On the Evolution and Impact of Mobile Botnets in Wireless Networks

Zhuo Lu, *Member, IEEE,* Wenye Wang, *Senior Member, IEEE,* and Cliff Wang, *Senior Member, IEEE*

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 mobile services, it can, at the same time, indeed escalate the risk of services being disrupted by mobile botnets.

Index Terms—Mobile botnet, malware, proximity propagation, wireless networks, denial-of-service, modeling and evaluation.

1 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

- *Zhuo Lu is with the Department of Computer Science, University of Memphis. Email: zhuo.lu@memphis.edu.*
- *Wenye Wang is with the Department of Electrical and Computer Engineering, North Carolina State University. Email: wwang@ncsu.edu.*
- *Cliff Wang is with Army Research Office, Research Triangle Park, NC. Email: cliff.wang@us.army.mil.*

An earlier version of the work was published in IEEE INFOCOM 2014. The research in this paper was supported by NSF CNS-142315 and CNS-1018447.

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 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 $1/\sqrt{B \log \frac{1}{1-\sigma}}$, 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 remainder of this paper is organized as follows. In Section 2, we introduce preliminaries and models. In Sections 3 and 4, we investigate how a mobile botnet evolves and what its impact is. In Section 5, we present related work. Finally, we conclude in Section 6.

2 Preliminaries and Models

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

2.1 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.

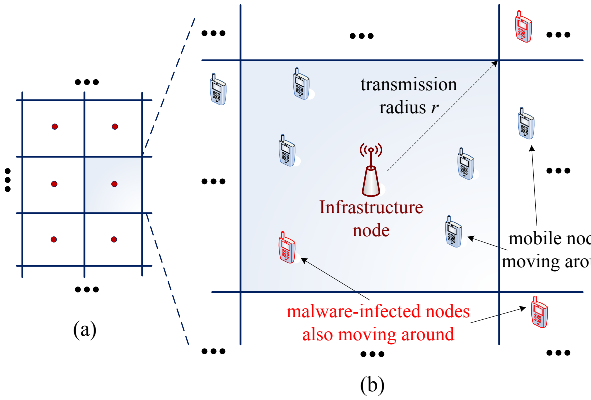

transmission radius r
Infrastructure node
mobile nodes moving around
malware-infected nodes also moving around
(a)
(b)

Fig. 1. Network architecture: infrastructure nodes and 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.

2.2 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].

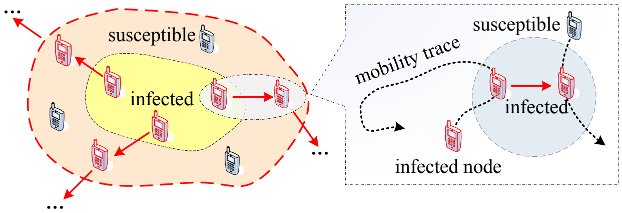


Fig. 2. Mobile botnet evolution over time via proximity infection.

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.

2.2.1 *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 the following generic mobility mode.

Definition 1: For a mobile node m_i, there exist a home point h_{m_i}, which is independently and uniformly distributed over region Ω. A mobility radius α for m_i is defined such that m_i moves around h_{m_i} with probability density function $\Psi(x)$, which is invariant in all directions and satisfies

$$\begin{cases} \Psi(x) > 0 & \text{when } \|x - h_{m_i}\| \le \alpha, \\ \Psi(x) = 0 & \text{otherwise.} \end{cases}$$

In addition, all mobile nodes move around their home points according to independent stationary processes.

It is worth mentioning that a home point in this paper is simply an anchor point for the mathematical model to specify the mobile range of a user. It does not mean that the user will be around this point more frequently. A mobile node m_i can frequently visit several places, such as workplace, school, and mall, as long as they are in the mobile range specified by the home point h_{m_i} and the radius α.

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.

2.3 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 2: 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 2, 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 3: 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 2 and 3, 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 $t > 0$?

3 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.

3.1 The Average Size and Evolution Speed

From Definition 3, we know that the evolution speed of a botnet $V(t)$ is based on the average size $\mathbb{E}|S(t)|$. Thus, it is essential to investigate the size of a mobile botnet at time t. We first prove the following lemma that will be used later.

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

Proof: As an infected node moves around in the network, it is possible to infect another moving node whose home point has a distance to the infected node no larger than $2\alpha + r$ since they always have a chance to move within each other's transmission range, as illustrated in Fig. 3. Thus, two nodes whose home points have a distance $(2a + r)$ always have a chance to meet each other in the network. It has already been shown in [17] that the inter-meeting time between two mobile nodes is exponentially distributed. Thus, our network with the generic mobility model is mathematically equivalent to a new network with node transmission range $r' = (2a+r)$, in which each link between two nodes that are located at their home points is an on-off process with on and off durations exponentially distributed. We differentiate the two networks as the original and new networks. Because of the mathematical equivalence, in what follows, we consider malware infection in the new network.

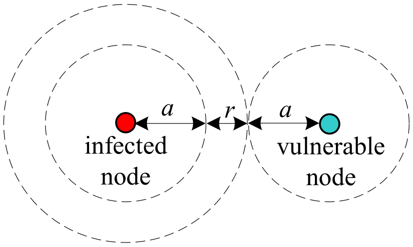


Fig. 3. The maximum possible range that a node can infect the other.

We discretize the new network into squares with side length cr' (where c is a constant) as shown on the left-hand side of Fig. 4. We say that a square is open if there exists at least a vulnerable node in the square and say it is closed otherwise. Then, the probability that a square is open can be written as

$$p = 1 - e^{-\kappa\lambda r'^2 c^2}. \tag{1}$$

We then map the new network into a discrete edge-percolation model on the right-hand side of Fig. 4. It has been shown that when p is sufficiently large (which is equivalent to say $\kappa\lambda r'^2$ sufficiently large or $\kappa\lambda(2a+r)^2 = \Omega(1)$), there are a large number of horizontal and vertical highways in the network [18]. A highway is a connected path between the two edges of the network, as shown on the left-hand side of Fig. 5. This means that as the number of total nodes n goes to infinity, each highway in our scenario is a path connected by infinity number of vulnerable nodes. In addition, the distance between two adjacent highways is $O(1)$. More details about the edge-percolation model and highway phenomena can be found in [18].

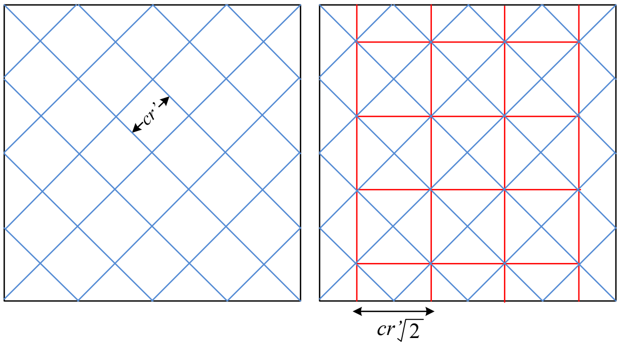


Fig. 4. Discretization of the new network.

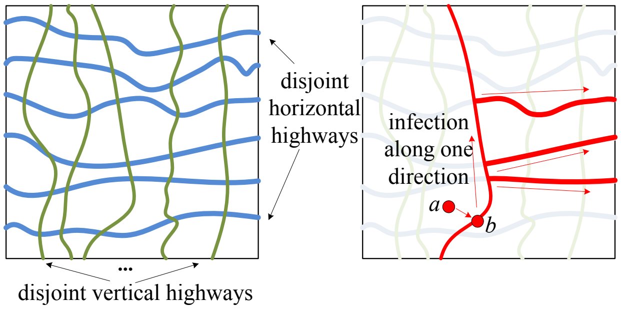

disjoint horizontal highways
...
disjoint vertical highways
infection along one direction
a
b

Fig. 5. Horizontal and vertical highways in the network.

We design a particular malware infection process to derive the lower bound of the size of botnet $|S(t)|$ under proximity infection. As shown on the right-hand side of Fig. 5, the initially infected node a first sends the malware to node b that is on a particular highway (called backbone). Then, node b continues to infect more vulnerable nodes along the backbone in one direction only (e.g., the north direction in Fig. 5). When it crosses the first horizontal highway (called branch) connected with infinite number of vulnerable nodes, the infection continues along both the backbone and the branch. As the time passes, we can expect that more branches start the infection process. We also assume that a branch only infects nodes in one direction (e.g., the east direction in Fig. 5) and cannot create more branches. Therefore, the infection process contains one backbone and an increasing number of branches over time.

Let $S'(t)$ be the set of the infected nodes on the backbone and branches at time t, and it is obvious that $|S'(t)| \leq |S(t)|$. In what follows, we are interested in finding out $|S'(t)|$.

Without loss of generality, assume that the first node on the backbone (e.g., node b in Fig. 5) is infected at time 0, and starts to infect other nodes since then. Recall that in our new network, each link is on and off with duration exponentially distributed. This indicates that the i-th infection takes time duration X_i exponentially distributed. Thus, the number of infected nodes on the backbone $N(t)$ is a Poisson process.

Denote by $N'(t)$ the number of branches up to time t, $N'(t)$ is also a Poisson process because the distance

[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.

between two adjacent highways is $O(1)$ and accordingly the time duration of an infection is exponentially distributed.

Denote by $X'_i(t)$ the number of infected nodes on branch i. Using a similar argument, we can show that $X'_i(t)$ is a delayed Poisson process satisfying

$$X'_i(t) = \begin{cases} 0 & t \le \sum_{j=1}^{i} Y_j \\ X_i(\tau) & \text{otherwise}, \end{cases} \tag{2}$$

where $X_i(\tau)$ is a Poisson process with $\tau = t - \sum_{j=1}^{i} Y_j \ge 0$, and Y_j is the time duration between the starting times of branches $i-1$ and i, and $\mathbb{E}(Y_j) = \Delta$.

Thus, $|S'(t)|$ is the sum of the number of infected nodes on the backbone $N(t)$ and the numbers of infected nodes on all branches from 1 to $N'(t)$; i.e.,

$$\begin{aligned} |S'(t)| &= N(t) + \sum_{i=1}^{N'(t)} X'_i(t) \\ &= N(t) + \sum_{i=1}^{N'(t)} X_i(t - \Sigma_{j=1}^{i} Y_j). \end{aligned} \tag{3}$$

Because $N'(t)$ is a Poisson process, we know from the strong law of large numbers for renewal processes that $\lim_{t\to\infty} N'(t)/t = 1/\Delta$. This indicates that $N'(t) \ge \lfloor \beta t \rfloor$ for any $\beta < 1/\Delta$ and t sufficiently large. Inserting it into (3) yields

$$|S'(t)| \ge N(t) + \sum_{i=1}^{\lfloor \beta t \rfloor} X_i(t - \Sigma_{j=1}^{i} Y_j). \tag{4}$$

Since $\sum_{i=1}^{\lfloor \beta t \rfloor} X_i(t - \Sigma_{j=1}^{i} Y_j)$ is an increasing sequence, we further obtain from the monotone convergence theorem that

$$\begin{aligned} \mathbb{E}|S'(t)| &\ge \mathbb{E}(N(t)) + \mathbb{E}\left(\sum_{i=1}^{\lfloor \beta t \rfloor} X_i(t - \Sigma_{j=1}^{i} Y_j) \right) \\ &= \mathbb{E}(N(t)) + \sum_{i=1}^{\lfloor \beta t \rfloor} \mathbb{E}(X_i(t - \Sigma_{j=1}^{i} Y_j)) \\ &= \sigma t + \sum_{i=1}^{\lfloor \beta t \rfloor} \delta(t - i\Delta) \\ &= \sigma t + \delta \lfloor \beta t \rfloor \left(t - \Delta \frac{\lfloor \beta t \rfloor}{2} \right) \\ &= \Theta(t^2). \end{aligned} \tag{5}$$

Note that (5) is obtained under the new network model. Since we have shown that the original network model is mathematically equivalent to the new network, we conclude that when $\kappa\lambda(2a + r)^2 = \Omega(1)$,

$$\mathbb{E}|S(t)| \ge \mathbb{E}(|S'(t)|) \ge \Theta(t^2); \tag{6}$$

i.e., $\mathbb{E}|S(t)| = \Omega(t^2)$, under the original network model. □

Given Lemma 1, we are ready to prove the following main results.

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

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

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

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

Part I: 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 [19] 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

$$Z_{i+1} = \sum_{j=1}^{Z_i} Y_{i,j}, \tag{7}$$

where $Y_{i,j}$ is the number of nodes infected directly by the j-th infected node of generation i.

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 $Y_{1,1} \le Y'_{1,1}$ at any time. Similarly, we have i.i.d. random variables $\{Y'_{i,j}\}$ that satisfy

$$Y_{i,j} \le Y'_{i,j} \quad \text{for any } i, j > 0. \tag{8}$$

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

$$\mu = \mathbb{E}(Y'_{i,j}) = \gamma\kappa\lambda\pi(2\alpha + r)^2 \tag{9}$$

by the thinning theorem [20], 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

$$Z'_{i+1} = \sum_{j=1}^{Z'_i} Y'_{i,j}. \tag{10}$$

It follows from (8) that $Z_i \le Z'_i$ 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 $\sum_{i=1}^{\infty} Z'_i = \frac{1}{1-\mu}$ 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

$$\mathbb{E}|\mathcal{S}(t)| \le \sum_{i=1}^{\infty} \mathbb{E}(Z'_i) = \frac{1}{1-\mu} = \Theta(1), \tag{11}$$

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

$$\kappa\lambda(2\alpha+r)^2 = O(1). \tag{12}$$

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

$$\mathbb{E}|\mathcal{S}(t)| = \Omega(t^2) \tag{13}$$

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 $\frac{(2\alpha+r)t}{\delta_1}$. In this range, the average number of vulnerable nodes is $\kappa\lambda\pi\left(\frac{(2\alpha+r)t}{\delta_1}\right)^2$, showing that

$$\mathbb{E}|\mathcal{S}(t)| = O(t^2). \tag{14}$$

Combining this upper bound (14) with the lower bound (13), we obtain that $\mathbb{E}|\mathcal{S}(t)| = \Theta(t^2)$ when

$$\kappa\lambda(2\alpha+r)^2 = \Omega(1). \tag{15}$$

□

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 3, 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. The intuitive reason in the difference is that malware can use TCP/UDP to reach almost any computer connected to the Internet (therefore resulting in a much faster propagation speed); however, under proximity infection, malware can only reach wireless neighbors because of limited wireless transmission range.

In addition, the following corollary follows immediately.

Corollary 2 (Parallel Infection): If there are a constant number of initially infected nodes that start a parallel infection process, the results in Theorem 1 still hold.

Proof: Suppose that there are n_0 initially infected nodes in the network. Recall that the set of all infected nodes is denoted as $\mathcal{S}(t)$ at time t.

Now consider the case that only the j-th ($j \in [1, n_0]$) node in all initially infected nodes starts the infection process. In this case, the set of infected nodes is denoted as $\mathcal{S}_j(t)$. Then, it is obvious that

$$|\mathcal{S}_1(t)| \le |\mathcal{S}(t)| \le \sum_{j=1}^{n_0} |\mathcal{S}_j(t)|.$$

This in turn yields

$$\mathbb{E}|\mathcal{S}_1(t)| \le \mathbb{E}|\mathcal{S}(t)| \le \sum_{j=1}^{n_0} \mathbb{E}|\mathcal{S}_j(t)|,$$

and thus

$$\mathbb{E}|\mathcal{S}_1(t)| \le \mathbb{E}|\mathcal{S}(t)| \le n_0\mathbb{E}|\mathcal{S}_j(t)| = \Theta(1)\mathbb{E}|\mathcal{S}_1(t)|,$$

which means that the number of infected nodes under parallel infection is on the same order of that under single infection process. This completes the proof. □

Remark 3: Corollary 2 shows that making multiple attackers propagate the malware at the same time cannot significantly improve the propagation speed by order of magnitude.

3.2 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 (10) that satisfies

$$\mathbb{P}(|\mathcal{S}(\infty)| > L) \quad \leq \quad \mathbb{P}\left(\sum_{i=1}^{\infty} Z_i' > L\right). \tag{16}$$

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 [19]), we obtain

$$\mathbb{P}\left(\sum_{i=1}^{\infty} Z_i' = l\right) = \frac{\mathbb{P}\left(\sum_{i=1}^{l} Y_{l,i}' = l - 1\right)}{l}, \tag{17}$$

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 [21], [22]. Thus, it holds for $Y_{l,i}'$ that

$$\mathbb{P}\left(\sum_{i=1}^{l} Y_{l,i}' = l - 1\right) = \frac{(l\mu)^{l-1} e^{-l\mu}}{(l-1)!}, \tag{18}$$

where μ is given in (9). Inserting (18) into (17) yields

$$\mathbb{P}\left(\sum_{i=1}^{\infty} Z_i' = l\right) = \frac{(l\mu)^{l-1} e^{-l\mu}}{l!}. \tag{19}$$

Applying Stirling's formula

$$l! = \Theta(1) l^{l+\frac{1}{2}} e^{-l}$$

to (19), we obtain

$$\mathbb{P}\left(\sum_{i=1}^{\infty} Z_i' = l\right) = \Theta(1) l^{-\frac{3}{2}} \mu^{l-1} e^{-l(\mu-1)}. \tag{20}$$

Therefore, it follows from (20) that

$$\begin{aligned} &\lim_{l\to\infty} \frac{\log \mathbb{P}(\sum_{i=1}^{\infty} Z_i' = l)}{l} \\ &= \lim_{l\to\infty} \frac{\Theta(1) - \frac{3}{2}\log l + (l-1)\log\mu - (\mu - 1)}{l} \\ &= \log\mu - \lim_{l\to\infty} \frac{1.5 \log l}{l} = \Theta(1), \end{aligned} \tag{21}$$

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

Remark 4: Theorem 2 shows that if $\kappa\lambda(2\alpha + r)^2$ is sufficiently small, the distribution of the size of a mobile botnet exhibits at least exponential decay; i.e., its tail distribution is bounded from above by an exponential distribution. 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.

3.3 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. If a node moves into the wireless transmission range of an infected node and at the same time it is vulnerable, it will become an infected node that starts to infect others.

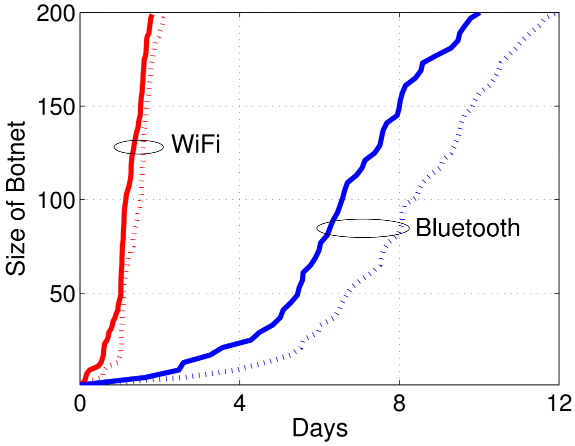


Fig. 6. Size of the botnet over time for two starting nodes: node "abmuyawm" – solid line, node "oafhynu" – dotted line.

In the first experiment, we use the EPFL data set [23], 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. 6 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. 6 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. 6 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. 7 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

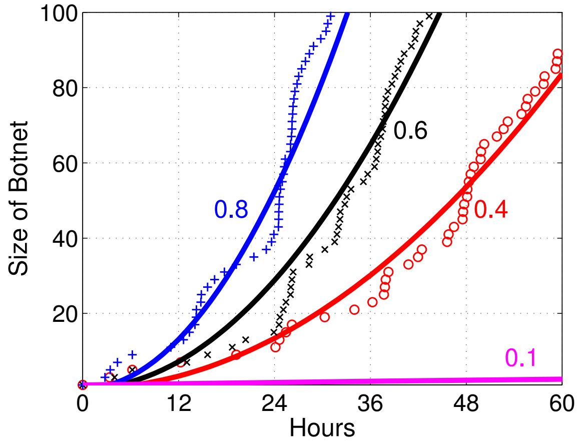


Fig. 7. Size of the botnet over time for vulnerable ratio κ=0.1, 0.4, 0.6, and 0.8.

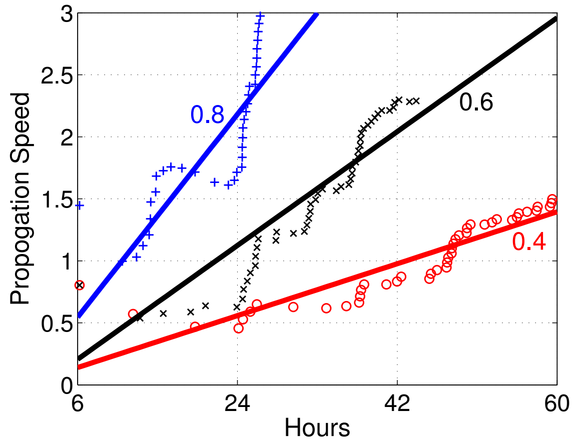


Fig. 8. Propagation speed over time for vulnerable ratio κ=0.4, 0.6, and 0.8.

curve-fit the experimental data in Fig. 7 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. 7 due to the X-axis limit). In addition, Fig. 8 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. 8 that the evolution speed shows the trend of linear increase (not strictly linear, but in the order sense) for different vulnerability ratios.

Fig. 9 shows the botnet size with different numbers of initially infected nodes. We can observe in Fig. 9 that more initially infected nodes lead to faster infection in the network: as the number of initially infected nodes goes from 2 to 4, the infection process exhibits approximately a sharper quadratic curve. This indicates that more initially infected nodes speeds up proximity infection by a constant order of magnitude.

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.

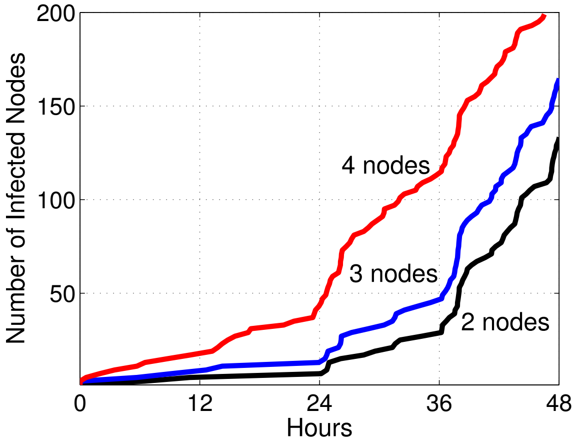


Fig. 9. Size of the botnet over time with different numbers of initially infected nodes.

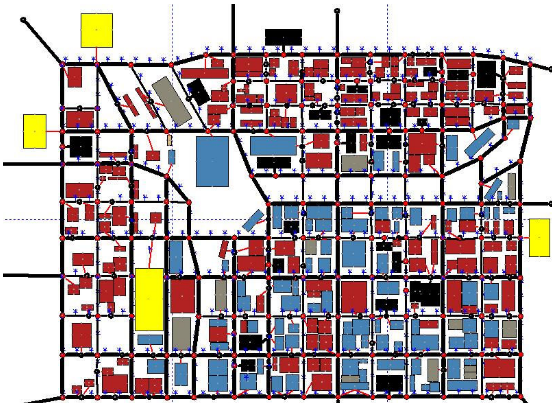


Fig. 10. 2km×2km map in downtown Chicago used in experiments (Courtesy of [24]).

In order to show how malware can propagate *without sufficient* mobility, we use the UDelModels [24] 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. 10. Detailed setups are shown in Table 1.

Fig. 11 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. 11 that when the mobility radius α is 100m, 500m, or 1km, the botnet size also exhibits quadratic growth over time, similar to Fig. 7. However, when α=10m, the botnet size does not increase as time increases, indicating the malware propagation will stop eventually due to insufficient mobility.

Fig. 12 shows the tail distribution of the eventual botnet size when α=10m on linear-log scales. We can observe from Fig. 12 that the tail distribution of the

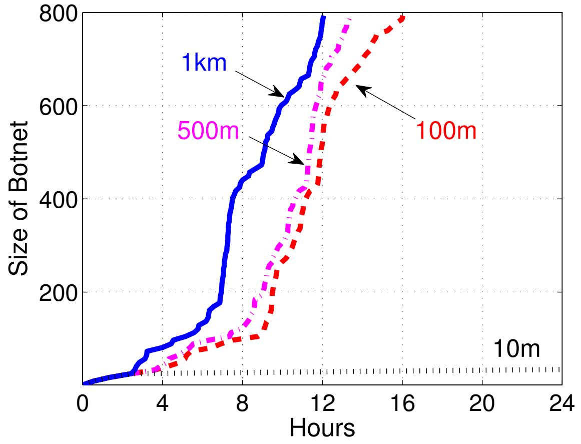


Fig. 11. Size of the botnet over time with mobility radius α=10, 100m, 500m, and 1km.

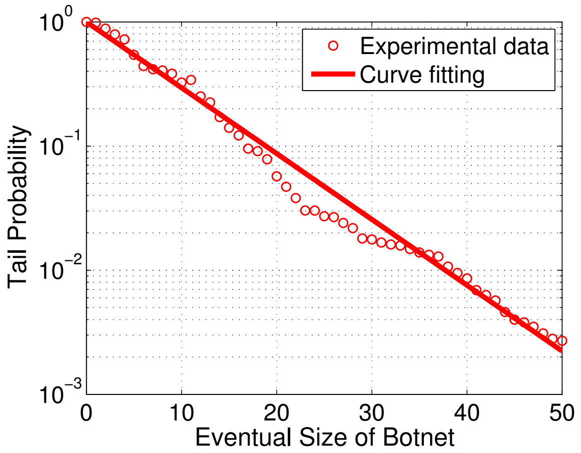


Fig. 12. The tail distribution of the eventual botnet size exhibits an exponential decay when mobility radius α = 10m.

botnet size exhibits approximately a straight line. As any exponential function exhibits a straight line on linear-log scales, Fig. 12 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.

4 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.

TABLE 1
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

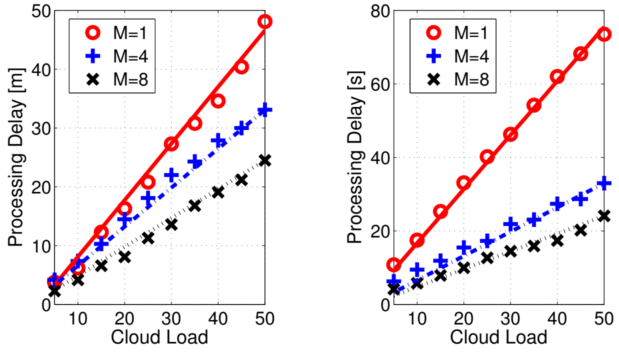
[1] Each node's mobility trace is generated based on a partial map with a given mobility radius in UDelModels.

4.1 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 [25], [26]: 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.

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) [25], [27]: 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 [28], [29]. 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 [28] and Storm [29] platforms.

- Hadoop [28] is an open-source cloud computing framework that allows for the processing of large



(a) Hadoop

(b) Storm

Fig. 13. The processing delay versus constant cloud load L in Hadoop and Storm with different numbers of computers M used in the cloud.

data sets across clusters of computers. Hadoop is now widely used in Google, Yahoo and Facebook.

- Storm [29] is another open-source distributed computation framework, aimed at offering real-time data processing capabilities. Storm has been deployed in Twitter and Groupon.

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. 13 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 [26]. Combining the constant load observation in Fig. 13 and the heavy tail property, we define the following stochastic cloud processing model.

Definition 4 (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 $\frac{L(t)}{C}$, where $\beta(t)$ is some positive power-law parameter at time t, and $\theta(d)$ is a slowly-varying function satisfying $\lim_{d\to\infty} \frac{\theta(cd)}{\theta(d)} = 1$ for constant c.

4.2 Impact of A Botnet on Mobile Services

After we formulate the service processing model in Definition 4, 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 5 (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.,

$$T_l = \sup\{t \geq 0 : \mathbb{P}(D_p < d) > \sigma\}. \tag{22}$$

With the metric of last chipper time in (22), 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

$$T_l = O\left(1/\sqrt{B \log \frac{1}{1-\sigma}}\right), \tag{23}$$

where B is the network bandwidth.

Proof: According to Definitions 4 and 5, we have

$$\begin{aligned} T_l &= \sup\{t \geq 0 : \theta(d)d^{-\beta(t)} > 1-\sigma\} \\ &\leq \sup\{t \geq 0 : \sup\{\theta(d)\}d^{-\beta(t)} > 1-\sigma\}, \end{aligned} \tag{24}$$

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 $\Theta(1)\frac{\beta(t)-1}{\beta(t)-2}$. From Definition 4, the average load can be written as

$$L = C\Theta(1)\frac{\beta(t)-1}{\beta(t)-2}. \tag{25}$$

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.,

$$L = L_l + L_a. \tag{26}$$

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. There are two major wireless access schemes: collision-free scheme (e.g., TDMA and FDMA), and collision-avoidance scheme (e.g., IEEE 802.11 DCF).

In the former, network bandwidth B is partitioned into orthogonal channels, each of which is used by only one node. In the latter, B is shared among all nodes, which use a random backoff algorithm (e.g., binary exponential backoff) to access the wireless channel. 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

$$L_a = C\mathbb{E}(|\mathcal{S}(t)|O(B)) = Ct^2O(B). \tag{27}$$

Then, It follows from (25), (26), and (27) that

$$\beta(t) = 2 + \frac{1}{t^2O(B)}. \tag{28}$$

Inserting (28) into (24) 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 5: 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.

4.3 Experimental Evaluation

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

4.3.1 System Setups

We set up a small-scale cloud that consists of 8 computers running over the Storm framework [29]. As shown in Fig. 14, the cloud is connected to a simulation server that simulates a wireless network environment.

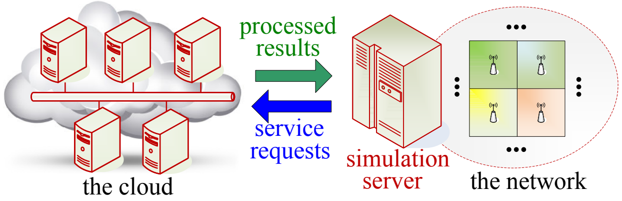


Fig. 14. A small-scale cloud is connected to a network simulation server.

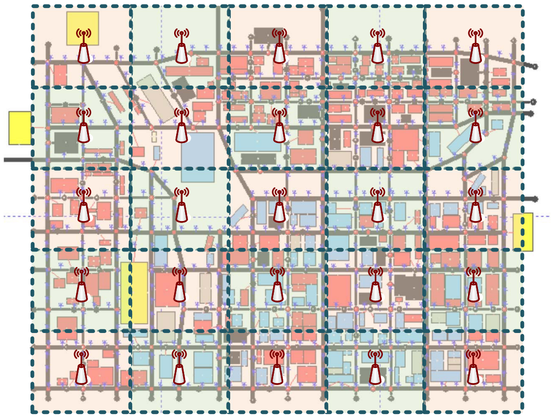


Fig. 15. 25 access points are placed with equal space on the map.

Network Setup: We place 25 access points with equal space on the 2km×2km map shown in Fig. 15 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 3.3. 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 [30], [31]: 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 κ = 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.

4.3.2 Experimental Results and Discussions

Fig. 16 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. 16 that the last chipper time does decrease as B increases. For example,

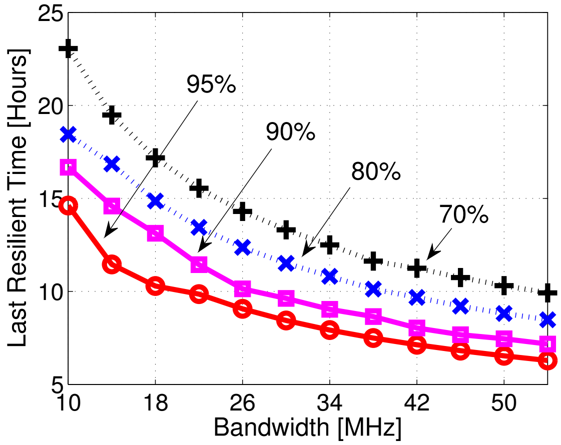


Fig. 16. Last chipper time versus network bandwidth with different service requirements.

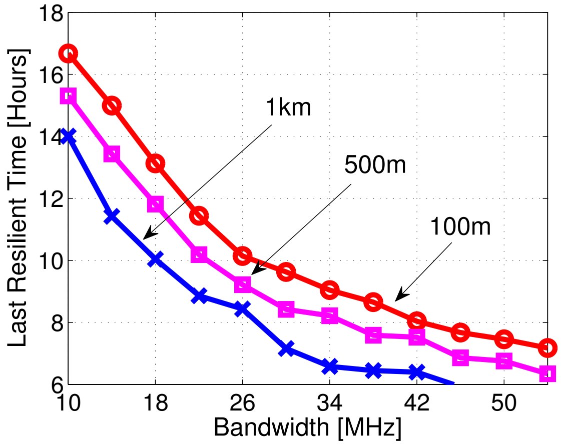


Fig. 17. Last chipper time versus network bandwidth with different mobility radii.

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. 17 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. 17 that regardless of different mobility radii, the last chipper time always decreases as network bandwidth B increases. Second, Fig. 17 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. 16 and 17 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.

5 Related Work

Many papers (e.g., [7], [8]) have studied the spreading of computer viruses/worms on the Internet using deterministic epidemic modeling. Recently, the epidemic modeling has been adopted in [9], [10] to study how mobile malware can spread via proximity infection. However, the pre-condition in deterministic epidemic modeling that a node can infect any other node with equal probability must always hold, which is not guaranteed in a practical mobile scenario with spatial heterogeneity. Experiments on mobile malware propagation have yielded opposite conclusions with distinct setups. For example, based on various mobility models and network setups, the works in [11]–[13], [32] showed that mobile malware via proximity infection can keep compromising vulnerable nodes as time goes, thereby leading to epidemics. In contrast, it is concluded in [14] that proximity infection only affects a limited number of nodes in realistic urban environments with relatively low node vulnerability ratios. Our findings of the dichotomy of mobile malware propagation have well explained the discrepancy in existing results. In addition, to the best of our knowledge, we are the first to show that proximity infection yields the fastest propagation rate of quadratic growth for mobile botnets.

The potential denial-of-service impact of mobile botnets has been considered in few recent papers. In [2], the authors showed that a mobile botnet with 11750 compromised phones can cause a reduction of throughput of more than 90% to area-code sized regions supported by current cellular infrastructures. Whether or how a botnet can achieve such a large size is not detailed in the paper. The work of [1] predicted that mobile malware that aims to launch denial-of-service attacks will eventually be found in practice with the increasingly improved bandwidth and power of mobile devices. The last chipper time used is our work quantitatively offers estimation on how mobile malware can benefit from the increased bandwidth in future wireless systems.

6 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] 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.

[18] M. Franceschetti, O. Dousse, D. N. Tse, and P. Thira, "Closing the gap in the capacity of wireless networks via percolation theory," *IEEE Trans. Information Theory*, vol. 53, no. 3, pp. 1009–1018, 2007.

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

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

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

[22] 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.

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

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

[25] E. Zohar, I. Cidon, and O. Mokryn, "The power of prediction: Cloud bandwidth and cost reduction," in *Proc. of ACM SIGCOMM '11*, 2011.

[26] J. Tan, X. Meng, and L. Zhang, "Performance analysis of coupling scheduler for MapReduce/Hadoop," in *Proc. of INFOCOM '12*, 2012.

[27] 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.

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

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

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

[31] 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.

[32] K. Channakeshava, D. C. D., K. Bisset, V. Kumar, and M. Marathe, "EpiNet: A simulation framework to study the spread of malware in wireless networks," in *Proc. of International Conference on Simulation Tools and Techniques*, 2009.

Zhuo Lu received his Ph.D. degree in the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh NC, in 2013. He was a Research Scientist at Intelligent Automation Inc, Rockville MD from 2013 to 2014. He has been an Assistant Processor with the Department of Computer Science, University of Memphis Since 2014. His research interests include network and mobile security, cyber-physical system and data analytics.

Wenye Wang received the M.S.E.E. degree and Ph.D. degree in computer engineering from the Georgia Institute of Technology, Atlanta, in 1999 and 2002, respectively. She is a Professor with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh NC. Her research interests include mobile and secure computing, modeling and analysis of wireless networks, network topology, and architecture design. Dr. Wang has been a Member of the Association for Computing Machinery (ACM) since 1998, and a Member of the Eta Kappa Nu and Gamma Beta Phi honorary societies since 2001. She is a recipient of the NSF CAREER Award 2006. She is the co-recipient of the 2006 IEEE GLOBECOM Best Student Paper Award - Communication Networks and the 2004 IEEE Conference on Computer Communications and Networks (ICCCN) Best Student Paper Award.

Cliff Wang graduated from North Carolina State University with a PhD degree in computer engineering in 1996. He currently serves as the computing sciences division chief for the Army Research Office. He is also appointed as an adjunct faculty member of computer science in the College of Engineering at North Carolina State University. Dr. Wang has been carrying out research in the area of computer vision, medical imaging, high speed networks, and most recently information security.