Modeling Propagation Dynamics of Bluetooth Worms

Guanhua Yan and Stephan Eidenbenz*
Information Sciences (CCS-3)
Los Alamos National Laboratory
{ghyan, eidenben}@lanl.gov

Abstract

The growing popularity of mobile devices in the last few years has made them attractive to virus and worm writers. One communication channel exploited by mobile malware is the Bluetooth interface. In this paper, we present a detailed analytical model that characterizes the propagation dynamics of Bluetooth worms. Our model captures not only the behavior of the Bluetooth protocol but also the impact of mobility patterns on the Bluetooth worm propagation. Validation experiments against a detailed discrete-event Bluetooth worm simulator reveal that our model predicts the propagation dynamics of Bluetooth worms with high accuracy.

1 Introduction

The last decade has witnessed a surge of wireless mobile devices such as cellular phones and PDAs, and the rapid proliferation of new services targeted at them. With the prevalence of these mobile devices in our lives, the reality that virus and worm writers are developing mobile malware propagating on them becomes increasingly haunting . Common to many existing mobile viruses and worms is that they leverage Bluetooth capabilities to propagate themselves. Bluetooth, a short-range radio technology aimed at connecting wireless devices with low power consumption, has a wide range of applications, such as wireless headsets, dial-up networking, and peer-to-peer file sharing. The market for Bluetooth devices has grown tremendously in recent years: world-wide, 272 million Bluetooth-enabled devices were shipped in 2005, twice as many as in 2004 [13].

Internet worms, which have been rampant for more than two decades, are nothing new to us. Bluetooth worms significantly depart from Internet worms in three major ways. First, the limited transmission range of a Bluetooth device leads to a proximity-based infection mechanism: a Bluetooth device controlled by a worm can only infect neighbors within its radio range. This differs from Internet worms that can scan the whole IP address space for susceptible victims. Second, the bandwidth available to Bluetooth devices is usually much narrower than those of Internet links. Finally, owing to the mobility of Bluetooth devices and their limited transmission range, the underlying network topology on which Bluetooth worms propagate is much more dynamic than that of the Internet.

Although there have been substantial efforts on modeling Internet worms, the fundamental differences between Bluetooth and Internet worms call for a new approach to modeling Bluetooth worm propagation. In this paper, we propose a detailed analytical model that characterizes the propagation dynamics of Bluetooth worms. The input parameters fed to this model consist of a few commonly used statistical metrics that describe the underlying mobility patterns, such as average node degree, average node meeting rate and the link duration distribution, and some control parameters used by the Bluetooth worms. The development of the model is based on detailed analysis of both the Bluetooth protocol and the impact of the mobility pattern on the worm behavior. Validation experiments against a detailed discrete-event Bluetooth worm simulator reveal that the model can accurately predict the propagation dynamics of Bluetooth worms, with relative errors smaller than 10% in most cases.

The remainder of this paper is structured as follows. Section 2 gives a simple behavior model of a typical Bluetooth worm. Section 3 discusses the modeling methodology used in this paper. Section 4 gives a model of the inquiry phase in a worm infection cycle, including how many neighbors can be discovered and how long it can take. Section 5 models the period that a worm contacts each neighbor it has discovered and attempts to infect each one of them. In Section 6, we describe a model that estimates the infection curve from the analysis of a single infection cycle. Section 7 presents experimental

[*] Los Alamos National Laboratory Publication No. LA-UR-06-3848

27th International Conference on Distributed Computing Systems (ICDCS'07)
0-7695-2837-3/07 $20.00 © 2007 IEEE

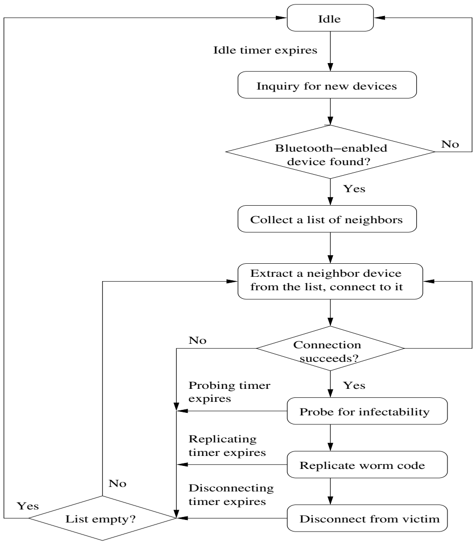


Figure 1. Infection Cycle of A Bluetooth Worm

Input	Explanation
λ_{ne}	Avg. meeting rate of neighbors
J_{in}	Avg. node degree
$F_L(\tau)$	CDF of link durations
N_{dev}	Number of devices
S_{dev}	Size of the area in which devices move
T^{to}_{inq}	Inquiry timeout value
N^{to}_{inq}	Max. number of inquiry responses expected
T^{to}_{conn}	Connection establishing timeout value
T^{to}_{prb}	Probing timeout value
T^{to}_{rep}	Worm replication timeout value
T^{to}_{disc}	Disconnection timeout value
T^{to}_{idle}	Duration of the idle phase
S_{prb}	Size of the probing packet
S_{worm}	Size of the worm code body

Table 1. Input parameters fed to the model

results of model validation. Section 8 introduces related work and Section 9 concludes the paper.

2 Behavior of Bluetooth Worms

The infection cycle of a typical Bluetooth worm can break down into two phases, illustrated in Fig. 1. When a Bluetooth worm is activated, it starts searching for Bluetooth-enabled devices in its vicinity. In this phase, the worm broadcasts Bluetooth inquiry packets and waits for responses. Because of the uncertainty about how many responses will be received, the worm has parameters as the expected number of responses, N^{to}_{inq}, and the maximum amount of time it wants to wait, T^{to}_{inq}. If N^{to}_{inq} responses are received before T^{to}_{inq} time units elapse, the worm stops the inquiry phase on the arrival of the N^{to}_{inq}-th response and then enters the next phase; otherwise, regardless of the number of responses it receives, the worm terminates the inquiry phase immediately after T^{to}_{inq} time units elapse.

Once the worm has collected a list of Bluetooth-enabled devices in its communication range, it iterates through the list, attempting the following steps with each neighbor device: establish a connection to it (*Step 1*), probe infection possibility (*Step 2*), replicate the worm code onto the victim device (*Step 3*), and disconnect from it (*Step 4*). Due to link instability in mobile networks, each of these steps may fail without notice from the other end. Hence, a timer is scheduled in each step, allowing the worm to detect possible connection failures. The maximum amount of times the worm is willing to wait in Steps 1, 2, 3, and 4 are denoted by T^{to}_{conn}, T^{to}_{prb}, T^{to}_{rep}, and T^{to}_{disc} respectively.

In Step 1, establishing a connection to a nearby device involves the *paging* process in the Bluetooth communication. We refer the reader to [4] for the details in this process. In Step 2, whether a device is infectable hinges on the vulnerability the worm exploits. We model this process by distinguishing three types of replies from a probed device: A REJECTED reply indicates that the probed device is *insusceptible*, an UNINFECTED reply indicates that the probed device is *susceptible and uninfected*, and an INFECTED reply indicates that the probed device is *susceptible but infected*. The last type of replies may not reflect the behavior of some Bluetooth worms, but this can be easily modified in our Bluetooth worm model. In Step 3, the time needed to replicate the worm code onto the victim depends on both the Bluetooth packet type and the size of the worm code.

Once all the devices on the neighbor list have been iterated, the worm remains inactive for T^{to}_{idle} time units. After the idle phase finishes, the worm enters another infection cycle and the process repeats.

3 Modeling Methodology

Out model characterizing the propagation dynamics of Bluetooth worms is deterministic and advances time in a discrete fashion. Let $i(t)$ be the average density of infected devices in the network being considered at time t. We assume that the worm starts propagating at time t_0 with the initial infection density as $i(t_0)$. Given the knowledge of the worm propagation state $i(t_k)$ at time t_k, the model determines the next time point t_{k+1} and the new worm propagation state $i(t_{k+1})$. Let $T_{cycle}(t)$ be the duration of an infection cycle that starts at time t. We then choose the time step size $t_{k+1} - t_k$ as $T_{cycle}(t_k)$. Moreover, between any two successive time points t_k and t_{k+1}, we use the following logistic equation to approximate the worm propagation curve:

$$\frac{\mathrm{d}i(t)}{\mathrm{d}t} = \beta(t) \cdot i(t) \cdot (\rho(t) - i(t)), \tag{1}$$

where $\rho(t)$ and $\beta(t)$ are the average device density and the pairwise infection rate at time t respectively.

To derive $T_{cycle}(t)$ and $\beta(t)$, we make the following assumptions: (1) all individual devices are homogeneously mixed; (2) the behavior of an infected device at time t is a deterministic function of the device density (i.e, $\rho(t)$), the worm propagation progress (i.e., $i(t)$) and the statistical properties of device mobilities; (3) all infected devices at time t have an identical infection cycle, except that they can be at different phases in the infection cycle. We note that the first assumption may not hold under some mobility patterns. For instance, the well-known random waypoint model leads to higher device mixing ratio at the center of the area than that in the bordering region. This problem can be solved by extending our approach to a spatial-temporal model, which divides a large area into multiple patches and updates the worm propagation status in each patch separately.

By assuming that individual devices are homogeneously mixed, we can abstract the underlying mobility model into a few statistical metrics. Fig. 1 gives a list of these metrics and their explanations. Note that all these metrics except the size of the area (i.e., S_{dev}) can be time variant. For clarity, we omit their time indices in the table. $\rho(t)$, the device density at time t, is actually $N_{dev}(t)/S_{dev}(t)$. Moreover, the statistical metrics that describe the mobility pattern form one part of the input parameters fed to our model, besides the Bluetooth worm parameters as discussed in Section 2. The Bluetooth worm parameters are also listed in Fig. 1.

In the following discussion, we first focus on the analysis of a single infection cycle starting time t, from which we derive the duration of the infection cycle (i.e., $T_{cycle}(t)$) and the number of new infections out of the infection cycle. We use $\alpha(t)$ to denote the latter. We then discuss how to derive $\beta(t)$ from $\alpha(t)$ and use Eq. (1) to estimate the worm propagation curve.

4 Modeling The Inquiry Phase

Consider an infective device starting its inquiry phase at time t. Without loss of generality, we number it as device 0. Let $T_{inq}(t)$ be the average duration of the inquiry phase at time t.

4.1 Number of Neighbors

We distinguish two different classes of neighbors. First, at the exact moment when device 0 starts its inquiry phase, some neighbors may be in its radio range. We call such neighbors *instantaneous neighbors* of device 0. Their average number at time t is actually $J_{in}(t)$ shown in Fig. 1, the average node degree at time t. As time goes by, some of these instantaneous neighbors may move out of its radio range and at the same time some new neighbors may enter its radio range. These new neighbors are called *contingent neighbors* of device 0, whose number we denote by $J_{co}(t)$. Apparently, $J_{co}(t)$ depends on how long the inquiry phase lasts. Here, we assume that the interarrival time between these new neighbors are exponentially distributed. Hence, the arrival process of new neighbors is a Poisson process. This assumption will also be used later in Section 4.4 to derive the mean duration of the inquiry phase. Let $\lambda_{ne}(t)$ be the arrival rate of new neighbors. Using the PASTA (Poisson Arrivals See The Average) property of the Poisson process, the number of neighbors, $H_{inq}(t)$, that device 0 meets in its inquiry phase starting at time t is

$$H_{inq}(t) = J_{in}(t) + J_{co}(t), \text{where } J_{co}(t) = \lambda_{ne}(t) \cdot T_{inq}(t).$$

4.2 Neighbor Discovery Probability

Not all the neighbors that the infective device meets in its inquiry phase can be discovered by it. As a neighbor receives an inquiry packet transmitted at the same frequency as the one that it is hopping on to receive inquiry packets, it backs off for a random period of time before responding to the inquiry device. A neighbor has to stay in the inquiry device's radio range long enough to get discovered. Let D be the time that an inquiry device needs to discover a neighbor in its radio range. The distribution function of D depends on how many devices that are performing inquiry operation simultaneously and analytically deriving the distribution function for D is infeasible [9]. We thus resort to simulation for an empirical solution to $\bar{D}(k)$, the average time needed to discover a neighbor given that k devices are performing inquiry simultaneously. We apply the linear least squares regression method and get the following equation: $\bar{D}(k) = 0.3322 \cdot k + 2.2325$. When k is 1, $\bar{D}(1)$ is 2.5647 seconds; it is very close to 2.292 seconds, the expected inquiry time derived from mathematical analysis [9]. We further assume that the discovery time $D(k)$ is uniformly distributed between 0 and $2\bar{D}(k)$. The observations from the simulation results confirm that it is a reasonable approximation. Thus, the probability density function of $D(k)$, denoted by $f_{D(k)}(\tau)$, is $\frac{1}{2\bar{D}(k)}$.

The number of devices that perform inquiry simultaneously increases as the network is populated with more infected devices. We use $m(t)$ to denote the average number of devices that perform inquiry simultaneously in device 0's radio range at time t. Recall that $T_{cycle}(t)$ denotes the total duration of an infection cycle starting at time t. The probability that an infected device is in the inquiry phase, denoted by $P^{inq}_{inf}(t)$, is thus $\frac{T_{inq}(t)}{T_{cycle}(t)}$. We thus have: $m(t) = i(t) \cdot \pi r^2_{ra} \cdot P^{inq}_{inf}(t)$, where r_{ra} is the radio range of a Bluetooth device.

We now calculate the discovery probability of a neighbor that device 0 meets in its inquiry phase starting at time t. We use random variable $L(t)$ to denote the duration of a link and $f_{L(t)}(\tau)$ to denote the probability density function of the link duration at time t. We can not simply let the discovery probability be $\mathbb{P}\{L(t) \geq D(m(t))\}$ because the inquiry phase initiated by device 0 may not start at exactly the same time as that when the link appears. We thus introduce notation $T_{gap}(t)$ to be $t_s^{link} - t_s^{inq}$, where t_s^{link} and t_s^{inq} are the starting times of the link and the inquiry phase respectively. Satisfying either of the following two propositions leads to a link between the two devices during the inquiry phase of device 0:

A_1: $T_{gap}(t) < 0$, and $T_{gap}(t) + L(t) > 0$.

A_2: $T_{gap}(t) \geq 0$, and $T_{gap}(t) < T_{inq}(t)$;

Proposition A_1 corresponds to the instantaneous neighbors met by device 0 in its inquiry phase and proposition A_2 corresponds to its contingent neighbors. Let P_{A_1} and P_{A_2} denote the probabilities that propositions A_1 and A_2 are true respectively. We then have:

$$\begin{aligned} P_{A_1} &= \mathbb{P}\{T_{gap}(t) + L(t) > 0 \wedge T_{gap}(t) < 0\} \quad (2) \\ P_{A_2} &= \mathbb{P}\{0 \leq T_{gap}(t) < T_{inq}(t)\} \quad (3) \end{aligned}$$

In order for a neighbor to be discovered by device 0, the link should overlap with the inquiry phase for at least $D(m(t))$. Satisfying the following two propositions enables device 0 to discover that neighbor:

B_1: $T_{gap}(t) < 0$, $0 \leq D(m(t)) \leq T_{inq}(t)$, and $T_{gap}(t) + L(t) \geq D(m(t))$.

B_2: $T_{gap}(t) \geq 0$, $0 \leq D(m(t)) \leq T_{inq}(t)$, $L(t) \geq D(m(t))$, and $T_{gap}(t) + D(m(t)) \leq T_{inq}(t)$;

Similarly, propositions B_1 and B_2 correspond to the instantaneous neighbors and the contingent neighbors that device 0 discovers in its inquiry phase respectively. Let P_{B_1} and P_{B_2} denote the probabilities that propositions B_1 and B_2 are true respectively. We have:

$$\begin{aligned} P_{B_1} &= \mathbb{P}\{T_{gap}(t) < 0 \wedge D(m(t)) \leq T_{inq}(t) \wedge \\ &\quad T_{gap}(t) + L(t) \geq D(m(t))\} \quad (4) \\ P_{B_2} &= \mathbb{P}\{0 \leq T_{gap}(t) \leq T_{inq}(t) - D(m(t)) \wedge \\ &\quad L(t) \geq D(m(t)) \wedge D(m(t)) \leq T_{inq}(t)\} \quad (5) \end{aligned}$$

Let $P_{dsc}^{in}(t)$ and $P_{dsc}^{co}(t)$ be the probability that an instantaneous neighbor and a contingent neighbor can be discovered by device 0 respectively. Clearly, we have: $P_{dsc}^{in}(t) = \frac{P_{B_1}}{P_{A_1}}$ and $P_{dsc}^{co}(t) = \frac{P_{B_2}}{P_{A_2}}$.

The computation of P_{A_1}, P_{A_2}, P_{B_1}, and P_{B_2} requires the knowledge of the distributions of both $T_{gap}(t)$ and $L(t)$. The latter is dictated by the mobility model that governs how devices move. Let Φ_l be the maximum link duration derived from the mobility model. We assume that $T_{gap}(t)$ is uniformly distributed between $-\Phi$ and Φ, where Φ is $\max(\Phi_l, T_{inq}^{to})$. Typically, Φ_l is much larger than T_{inq}^{to}. Hence, the probability density function of $T_{gap}(t)$, denoted by $f_{T_{gap}(t)}(\tau)$, is $\frac{1}{2\Phi}$. We thus have

$$\begin{aligned} P_{B_1} &= \frac{1}{2\bar{D}(m(t))} \times \frac{1}{2\Phi} \times \\ &\quad \int_0^{\Phi} \mathrm{d}s \int_0^{\min\{T_{inq}(t), 2\bar{D}(m(t))\}} \mathbb{P}\{L(t) \geq v + s\}\mathrm{d}v; \\ P_{B_2} &= \frac{1}{2\bar{D}(m(t))} \times \frac{1}{2\Phi} \times \\ &\quad \int_0^{T_{inq}(t)} (T_{inq}(t) - v)\mathbb{P}\{L(t) \geq v\}\mathrm{d}v. \quad (6) \end{aligned}$$

4.3 Number of Inquiry Responses

Infected neighbors are discovered by the inquiry device with a different probability from uninfected neighbors, because an infected neighbor in the inquiry or paging state can not respond to the inquiry. Let $P_{inf}^{av}(t)$ denote the probability that an infected neighbor is not in inquiry or paging mode. We also use $T_{page}(t)$ to denote the total time that device 0 spends on paging the neighbors it has discovered. We then have

$$P_{inf}^{av}(t) = 1 - \frac{T_{inq}(t) + T_{page}(t)}{T_{cycle}(t)}. \quad (7)$$

Even though a neighbor is not infected or is not in either inquiry or paging mode, it may be contacted by another infected device. According to the Bluetooth protocol state transition diagram [10], if a neighbor is being contacted by other infected devices and is thus not in the CONNECTION state or the STANDBY state, it can not respond to the inquiry from device 0. Let $P_{rsp}(t)$ denote the probability that an uninfected device or an infected device not in the inquiry or paging mode responds to the inquiry of device 0. It is difficult, though, to derive a precise analytical model for $P_{rsp}(t)$. For simplicity, we assume that a neighbor does not respond to the inquiry of device 0 if there exists another infected device that is paging it or has already established a connection to it. Consider any neighbor of device 0 that is either not infected or an infected device not in the inquiry or paging mode. Suppose it is device k. We use $N_{proc}(t)$ to denote the average number of infected devices in device k's radio range that are actively processing the neighbors that they have discovered. We also use $T_{proc}(t)$ to denote the total time that an infected device spends on processing the neighbors it has discovered. We have

$$N_{proc}(t) = \frac{T_{proc}(t)}{T_{cycle}(t)} \cdot i(t) \cdot \pi r_{ra}^2 \quad (8)$$

To derive an approximate formula for $P_{rsp}(t)$, we consider a static case in which no devices move. Then, the neighbors that these $N_{proc}(t)$ devices are contacting should be located within $2r_{ra}$ distance to device k and they are either uninfected or infected but idle. Let

$M_{proc}(t)$ be the average number of devices that these $N_{proc}(t)$ devices can possibly be processing. We have:

$$M_{proc}(t) = (\rho(t) - i(t) + \frac{T_{idle}^{to}}{T_{cycle}(t)} \cdot i(t)) \cdot \pi(2r_{ra})^2. \quad (9)$$

If device k is able to respond to the inquiry of device 0, it should not be contacted by any of the $N_{proc}(t)$ infected devices. For each of the $N_{proc}(t)$ infected devices, the probability that it does not contact device k is $\frac{M_{proc}(t)-1}{M_{proc}(t)}$. Hence, it immediately follows that

$$P_{rsp}(t) = (\frac{M_{proc}(t) - 1}{M_{proc}(t)})^{N_{proc}(t)}. \quad (10)$$

Now we calculate $R(t)$, the average number of neighbors that device 0 can discover in its inquiry phase. We treat instantaneous neighbors and contingent neighbors differently because their discovery probabilities are not the same. Let $N_{rsp}^{in}(t)$ and $N_{rsp}^{co}(t)$ denote the average number of instantaneous neighbors and contingent neighbors discovered by device 0 respectively. For brevity, we also introduce another notation $\hbar(t)$ as follows:

$$\hbar(t) = \frac{\rho(t) - i(t)}{\rho(t)} + \frac{i(t)}{\rho(t)} \cdot P_{inf}^{av}. \quad (11)$$

We then have:

$$\begin{aligned} N_{rsp}^{in}(t) &= J_{in}(t) \cdot P_{dsc}^{in}(t) \cdot \hbar(t) \cdot P_{rsp}(t), \quad (12) \\ N_{rsp}^{co}(t) &= J_{co}(t) \cdot P_{dsc}^{co}(t) \cdot \hbar(t) \cdot P_{rsp}(t). \quad (13) \end{aligned}$$

As the total number of neighbors that device 0 can discover should not exceed N_{inq}^{to}, the number of neighbors discovered in the inquiry phase, i.e., $R(t)$, can be established as follows:

$$R(t) = \min\{N_{inq}^{to}, N_{rsp}^{in}(t) + N_{rsp}^{co}(t)\}. \quad (14)$$

4.4 Duration of The Inquiry Phase

The duration of the inquiry phase is related to how many instantaneous neighbors device 0 can discover. If $N_{rsp}^{in}(t)$ is equal to or greater than N_{inq}^{to}, then device 0 does not need to wait for the appearance of contingent neighbors. Hence, the duration of the inquiry phase is simply $\bar{D}(m(t))$. We thus have the following:

$$T_{inq}(t) = \bar{D}(m(t)), \text{ if } N_{rsp}^{in}(t) \geq N_{inq}^{to}. \quad (15)$$

On the other hand, if $N_{rsp}^{in}(t)$ is smaller than N_{inq}^{to}, then device 0 has to discover more contingent neighbors to fill the gap between them. In this case, computing the duration of the inquiry phase requires the knowledge of how device 0 meets its neighbors. As in Eq. (2), we assume that links between device 0 and its neighbors appear according to Poisson process at arrival rate $\lambda_{ne}(t)$. Moreover, we also assume that all devices are homogeneously mixed so that among $H_{inq}(t)$ neighbors, the number of infected and uninfected devices are proportional to their fractions in the whole network. Hence, the original Poisson process can be split into two subprocesses which both are Poisson processes. The first one has only uninfected devices and their arrival rate, denoted by $\lambda_1(t)$, is

$$\lambda_1(t) = \frac{\rho(t) - i(t)}{\rho(t)} \cdot \lambda_{ne}(t) \cdot P_{rsp}(t). \quad (16)$$

The second subprocess consists of only infected devices and their arrival rate is $\frac{i(t)}{\rho(t)} \cdot \lambda_{ne}(t)$. Since the probability that an infected device can respond to the inquiry by device 0 is $P_{inf}^{av}(t) \cdot P_{rsp}(t)$, all such devices form another Poisson process and its arrival rate, denoted by $\lambda_2(t)$, is

$$\lambda_2(t) = \frac{i(t)}{\rho(t)} \cdot \lambda_{ne}(t) \cdot P_{inf}^{av} \cdot P_{rsp}(t). \quad (17)$$

As two Poisson processes merge into a new Poisson process, all the neighbors that can respond to the inquiry of device 0, including both infected and uninfected devices, form another Poisson process. Moreover, recall that the discovery probability of a contingent neighbor is $P_{dsc}^{co}(t)$. The process after random selection with probability $P_{dsc}^{co}(t)$ is still a Poisson process and its arrival rate, denoted by $\lambda(t)$, is $(\lambda_1(t) + \lambda_2(t)) \cdot P_{dsc}^{co}(t)$.

Let Z_n be the time needed for device 0 to collect n neighbors and $z_n(s)$ be its probability density function. We then have [6]

$$z_n(s) = \frac{\lambda(t)(\lambda(t)s)^{n-1}e^{-\lambda(t)s}}{(n-1)!}. \quad (18)$$

Since $N_{rsp}^{in}(t)$ instantaneous neighbors have already been discovered, device 0 only needs to find $N_{inq}^{to} - N_{rsp}^{in}(t)$ contingent neighbors, unless the inquiry timer expires before it does so. Therefore, if $N_{rsp}^{in}(t) < N_{inq}^{to}$, the average duration of the inquiry phase is

$$\begin{aligned} T_{inq}(t) = \quad & \tfrac{1}{2\bar{D}(m(t))} \int_0^{2\bar{D}(m(t))} ((1 - \int_0^{T_{inq}^{to}-v} z_\epsilon(t)\mathrm{d}t) \cdot T_{inq}^{to} \\ & + \int_0^{T_{inq}^{to}-v} (t+v) \cdot z_\epsilon(t)\mathrm{d}t)\mathrm{d}v, \quad (19) \end{aligned}$$

where ϵ is $N_{inq}^{to} - N_{rsp}^{in}(t)$. Note that in Eq. (19), we integrate from 0 to $T_{inq}^{to} - v$ on t. This is because the link between a contingent neighbor and device 0 must last at least $D(m(t))$ time units before it is discovered.

5 Modeling Neighbor Processing Phase

For ease of explanation, we number the infective device under consideration as device 0 and all the neighbors discovered from 1 to $R(t)$. In order for the worm to infect device k, where $1 \leq k \leq R(t)$, it has to wait until all neighbor devices numbered before neighbor k have been processed. We use $\tau_s^{(k)}(t)$ to denote the duration of the period that starts when device 0 starts its inquiry phase and ends when it starts to process neighbor k. Obviously, we always have the following:

$$\tau_s^{(1)}(t) = T_{inq}(t). \quad (20)$$

27th International Conference on Distributed Computing Systems (ICDCS'07)
0-7695-2837-3/07 $20.00 © 2007 IEEE

5.1 Establishing A Connection

We first model the probability that a neighbor discovered is pageable. Let $P_i(t)$ and $P_u(t)$ denote the fraction of infected devices and uninfected devices among all the neighbors discovered by device 0 respectively. According to the discussion in Section 4.3, infected devices in inquiry or paging mode do not respond to the inquiry of device 0. Hence, we have

$$P_i(t) \quad = \quad \frac{P^{av}_{inf}(t) \cdot i(t)}{P^{av}_{inf}(t) \cdot i(t) + \rho(t) - i(t)} \qquad (21)$$

$$P_u(t) \quad = \quad \frac{\rho(t) - i(t)}{P^{av}_{inf}(t) \cdot i(t) + \rho(t) - i(t)} \qquad (22)$$

For any of these infected neighbors, if it is in the inquiry or paging mode, device 0 can not successfully establish a connection to it. Furthermore, for any neighbor collected by device 0, if there is another infected device also connecting to it, device 0 may not be able to establish a connection to it successfully. Deriving the precise probability that a device is pageable is difficult. For simplicity, we assume that if there exists another infected device in contact with neighbor k, neighbor k is not pageable. Let $P^{pos}_{page}(t)$ denote the probability that a neighbor discovered by device 0 is pageable and $P^{neg}_{page}(t)$ denote the probability that a neighbor discovered by device 0 is not pageable. We then have

$$\begin{aligned} P^{pos}_{page}(t) &= (P_i(t) \cdot P^{av}_{inf}(t) + P_u(t)) \cdot P_{rsp}(t), \quad (23) \\ P^{neg}_{page}(t) &= P_i(t) \cdot (1 - P^{av}_{inf}(t) + P^{av}_{inf}(t) \cdot (1 - \\ &\quad P_{rsp}(t))) + P_u(t) \cdot (1 - P_{rsp}(t)). \quad (24) \end{aligned}$$

Let $P^{page}_{i}(t)$ and $P^{page}_{u}(t)$ denote the proportions of infected devices and uninfected devices among all pageable neighbors respectively. Obviously,

$$P^{page}_{i}(t) \quad = \quad \frac{P_i(t) \cdot P^{av}_{inf}(t)}{P_i(t) \cdot P^{av}_{inf}(t) + P_u(t)}, \qquad (25)$$

$$P^{page}_{u}(t) \quad = \quad \frac{P_u(t)}{P_i(t) \cdot P^{av}_{inf}(t) + P_u(t)}. \qquad (26)$$

Packet losses due to channel congestion (e.g., cochannel interference and adjacent channel interference) can increase the duration of the connection establishing process. In our model, we take this into consideration. Let $\bar{\tau}_{conn}$ be the average duration of successfully establishing a connection between two devices in a loss-free environment. We model the connection establishing process as a two-way handshake: the paging device sends out a packet with the paged device's access code requesting a connection and the paged device replies with a new packet carrying its own access code. An iteration of two-way handshake fails if either of the packets gets dropped. Let $P^{page}_{loss}(t)$ denote the paging packet loss rate at time t. In the following, we compute $T^{good}_{conn}(t)$, the average time needed for successfully establishing a connection provided that the paging packet loss rate is $P^{page}_{loss}(t)$. The loss probability of a packet is related to its size. As a paging response packet has the same size as a paging packet, the loss probability of paging response packets is also $P^{page}_{loss}(t)$. We refer readers to [16] for details of computing $P^{page}_{loss}(t)$. Let s be the maximum number of iterations allowed. Assuming both error-free transmission and no estimate of the slave's native clock by the paging device, if the paging procedure uses the R1 mode [10], the mean duration of the paging process is 1.28 seconds and its maximum duration is 2.56 seconds [5]. If an iteration of two-way handshake fails, the paging device wastes 2.56 seconds and has to wait for the next iteration. Let $\delta(t)$ be $(1-P^{page}_{loss}(t))^2$. Then, the average duration of a successful paging process is

$$T^{good}_{conn}(t) = 2.56((\frac{1}{\delta(t)} - (\frac{1}{\delta(t)} + s)(1-\delta(t))^s) - 1.28, \quad (27)$$

where $s = \lfloor \frac{T^{to}_{conn}}{2.56} \rfloor$.

A necessary condition for device 0 to establish a connection to neighbor k successfully is that the link between these two devices should be long enough such that the connection establishing process can be finished. Hence, the following proposition should be satisfied:

$$Q_1 : L(t) + T_{gap}(t) \geq \tau^{(k)}_s(t) + T^{good}_{conn}(t).$$

Before device 0 connects to neighbor k, the prior knowledge is that it must have already discovered this neighbor in its inquiry phase. Hence, we know that either propositions B_1 or B_2 must be true. Then, the probability that device 0 can connect to neighbor k successfully is

$$P^{succ}_{conn}(t, \tau^{(k)}_s(t)) \quad = \quad P^{pos}_{page}(t) \cdot \mathbb{P}\{Q_1 \mid B_1 \vee B_2\} \quad (28)$$

To simplify Eq. (28), we introduce proposition Q_0:

$$Q_0 : T_{gap}(t) \leq T_{inq}(t) - D(m(t)) \wedge D(m(t)) \leq T_{inq}(t).$$

After some logic computation, we have:

$$Q_1 \wedge (B_1 \vee B_2) = Q_0 \wedge Q_1 \qquad (29)$$

Hence, Eq. (28) can be rewritten as

$$P^{succ}_{conn}(t, \tau^{(k)}_s(t)) = P^{pos}_{page}(t) \cdot \frac{\mathbb{P}\{Q_0 \wedge Q_1\}}{\mathbb{P}\{B_1 \vee B_2\}} \qquad (30)$$

If device 0 fails to establish a connection to neighbor k, it has to wait until the connection establishing timer expires, which lasts T^{to}_{conn} time units. Let $P^{fail}_{conn}(t, \tau^{(k)}_s(t))$ denote the probability that device 0 fails to establish a connection to neighbor k provided that device 0 starts to process neighbor k at time $\tau^{(k)}_s(t)$. We thus have: $P^{fail}_{conn}(t, \tau^{(k)}_s(t)) = 1 - P^{succ}_{conn}(t, \tau^{(k)}_s(t))$.

In Eqs. (30), $\mathbb{P}\{B_1 \vee B_2\}$ equals $\mathbb{P}\{B_1\} + \mathbb{P}\{B_2\}$, and $\mathbb{P}\{Q_0 \wedge Q_1\}$ can be written as an expression of $\mathbb{P}\{T_{gap}(t) + D_{m(t)} \leq T_{inq}(t) \wedge D_{m(t)} \leq T_{inq}(t) \wedge L(t) + T_{gap}(t) \geq Y(t)\}$, where $Y(t) \geq T_{inq}(t)$. We refer readers to [16] for details of computing such an expression.

5.2 Probing for Infection Possibility

If device 0 succeeds in establishing a connection to neighbor k, it probes whether it is infected. It is obvious that the probing process can be prolonged because of channel congestion. Let $\eta(t)$ be the average data throughput at time t. We refer readers to [16] for details of computing $\eta(t)$. Recall that the total number of bytes in the probing packet and replying packet is S_{prb}. The average time needed for a successful probing process is $\frac{S_{prb}}{\eta(t)}$. Therefore, in order for the probing process finishes successfully, the following must hold:

$$Q_2 : L(t) + T_{gap}(t) \geq \tau_s^{(k)}(t) + T_{conn}^{good}(t) + \frac{S_{prb}}{\eta(t)}. \quad (31)$$

The prior knowledge for device 0 to probe the k-th neighbor is that device 0 successfully establishes a connection to it. Hence, $(B_1 \vee B_2) \wedge Q_1$ must be true. If the probing process succeeds, the duration of the probing phase is $\frac{S_{prb}}{\eta(t)}$; otherwise, the probing timer expires and the probing phase thus lasts T_{prb}^{to} time units. Furthermore, the probability that device 0 attempts to probe the k-th neighbor is $\mathbb{P}\{Q_1 | B_1 \vee B_2\}$. Let $P_{prb}^{succ}(t, \tau_s^{(k)}(t))$ denote the probability that device 0 successfully probes the infection state of device k. Obviously, $Q_1 \wedge Q_2 = Q_2$. Some logic computation leads to the following:

$$Q_2 \wedge (B_1 \vee B_2) = Q_0 \wedge Q_2. \quad (32)$$

By applying Bayes's rule, we have:

$$P_{prb}^{succ}(t, \tau_s^{(k)}(t)) \quad = \quad P_{page}^{pos}(t) \cdot \frac{\mathbb{P}\{Q_0 \wedge Q_2\}}{\mathbb{P}\{B_1 \vee B_2\}}. \quad (33)$$

Let $P_{prb}^{fail}(t, k)$ denote the probability that device 0 fails to probe the infection state of device k. Then,

$$P_{prb}^{fail}(t, \tau_s^{(k)}(t)) = P_{page}^{pos}(t) \cdot \frac{\mathbb{P}\{Q_0 \wedge Q_1\} - \mathbb{P}\{Q_0 \wedge Q_2\}}{\mathbb{P}\{B_1 \vee B_2\}}.$$

5.3 Replicating The Worm Code

After the probing step, if device 0 finds that neighbor k is not infected, it tries to replicate the worm code onto the victim. The prior knowledge for device 0 to replicate the worm code onto neighbor k includes: (1) device 0 establishes a connection to neighbor k successfully; (2) device 0 receives the reply to its probing packet from neighbor k; (3) neighbor k has not been infected. The probability that all these three conditions are satisfied, denoted by $P_{rep}^{prior}(t, \tau_s^{(k)}(t))$ is

$$P_{rep}^{prior}(t, \tau_s^{(k)}(t)) = P_{page}^{pos}(t) \cdot \frac{\mathbb{P}\{Q_0 \wedge Q_2\}}{\mathbb{P}\{B_1 \vee B_2\}} \cdot P_u^{page}(t). \quad (34)$$

Let $P_{inf}^{prb}(t, \tau_s^{(k)}(t))$ be the probability that device 0 finds that neighbor k has already been infected. Then,

$$P_{inf}^{prb}(t, \tau_s^{(k)}(t)) = P_{page}^{pos}(t) \cdot \frac{\mathbb{P}\{Q_0 \wedge Q_2\}}{\mathbb{P}\{B_1 \vee B_2\}} \cdot P_i^{page}(t). \quad (35)$$

The average time needed to replicate the code successfully is $S_{worm}/\eta(t)$. The following must be true if the worm code is successfully copied onto neighbor k:

$$Q_3 : L(t) + T_{gap}(t) \geq \tau_s^{(k)}(t) + T_{conn}^{good}(t) + \frac{S_{prb}}{\eta(t)} + \frac{S_{worm}}{\eta(t)}.$$

Similar to Eqs. (29) and (32), we have the following

$$Q_3 \wedge (B_1 \vee B_2) = Q_0 \wedge Q_3. \quad (36)$$

It is also possible that worm code replication fails because neighbor k moves out of the radio range or the packet loss rate is too high. Let $P_{rep}^{succ}(t, \tau_s^{(k)}(t))$ denote the probability that the worm code can be successfully replicated onto the victim. It is actually $P_{rep}^{prior}(t, \tau_s^{(k)}(t)) \cdot \mathbb{P}\{Q_3 \mid Q_2 \wedge (B_1 \vee B_2)\}$. Hence,

$$P_{rep}^{succ}(t, \tau_s^{(k)}(t)) = P_{page}^{pos}(t) \cdot P_u^{page}(t) \cdot \frac{\mathbb{P}\{Q_0 \wedge Q_3\}}{\mathbb{P}\{B_1 \vee B_2\}}. \quad (37)$$

If the worm code is not successfully replicated before the code replication timer expires, the worm code replication process takes T_{rep}^{to} time units. Let $P_{rep}^{fail}(t, \tau_s^{(k)}(t))$ be the probability that device 0 fails to copy the worm code onto the victim. Similarly, we have:

$$P_{rep}^{fail}(t, \tau_s^{(k)}(t)) =$$

$$P_{page}^{pos}(t) \cdot P_u^{page}(t) \cdot \frac{\mathbb{P}\{Q_0 \wedge Q_2\} - \mathbb{P}\{Q_0 \wedge Q_3\}}{\mathbb{P}\{B_1 \vee B_2\}}. \quad (38)$$

5.4 Total Time Spent On Processing All The Neighbors Discovered

The total time spent on processing neighbor k by device 0 depends on multiple conditions, including whether device 0 can successfully establish a connection to it, whether device 0 can successfully probe its infection state, whether neighbor k has already been infected, and whether device 0 can successfully copy the worm code onto it if it is found to be uninfected. Let $\vec{V}$ be a vector of 5 elements. We define function $\Omega(t, k, \tau_s^{(k)}(t), \vec{V})$ recursively as follows:

$$\Omega(t, k, \tau_s^{(k)}(t), \vec{V}) = \begin{cases} 0 & \text{if } k > R(t), \\ \omega & \text{if } k \leq R(t), \end{cases} \quad (39)$$

where $\omega = P_{conn}^{fail}(t, \tau_s^{(k)}(t)) \cdot$
$(\vec{V}[1] + \Omega(t, k+1, \tau_s^{(k)}(t) + T_{conn}^{to})) + P_{prb}^{fail}(t, \tau_s^{(k)}(t)) \cdot$
$(\vec{V}[2] + \Omega(t, k+1, \tau_s^{(k)}(t) + T_{conn}^{good}(t) + T_{prb}^{to})) +$
$P_{inf}^{prb}(t, \tau_s^{(k)}(t)) \cdot (\vec{V}[3] + \Omega(t, k+1, \tau_s^{(k)}(t) + T_{conn}^{good}(t)$
$+ \frac{S_{prb}}{\eta(t)})) + P_{rep}^{succ}(t, \tau_s^{(k)}(t)) \cdot (\vec{V}[4] + \Omega(t, k+1, \tau_s^{(k)}(t)$
$+ T_{conn}^{good}(t) + \frac{S_{prb}}{\eta(t)} + \frac{S_{worm}}{\eta(t)})) + P_{rep}^{fail}(t, \tau_s^{(k)}(t)) \cdot$
$(\vec{V}[5] + \Omega(t, k+1, \tau_s^{(k)}(t) + T_{conn}^{good}(t) + \frac{S_{prb}}{\eta(t)} + T_{rep}^{to}))$.

Then, $T_{proc}(t)$, the total time that device 0 spends on processing all the neighbors it has discovered, is

$$T_{proc}(t) = \Omega(t, 1, T_{inq}(t), \vec{V}_{proc}(t)). \quad (40)$$

where $\vec{V}_{proc}(t) = \langle T_{conn}^{to}, T_{conn}^{good}(t) + T_{prb}^{to}, T_{conn}^{good}(t) + \frac{S_{prb}}{\eta(t)}, T_{conn}^{good}(t) + \frac{S_{prb}}{\eta(t)} + \frac{S_{worm}}{\eta(t)}, T_{conn}^{good}(t) + \frac{S_{prb}}{\eta(t)} + T_{rep}^{to} \rangle$.

The total duration of an infection cycle starting at time t, $T_{cycle}(t)$, is given by:

$$T_{cycle}(t) = T_{inq}(t) + T_{proc}(t) + T_{idle}^{to}. \quad (41)$$

Once the worm code has been successfully replicated onto a victim, a new device has been infected. Recall that $\alpha(t)$ denotes the number of new infections out of an infection cycle starting at time t. Then, we have

$$\alpha(t) = \Omega(t, 1, T_{inq}(t), \langle 0, 0, 0, 1, 0 \rangle). \quad (42)$$

6 Modeling The Infection Curve

We model the Bluetooth worm infection curve using the logistic equation with variable infection rates. By assuming that individuals are homogeneously mixed, the model can be written as the differential equation given in Eq. (1). Now we estimate $\beta(t)$, the pairwise infection rate. Consider the $T_{cycle}(t)$ time units after time t. As the number of new infections out of each infection cycle is $\alpha(t)$, we can approximate $\beta(t)$ from the follows:

$$\begin{aligned} \frac{\mathrm{d}i(t)}{\mathrm{d}t} &= \beta(t) \cdot i(t) \cdot (\rho(t) - i(t)) = \frac{\alpha(t)}{T_{cycle}(t)} \cdot i(t) \\ \Longrightarrow \beta(t) &= \frac{\alpha(t)}{(\rho(t) - i(t)) \cdot T_{cycle}(t)}. \quad (43) \end{aligned}$$

The worm propagation curve can thus be modeled as:

$$i(t + \Delta t) = \frac{i(t) \cdot \rho(t)}{i(t) + (\rho(t) - i(t))e^{-\beta(t)\cdot\rho(t)\cdot\Delta t}} \quad (44)$$

Hence, after an infection cycle, the new density of infected devices is

$$i(t + T_{cycle}(t)) = \frac{i(t) \cdot \rho(t)}{i(t) + (\rho(t) - i(t))e^{\frac{-\alpha(t)\cdot\rho(t)}{\rho(t)-i(t)}}}. \quad (45)$$

Eq. (45) directly leads to an approach to compute the whole infection curve. Let t_0 be 0. We assume that at time t_0, there is only one single infected device. Hence, $i(t_0)$ is $\rho(t_0)/N_{dev}(t_0)$, where $N_{dev}(t)$ denotes the total number of devices at time t. Starting from t_0, we compute $T_{cycle}(t_k)$ and $\alpha(t_k)$, for $k \geq 0$ and then recursively update t_{k+1} and $i(t_{k+1})$ as follows:

$$\begin{aligned} t_{k+1} &= t_k + T_{cycle}(t_k), \quad (46) \\ i(t_{k+1}) &= \frac{i(t_k) \cdot \rho(t_k)}{i(t_k) + (\rho(t_k) - i(t_k))e^{\frac{-\alpha(t_k)\cdot\rho(t_k)}{\rho(t_k)-i(t_k)}}}. (47) \end{aligned}$$

There are a few problems with the above approach. First, at the early phase of the worm propagation, infected devices tend to cluster together because it takes some time for them to diffuse into each region of the area. A fundamental assumption underlying the logistic model is that infected and uninfected devices are homogeneously mixed. This problem manifests itself especially when a small number of devices are sparsely distributed in a large area. To fix this problem, we set the low bound on the density of infected devices as follows. Consider an infected device starting its inquiry at time t. We assume that it moves along a straight line during its inquiry phase. The area that its radio signal covers during its inquiry phase, denoted by $S_{inq}(t)$, is:

$$S_{inq}(t) = \pi r_{ra}^2 + 2 \cdot r_{ra} \cdot v(t) \cdot T_{inq}(t), \quad (48)$$

where $v(t)$ is the average device speed at time t. In the area covered by the infected device, there exists at least one infected device, which is itself. Let $i'(t)$ be $\max\{i(t), \frac{1}{S_{inq}(t)}\}$. When we compute $T_{cycle}(t_k)$ and $\alpha(t_k)$, we use $i'(t)$ to replace $i(t)$ and $i(t_{k+1})$ is updated as follows instead of using Eq. (47):

$$i(t_{k+1}) = \frac{i(t_k) \cdot \rho(t_k)}{i'(t_k) + (\rho(t_k) - i'(t_k))e^{\frac{-\alpha(t_k)\cdot\rho(t_k)}{\rho(t_k)-i'(t_k)}}}. \quad (49)$$

The second problem with Eqs. (46) and (47) is related with the assumption that new infections out of a single infection cycle is evenly distributed in the infection cycle. If a large fraction of devices have already been infected, this assumption is reasonable because the phase of an infected device in the infection cycle can be random. At the early stage of the infection, however, a newly infected device immediately enters the active scanning mode. Hence, Eq. (47) tends to underestimate the worm propagation speed at its early phase. Moreover, if $\beta(t_k)$ is larger, there are more new infections out of a single infection cycle and the estimation error is thus larger. We thus reduce $T_{cycle}(t_k)$ based on $\beta(t_k)$ in the first few iterations. The adjusted model on computing $T_{cycle}(t_k)$ is given as follows:

$$T_{cycle}(t_k) = \begin{cases} T_{inq}(t_k) + T_{proc}(t_k) + e^{-2\cdot\beta(t_k)} \cdot T_{idle}^{to}, & \text{if } k < 3, \\ T_{inq}(t_k) + T_{proc}(t_k) + T_{idle}^{to}, & \text{if } k \geq 3. \end{cases}$$

The third problem with the model is that it computes the worm growth rate based on the infection state at a time point t_k and then assumes this growth rate stays unaltered until an infection cycle starting at time t finishes at time t_{k+1}. For those infected devices that start their infection cycle after time t but before time t_{k+1}, α is overestimated. To overcome this flaw in the model, we further readjust the computation of $i(t_{k+1})$ as follows. First, we compute $\alpha(t_k)$ as before. Then, we estimate the density of infected devices at time t_x, where

$$t_x = t_k + T_{cycle}(t_k) - T_{proc}(t_k). \quad (50)$$

Actually, t_x is the latest time when an infected device finishes its inquiry phase so that it can finish processing all the neighbors discovered no later than $t_k + T_{cycle}(t_k)$. The estimated infection state at time t_x is

$$i(t_x) = \frac{i(t_k) \cdot \rho(t_k)}{i'(t_k) + (\rho(t_k) - i'(t_k))e^{\frac{-\alpha(t_k)\cdot\rho(t_k)}{\rho(t_k)-i'(t_k)} \cdot \frac{t_x - t_k}{T_{cycle}(t_k)}}}.$$

Based on the estimated infection state at time t_x, we can compute $\alpha(t_x)$. We define α' as follows

$$\alpha' = \frac{\rho(t_k) - i(t_k)}{\rho(t_k)} \cdot \alpha(t_k) + \frac{i(t_k)}{\rho(t_k)} \cdot \alpha(t_x). \quad (51)$$

The new equation to compute $i(t_{k+1})$ becomes

$$i(t_{k+1}) = \frac{i(t_k)\cdot\rho(t_k)}{i'(t_k) + (\rho(t_k) - i'(t_k))e^{\frac{-\alpha'\cdot\rho(t_k)}{\rho(t_k)-i'(t_k)}}}.$$

Obviously, at the early stage of the worm propagation, α' is close to $\alpha(t_k)$ and at the late stage of the worm propagation, α' is close to $\alpha(t_x)$. This can be explained as follows. In the logistic model, at the early stage of the worm propagation, the worm infection curve is convex and the average number of infected devices between time t_k and t_x is smaller than the $(i(t_k) + i(t_x))/2$; hence, choosing α' closer to $\alpha(t_k)$ achieves a better estimate. Similarly, at the late stage of the worm propagation, the infection curve turns concave in the logistic model and the average number of infected devices between time t_k and t_x is larger than the $(i(t_k)+i(t_x))/2$, which suggests that choosing α' closer to t_x leads to a better estimate.

7 Experiments

The system of equations proposed to characterize the Bluetooth worm propagation is comprehensive, covering both dynamics of the Bluetooth protocol behavior and statistical properties of mobility patterns. It is, however, not easy to solve these equations analytically. On the other hand, closed-form analytical solutions to the statistical properties of even simple mobility models (e.g., random walk model) can be hairy. Under such circumstance, we resort to numerical solutions. We have implemented our model as a system of equations in Octave [8] . We use function *fsolve*, a nonlinear equations solver provided in Octave, to derive the solutions numerically. We use the ns-2 network simulator [2], extended with the detailed UCBT Bluetooth simulation module [1].

To evaluate the accuracy of the model, we conduct experiments with different mobility and Bluetooth worm parameter settings. In all the experiments, a Bluetooth device moves in a square area according to the random walk model, in which it updates its direction and speed every 30 seconds. The speed of a device is uniformly chosen between 1 and 2 meters per second. The average device speed in our experiments is roughly the same of pedestrians. Fig. 2 presents N_{dev}, S_{dev}, λ_{ne}, J_{in} used in our experiments and Fig. 2 depicts the CDFs of link durations corresponding to the four mobility scenarios. We notice that the CDFs of link durations produced from the four mobility scenarios are very close to each other. This is because devices in each of them move according to the same parameterized random walk model.

We use two sets of Bluetooth worm parameters, denoted by W_1 and W_2 respectively. In setting W_1, we have: $T^{to}_{inq} = 10.24$ (sec), $N^{to}_{inq} = 5$, and $T^{to}_{idle} = 20$ (sec); in setting W_2, we have: $T^{to}_{inq} = 5.12$ (sec), $N^{to}_{inq} = 3$, and $T^{to}_{idle} = 10$ (sec). For each of the 8 scenarios, we use ns-2 to simulate 20 sample runs, in which the initial infection is randomly chosen. Some other parameters in the experiments are given as follows: $r_{ra} = 10$ (m), $S_{worm} = 20000$ (bytes), $S_{prb} = 27$ (bytes), $T^{to}_{conn} = 5.12$ (sec), $T^{to}_{prb} = 1$ (sec), $T^{to}_{rep} = 10$ (sec), $T^{to}_{disc} = 0.1$ (sec).

ID	N_{dev}	S_{dev}	λ_{ne}	J_{in}
M_1	50	75×75 m^2	0.5239	2.4751
M_2	200	75×75 m^2	2.1199	10.0088
M_3	200	150×150 m^2	0.5753	2.6651
M_4	800	150×150 m^2	2.3089	10.6693

Table 2. Mobility parameter settings

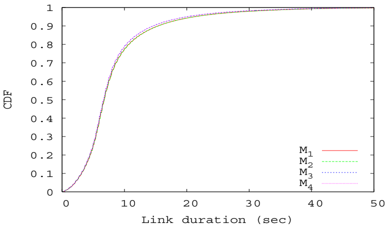


Figure 2. CDF of link durations

Figures 3 and 4 depict the fraction of infected devices as a function of propagation time derived from the model and that obtained from the simulation by averaging 20 sample runs for each scenario. Apparently, the infection curves produced from the model match well with the simulation results in most cases. The only exception happens under Mobility scenario M_3 and Bluetooth worm parameter setting W_2: the model slightly overestimates the worm propagation speed in the late stage of the worm propagation.

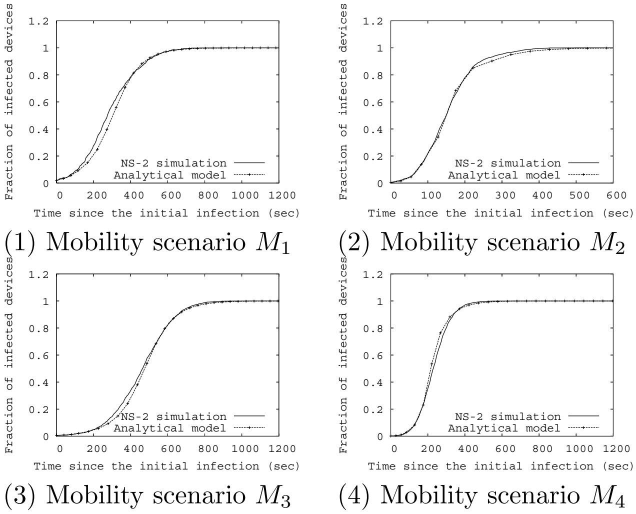
To quantify the prediction errors from the model, we consider the times needed to infect 20%, 40%, 60%, and 80% of the population. The model, due to its variable time steps, may not produce infection states at these points. We thus use a linear model to predict the infection states between consecutive time steps. We compute the relative errors on the times needed to infect 20%, 40%, 60%, and 80% of the population. The results show that in all the cases, the relative errors are smaller than 18%, and in most of the cases, the relative errors are below 10%. Hence, the model characterizes well the propagation process of Bluetooth worms.

8 Related Work

There have been substantial efforts in modeling the propagation dynamics of Internet worms recently. Staniford et al. used the logistic function to fit the propagation curve of the Code Red worm [11]. Zou et al. proposed a two-factor worm model to model the epidemic spreading of Internet worms [17].

So far, there are only a few papers that investigate the behavior of mobile worms. Bose et al. gave a comprehensive survey of existing Bluetooth viruses and

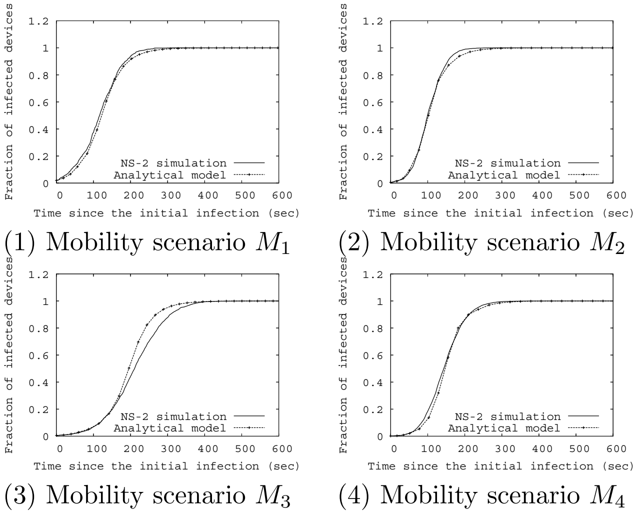
0-7695-2837-3/07 $20.00 © 2007 IEEE



(1) Mobility scenario M_1 (2) Mobility scenario M_2

(3) Mobility scenario M_3 (4) Mobility scenario M_4

Figure 3. Infection curves of under different mobility models (T_{inq}^{to} = 10.24s, N_{inq}^{to} = 5, T_{idle}^{to} = 20s)



(1) Mobility scenario M_1 (2) Mobility scenario M_2

(3) Mobility scenario M_3 (4) Mobility scenario M_4

Figure 4. Infection curves of under different mobility models (T_{inq}^{to} = 5.12s, N_{inq}^{to} = 3, T_{idle}^{to} = 10s)

worms in [3]. As a starting point of research on Bluetooth worms, simulations of the Bluetooth worm propagation have been pursued from different perspectives in [3][15][12]. In [14], Yan et al. investigated the impact of mobility patterns on the Bluetooth worm propagation. Mickens et al. proposed a probabilistic queueing model to model epidemic spreading in mobile environments [7].

9 Conclusions

Recently, Bluetooth worms have created growing security concerns over the data stored on mobile devices such as cellular phones and PDAs. This paper proposes a detailed model that characterizes the propagation dynamics of Bluetooth worms. We have used our model for a large-scale scenario roughly modeling metropolitan Los Angeles (4M people with Bluetooth devices on 500 square miles). After setting model parameters accordingly ($\lambda_{ne} = 0.2108$ and $J_{in} = 0.2372$), our model predicts that the time it would take to infect 99% of the devices is slightly less than one hour.

References

[1] http://www.ececs.uc.edu/ cdmc/ucbt/ucbt.html.

[2] http://www.isi.edu/nsnam/ns/index.html.

[3] A. Bose and K. G. Shin. On mobile viruses exploiting messaging and Bluetooth services. In *Proceedings of SecureComm'06*, August 2006.

[4] J. Bray and C. Sturman. *Bluetooth: Connect Without Cables*. Prentice Hall, December 2000.

[5] A. Busboom, I. Herwono, M. Schuba, and G. Zavagli. Unambiguous device identification and fast connection setup in Bluetooth. In *Proceedings of the European Wireless 2002*, Florence, Italy, 2002.

[6] J. N. Daigle. *Queueing Theory with applications to Packet Telecommunication*. Springer, 2005.

[7] J. W. Mickens and B. D. Noble. Modeling epidemic spreading in mobile environments. In *Proceedings of the 4th ACM workshop on Wireless security*, Sept. 2005.

[8] http://www.gnu.org/software/octave/.

[9] B. S. Peterson. *Device Discovery in Frequency Hopping Wireless Ad Hoc Networks*. PhD thesis, Air Force Institute of Technology, 2004.

[10] Specification of the Bluetooth system: Core, version 1.1, February 2001.

[11] S. Staniford, V. Paxson, and N. Weaver. How to 0wn the internet in your spare time. In *Proceedings of the 11th USENIX Security Symposium (Security '02)*, 2002.

[12] J. Su, A. G. Miklas K. K. W. Chan, K. Po, A. Akhavan, S. Saroiu, E. d. Lara, and A. Goel. A preliminary investigation of worm infections in a Bluetooth environment. In *Proceedings of WORM'06*, 2006.

[13] C. Taylor and N. Mawston. Bluetooth market doubles: CSR still gaining momentum. http://www.strategyanalytics.net/, December 2005.

[14] G. Yan, L. Cuellar, S. Eidenbenz, H. D. Flores, N. Hengartner, and V. Vu. Bluetooth worm propagation: Mobility pattern matters! In *Proceedings of ACM ASIACCS'07*, March 2007.

[15] G. Yan and S. Eidenbenz. Bluetooth worms: Models, dynamics, and defense implications. In *Proceedings of ACSAC'06*, December 2006.

[16] G. Yan and S. Eidenbenz. Modeling propagation dynamics of Bluetooth worms. Technical Report LA-UR-06-8586, Los Alamos National Laboratory, 2006.

[17] C. C. Zou, W. Gong, and D. Towsley. Code red worm propagation modeling and analysis. In *Proceedings ACM CCS'02*, October 2002.