the malware spreading rates. Different from Fig. 1 (a), the number of infected nodes increases with the spreading rate. From these results, we can obtain that our OPT has a larger gain on reducing the number of infected nodes than other heuristic algorithms, which shows its efficiency.

B. Distributed Algorithm

In this subsection, we present the simulation results for our distributed algorithm to address the following goals: a) to demonstrate that our distributed algorithm converges to the optimal system performance in realistic environment settings, b) to demonstrate that our scheme of deploying the defense system achieves good performance of preventing the malware propagation under the real-world mobility traces. In order to achieve these two points, we cover a broad set of parameters as follows: a) extensive mobility models of both real- and



(a) (b)

Fig. 3. System performance of malware infected ratio under different mobility models of (a) Random Walk; and (b) Random Waypoint.

synthetic-traces, in which real-traces include both human and taxi mobility traces, while synthetic-traces include the Random WayPoint (RWP) and Random Walk (RW) model, and b) various compared schemes including our centralized optimal greedy algorithm, UR and IF. According to the malware propagation, we use the opportunistic contacts between nodes to spread the proximity malware, while use the phone books generated by the social mode in [15] to spread the MMS malware. More specifically, the infected nodes will transmit the malware to the nodes in its phone book one by one, and the time interval and malware transmission and receiving time are set as exponential distribution. While if they encounter other nodes in proximity, others will be infected immediately.

1) Mobility Model Simulation: We now simulate the greedy algorithm and distributed algorithm under the mobility model of RW and RWP. We first use a network with 500 nodes, and the malware distribution follows the market sharing mentioned above but merges the smallest into the other type. In the simulation, v_0 is set to 50% of the whole nodes, and 10% nodes are set as the helpers with uniform random storage for 1 to 4 signatures. Since we have proved that our greedy algorithm gives the optimal system performance, we use it to compare with the distributed algorithm.

We show the deviation of the number of helpers for each kind of malware (u_1 to u_4) between the greedy algorithm and distributed algorithm in Fig. 2. From the result, we can see that with the increase in time, the deviation converges to 0 in almost all cases. Therefore, this demonstrates the convergence of our distribution algorithm to the optimal signature distribution. Second, we show the malware infected ratio of nodes against time in Fig. 3. From the result, we can see that the greedy algorithm provides better performance than the distributed algorithm when the time is short. But the distributed algorithm approaches to the performance of the greedy algorithm with the increase of time. When time is long enough, these two schemes have the same performance. Therefore, we can conclude that our distribution algorithm approaches the optimal system performance.

2) Real Contact Traces: In order to show the efficiency of our scheme in real mobility environments, we now use real traces for simulation. We use two traces, one is the human mobility contact trace from the Reality Project of MIT [14], the other is the taxi GPS trace of Shanghai [16]. The trace information is given by Table 1. From the features, we can see that they cover a large diversity of DTN environments, from disperse university campus (*Reality*) to concentrated road site (*Shanghai*), with the experiment period from 98 days (*Reality*) to 30 days (*Shanghai*). For the simulated algorithm, we compare our distributed algorithm, denoted DOPT, with OPT, UR and IF. We set all nodes are infected at first, we use 15% nodes as helper to distribute the signatures.

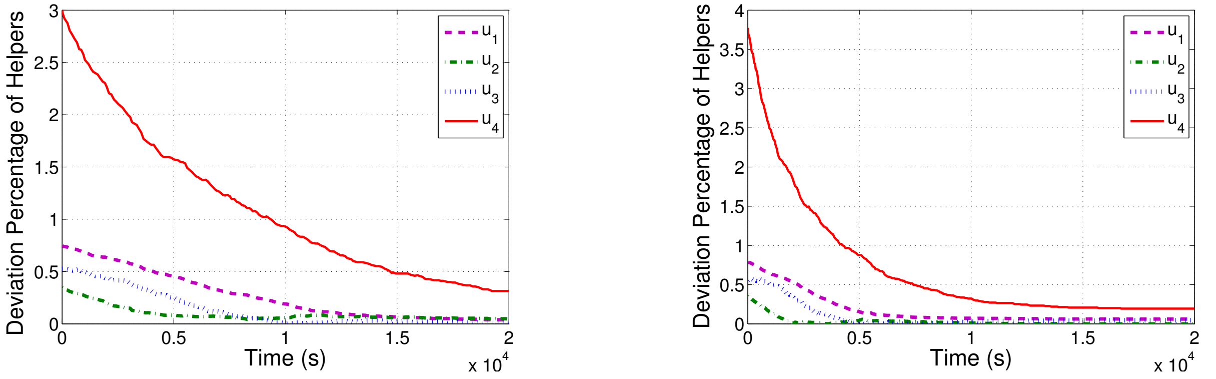
TABLE I
TRACE SUMMARY

Trace	Reality	Shanghai Taxi
Network Type	Bluetooth	GPS
Device	Phone	GPS device
Number of devices	98	2000
Duration (days)	246	30

The results are shown in Fig. 4. From the results, we can see that IF and UR performs worse than our greedy algorithm and distributed algorithm DOPT. Comparing OPT and DOPT, we can observe that DOPT is much closer to the optimal system performance provided by OPT with the increase of time. Therefore, we can conclude that the proposed scheme for the signature distribution achieves good performance of preventing malware propagation under the real-world environments.

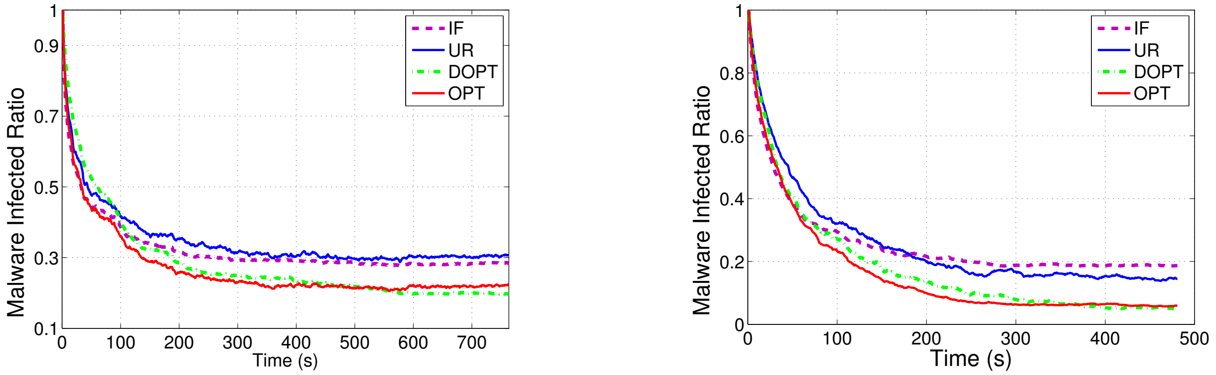
VI. Related Work

With the growth of SMS/MMS, mobile games, mobile commerce and mobile peer-to-peer file sharing, a number of studies have demonstrated the threat of malware propagation on mobile phones through proximity contacts by short-range radio interface and SMS/MMS messages. They can be in general categorized into two main types. One class of works focuses on analyzing the proximity malware spreading. Su et al. [17] demonstrate that malware propagation via Bluetooth is viable by analyzing Bluetooth traces . Yan et al. [18], [19] develop a simulation and analytic model for Bluetooth worms, and show that mobility has a significant impact on the propagation dynamics. The other class focuses on the malware spreading by SMS/MMS. Fleizach et al. [20] evaluate the speed and severity of malware spreading by cell phone address books. Zhu et al. [5] study the characteristics of slow start and exponential propagation exhibited by MMS malware. Besides, a small amount of works also look at both MMS and proximity malware. For example, Bose and Skin [21] investigate the propagation of mobile worms and viruses using data from a real-life SMS customer network, and they reveal that hybrid worms using both MMS and proximity scanning can spread rapidly within cellular networks. Wang et al. [1] model the mobility of mobile phone users by analyzing a trace of 6.2 million mobile subscribers from a service provider. They study the fundamental spreading patterns that characterize a mobile virus outbreak and find that the greatest danger is posed by hybrid viruses that take advantage of both proximity and MMS. Obtaining the insights of these two works, our model considers both the MMS and proximity propagation in our defense system design.



(a) (b)

Fig. 2. Convergence of the helper under different mobility models of (a) Random Walk; and (b) Random Waypoint.



(a) (b)

Fig. 4. System performance of malware infected ratio with real trace of (a) Reality human trace; and (b) Shanghai vehicle trace.

For performance evaluation and modeling of mobile malware spreading, the epidemic model, based on the classical Kermack-Mckendrick model [22] traditionally used in wired networks, has been extensively used in [6], [23], [4], [1] etc. Actually, the system performance of the epidemic model can be approximated by the Ordinary Differential Equations (ODE) with a well known technique called fluid model [9] being widely used to model the epidemic forwarding in DTN [9], [10], [24]. In the fluid model, the solution of the ODE converges in probability to the system's sample paths. These works show that when the number of nodes in a network is large, the deterministic epidemic models can successfully represent the dynamics of malware spreading, which is demonstrated by simulations and matching with actual data. We use a ODE model to analyze and design the signature distribution problem in the malware defense system. Therefore, our model in this work is reasonable.

Recently, some malware coping schemes have been proposed to defend mobile devices against malware propagation. In order to prevent the malware spreading by MMS/SMS, Zhu et al. [5] propose a counter-mechanism to stop the propagation of a mobile worm by patching an optimal set of selected phones by extracting a social relationship graph between mobile phones via an analysis of the network traffic and contact books. This approach only targets the MMS spreading malware and has to be centrally implemented and deployed in the service provider's network. In order to defend mobile networks from proximity malware by Bluetooth, Gjergji et al. [6] explore three strategies, including local detection, proximity signature dissemination and broadcast signature dissemination. For detecting and mitigating proximity malware, Li et al. [7] propose a community-based proximity malware coping scheme by utilizing the social community structure reflecting a stable and controllable granularity of security. These two works both target the proximity malware. The former one has the limitations that signature flooding costs too much and the local view of each nodes constrains the global optimal solution. Although the aftermath scheme integrates short-term coping components to deal with individual malware and long-term evaluation components to offer vulnerability evaluation towards individual nodes, the social community information still need to be obtained in a centralized way. There are significant differences between these works and our work. First, our scheme targets both the MMS and proximity malware at the same time, and consider the problem of signature distribution. Second, all these works assume that malware and devices are homogeneous, we take the heterogeneity of devices into account to deploy the system and consider the system resource limitations. Third, our scheme is a distributed algorithm, and approaches to the system optimal solution.

VII. Conclusions

In this paper, we investigate the problem of optimal signature distribution to defend mobile networks against the propagation of both proximity and MMS based malware. We introduce a distributed algorithm that can closely approach the optimal system performance of a centralized solution. Through extensive simulations, we demonstrate the efficiency of our defense scheme in significantly reducing the amount of infections in the system.

At the same time, a number of open questions remain unanswered. For example, the malicious nodes may inject some dummy signatures targeting no malware into the network and induce denial-of-service attacks to the defense system. Therefore, security and authentication mechanisms should be considered. From the aspect of malware, since some sophisticated malware that can bypass the signature detection would emerge with the development of the defense system, new defense mechanisms will be required. We are continuing to cover these topics in the future work.

Acknowledgment

This work was supported by National Major Scientific and Technological Specialized Project (No. 2010ZX03004-002-02), National Basic Research Program (No. 2007CB310701), National High Technology Research and Development Program (No. 2008AA01Z107, No. 2008AA01A331 and No. 2009AA011205), NSFC (No. 61021001), PCSIRT and TNLIST.

References

[1] P. Wang, M. Gonzalez, C. Hidalgo, and A. Barabasi, "Understanding the spreading patterns of mobile phone viruses," *Science*, vol. 324, no. 5930, p. 1071, 2009.

[2] M. Hypponen, "Mobile Malwar," in *Proc. of 16 USENIX Security Symposium*, 2007.

[3] G. Lawton, "On the trail of the Conficker worm," *Computer*, vol. 42, no. 6, pp. 19–22, 2009.

[4] M. Khouzani, S. Sarkar, and E. Altman, "Maximum Damage Malware Attack in Mobile Wireless Networks," in *Proc. of IEEE INFOCOM*, 2010.

[5] Z. Zhu, G. Cao, S. Zhu, S. Ranjan, and A. Nucci, "A Social Network Based Patching Scheme for Worm Containment in Cellular Networks," in *Proc. of IEEE INFOCOM*, 2009.

[6] G. Zyba, G. Voelker, M. Liljenstam, A. Méhes, and P. Johansson, "Defending mobile phones from proximity malware," in *Proc. of IEEE INFOCOM*, 2009.

[7] F. Li, Y. Yang, and J. Wu, "CPMC: An Efficient Proximity Malware Coping Scheme in Smartphone-based Mobile Networks," in *Proc. of IEEE INFOCOM*, 2009.

[8] P. Brémaud, *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*. Springer Verlag, 1999.

[9] E. Altman, G. Neglia, F. De Pellegrini, and D. Miorandi, "Decentralized Stochastic Control of Delay Tolerant Networks," in *Proc. of IEEE INFOCOM*, 2009.

[10] L. Hu, J. Boudec, and M. Vojnovic, "Optimal channel choice for collaborative ad-hoc dissemination," in *Proc. of IEEE INFOCOM*, 2010.

[11] J. Reich and A. Chaintreau, "The age of impatience: optimal replication schemes for opportunistic networks," in *Proc. of ACM CoNext*, 2009, pp. 85–96.

[12] A. Picu and T. Spyropoulos, "Distributed stochastic optimization in opportunistic networks: the case of optimal relay selection," in *Proc. of the 5th ACM workshop on Challenged networks*, 2010, pp. 21–28.

[13] S. Ioannidis, L. Massoulié, and A. Chaintreau, "Distributed Caching over Heterogeneous Mobile Networks," in *Proc. of ACM SIGMETRICS*, 2010.

[14] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay tolerant networks," in *Proc. of ACM MobiHoc*, 2008.

[15] J. Kumpula, J. Onnela, J. Saramäki, K. Kaski, and J. Kertész, "Emergence of communities in weighted networks," *Physical review letters*, vol. 99, no. 22, p. 228701, 2007.

[16] S. J. U. Traffic Information Grid Team, Grid Computing Center, "Shanghai taxi trace data," *http://wirelesslab.sjtu.edu.cn/*.

[17] J. Su, K. Chan, A. Miklas, K. Po, A. Akhavan, S. Saroiu, E. de Lara, and A. Goel, "A preliminary investigation of worm infections in a bluetooth environment," in *Proc. of the 4th ACM workshop on Recurring malcode*, 2006, p. 16.

[18] G. Yan and Eidenbenz, "Modeling Propagation Dynamics of Bluetooth Worms," *IEEE Transactions on Mobile Computing*, vol. 8, no. 3, p. 1071, 2008.

[19] G. Yan, H. Flores, L. Cuellar, N. Hengartner, S. Eidenbenz, and V. Vu, "Bluetooth worm propagation: mobility pattern matters!" in *Proc. of ACM symposium on Information, computer and communications security*, 2007, p. 44.

[20] C. Fleizach, M. Liljenstam, P. Johansson, G. Voelker, and A. Mehes, "Can you infect me now? malware propagation in mobile phone networks," in *Proc. of ACM workshop on Recurring malcode*, 2007, p. 68.

[21] A. Bose and K. Shin, "On mobile viruses exploiting messaging and bluetooth services," in *Securecomm and Workshops*, 2006, pp. 1–10.

[22] D. Daley and J. Gani, *Epidemic modelling: an introduction*. Cambridge Univ Press, 2001.

[23] J. Mickens and B. Noble, "Modeling epidemic spreading in mobile environments," in *Proc. of the 4th ACM workshop on Wireless security*, 2005, p. 86.

[24] T. B. Eitan Altman, Prakash Azad and F. D. Pellegrini, "Optimal Activation and Transmission Control in Delay Tolerant Networks," in *Proc. of IEEE INFOCOM*, 2010.

Appendix

A. Proof of Lemma 1

Proof: Perform the variable substitution of $z = \lambda_{k2}u_k - \lambda_{k1}v_k$,, then $H_k(u_k)$ becomes

$$F(z) = \frac{-z}{\lambda_{k1}\left(1-(1+Bz)\,e^{zT}\right)}, \tag{13}$$

where constant $B = \frac{1}{\lambda_{k1}v_k^0}$. We investigate the monotonicity and concavity-convexity of $F(z)$ first. Define function $H(z)$ as $H(z) = 1-(1+Bz)e^{zT}$, then we get $F(z) = -z/(\lambda_{k1}H(z))$. Calculate the first and second order derivatives of $F(z)$, we have

$$\frac{dF(z)}{dz} = \frac{-H(z)+zH'(z)}{\lambda_{k1}H^2(z)}$$

$$\frac{d^2F(z)}{dz^2} = \frac{zH''(z)H(z) - 2H'(z)\left(-H(z)+zH'(z)\right)}{\lambda_{k1}H^3(z)}$$

Define $G_1(z) = -H(z) + zH'(z)$, so that $\frac{dF(z)}{dz} = \frac{G_1(z)}{\lambda_{k1}H^2(z)}$. Next we will show $G_1(z)$ is non-positive for all $z \in \mathbb{R}$. Considering the definition of $H(z)$, we know

$$G_1(z) = -1 + e^{zT}(1 - zT(1+Bz)),$$

thus

$$\frac{dG_1(z)}{dz} = -e^{zT}(zT^2(1+Bz) + 2BTz).$$

Now it is clear that $\frac{dG_1(z)}{dz} > 0$ if and only if $z \in (-\frac{T+2B}{BT}, 0)$; consequently, $G_1(z)$ is decreasing on interval $\left(-\infty, -\frac{T+2B}{BT}\right]$ as well as on $[0, \infty)$, respectively , and $G_1(z)$ is increasing on interval $\left[-\frac{T+2B}{BT}, 0\right]$. Combined with the facts that $G_1(0) = 0$ and $\lim\limits_{z \to -\infty} G_1(z) = -1$, we have already proved $G_1(z) \le 0$. Consequently,

$$\frac{dF(z)}{dz} \le 0,$$

so F(z) is decreasing for $z \in \mathbb{R}$.

Similarly, define

$$G_2(z) = zH''(z)H(z) - 2H'(z)\left(-H(z) + zH'(z)\right),$$

then $\frac{d^2F(z)}{dz^2} = \frac{G_2(z)}{\lambda_{k1}H^3(z)}$. For simplicity, we focus on the scenario where $z > 0$ and T is large enough. It is easy to achieve $H(z) < 0$ for $z > 0$, and next we consider the sign of $G_2(z)$. The analytical expression of $G_2(z)$ is as following:

$$\begin{aligned} G_2(z) = & - e^{zT}(BT^2z^2 + (T^2 + 4BT)z + 2B + 2T \\ & + e^{zT}\left(B^2T^2z^3 + 2BT^2z^2 + (T^2 - 2BT)z - 2B - 2T\right)\right). \end{aligned}$$

If T satisfies $T \ge \frac{1+\sqrt{1+2Bz}}{z(Bz+1)}$, one can know quadratic function

$$\left(B^2T^2z^3 + 2BT^2z^2 + (T^2 - 2BT)z - 2B - 2T\right)$$

of variable T is non-negative. Combined with the fact B, T and z are all positive, it can be obtained that $G_2(z) < 0$ and

$$\frac{d^2F(z)}{dz^2} > 0.$$

Thus, we have proved that $F(z)$ is decreasing for $z \in \mathbb{R}$ and convex when $z > 0$ and $T \ge \frac{1+\sqrt{1+2Bz}}{z(Bz+1)}$. At the same time, z is a linear function of u_k with the slope $\lambda_{k2} > 0$, so the monotonicity and concavity-convexity of $F(z)$ is equivalent with those of $H_k(u_k)$. Finally, we arrive at the conclusion that $H_k(u_k)$ is decreasing for $z \in \mathbb{R}$ and convex when $z > 0$ and $T \ge \frac{1+\sqrt{1+2Bz}}{z(Bz+1)}$. ■

B. Proof of Lemma 2

Proof: Calculate the second-order derivative of $F_k(u_k)$ according to the chain rule,

$$\frac{\mathrm{d}^2F_k}{\mathrm{d}u_k^2} = \frac{\mathrm{d}}{\mathrm{d}u_k}\left(\frac{\mathrm{d}G_k}{\mathrm{d}y} \cdot \frac{\mathrm{d}H_k}{\mathrm{d}u_k}\right) = \frac{\mathrm{d}^2G_k}{\mathrm{d}y^2}\left(\frac{\mathrm{d}H_k}{\mathrm{d}u_k}\right)^2 + \frac{\mathrm{d}G_k}{\mathrm{d}y}\frac{\mathrm{d}^2H_k}{\mathrm{d}u_k^2}.$$

Since $H_k(u_k)$ is an increasing and strictly convex function of u_k and $G_k(h_k)$ is a non-increasing function, $G_k(H_k(u_k))$ is an increasing and concave function of u_k, which proves the Lemma.

■

C. Proof of Lemma 3

Proof: First, we prove $\widehat{\delta_j} \ge \widehat{\delta_{j+1}}$. If $K_2(j) = K_2(j+1)$, it can be deduced that $u_{K_2(j+1)} = u_{K_2(j)} + 1$ since the update of $u_{K_2(j)}$ is performed at the <jth> step. From the concavity of $F_k(u_k)$, we know $F_k(u_k + 1) - F_k(u_k) \ge F_k(u_k + 2) - F_k(u_k + 1)$, and thus $\Delta_k(u_k) \ge \Delta_k(u_k + 1)$; substitute k and u_k with $K_2(j)$ and $u_{K_2(j)}$, respectively, it is finally achieved that $\widehat{\delta_j} \ge \widehat{\delta_{j+1}}$. If $K_2(j) \ne K_2(j+1)$, we know $u_{K_2(j+1)}$ remains the same between the <jth> and <$(j+1)$th> step of the algorithm, thus $\Delta F_{K_2(j+1)}$ is the same for the <jth> and <$(j+1)$th> step; since $K_2(j) = \mathrm{argmax}_k\{\Delta F_k \mid k \in \mathbb{K}, at\ the j\mathrm{th}\ step\}$, we know $\Delta F_{K_2(j)}$ is no less than $\Delta F_{K_2(j+1)}$ at the <jth> step, consequently, we have

$$\widehat{\delta_j} = \Delta F_{K_2(j)} \ge \Delta F_{K_2(j+1)} = \widehat{\delta_{j+1}}. \tag{14}$$

Thus, it is obtained that

$$\widehat{\delta_1} \ge \widehat{\delta_2} \ge \widehat{\delta_3} \ge \ldots \ge \widehat{\delta_n} \ge \ldots$$

and we have $\widehat{\delta_j} \le \widehat{\Delta_j}$ by the definition of $\widehat{\Delta_j}$.

Next we obtain $\widehat{\delta_j} = \widehat{\Delta_j}$ by proof of contradiction. If it is not true, we can denote $j_0 = \min\{j \mid \widehat{\delta_j} \ne \widehat{\Delta_j}\}$. Since we have already proved that $\widehat{\delta_j} \le \widehat{\Delta_j}$, it can be obtained $\widehat{\delta_{j_0}} < \widehat{\Delta_{j_0}}$. Without loss of generality, we can assume that if $\widehat{\Delta_j} = \widehat{\Delta_{j+1}}$, the indexes will satisfy $K_1(j) < K_1(j+1)$ or $K_1(j) = K_1(j+1)$ and $U_1(j) < U_1(j+1)$. Furthermore, from (8) we have $\widehat{\Delta_{j_0}} = \Delta_{K_1(j_0)}(U_1(j_0))$. From our proof in the previous paragraph, it can be attained that $\Delta_{K_1(j_0)}(q) \ge \Delta_{K_1(j_0)}(U_1(j_0))$ for all $q < U_1(j_0)$. Since j_0 is the minimum and from the assumption of indexes above, we can know $\Delta_{K_1(j_0)}(q),\ 1 \le q < U_1(j_0)$ all appear in $\widehat{\Delta_j},\ j < j_0$, and thus they appear in $\widehat{\delta_j},\ j < j_0$. Consider the <$j_0$th> step of algorithm, it can be deduced that $u_{K_1(j_0)} = U_1(j_0)$ from above analysis, thus $K_1(j_0) = \mathrm{argmax}_k\{\Delta F_k \mid k \in \mathbb{K},\ at\ the\ j_0\mathrm{th}\ step\}$ because $\Delta F_{K_1(j_0)} = \widehat{\Delta_{j_0}}$ and $\widehat{\Delta_{j_0}} \ge \widehat{\Delta_{j'}}$ when $j' > j_0$. Consequently, from the algorithm we know $\widehat{\delta_{j_0}} = \widehat{\Delta_{j_0}}$, which lead a contradiction. Thus, we have finished the proof of *Lemma* 3.

■

D. Proof of Lemma 4

Proof: From Equation (12) in the description of Metropolis sampler, we have

$$\beta = \exp\left(\frac{U(x') - U(x)}{T_n}\right).$$

Since $U(x) = \sum\limits_{k \in \mathbb{K}} w_k F_k(u_k)$ and the difference between x' and x only lies in x_i and $x'_i = x_i - 1_{i,c} + 1_{i,c'}$, we have

$$U(x) = \sum_{k \in \mathbb{K}, k \ne c, c'} w_k F_k(u_k) + w_c F_c(u_c) + w_{c'} F_{c'}(u_{c'})$$

$$U(x') = \sum_{k \in \mathbb{K}, k \ne c, c'} w_k F_k(u_k) + w_c F_c(u'_c) + w_{c'} F_{c'}(u'_{c'})$$

where $u'_c = \sum\limits_{s \in \mathbb{S}} x'_{s,c}$. At the same time, note that $u'_c = u_c - 1$ and $u'_{c'} = u_{c'} + 1$, we have $U(x') - U(x) =$

$$w_c\left(F_c(u_c) - F_c(u_c - 1)\right) + w_{c'}\left(F_{c'}(u_{c'}) - F_{c'}(u_{c'} + 1)\right),$$

which proves the Lemma. ■