How Can Botnets Cause Storms? Understanding the Evolution and Impact of Mobile Botnets

Zhuo Lu Wenye Wang
Department of Electrical and Computer Engineering
North Carolina State University, Raleigh NC 27606
Emails: {zlu3, wwang}@ncsu.edu

Cliff Wang
Army Research Office
Research Triangle Park, NC 27709
Email: cliff.wang@us.army.mil

***Abstract*—A botnet in mobile networks is a collection of compromised nodes due to mobile malware, which are able to perform coordinated attacks. Different from Internet botnets, mobile botnets do not need to propagate using centralized infrastructures, but can keep compromising vulnerable nodes in close proximity and evolving organically via data forwarding. Such a distributed mechanism relies heavily on node mobility as well as wireless links, therefore breaks down the underlying premise in existing epidemic modeling for Internet botnets.**

In this paper, we adopt a stochastic approach to study the evolution and impact of mobile botnets. We find that node mobility can be a trigger to botnet propagation storms: the average *size* (i.e., number of compromised nodes) of a botnet increases quadratically over time if the mobility range that each node can reach exceeds a threshold; otherwise, the botnet can only contaminate a limited number of nodes with average size always bounded above. This also reveals that mobile botnets can propagate at the fastest rate of quadratic growth in size, which is substantially slower than the exponential growth of Internet botnets. To measure the denial-of-service impact of a mobile botnet, we define a new metric, called *last chipper time*, which is the last time that service requests, even partially, can still be processed on time as the botnet keeps propagating and launching attacks. The last chipper time is identified to decrease at most on the order of $1/\sqrt{B}$, where B is the network bandwidth. This result reveals that although increasing network bandwidth can help with mobile services; at the same time, it can indeed escalate the risk for services being disrupted by mobile botnets.

I. Introduction

With the proliferation of smart handheld devices and the exploded number of malware on mobile platforms, a mobile botnet [1], [2], which is a collection of compromised (or infected) mobile nodes. that can perform coordinated attacks, no longer occurs in theory, but comes into practice. For example, *Ikee.B* [3] in 2009 was found to include command and control logic to render a number of infected iPhones under the control. In 2012, Symantec found a large botnet *Android.Bmaster* [4] in China that had infected an estimate of hundreds of thousands of Android phones. As a result, mobile botnets have already become one of the most serious security threats to today's mobile networks and applications.

A mobile botnet can compromise vulnerable nodes by sending malware via centralized infrastructures (e.g., using short and multimedia message services [1], [4], [5]). However, to eschew increasingly enhanced monitoring of cellular infrastructures, a stealthy way for propagation is to stay off the radar and spread to vulnerable nodes nearby, which has been adopted in existing malware, such as Mabir, Lansco and CPMC [6]. A challenging question is how botnets propagate via such proximity infection, especially how they behave in mobile networks compared with their forerunners in the Internet.

Extensive works have investigated Internet malware propagation using epidemic modeling (e.g., [7], [8]), which presumes a condition that an infected node can compromise other vulnerable nodes with equal probability. A few studies [9], [10] have adapted epidemic modeling to characterize mobile malware based on simplistic random movements, where the equal-probability assumption still holds. These prior efforts conclude that using proximity infection, malware can continue infecting more nodes without using infrastructures, thereby leading to severe epidemics. This result is also observed by a number of experiments [11]–[13]. Interestingly, however, a recent paper [14] draws an opposite conclusion based on simulations that proximity infection only affects a limited number of nodes and is far less concerning in urban environments where node susceptibility is relatively low. These somewhat discrepant results may be due to different system setups, such as transmission range and random mobility. Nonetheless, the primary reason is still unclear. As a result, it is not yet fully understood *how proximity infection can cause a botnet propagation storm and what the impact is in mobile networks*.

In this paper, we are motivated to address this open question by considering a practical scenario with heterogeneous mobility, in which nodes are more likely to move around in certain areas. Such heterogeneity inevitably breaks the premise of equal-probability infection used in existing epidemic modeling [9], [10]. Thus, we take a stochastic approach to study how a mobile botnet evolves. In particular, we denote by $\mathcal{S}(t)$ the set of infected nodes in a mobile botnet at time t. The botnet originates from an initially infected node that starts to move around and compromise nearby vulnerable nodes at time 0. We are interested in how the *botnet size* $|\mathcal{S}(t)|$ (defined as the number of infected nodes in the botnet) increases over time t.

Our results reveal an interesting dichotomy of mobile botnet propagation: the average size of a mobile botnet $\mathbb{E}|\mathcal{S}(t)|$ either grows quadratically over time t or is always bounded above. In particular, given node density λ, wireless transmission range r, and mobility radius α that is the maximum range that a node

The work was sponsored by ARO staff research grant W911NF-07-R-0001-05 and NSF Career Award CNS-0546289.

978-1-4799-3360-0/14/$31.00 © 2014 IEEE

can reach, we find that as long as $\lambda(2\alpha{+}r)^2$ exceeds a threshold, $\mathbb{E}|\mathcal{S}(t)|$ is a *quadratical* function of t; otherwise, $|\mathcal{S}(t)|$ is *finite almost surely* with eventual size $|\mathcal{S}(\infty)|$ exponentially distributed. This means that with fixed network setups λ and r, sufficient mobility (i.e., mobility radius α becomes large) can provoke mobile botnet propagation from limited infection to epidemics. Therefore, our findings not only serve as a bridge to connect two discrepant results in the literature, but also reveal that mobile botnets via proximity infection can propagate at the fastest rate of quadratic growth, which is much slower than the exponential growth of Internet botnets.

In order to measure the denial-of-service impact of a mobile botnet with quadratic growth in size, we define *last chipper time*, the last time moment that a required ratio σ of service requests from mobile nodes to a service center can still be processed on time, while the botnet keeps propagating and attacking. We find that the last chipper time decreases at most on the order of $1/\sqrt{B\log(1/(1{-}\sigma))}$, where B is the network bandwidth. Based on this, we can *quantitatively* assess how increasing network bandwidth induces the risk of botnets to disrupt mobile services. For example, the bandwidth of current cellular networks is expected to increase 10 times from LTE to LTE advanced, a mobile botnet, in the fastest case, needs to propagate only one third (i.e., $1/\sqrt{10}$) of the time that it spends in LTE to disrupt the same service in LTE advanced.

The reminder of this paper is organized as follows. In Section II, we introduce preliminaries and models. In Sections III and IV, we investigate how a mobile botnet evolves and what its impact is. Finally, we conclude in Section V.

II. Preliminaries and Models

In this section, we first present the models used in this paper, then formulate the research problem.

A. Network and Mobile Users

We consider a hybrid mobile network with two distinct types of nodes: mobile nodes that are common users moving around in the network, and infrastructure nodes that are base stations or access points to provide mobile services to mobile nodes.

There are n mobile nodes distributed independently and uniformly on a torus surface $\Omega = [0, \sqrt{\frac{n}{\lambda}}]^2$ for some node density λ. Infrastructure nodes form square cells in the network, as shown in Fig. 1(a). They have the wireless network interface that offers wireless access to mobile nodes. In addition, they are interconnected with each other via high-speed wireline networks and are also connected to a data service center that processes service requests from mobile nodes.

Mobile nodes are able to communicate directly with each other, and can also communicate with their nearest infrastructure nodes for mobile services. As shown in Fig. 1(b), the transmission ranges of mobile and infrastructure nodes are the same and denoted by r. The network bandwidth B is shared among all mobile and infrastructure nodes. Mobile nodes consist of legitimate nodes and malicious nodes that are compromised by malware and attempt to infect other mobile nodes in the network. Infrastructure nodes, on the other hand, are invulnerable to malware infection.

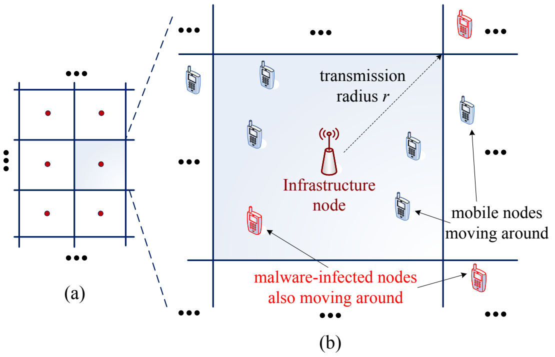


Fig. 1. Network architecture: infrastructure nodes and mobile nodes.

B. Mobile Malware and Botnet

When a mobile node is infected by malware, it may not behave legitimately. Generally speaking, mobile malware is malicious software on mobile platforms that attempts to take control of a device and copy itself to other susceptible devices, which is called malware propagation [1], [3]. More dangerously, if mobile nodes are infected by the same malware, they can form a mobile botnet [2], [3] that is a collection of compromised mobile devices under the same control. Mobile botnets have already been found in practice, such as *Ikee.B* in 2009 [3] and *Android.Bmaster* in 2011 [4]. In essence, a mobile botnet can be formed in the following two ways: (i) propagation through infrastructures (malware sending its copies using short/multimedia message services or advertising its applications (APPs) on mobile markets [1], [4], [5]), (ii) proximity infection (a compromised node sending malware to nearby nodes using peer-to-peer wireless links [6], [14]).

Although botnet propagation is very fast through infrastructures, it can be easily ceased by increasingly enhanced security systems at infrastructures (e.g., Google's Android kill switch). Hence, a stealthy and safe way for propagation is to infect vulnerable nodes nearby, because such proximity infection can easily persist and remain undetected due to the nature of decentralized infection and the dynamic network topology. The proximity infection mechanism has already been found in existing malware, such as Mabir, Lansco and CPMC [6].

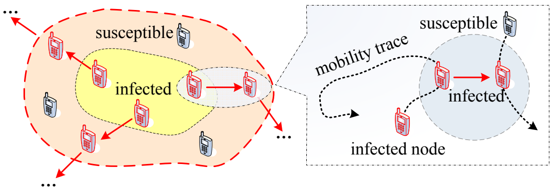


Fig. 2. Mobile botnet evolution over time via proximity infection.

Accordingly, we focus on the scenario in which malware intends to use proximity infection to form a botnet. We consider the malware infection process starting from one initially infected node that attempts to propagate malware to other vulnerable nodes in the network. As shown in Fig. 2, a compromised node propagates malware to the other node when (i) the two nodes must move into each other's wireless transmission range r; (ii) the other node must be susceptible to malware (a vulnerability ratio $\kappa \in (0,1)$ is used to denote the probability that a node is vulnerable); and (iii) the required infection time (how long it takes to infect a node) is randomly distributed in a range $[\delta_1, \delta_2]$. This is because the spread of malware requires some time for user or application interaction. If two nodes move out of each other's range and have no time to finish the interaction, a node cannot be infected even if it is vulnerable. Thus, our model also accommodates the case of limited contact or interaction time.

1) Node Mobility: Mobility plays an essential role in the performance of mobile applications, and accordingly has substantial impacts on malware propagation [13]. We consider a generic mobility model that accounts for a practical scenario of spatial heterogeneity, in which mobile nodes are more likely to stay in certain areas (e.g., their homes or offices) and less likely to be in others. In particular, similar to existing works [15], [16], we define that for a mobile node m_i, there exist a home point h_{m_i}, which is independently and uniformly distributed over region Ω. We also define a mobility radius α for m_i such that m_i moves around h_{m_i} with probability density function $\Psi(x)$, which is invariant in all directions and satisfies $\Psi(x)>0$ when $\|x-h_{m_i}\| \leq \alpha$, and $\Psi(x)=0$ otherwise. In addition, all mobile nodes move around their home points according to independent stationary processes.

We assume that malware can only compromise the software in a vulnerable node, but cannot decide the node's movement since mobility is usually determined by human beings.

C. Problem Formulation

As the initially infected node moves around and intends to spread malware to other vulnerable nodes starting from time 0, it can be expected that more and more nodes are infected and repeat the same infection process in the network. Therefore, a large-scale mobile botnet might be built from the scratch with sufficient time. Such a botnet could be very detrimental to mobile users as well as mobile service operations.

In order to understand the potential impact of a mobile botnet, we first need to investigate how it evolves over time; i.e., we are interested in how many nodes in total have been infected at a particular time t. To proceed, we define the size of a mobile botnet as follows.

Definition 1: A mobile botnet, denoted by $\mathcal{S}(t)$, is the set of all malware-infected nodes at time t. The size of the botnet $|\mathcal{S}(t)|$ is defined as the total number of nodes in $\mathcal{S}(t)$.

With Definition 1, we further characterize how fast a mobile botnet can spread malware in the network. Specifically, we define the evolution speed of a botnet in the following.

Definition 2: The evolution speed of a mobile botnet, denoted by $V(t)$, is defined as $V(t) = \mathbb{E}|\mathcal{S}(t)|/t$, where $\mathbb{E}|\mathcal{S}(t)|$ is the average number of nodes in $\mathcal{S}(t)$ at time t.

Given Definitions 1 and 2, we formally state our research problem: for a mobile botnet originated from one initially infected node at time 0, what its size $|\mathcal{S}(t)|$ and evolution speed $V(t)$ are at time $t > 0$?

III. How Does A Mobile Botnet Evolve over Time?

In this section, we first investigate the size of a mobile botnet $|\mathcal{S}(t)|$ and its evolution speed $V(t)$, then use mobility traces to show botnet propagation in realistic environments.

A. The Average Size and Evolution Speed

From Definition 2, we know that the evolution speed of a botnet $V(t)$ is based on the average size $\mathbb{E}|S(t)|$. Thus, we first investigate the size of a mobile botnet at time t.

Theorem 1 (Size of a mobile botnet): For a mobile botnet, its average size $\mathbb{E}|\mathcal{S}(t)|$ at time t can be written as

$$\mathbb{E}|\mathcal{S}(t)| = \begin{cases} \Theta(1) & if\ \kappa\lambda(2\alpha+r)^2 = O(1) \\ \Theta(t^2) & if\ \kappa\lambda(2\alpha+r)^2 = \Omega(1), \end{cases}$$

where κ is the vulnerability ratio, λ is the node density, α is the mobility radius, and r is the wireless transmission range.[1]

Proof: This theorem consists of two parts. We first consider the $\mathbb{E}|\mathcal{S}(t)| = \Theta(1)$ part, then the $\mathbb{E}|\mathcal{S}(t)| = \Theta(t^2)$ one.

Without loss of generality, assume that mobile node m_1 is the initially infect node that moves around in the network and attempts to infect vulnerable nodes as many as possible. Once a node is infected by node m_1, it will also start to infect others. This means that this node can be considered as an offspring of node m_1. Thus, proximity infection can be modeled based on a branching process [17] that characterizes how a population evolves from generations to generations.

We consider node m_1 as the only node in the 1st generation, the nodes directly infected by node m_1 as the 2nd generation, and so on. Now construct a branching process $\{Z_i\}$ satisfying $Z_{i+1} = \sum_{j=1}^{Z_i} Y_{i,j}$, where $Y_{i,j}$ is the number of nodes infected directly by the j-th infected node of generation i.

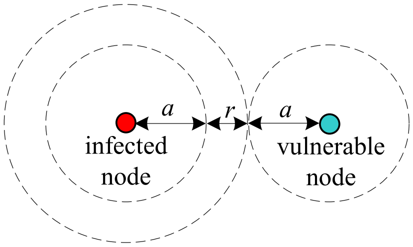


Fig. 3. The maximum possible range that a node can infect the other.

First take a look at node m_1 (i.e., the 1st infected node of generation 1). As shown in Fig. 3, it is impossible for node m_1 to infect a node whose home point has a distance to m_1's larger than $2\alpha + r$ since there is no way for the node to move into m_1's contact region. Let $Y'_{1,1}$ be the total number of vulnerable nodes that are able to move into the contact region of node m_1. Then, it always holds that $Y_{1,1} \leq Y'_{1,1}$ at any time. Similarly, we have i.i.d. random variables $\{Y'_{i,j}\}$ that satisfy

$$Y_{i,j} \leq Y'_{i,j} \qquad \text{for any } i, j > 0. \tag{1}$$

[1] We say $f(x) = O(g(x))$ if $\exists\ x_0$ and $c > 0$ such that $f(x) \leq cg(x)$ $\forall x > x_0$. Similarly, $f(x) = \Omega(g(x))$ if $f(x) \geq cg(x)$. Finally, we say $f(x) = \Theta(g(x))$ if $f(x) = O(g(x))$ and $f(x) = \Omega(g(x))$ at the same time.

Note that $Y'_{i,j}$ denotes the total number of vulnerable nodes that can move into the contact region of the i-th infected node of generation j with radius $2\alpha+r$. This indicates that the mean of $Y'_{i,j}$ satisfies $\mu = \mathbb{E}(Y'_{i,j}) = \gamma\kappa\lambda\pi(2\alpha+r)^2$ by the thinning theorem [18], where $\gamma > 0$ is the probability that an infected node has no enough time to infect a vulnerable node when they meet each other (i.e., their contact time is smaller than the required infection time randomly distributed in $[\delta_1, \delta_2]$).

Construct a Galton-Watson process $\{Z'_i\}$ satisfying

$$Z'_{i+1} = \sum_{j=1}^{Z'_i} Y'_{i,j}. \tag{2}$$

It follows from (1) that $Z_i \leq Z'_i$ for $i > 0$. From the branching property, it holds for generations $i+1$ and i that $\mathbb{E}(Z'_{i+1}) = \mu\mathbb{E}(Z'_i)$, and the average total number of nodes $\sum_{i=1}^{\infty} Z'_i = 1/(1-\mu)$ when $\mu < 1$. Thus, if $\mu < 1$ (i.e., $\gamma\kappa\lambda\pi(2\alpha+r)^2 < 1$), the average botnet size can be written as

$$\mathbb{E}|\mathcal{S}(t)| \leq \sum_{i=1}^{\infty} \mathbb{E}(Z'_i) = 1/(1-\mu) = \Theta(1), \tag{3}$$

which completes the $\mathbb{E}|\mathcal{S}(t)| = \Theta(1)$ part after we rewrite the condition $\gamma\kappa\lambda\pi(2\alpha+r)^2 < 1$ as $\kappa\lambda(2\alpha+r)^2 = O(1)$.

Next, we move on to the $\mathbb{E}|\mathcal{S}(t)| = \Theta(t^2)$ part. First, it follows from Lemma 1 in Appendix A that the average size of the botnet satisfies

$$\mathbb{E}|\mathcal{S}(t)| = \Omega(t^2) \tag{4}$$

for $\kappa\lambda(2\alpha+r)^2 = \Omega(1)$.

Thus, it suffices to show that $\mathbb{E}|\mathcal{S}(t)|$ is upper bounded by a quadratic function of t at the same time, i.e., $\mathbb{E}|\mathcal{S}(t)| = O(t^2)$. Note that it takes at least a time period δ_1 to propagate the malware from one node to the other. At time t, the farthest distance the malware can propagate is $(2\alpha+r)t/\delta_1$. In this range, the average number of vulnerable nodes is $\kappa\lambda((2\alpha+r)t/\delta_1)^2$, showing that $\mathbb{E}|\mathcal{S}(t)| = O(t^2)$. Combining this upper bound with the lower bound in (4), we obtain that $\mathbb{E}|\mathcal{S}(t)| = \Theta(t^2)$ when $\kappa\lambda(2\alpha+r)^2 = \Omega(1)$. □

Remark 1: Theorem 1 reveals interesting phenomena of mobile botnet propagation: a mobile botnet can either exhibit quadratic growth in its size over time, or have a limited size without persistent propagation. The key factor that determines which type of propagation the botnet has is the value of $\kappa\lambda(2\alpha+r)^2$. When the value is larger than some constant, the average total number of infected nodes keeps increasing quadratically; when the value is less than some constant, only a limited number of nodes can be infected in the network.

Given fixed network setups (i.e., node density λ and wireless transmission rage r), Theorem 1 indicates that sufficient mobility (i.e., mobility radius α is sufficiently large) always guarantees the quadratic growth in size for a mobile botnet. In this case, more and more nodes become infected as time goes, which has been observed in [9]–[13]. On the other hand, given fixed mobility models, sufficiently small vulnerability ratio κ ensures the limited propagation of a mobile botnet, which well explains the opposite results in [14]. We also note that there may exist a unique threshold of $\kappa\lambda(2\alpha+r)^2$ to trigger the $\Theta(t^2)$ propagation. However, its exact value could be mathematically intractable to find.

With Theorem 1, the results on the evolution speed of a mobile botnet are presented in the following.

Corollary 1 (Botnet evolution speed): Given the conditions in Theorem 1, it holds for the evolution speed of a mobile botnet $V(t)$ that $V(t) = \Theta(1/t)$ or $V(t) = \Theta(t)$.

Proof: According to Definition 2, we obtain the evolution speed $V(t) = \mathbb{E}|\mathcal{S}(t)|/t$. Then, the results of $V(t) = \Theta(1/t)$ or $V(t) = \Theta(t)$ follow immediately from Theorem 1. □

Remark 2: It is well known that the malware propagation speed on the Internet increases exponentially over time. Our results quantitatively show that mobile malware via proximity infection propagates with at most linearly increasing speed, which is significantly less than its counterpart on the Internet.

B. Stochastic Bound

According to Theorem 1, we know that the average size of a mobile botnet with $\Theta(1)$ propagation is always bounded above even if the time goes to infinity. In this case, we are also interested in what the distribution of its eventual size is, which is given in the following.

Theorem 2: The tail distribution of the eventual size of a botnet $\mathbb{P}(|\mathcal{S}(\infty)| > L)$ decays at least exponentially fast when $\kappa\lambda(2\alpha+r)^2 = O(1)$.

Proof: Recall that we have already constructed a process in (2) that satisfies

$$\mathbb{P}(|\mathcal{S}(\infty)| > L) \quad \leq \quad \mathbb{P}(\sum_{i=1}^{\infty} Z'_i > L). \tag{5}$$

Then, it suffices to show that the distribution of $\sum_{i=1}^{\infty} Z'_i$ decays exponentially fast.

First, according to the total progeny theorem (Proposition 3.4 in [17]), we obtain

$$\mathbb{P}(\sum_{i=1}^{\infty} Z'_i = l) = \mathbb{P}(\sum_{i=1}^{l} Y'_{l,i} = l-1)/l, \tag{6}$$

where $Y'_{l,i}$ is the number of vulnerable nodes whose home points fall into a circle with radius $2\alpha+r$. With the network size scaling, node distribution can be represented as a Poisson point process [19], [20]. Thus, it holds for $Y'_{l,i}$ that $\mathbb{P}(\sum_{i=1}^{l} Y'_{l,i} = l-1) = \frac{(l\mu)^{l-1}e^{-l\mu}}{(l-1)!}$. Inserting it into (6) yields

$$\mathbb{P}(\sum_{i=1}^{\infty} Z'_i = l) = (l\mu)^{l-1}e^{-l\mu}/l!. \tag{7}$$

Applying Stirling's formula ($l! = \Theta(1)l^{l+\frac{1}{2}}e^{-l}$) to (7), we obtain

$$\mathbb{P}(\sum_{i=1}^{\infty} Z'_i = l) = \Theta(1)l^{-\frac{3}{2}}\mu^{l-1}e^{-l(\mu-1)}. \tag{8}$$

Therefore, it follows from (8) that

$$\begin{aligned}\lim_{l\to\infty}\frac{\log\mathbb{P}(\sum_{i=1}^{\infty}Z'_i = l)}{l} &= \lim_{l\to\infty}\frac{\Theta(1)-\frac{3}{2}\log l+(l-1)\log\mu-(\mu-1)}{l}\\ &= \log\mu - \lim_{l\to\infty} 1.5\log l/l = \Theta(1),\end{aligned} \tag{9}$$

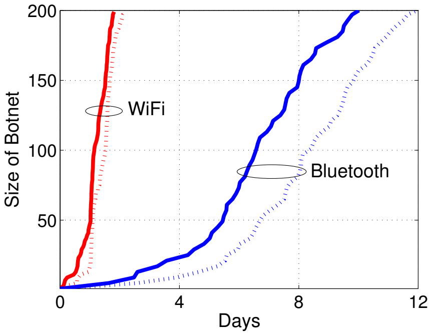


Fig. 4. Size of the botnet over time for two starting nodes: node "abmuyawm" – solid line, node "oafhynu" – dotted line.

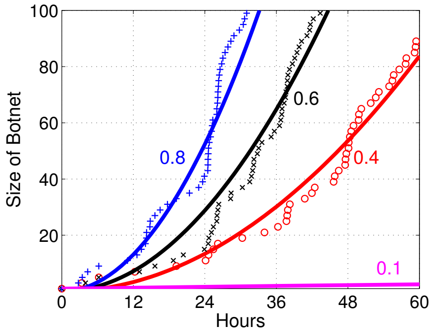


Fig. 5. Size of the botnet over time for vulnerable ratio κ=0.1, 0.4, 0.6, and 0.8.

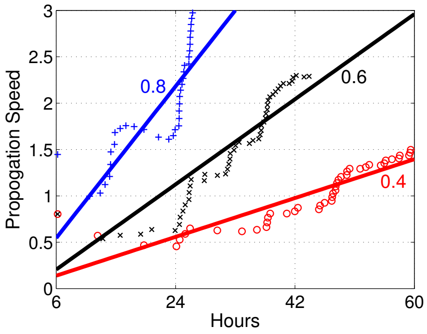


Fig. 6. Propagation speed over time for vulnerable ratio κ=0.4, 0.6, and 0.8.

showing that $\mathbb{P}(\sum_{i=1}^{\infty} Z_i')$ decays exponentially, which completes the proof. □

Remark 3: Theorem 2 shows that if $\kappa\lambda(2\alpha + r)^2$ is sufficiently small, the distribution of the size of a mobile botnet exhibits exponential decay. In this case, it is quite unlikely that a botnet can infect a large number of nodes in the network and cause severe impacts on mobile services.

C. Experimental Evaluation

In addition to theoretical analysis, we use experiments based on mobility traces to investigate mobile botnet propagation in realistic environments. In our experiments, we generate mobile nodes on a fixed-size map. Each node moves around according to realistic mobility traces. We randomly choose one node as the initially infected node that attempts to propagate malware to other vulnerable nodes.

In the first experiment, we use the EPFL data set [21], which contains mobility traces of taxi cabs in San Francisco. We generate 300 mobile nodes based on the 300 cab traces during a 12-day time period. The experiment starts at time 0 and we are interested in how many nodes are infected as time goes.

Fig. 4 shows the botnet size (i.e., the number of total infected nodes) versus elapsed time with different initially infected nodes (cabs "abmuyawm" in solid line and "oafhynu" in dotted line), different transmission ranges (100m WiFi and 10m bluetooth) and a constant vulnerability ratio κ=0.8 (i.e. 240 out of 300 nodes are vulnerable). It is noted from Fig. 4 that malware propagation with WiFi is substantially faster than that with bluetooth since WiFi has a much larger transmission range than bluetooth. Moreover, we observe in Fig. 4 that the botnet size as a function of elapsed time exhibits approximately parabolic curves especially for the two bluetooth cases, meaning that the botnet size is on the same order of a quadratic function of time t, i.e., $\Theta(t^2)$.

In order to further evaluate the WiFi cases, we perform a set of experiments. Fig. 5 shows the botnet size versus elapsed time for distinct vulnerability ratios (κ=0.1, 0.4, 0.6, and 0.8). The initially infected node is set to be cab "abmuyawm" in the traces, and all nodes use WiFi to propagate malware. We use a quadratic function to curve-fit the experimental data in Fig. 5 and find that the data shows the good trend of quadratic increase (even for the κ=0.1 case with sufficient time, which is not depicted in Fig. 5 due to the X-axis limit). In addition, Fig. 6 depicts the evolution speed as a function of time with vulnerability ratio κ=0.4, 0.6, and 0.8. It is observed from Fig. 6 that the evolution speed shows the trend of linear increase (not strictly linear, but in the order sense) for different vulnerability ratios.

It is worth mentioning that during our experiments, we find that malware can always infect all vulnerable nodes eventually. The reason is that the mobility traces in the EPFL data set are based on taxi cabs, which move around *sufficiently* on the map of San Francisco. In other words, the mobility radius α is large enough so that mobility has already triggered the $\Theta(t^2)$ propagation in Theorem 1.

In order to show how malware can propagate *without sufficient* mobility, we use the UDelModels [22] to generate mobility traces. UDelModels is a tool that can generate realistic human mobility for downtown metropolitan areas with configurable parameters. The map used in our experiments is a 2km×2km map in downtown Chicago as shown in Fig. 7. Detailed setups are shown in Table I.

TABLE I
UDELMODELS-BASED EXPERIMENT SETUP.

Number of walking nodes:	2000
Moving speed	[1, 4]
Pause time distribution	Exponential
Wireless transmission	Bluetooth (10m)
Vulnerability ratio	60%
Running time	24 hours
Mobility radius[1]	10m, 100m, 500m, 1km

[1] Each node's mobility trace is generated based on a partial map with a given mobility radius in UDelModels.

Fig. 8 illustrates the botnet size as a function of the elapsed time with vulnerability ratio κ=60% and mobility radius α=10m, 100m, 500m, and 1km. We note from Fig. 8 that when the mobility radius α is 100m, 500m, or 1km, the botnet size also exhibits quadratic growth over time, similar to Fig. 5. However, when α=10m, the botnet size does not increase as

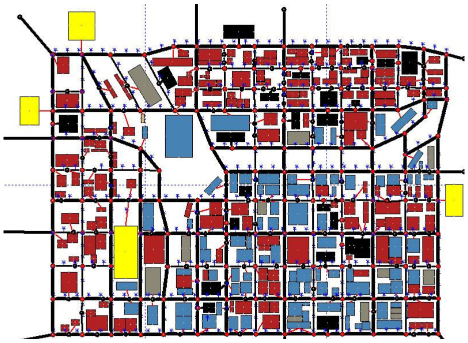


Fig. 7. 2km×2km map in downtown Chicago used in experiments (Courtesy of [22]).

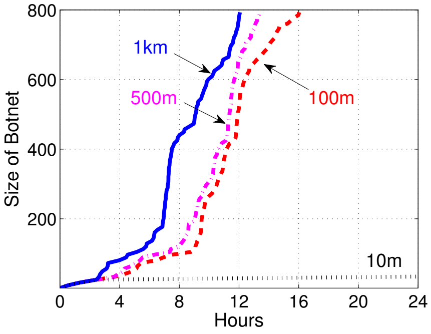


Fig. 8. Size of the botnet over time with mobility radius α=10, 100m, 500m, and 1km.

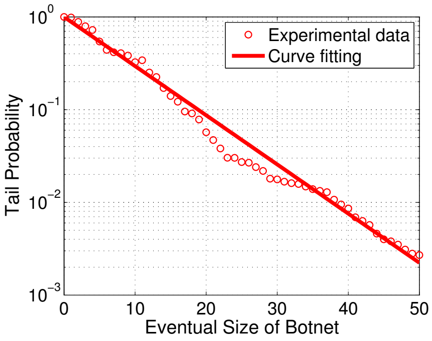


Fig. 9. The tail distribution of the eventual botnet size exhibits an exponential decay when mobility radius α = 10m.

time increases, indicating the malware propagation will stop eventually due to insufficient mobility.

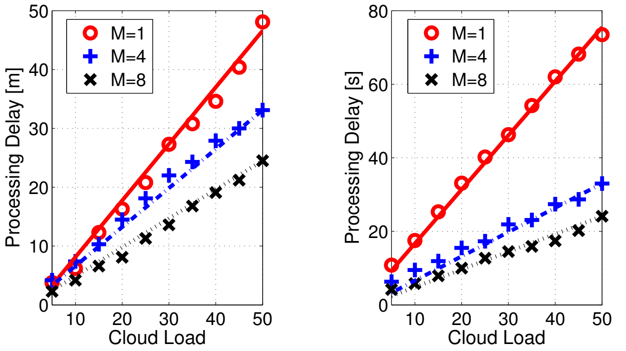
Fig. 9 shows the tail distribution of the eventual botnet size when α=10m on linear-log scales. We can observe from Fig. 9 that the tail distribution of the botnet size exhibits approximately a straight line. As any exponential function exhibits a straight line on linear-log scales, Fig. 9 demonstrates that *without sufficient* mobility, the botnet propagation can eventually stop with final size exponentially distributed, which validates the theoretical prediction in Theorem 2. Due to the exponential decay of the size of such a botnet, we can expect that it is not likely to infect a very large number of vulnerable nodes and make significant impacts.

IV. What Is the Impact of a Mobile Botnet?

By compromising mobile nodes, a mobile botnet can lead to either individual impacts (e.g., blocking the use of mobile devices [1]), or global impacts (e.g., denial-of-service attacks [2]). From the perspective of reliable network operations, the denial-of-service impact is much more severe than the individual impacts. Therefore, in the following, we focus on the denial-of-service impact of a mobile botnet. Our objective is to investigate what is the impact of a botnet, in which all compromised nodes flood service requests to a service provider to launch denial-of-service attacks. We first model how service requests from mobile nodes are processed, then propose the metric of last chipper time to measure the impact.

A. Modeling Mobile Service Processing

When mobile nodes move around in the network, they connect to a service provider via infrastructure nodes for service requesting and processing, as shown in Fig. 1. When a service request is delivered to a service provider, it will be immediately processed by the service processing center. Nowadays, many service processing centers feature a cloud computing paradigm [23], [24]: the data processing will be partitioned into different tasks, which are assigned to distinct computing units; then outputs of all tasks are combined. In this paper, we also consider such a cloud processing model as our mobile service application. In what follows, we will use the cloud and the service processing center interchangeably to denote the entity that processes service requests from mobile nodes.



(a) Hadoop

(b) Storm

Fig. 10. The processing delay versus constant cloud load L in Hadoop and Storm with different numbers of computers M used in the cloud.

At first glance, it appears that performance modeling for cloud processing is similar to a conventional waiting queue, in which one or few users can be served and the others are waiting in the queue. Nonetheless, cloud processing can be quite different in that the cloud supports concurrent processing (similar to the CPU sharing model) [23], [25]: when a service request arrives, the cloud directly allocates the shared computational resources (e.g., CPU time) for it instead of making the user waiting. Such a concurrent processing mechanism is widely used in current cloud processing frameworks [26], [27]. Therefore, a large amount of concurrent service requests can be processed in the cloud at the same time. The more the concurrent users (the heavier the cloud load), the longer the processing delay. To find out the relation between the cloud processing delay and the number of concurrent users, we adopt an experimental approach in a small-scale cloud based on the two popular Hadoop [26] and Storm [27] platforms.

We set up a small-scale cloud consisting of up to 8 computers with Intel Core i5 2.67GHz. The cloud is installed with Hadoop 1.0.2 and Storm 0.7.4. Fig. 10 shows the processing delay D_p as a function of constant cloud load L (which is the number of concurrent service requests being processed in the cloud at the same time) for different numbers of computers M. We can observe that for both Hadoop-based and Storm-based systems, there is approximately a linear relation between

D_p and L, i.e., $D_p \approx kL$, where the slope k is a decreasing function of M, showing that the more the computing resources in the cloud, the less the processing delay. Accordingly, we assume in this paper that $D_p = kL$ for any constant load L, and define $C = 1/k$ as the cloud capability, which can be considered as an indicator to represent the maximum number of service requests that can be finished in the cloud per second.

With parameter C, we can obtain $D_p = L/C$ for any constant load L. In practice, however, the cloud load L is a stochastic process due to network traffic dynamics, making the processing delay D_p a random variable. It has been shown that the cloud processing delay exhibits a heavy tail property [24]. Combining the constant load observation in Fig. 10 and the heavy tail property, we define the following stochastic cloud processing model.

Definition 3 (Service Processing): Let C be the cloud capability and $L(t)$ be the average cloud load at time t. The cloud processing delay D_p has a heavy tail, i.e., $\mathbb{P}(D_p > d) = \theta(d)d^{-\beta(t)}$ with mean $L(t)/C$, where $\beta(t)$ is some positive power-law parameter at time t, and $\theta(d)$ is a slowly-varying function satisfying $\lim_{d\to\infty} \theta(cd)/\theta(d) = 1$ for constant c.

B. Impact of A Botnet on Mobile Services

After we formulate the service processing model in Definition 3, we can investigate the impact of a mobile botnet on mobile services. We consider the scenario where all compromised nodes in a botnet flood service requests to the cloud. In particular, the botnet, by keeping infecting more nodes and flooding more requests, can gradually increase the cloud load and reduce service availability for legitimate services. This means that for any real-time mobile service, the probability that a legitimate service request is processed on time is gradually decreased. We are interested in how fast such a botnet attack process can take down the service. As a result, we define the metric of last chipper time as follows.

Definition 4 (Last Chipper Time): If a mobile botnet starts propagation at time 0, the last chipper time T_l is the last time that a required ratio ($\sigma < 1$) of mobile service requests can still be processed on time under the botnet attack, i.e.,

$$T_l = \sup\{t \geq 0 : \mathbb{P}(D_p < d) > \sigma\}. \tag{10}$$

With the metric of last chipper time in (10), we state our main result on the impact of a mobile botnet.

Theorem 3: If a mobile botnet can keep evolving in the network, the last chipper time T_l of a mobile service with requirement σ satisfies

$$T_l = O\left(1/\sqrt{B \log(1/(1-\sigma))}\right), \tag{11}$$

where B is the network bandwidth.

Proof: According to Definitions 3 and 4, we have

$$\begin{aligned} T_l &= \sup\{t \geq 0 : \theta(d)d^{-\beta(t)} > 1-\sigma\} \\ &\leq \sup\{t \geq 0 : \sup\{\theta(d)\}d^{-\beta(t)} > 1-\sigma\}, \end{aligned} \tag{12}$$

where $\sup\{\theta(d)\} = \Theta(1)$ (property of slowly-varying functions) and $\beta(t)$ is the power-law parameter at time t. From the power-law property, the average processing delay can be represented as $\Theta(1)(\beta(t)-1)/(\beta(t)-2)$. From Definition 3, the average load can be written as

$$L = C\Theta(1)(\beta(t)-1)/(\beta(t)-2). \tag{13}$$

On the other hand, the average load L is the sum of the average load of legitimate requests L_l and the average load induced by attacks L_a, i.e.,

$$L = L_l + L_a. \tag{14}$$

To calculate L_a, we first obtain from Theorem 1 that the average botnet size $\mathbb{E}|\mathcal{S}(t)|$ is at most $\Theta(t^2)$.

In addition, compromised nodes can flood service requests to the service processing center. How many service requests they can exactly send to the center depends on the network access schemes and network bandwidth B. No matter what access scheme the network has, the maximum bandwidth available for a node is always no greater than network bandwidth B, which indicates the rate of flooded requests at each compromised node is always upper bounded by $O(B)$.

Therefore, the average load induced by attacks L_a at the service processing center is at most

$$L_a = C\mathbb{E}(|\mathcal{S}(t)|O(B)) = Ct^2O(B). \tag{15}$$

Then, It follows from (13), (14), and (15) that

$$\beta(t) = 2 + 1/(t^2O(B)). \tag{16}$$

Inserting (16) into (12) completes the proof. □

Theorem 3 shows that if a botnet can keep evolving in the network, the last chipper time decreases at most on the order of $1/\sqrt{B}$. It has already been predicted in existing work [1] that the risk of mobile malware attack increases with the improved bandwidth in future wireless networks. Theorem 3 gives an interesting assessment on how such a risk is boosted. For example, LTE advanced is planned to improve the LTE uplink speed 10 times (from 50 Mbps to 500 Mbps). It follows from Theorem 3 that for the same mobile service, its last chipper time in LTE advanced will become around one third of the time in LTE ($1/\sqrt{10} \approx 1/3$). This means that in order to make some impact in LTE advanced, a botnet only needs to propagate one third of the time that it spends in LTE.

Remark 4: It is worthy of note that the decrease on the order of $1/\sqrt{B}$ of the last chipper time relies on the condition that all infected nodes attempt to saturate the network channel to launch attacks. If they attack at a constant rate that does not depend on B, the last chipper time should not be affected by B. Therefore, practical networks must always deploy attack detection and rate-limiting schemes to prevent infected nodes from flooding service requests at the saturated rate. However, we do believe that the decrease on the order of $1/\sqrt{B}$ represents the worst-case scenario that should be considered for any risk assessment of mobile botnets.

C. Experimental Evaluation

We also use experiments to measure the last chipper time. We first present the setups, then discuss the results.

1) System Setups: We set up a small-scale cloud that consists of 8 computers running over the Storm framework [27]. As shown in Fig. 11, the cloud is connected to a simulation server that simulates a wireless network environment.

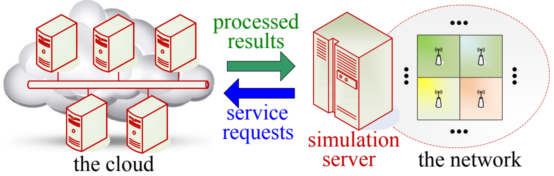


Fig. 11. A small-scale cloud is connected to a network simulation server.

Network Setup: We place 25 access points with equal space on the 2km×2km map shown in Fig. 7 to provide full wireless coverage with 802.11 DCF. The transmission range of access points and mobile nodes is 300 m. The network bandwidth varies from 1 to 54 Mbps. Mobile nodes move around based on UDelModels traces in Section III-C. They send service requests to their nearest access points. These service requests are delivered from the simulation server to the cloud for real-time processing. Then, the processed results in the cloud are sent back to mobile nodes in the simulation environment.

Service Setup: Mobile nodes use a location-aware service [28], [29]: they send their location/mobile sensing data via access points to the cloud, and obtain processed results from the cloud every 5 s. The size of service requests is 800 bytes, the size of processed results is 1200 bytes, and the processing delay requirement for each request is 2 s at the cloud.

Botnet Setup: The vulnerability ratio $\kappa = 60\%$, We randomly choose one node in the network as the initially infected node that propagates malware to others at time 0. To launch denial-of-service attacks, all infected nodes attempt to saturate the network channel by keep sending service requests to the cloud.

2) Experimental Results and Discussions: Fig. 12 shows the last chipper time as a function of network bandwidth B for service requirement σ= 70%, 80%, 90%, and 95%. The mobility radius of each node is 100m. We can observe from Fig. 12 that the last chipper time does decrease as B increases. For example, for requirement σ=95%, when B goes from 10MHz to 40MHz (4 times), the last chipper time decreases from 14.6 hours to 7.5 hours (almost halved). This can be well predicated in Theorem 3: the last chipper time T_l can be written as $O(1/\sqrt{B})$, and if B increases 4 times, T_l will be reduced to one half of the original value.

Fig. 13 illustrates the last chipper time as a function of network bandwidth B for mobility radius α=10m, 500m, and 1km. The service requirement is set to be σ=90%. First, we see from Fig. 13 that regardless of different mobility radii, the last chipper time always decreases as network bandwidth B increases. Second, Fig. 13 shows that more node movement does help the propagation of malware infection, and the last chipper time decreases accordingly with α becoming larger.

We conclude from Figs. 12 and 13 that the last chipper time is $O(1/\sqrt{B})$, as predicted in Theorem 3, and the more the mobility radius, the smaller the last chipper time.

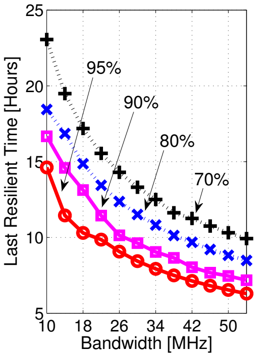


Fig. 12. Last chipper time with different service requirements.

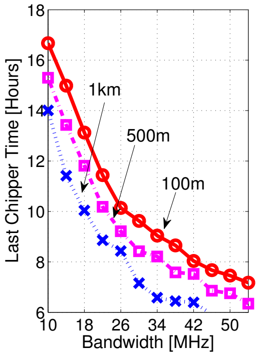


Fig. 13. Last chipper time with different mobility radii.

V. Conclusions

In this paper, we investigated how mobile botnets evolve via proximity infection and their impacts. We found that the size of a mobile botnet can either increase quadratically over time or be exponentially distributed with finite mean. In addition, we also defined the metric of last chipper time to measure the last time that a mobile service is still feasible under botnet attacks. Our findings in this paper not only provide a theoretical foundation to explain discrepant experimental results of mobile malware propagation in the literature, but also offer quantitative risk assessment on potential denial-of-service impacts of botnet attacks in mobile networks.

References

[1] A. P. Felt, M. Finifter, E. Chin, S. Hanna, and D. Wagner, “A survey of mobile malware in the wild,” in *Proc. of ACM workshop on Security and Privacy in Smartphones and Mobile Devices*, 2011.

[2] P. Traynor, M. Lin, M. Ongtang, V. Rao, T. Jaeger, P. McDaniel, and T. L. Porta, “On cellular botnets: Measuring the impact of malicious devices on a cellular network core,” in *Proc. of ACM Conference on Computer and Communications Security (CCS)*, 2009.

[3] C. Mulliner and J. P. Seifert, “Rise of the iBots: Owning a telco network,” in *Proc. of International Conference on Malicious and Unwanted Software*, Oct. 2010.

[4] Symantec Blog, “Android.Bmaster: A million-dollar mobile botnet,” Feb. 2012.

[5] Z. Zhu, G. Cao, S. Zhu, S. Ranjany, and A. Nucciy, “A social network based patching scheme for worm containment in cellular networks,” in *Proc. of IEEE INFOCOM*, 2009.

[6] F. Li, Y. Yang, and J. Wu, “CPMC: An efficient proximity malware coping scheme in smartphone-based mobile networks,” in *Proc. of IEEE INFOCOM*, 2010.

[7] J. Kim, S. Radhakrishnan, and S. K. Dhall, “Measurement and analysis of worm propagation on internet network topology,” in *Proc. of IEEE ICCCN*, 2004.

[8] P. V. Mieghem, J. Omic, and R. Kooij, “Virus spread in networks,” *IEEE/ACM Trans. Networking*, vol. 17, 2009.

[9] P. Wang, M. Gonzalez, C. Hidalgo, and A. Barabasi, “Understanding the spreading patterns of mobile phone viruses,” *Science*, 2009.

[10] G. Yan and S. Eidenbenz, “Modeling propagation dynamics of bluetooth worms (extended version),” *IEEE Trans. Mobile Computing*, Mar. 2009.

[11] J. Su, K. Chan, A. Miklas, K. Po, A. Akhvan, S. Saroiu, E. De Lara, and A. Goel, “A preliminary investigation of worm infections in a bluetooth environment,” in *Proc. of ACM WORM*, 2006.

[12] L. Carettoni, C. Merloni, and S. Zanero, “Studying bluetooth malware propagation: The bluebag project,” *IEEE Security and Privacy*, 2007.

[13] G. Yan, H. D. Flores, L. Cuellar, N. Hengartner, S. Eidenbenz, and V. Vu, “Bluetooth worm propagation: mobility pattern matters!” in *Proc. of ACM AsiaCCS*, 2007.

[14] N. Husted and S. Myers, “Why mobile-to-mobile wireless malware won’t cause a storm,” in *Proc. of USENIX LEET*, 2011.

[15] M. Garetto, P. Giaccone, and E. Leonardi, “Capacity scaling in delay tolerant networks with heterogeneous mobile nodes,” in *Proc. of ACM MobiHoc*, 2007.

[16] L. Sun and W. Wang, “On latency distribution and scaling: From finite to large cognitive radio networks under general mobility,” in *Proc. of IEEE INFOCOM*, 2012.

[17] A. Lambert, *Some aspects of discrete branching processes*. PrePrint, 2010.

[18] R. Meester and R. Roy, *Continuum Percolation*. Cambridge Univ. Press, 1996.

[19] M. Penrose, *Random Geometric Graphs*. Oxford Univ. Press, 2003.

[20] L. Sun and W. Wang, “On distribution and limits of information dissemination latency and speed in mobile cognitive radio networks,” in *Proc. of IEEE INFOCOM*, 2011.

[21] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, “CRAWDAD data set epfl/mobility (v. 2009-02-24),” Feb. 2009.

[22] UDelModels, http://www.udelmodels.eecis.udel.edu/.

[23] E. Zohar, I. Cidon, and O. Mokryn, “The power of prediction: Cloud bandwidth and cost reduction,” in *Proc. of ACM SIGCOMM ’11*, 2011.

[24] J. Tan, X. Meng, and L. Zhang, “Performance analysis of coupling scheduler for MapReduce/Hadoop,” in *Proc. of INFOCOM ’12*, 2012.

[25] B.-G. Chun and P. Maniatis, “Dynamically partitioning applications between weak devices and clouds,” in *Proc. of ACM workshop on Mobile Cloud Computing & Services*, Jun. 2010.

[26] Hadoop, http://hadoop.apache.org/.

[27] Storm, http://storm-project.net/.

[28] K. Kumar and Y.-H. Lu, “Cloud computing for mobile users: Can offloading computation save energy?” *Computer*, 2010.

[29] P. Angin and B. K. Bhargava, “Real-time mobile-cloud computing for context-aware blind navigation,” *International Journal of Next-Generation Computing*, vol. 2, 2011.

[30] H. Cai and D. Y. Eun, “Crossing over the bounded domain: From exponential to power-law inter-meeting time in MANET,” in *Proc. of ACM Mobicom*, 2007.

APPENDIX

Lemma 1: For a mobile botnet evolving in the network, its average size $\mathbb{E}|\mathcal{S}(t)| = \Theta(t^2)$ for $\kappa\lambda(2\alpha + r)^2$ larger than some constant, i.e., $\kappa\lambda(2\alpha + r)^2 = \Omega(1)$.

Proof: First we discretize the entire network into hexagonal cells with radius $2\alpha + r$. In what follows, we introduce necessary definitions to facilitate our proof. We call a cell is infected if there is at least one infected node (e.g., say node A) in the cell, and call an infected cell is open if there are at least one node in a nearby cell that is vulnerable to infection and whose home point is within the reachable distance (i.e., $2\alpha+r$) to the home point of the vulnerable node (i.e., node A). If a cell is not open, it is called closed. For two open cells, we say they are directly connected if they are neighbors to each other, and indirectly connected if there exists a path between them, on which all cells are open. Fig. 14 illustrates how we discretize the entire network into open and closed cells.

Without loss of generosity, assume that the initially infected node is in cell 0 in Fig. 14. A necessary condition for $\mathbb{E}|\mathcal{S}(t)| = \Theta(t^2)$ is that there must be infinitely many open cells (directly or indirectly) connected to cell 0 in order for malware propagation to go on. For example, malware in cell 0 shown in Fig. 14 is propagated to six neighbor cells (1–6), called the first-generation cells, in which cells 2–6 are open and cell 1 is closed. Then, malware in open cells 2–6 can be propagated farther to their neighbor cells 8–18 (the second-generation), in which cells 7, 8, 15, and 17 are closed. The open cells in the second generation can repeat the same infection process to the third generation, and so on.

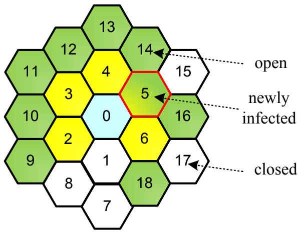


Fig. 14. Network discretization.

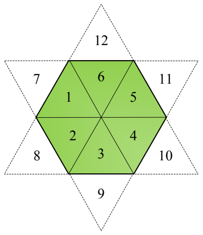


Fig. 15. Example of an open cell.

Denote by ρ the probability that a cell is infected by its neighbor and is also open (i.e., the newly infected cell can also propagate the malware to some other cells) and denote by S_1 the number of open neighbor cells in the first-generation cells. We can obtain S_1 is binomially distributed with parameters 6 and ρ, i.e., $S_1 \sim binomial(6, \rho)$. Let S_2 be the number of second-generation cells that are open. Then, conditioned on S_1, S_2 is also binomially distributed with parameters S_1 and ρ, i.e., $S_2 \sim binomial(c_1 S_1, \rho)$ for some constant c_1, where c_1 is called cell expansion ratio and $c_1 > 1$. Similarly, we have $S_{i+1} \sim binomial(c_i S_i, \rho)$, for all $i \geq 1$.

Accordingly, we have $\mathbb{E}(S_{i+1}|S_i) = c_i S_i \rho$, and the total number of connected open cells can be represented as $\sum_{i=1}^{\infty} S_i = \sum_{i=1}^{\infty} 6\rho^i \Pi_{j=1}^{i} c_j \geq 6\sum_{i=1}^{\infty}(\epsilon\rho)^i$, where $\epsilon = \min\{c_i\} > 1$. This shows that there will be infinitely many connected open cells if $\epsilon\rho \geq 1$, where ρ is the probability that a cell is open. For a particular cell, as shown in Fig. 15, it is surely open if there is at least one vulnerable node in each of areas 1–12. This implies that a cell is open with probability $\rho \geq (1-e^{-\sqrt{3}\kappa(2\alpha+r)^2/4})^{12}$. Therefore, there will be infinitely many connected open cells if $\epsilon(1-e^{-\sqrt{3}\kappa\lambda(2\alpha+r)^2/4})^{12} \geq 1$, which means that $\kappa\lambda(2\alpha + r)^2 = \Omega(1)$.

To further show how $|\mathcal{S}(t)|$ increases when $\kappa\lambda(2\alpha+r)^2 = \Omega(1)$, let $G(t)$ be the max number of cell generations that the infection process has reached at time t. Then, the total number of infected cells can be written as $\sum_{i=1}^{G(t)} S_i(t)$, where $S_i(t)$ is the number of infected cells at time t for generation i. Accordingly, we have

$$\mathbb{E}|\mathcal{S}(t)| \geq \mathbb{E}(\sum_{i=1}^{G(t)} S_i(t)). \tag{17}$$

The wait time between two cells depends on when two nodes in the cells meet each other and it has already been shown that in any bounded domain, the inter-meeting time of two nodes is exponential distributed [30]. Therefore, the wait time to propagate the malware from one cell to another is also exponentially distributed, based on which $G(t)$ can be shown as a continuous Markov process with intervals decaying exponentially fast. It follows from the elementary renewal theorem that $\lim_{t\to\infty} G(t)/t = \Theta(1)$ and therefore $G(t) = \Theta(t)$. Similarly, we can show that $S_i(t)/t = \Theta(1)$. Then, it follows from (17) that $\mathbb{E}|\mathcal{S}(t)| \geq t\mathbb{E}\left(\sum_{i=1}^{G(t)} S_i(t)/t\right) = t\mathbb{E}\left(\sum_{i=1}^{\Theta(t)} \Theta(1)\right) \geq t\Theta(t) = \Theta(t^2)$, which shows that $\mathbb{E}|\mathcal{S}(t)| = \Omega(t^2)$. □