Modeling Propagation Dynamics of Bluetooth Worms (Extended Version)

Guanhua Yan and Stephan Eidenbenz

Abstract—In the last few years, the growing popularity of mobile devices has made them attractive to virus and worm writers. One communication channel often 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. We further use our model to efficiently predict the propagation curve of Bluetooth worms in big cities such as Los Angeles. Our model not only sheds light on the propagation dynamics of Bluetooth worms but also allows one to predict spreading curves of Bluetooth worm propagation in large areas without the high computational cost of discrete-event simulation.

Index Terms—Bluetooth, Bluetooth worm, epidemic modeling, propagation dynamics.

1 Introduction

The last decade has witnessed a surge of wireless mobile devices such as cellular phones, PDAs, and headsets. With such prevalence of wireless mobile devices in our everyday life, more and more viruses and worms have surfaced on them. So far, we have seen more than 100 malware instances that propagate on various types of mobile devices [4]. Common to many existing mobile viruses and worms is that they leverage Bluetooth capabilities to propagate themselves. Bluetooth is a short-range radio technology aimed at connecting different wireless devices at low power consumption and at low cost. It 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 been growing tremendously: worldwide, 272 million Bluetooth-enabled devices have been shipped in 2005, twice as many as in 2004 [19].

Computer worms, which have been rampant in the Internet for more than two decades, are nothing new to us. Bluetooth worms significantly differ from Internet worms in three ways. First, the limited transmission range of a Bluetooth device leads to a proximity-based infection mechanism: a Bluetooth-enabled device controlled by the worm can only infect neighbors within its radio range. This differs from Internet worms that often scan the entire IP address space for susceptible victims. Second, the bandwidth available to Bluetooth devices is usually much narrower than those of Internet links. For instance, the maximum transmission rate of a device operating on the class 2 Bluetooth radio is 1 Mbps. Finally, due to the mobility and limited transmission ranges of Bluetooth devices, the underlying network topology on which Bluetooth worms propagate is much more dynamic than that of Internet worms.

Although there have been substantial efforts on modeling Internet worms [17], [24], [25], [20], 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 include some key statistical metrics that describe the underlying mobility patterns, such as average node degree, average node meeting rate, and the link duration distribution. The input parameters of the model also include control parameters used by the Bluetooth worms, such as how much time the Bluetooth worm spends at most on discovering new victims and how many victims it expects to discover within a single infection cycle, and how long the Bluetooth worm remains dormant before it is activated again for new infection attempts.

The development of the model rests 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 accurately predicts the propagation dynamics of Bluetooth worms, with relative errors smaller than 10 percent in most cases. To illustrate a use case of our model, we show how to set the input parameters to predict the spreading curve of a Bluetooth worm in a large metropolitan area such as Los Angeles with four million Bluetooth devices. The execution time to calculate the curve is only 30 minutes on a commodity PC. This is in contrast to the scaling limit of discrete-event simulators, such as ns-2, which can only simulate a few thousand Bluetooth devices within a reasonable amount of time. Hence, our model not only sheds light on the propagation dynamics of Bluetooth worms but also allows one to predict spreading curves of Bluetooth worm propagation in large areas without the high computational cost of discrete-event simulation.

The remainder of this paper is structured as follows: Section 2 briefly introduces the Bluetooth protocol. Section 3 presents a simple behavior model of a typical Bluetooth

• *The authors are with the Information Sciences Group (CCS-3), Los Alamos National Laboratory, PO Box 1663, MS B256, Los Alamos, NM 87545. E-mail: {ghyan, eidenben}@lanl.gov.*

Manuscript received 16 May 2008; revised 4 Aug. 2008; accepted 18 Aug. 2008; published online 5 Sept. 2008.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2008-05-0190. Digital Object Identifier no. 10.1109/TMC.2008.129.

worm, which includes three phases: inquiry phase, infection attempt phase, and inactive phase. Section 4 discusses the modeling methodology used in this paper. Section 5 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 6 models the period that a worm contacts each neighbor it has discovered and attempts to infect each one of them. Section 7 presents a model that captures the packet loss probability and the data throughput in a Bluetooth network as the worm spreads in it. In Section 8, we describe a model that estimates the infection curve from the analysis of a single infection cycle. Section 9 presents results of model validation and model-based prediction. Some further discussions are provided in Section 10. Finally, Section 11 gives an overview of related work, and Section 12 concludes this paper.

2 Bluetooth Primer

In this section, we present a brief overview of Bluetooth technology [3], [8]. Bluetooth is a short-range radio technology that is aimed at connecting different wireless devices at low power consumption and at low cost. Bluetooth is designed to work in areas with high densities of communicating devices and high-level radio-frequency noise from sources like microwave ovens and cordless phones. It operates in the 2.4-GHz frequency band and its channels are shared among devices through a time-division duplexing (TDD) scheme. Bluetooth also uses a frequency hopping scheme to reduce interference. A Bluetooth device can operate at any one of three power levels: power classes 1, 2, and 3. They correspond to ranges of 100, 10, and 0.1 m and maximum output powers of 20, 4, and 0 dBm, respectively.

When a Bluetooth device wants to find other devices in its vicinity, it broadcasts inquiry packets by hopping 3,200 times per second along a 32-channel inquiry hopping sequence. A nearby device in the *discoverable* mode listens on the same frequency sequence but moves forward its listening carrier every 1.28 seconds. When a device hears an inquiry packet, it backs off for a random period of time and then reenters the scanning state. When it receives another inquiry packet, it responds with a Frequency Hop Synchronization (FHS) packet. On the arrival of this packet, the inquirer device discovers the responder.

Once a device has discovered its neighboring devices, it may want to establish a connection with one or more of them. In order to set up a Bluetooth link with a neighbor device, it goes through the paging process. This process is similar to the inquiry process, except that the paging device explicitly specifies the receiver's address to indicate which device it wants to set up a connection with. After a connection is established, the pager device and the paged device are called the *master* and *slave* of the new link, respectively. In the connected state, the master and the slave can exchange normal data packets by hopping 1,600 times per second along a 79-channel frequency sequence decided by the master's local clock and its device address.

A master device can have up to seven slaves in a *piconet*, which is a collection of Bluetooth devices sharing the same channel. In a piconet, communications can only occur between the master and the slaves. In other words, two slaves in a piconet do not communicate with each other directly. The master device in a piconet regulates how the channel bandwidth is allocated to each slave. A piconet contains only a small number of Bluetooth devices, but multiple piconets can form a larger network, which is called a *scatternet*. A device can only be a master in at most one piconet in a scatternet, but it can act as slaves simultaneously in multiple piconets.

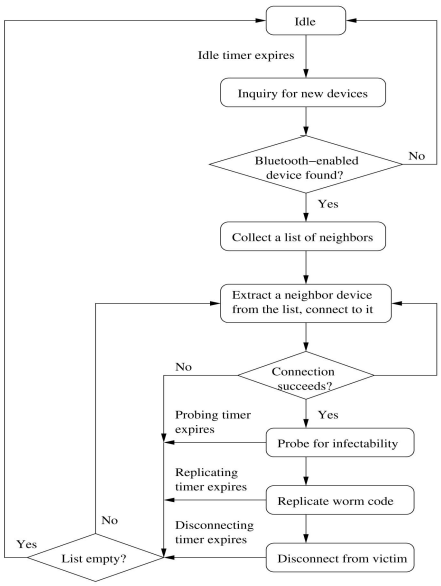

Fig. 1. Infection cycle of a Bluetooth worm.

A Bluetooth link has a maximum capacity of 1 Mbps. The Bluetooth specification defines two types of links: synchronous connection-oriented (SCO) links for voice communication and asynchronous connectionless (ACL) links for data communication. In this paper, we assume that a Bluetooth worm uses ACL links for propagation. ACL links use the Automatic Repeat Request (ARQ) scheme to recover lost packets. ACL links support six types of packets and DH1 is one of them. The maximum symmetric data throughput of an ACL link using DH1 packets is 172.8 Kbps.

3 Behavior of Bluetooth Worms

The infection cycle of a typical Bluetooth worm can break down into several phases, as 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 specifies the expected number of responses, N_{inq}^{to}, and the maximum amount of time it wants to wait, T_{inq}^{to}. If N_{inq}^{to} responses are received before T_{inq}^{to} time units elapse, the worm stops the inquiry phase on the arrival of the $N_{inq}^{to}\text{th}$ 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_{inq}^{to} time units.

Once the worm collects 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*). Owing 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, thus 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 [5] for the details in this process. In step 2, whether a device is infectable hinges on the vulnerability the worm exploits. For example, the Commwarrior worm probes each victim device for the availability of the "Obex Push" service; on a positive reply, the worm replicates itself onto the victim. In our model, we simplify 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 is contingent on both the Bluetooth packet type and the size of the worm code. In our model, we use DH1 packet type for transmission due to its simplicity for analysis. The size of the worm code varies from worm to worm. For example, the Cabir.H worm consists of about 7,000 bytes, but the Commwarrior.a!sys worm has 30,582 bytes. In our model, we use S_{worm} to denote the worm code size.

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.

4 Modeling Methodology

The model we propose for characterizing the propagation dynamics of Bluetooth worms is deterministic and advances the time in a discrete fashion. Let $i(t)$ denote 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 progress $i(t_k)$ at time t_k, where $k \geq 0$, the model determines the next time point t_{k+1} and the associated worm propagation status $i(t_{k+1})$. 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:

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

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

To derive both $T_{cycle}(t)$ 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, e.g., the distributed infective model [11] that divides the whole area into multiple patches and updates the worm propagation status in each patch separately.

	Input	Explanation
Mobility metrics	λ_{ne}	Avg. number of neighbors a device meets per sec.
	J_{in}	Avg. node degree
	$F_L(\tau)$	Cumulative distribution function of link durations
	N_{dev}	Number of devices
	S_{dev}	Size of the area in which devices move
Worm parameters	T^{to}_{inq}	Inquiry timeout value
	N^{to}_{inq}	Max. number of inquiry responses waited for
	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

Fig. 2. Input parameters fed to the model.

Assuming that individual devices are homogeneously mixed, we can abstract the underlying mobility model into a few statistical metrics. Fig. 2 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 3. The Bluetooth worm parameters are also listed in Fig. 2.

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 an 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 (1) to estimate the worm propagation curve.

5 Modeling the Inquiry Phase

We first model the inquiry phase. Consider an infective device starting inquiry at time t. Without loss of generality, we assume it is device 0. Let $T_{inq}(t)$ represent the average duration of the inquiry phase at time t.

5.1 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. 2, 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 is exponentially distributed. Hence, the arrival process of new neighbors is a Poisson process. This assumption will also be used later in Section 5.4 to derive the mean duration of the inquiry phase. Let $\lambda_{ne}(t)$ be the arrival rate of new neighbors. Using the Poisson Arrivals See The Average (PASTA) 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 equal to $J_{in}(t) + J_{co}(t)$, where $J_{co}(t) = \lambda_{ne}(t) \cdot T_{inq}(t)$.

5.2 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. The back-off mechanism in the Bluetooth protocol avoids the situation in which multiple neighbors respond to an inquiry packet in the same time slot. This implies that a neighbor has to stay in the inquiry device's radio range long enough to get discovered. Let D denote the time that an inquiry device needs to discover a neighbor in its radio range. In [15], it is shown that the distribution function of D actually depends on how many devices are performing inquiry operation simultaneously. Even if there is only one device searching for its neighbors, the solution provided in [15] is so complicated that it is difficult to incorporate it in our model; if there is more than one device performing inquiry, deriving the distribution function of D is infeasible [15]. Under such circumstances, we 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. More specifically, we simulate 11 Bluetooth devices, among which 10 of them periodically inquire for neighbors. All these 11 Bluetooth devices are static and located within each other's communication range. In the experiments, we consider a perfect situation in which no packets are dropped due to cochannel interference. We use the ns-2 simulator [2], extended with the UCBT Bluetooth simulation module [1], which provides a very detailed implementation of the full Bluetooth protocol stack, to derive the average duration between the time an inquiry is issued and the time a neighbor is discovered. We vary the number of Bluetooth devices that simultaneously scan for new neighbors from 1 to 10 in the experiments. We apply the linear least squares regression method to the simulation result and obtain the following equation:

$$\bar{D}(k) = 0.3322 \cdot k + 2.2325. \tag{2}$$

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 [15]. We do not consider stochastic variance in our model for simplicity, although it has been shown to be an important factor that affects propagation dynamics of Internet worms [13]. We further assume that the discovery time $D(k)$ is uniformly distributed between 0 and $2\bar{D}(k)$. The observations made 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

$$f_{D(k)}(\tau) = \frac{1}{2\bar{D}(k)}. \tag{3}$$

On the other hand, 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 $P_{inf}^{inq}(t)$, is thus

$$P_{inf}^{inq}(t) = \frac{T_{inq}(t)}{T_{cycle}(t)}. \tag{4}$$

It then immediately follows

$$m(t) = i(t) \cdot \pi r_{ra}^2 \cdot P_{inf}^{inq}(t), \tag{5}$$

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 link durations at time t. Here, a link means the period during which two devices remain each other's communication range. We cannot 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:

$$\begin{aligned} A_1 &: T_{gap}(t) < 0 \quad \text{and} \quad T_{gap}(t) + L(t) > 0, \\ A_2 &: T_{gap}(t) \geq 0 \quad \text{and} \quad T_{gap}(t) < T_{inq}(t). \end{aligned}$$

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

$$P_{A_1} = \mathbb{P}\{T_{gap}(t) + L(t) > 0 \wedge T_{gap}(t) < 0\}, \tag{6}$$

$$P_{A_2} = \mathbb{P}\{0 \leq T_{gap}(t) < T_{inq}(t)\}. \tag{7}$$

On the other hand, 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:

$$\begin{aligned} B_1 &: T_{gap}(t) < 0, 0 \leq D(m(t)) \leq T_{inq}(t) \text{ and} \\ &\quad T_{gap}(t) + L(t) \geq D(m(t)), \\ B_2 &: T_{gap}(t) \geq 0, 0 \leq D(m(t)) \leq T_{inq}(t), \\ &\quad L(t) \geq D(m(t)), \text{ and } T_{gap}(t) + D(m(t)) \leq T_{inq}(t). \end{aligned}$$

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. Hence,

$$P_{B_1} = \mathbb{P}\Big\{T_{gap}(t) < 0 \wedge D(m(t)) \le T_{inq}(t) \wedge T_{gap}(t) + L(t) \ge D(m(t))\Big\}, \tag{8}$$

$$P_{B_2} = \mathbb{P}\Big\{0 \le T_{gap}(t) \le T_{inq}(t) - D(m(t)) \wedge L(t) \ge D(m(t)) \wedge D(m(t)) \le T_{inq}(t)\Big\}. \tag{9}$$

Let $P^{in}_{dsc}(t)$ and $P^{co}_{dsc}(t)$ be the probability that an instantaneous neighbor and a contingent neighbor can be discovered by device 0, respectively. Clearly, we have the following:

$$P^{in}_{dsc}(t) = \frac{P_{B_1}}{P_{A_1}} \quad \text{and} \quad P^{co}_{dsc}(t) = \frac{P_{B_2}}{P_{A_2}}. \tag{10}$$

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 $\max(\Phi_l, T^{to}_{inq})$. Typically, Φ_l is much larger than T^{to}_{inq}. Hence, the probability density function of $T_{gap}(t)$, denoted by $f_{T_{gap}(t)}(\tau)$, is

$$f_{T_{gap}(t)}(\tau) = \frac{1}{2\Phi}, \quad \text{where } \Phi = \max\Big(\Phi_l, T^{to}_{inq}\Big). \tag{11}$$

Given that the probability density functions of $T_{gap}(t)$, $D(m(t))$, and $L(t)$ are $f_{T_{gap}(t)}(s)$, $f_{D(m(t))}(v)$, and $f_{L(t)}(\tau)$, respectively, we are able to compute P_{B_1} and P_{B_2} as follows:

$$\begin{aligned} P_{B_1} &= \int_{-\Phi}^{0} \mathrm{d}s \int_{0}^{\min\{T_{inq}(t), 2\bar{D}(m(t))\}} \mathrm{d}v \int_{v-s}^{\Phi} f_{T_{gap}(t)}(s) \cdot f_{D(m(t))}(v) \cdot f_{L(t)}(\tau)\mathrm{d}\tau \\ &= \frac{1}{2\bar{D}(m(t))} \cdot \frac{1}{2\Phi} \cdot \int_{0}^{\Phi} \mathrm{d}s \int_{0}^{\min\{T_{inq}(t), 2\bar{D}(m(t))\}} \mathbb{P}\{L(t) \ge v + s\}\mathrm{d}v, \end{aligned} \tag{12}$$

$$\begin{aligned} P_{B_2} &= \int_{0}^{T_{inq}(t)} \mathrm{d}v \int_{0}^{T_{inq}(t)-v} \mathrm{d}s \int_{v}^{\infty} f_{D(m(t))}(v) \cdot f_{T_{gap}(t)}(s) \cdot f_{L(t)}(\tau)\mathrm{d}\tau \\ &= \frac{1}{2\bar{D}(m(t))} \cdot \frac{1}{2\Phi} \cdot \int_{0}^{T_{inq}(t)} \big(T_{inq}(t) - v\big) \cdot \mathbb{P}\{L(t) \ge v\}\mathrm{d}v. \end{aligned} \tag{13}$$

In the above computation, we have used (3) and (11). $f_{L(t)}(\tau)$ is the input parameter of our model. In (12), we have $\int_{v-s}^{\Phi} f_{L(t)}(\tau)\mathrm{d}\tau = \mathbb{P}\{L(t) \ge v - s\}$ and then reverse the sign of s when doing integration on it. Similarly in (13), we have $\int_{v}^{\infty} f_{L(t)}(\tau)\mathrm{d}\tau = \mathbb{P}\{L(t) \ge v\}$.

5.3 Number of Inquiry Responses

If we ignore the impact of interference among different responses, the responses can be assumed to be independent. Hence, we can model it as a binomial process. Let n be the number of neighbors device 0 meets in its inquiry phase and p be the probability for each of them to be discovered. The average number of responses is then np. However, the assumption that all neighbors are discovered by the inquiry device with the same probability, no matter whether they have already been infected or not, does not necessarily hold, because an infected neighbor in the inquiry or paging state cannot respond to the inquiry. Let $P^{av}_{inf}(t)$ 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

$$P^{av}_{inf}(t) = 1 - \frac{T_{inq}(t) + T_{page}(t)}{T_{cycle}(t)}. \tag{14}$$

Even though a neighbor is not infected or is not in either inquiry or paging mode, it may be contacted by another infected device. The Bluetooth protocol state transition diagram [16] shows that only a neighbor in the CONNECTION state or the STANDBY state can move to the INQUIRY-SCAN state, in which it can be discovered by other devices. Hence, if a neighbor is being contacted by other infected devices and is thus not in these two states, it cannot transition to the INQUIRY-SCAN state and thus cannot 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

$$N_{proc}(t) = \frac{T_{proc}(t)}{T_{cycle}(t)} \cdot i(t) \cdot \pi r^2_{ra}. \tag{15}$$

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

$$M_{proc}(t) = \left(\rho(t) - i(t) + \frac{T^{to}_{idle}}{T_{cycle}(t)} \cdot i(t)\right) \cdot \pi(2r_{ra})^2. \tag{16}$$

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

$$P_{rsp}(t) = \left(\frac{M_{proc}(t)-1}{M_{proc}(t)}\right)^{N_{proc}(t)}. \quad (17)$$

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:

$$\hbar(t) = \frac{\rho(t)-i(t)}{\rho(t)} + \frac{i(t)}{\rho(t)} \cdot P_{inf}^{av}. \quad (18)$$

We then have

$$N_{rsp}^{in}(t) = J_{in}(t) \cdot P_{dsc}^{in}(t) \cdot \hbar(t) \cdot P_{rsp}(t), \quad (19)$$

$$N_{rsp}^{co}(t) = J_{co}(t) \cdot P_{dsc}^{co}(t) \cdot \hbar(t) \cdot P_{rsp}(t). \quad (20)$$

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:

$$R(t) = \min\left\{N_{inq}^{to}, N_{rsp}^{in}(t) + N_{rsp}^{co}(t)\right\}. \quad (21)$$

5.4 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:

$$T_{inq}(t) = \bar{D}(m(t)), \text{ if } N_{rsp}^{in}(t) \geq N_{inq}^{to}. \quad (22)$$

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. 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

$$\lambda_1(t) = \frac{\rho(t)-i(t)}{\rho(t)} \cdot \lambda_{ne}(t) \cdot P_{rsp}(t). \quad (23)$$

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

$$\lambda_2(t) = \frac{i(t)}{\rho(t)} \cdot \lambda_{ne}(t) \cdot P_{inf}^{av} \cdot P_{rsp}(t). \quad (24)$$

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

$$\lambda(t) = (\lambda_1(t) + \lambda_2(t)) \cdot P_{dsc}^{co}(t). \quad (25)$$

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 [7]

$$z_n(s) = \frac{\lambda(t)(\lambda(t)s)^{n-1}e^{-\lambda(t)s}}{(n-1)!}. \quad (26)$$

Since $N_{rsp}^{in}(t)$ instantaneous neighbors have already been found, device 0 only needs to find $N_{inq}^{to} - N_{rsp}^{in}(t)$ contingent neighbors. However, if the inquiry timer expires before it does so, it cannot find N_{inq}^{to} neighbors eventually. So, the average duration of the inquiry phase is

$$T_{inq}(t) = \frac{1}{2\bar{D}(m(t))} \int_0^{2\bar{D}(m(t))} \times \left(\left(1 - \int_0^{T_{inq}^{to}-v} z_\epsilon(t)\mathrm{d}t\right) \cdot T_{inq}^{to} + \int_0^{T_{inq}^{to}-v} (t+v) \cdot z_\epsilon(t)\mathrm{d}t\right)\mathrm{d}v, \quad (27)$$

where ϵ is $N_{inq}^{to} - N_{rsp}^{in}(t)$. Note that in (27), 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 by device 0.

6 Modeling the 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:

$$\tau_s^{(1)}(t) = T_{inq}(t). \quad (28)$$

6.1 Step of 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 and uninfected devices among all the neighbors discovered by device 0, respectively. According to the discussion in Section 5.3, infected devices in inquiry or paging mode do not respond to the inquiry of device 0. Hence, we have

$$P_i(t) = \frac{P_{inf}^{av}(t) \cdot i(t)}{P_{inf}^{av}(t) \cdot i(t) + \rho(t) - i(t)}, \tag{29}$$

$$P_u(t) = \frac{\rho(t) - i(t)}{P_{inf}^{av}(t) \cdot i(t) + \rho(t) - i(t)}. \tag{30}$$

For any of these infected neighbors, if it is in the inquiry or paging mode, device 0 cannot 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 $P_{page}^{pos}(t)$ denote the probability that a neighbor discovered by device 0 is pageable and $P_{page}^{neg}(t)$ denote the probability that a neighbor discovered by device 0 is not pageable. We then have

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

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

$$P_i^{page}(t) = \frac{P_i(t) \cdot P_{inf}^{av}(t)}{P_i(t) \cdot P_{inf}^{av}(t) + P_u(t)}, \tag{31}$$

$$P_u^{page}(t) = \frac{P_u(t)}{P_i(t) \cdot P_{inf}^{av}(t) + P_u(t)}. \tag{32}$$

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 also carrying the slave's access code. An iteration of two-way handshake fails if either of the packets gets dropped. Let $P_{loss}^{page}(t)$ denote the paging packet loss rate at time t. In the following, we compute $T_{conn}^{good}(t)$, the average time needed for successfully establishing a connection provided that the paging packet loss rate is $P_{loss}^{page}(t)$. As we discuss later in Section 7, the loss probability of a packet is related to its size. Since a paging response packet has the same size as a paging packet, the loss probability of paging response packets is also $P_{loss}^{page}(t)$. The computation of $P_{loss}^{page}(t)$ will be introduced later in Section 7. 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 [16], the mean duration of the paging process is 1.28 seconds and its maximum duration is 2.56 seconds [6]. 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 $(1 - P_{loss}^{page}(t))^2$. The following table illustrates the computation of the average time needed to establish a connection successfully:

Probability	Average duration of paging process
$\delta(t)$	1.28
$(1-\delta(t))\delta(t)$	$1.28 + 1 \times 2.56$
$(1-\delta(t))^2\delta(t)$	$1.28 + 2 \times 2.56$
$\dots$	$\dots$
$(1-\delta(t))^{s-1}\delta(t)$	$1.28 + (s-1) \times 2.56$

where $s = \lfloor \frac{T_{conn}^{to}}{2.56} \rfloor$. Hence, the average duration of a successful paging process is

$$T_{conn}^{good}(t) = ((1/\delta(t) - (1/\delta(t) + s)(1 - \delta(t))^s) \times 2.56 - 1.28.$$

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:

$$Q_1 : L(t) + T_{gap}(t) \geq \tau_s^{(k)}(t) + T_{conn}^{good}(t). \tag{33}$$

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 proposition B_1 or B_2 must be true. Then, the probability that device 0 can connect to neighbor k successfully is

$$\begin{aligned} P_{conn}^{succ}\Big(t, \tau_s^{(k)}(t)\Big) &= P_{page}^{pos}(t) \cdot \mathbb{P}\{Q_1 \mid B_1 \vee B_2\} \\ &= P_{page}^{pos}(t) \cdot \frac{\mathbb{P}\{Q_1 \wedge (B_1 \vee B_2)\}}{\mathbb{P}\{B_1 \vee B_2\}}. \end{aligned}$$

To further simplify the above equation, we introduce another proposition Q_0:

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

After applying some logic computation, we have $Q_1 \wedge (B_1 \vee B_2) = Q_0 \wedge Q_1$. Hence, $P_{conn}^{succ}(t, \tau_s^{(k)}(t))$ can be rewritten as

$$P_{conn}^{succ}\Big(t, \tau_s^{(k)}(t)\Big) = P_{page}^{pos}(t) \cdot \frac{\mathbb{P}\{Q_0 \wedge Q_1\}}{\mathbb{P}\{B_1 \vee B_2\}}. \tag{34}$$

If device 0 fails to establish a connection to neighbor k, it has to wait until the connection establishing timer expires, which lasts T_{conn}^{to} time units. Let $P_{conn}^{fail}(t, \tau_s^{(k)}(t))$ 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 $\tau_s^{(k)}(t)$. It immediately follows:

$$\begin{aligned} P_{conn}^{fail}\Big(t, \tau_s^{(k)}(t)\Big) &= 1 - P_{conn}^{succ}\Big(t, \tau_s^{(k)}(t)\Big) \\ &= 1 - P_{page}^{pos}(t) \cdot \frac{\mathbb{P}\{Q_0 \wedge Q_1\}}{\mathbb{P}\{B_1 \vee B_2\}}. \end{aligned} \tag{35}$$

In (34) and (35), $\mathbb{P}\{B_1 \vee B_2\}$ is actually equal to $\mathbb{P}\{B_1\} + \mathbb{P}\{B_2\}$ and $\mathbb{P}\{Q_0 \wedge Q_1\}$ can be written as an expression of $\mathbb{P}\{T_{gap}(t) + D_{m(t)} \le T_{inq}(t) \wedge D_{m(t)} \le T_{inq}(t) \wedge L(t) + T_{gap}(t) \ge Y(t)\}$, where $Y(t) \ge T_{inq}(t)$. It can be computed as follows:

$$\begin{aligned}
&\mathbb{P}\Big(T_{gap}(t) + D_{m(t)} \le T_{inq}(t) \wedge D_{m(t)} \le T_{inq}(t) \wedge L(t) \\
&\quad + T_{gap}(t) \ge Y(t)\Big) \\
&= \int_{-\Phi}^{T_{inq}(t)} \mathrm{d}s \int_{0}^{\min\{T_{inq}(t), 2\bar{D}(m(t)), T_{inq}(t)-s\}} \mathrm{d}v \\
&\quad \int_{Y(t)-s}^{\Phi} f_{T_{gap}(t)}(s) \cdot f_{D(m(t))}(v) \cdot f_{L(t)}(\tau)\mathrm{d}\tau \\
&= \frac{1}{2\bar{D}(m(t))} \cdot \frac{1}{2\Phi} \cdot \min\{T_{inq}(t), 2\bar{D}(m(t))\} \\
&\quad \cdot \int_{-\Phi}^{0} \cdot\, \mathbb{P}(l \ge Y(t) - s)\mathrm{d}s + \frac{1}{2\bar{D}(m(t))} \cdot \frac{1}{2\Phi} \\
&\quad \cdot \int_{0}^{T_{inq}(t)} \min\{2\bar{D}(m(t)), T_{inq}(t) - s\}\mathbb{P}(l \ge Y(t) - s)\mathrm{d}s).
\end{aligned} \tag{36}$$

6.2 Step of 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. The computation of $\eta(t)$ will be introduced in Section 7. Recall that the total number of bytes in the probing packet and replying packet is S_{prb}. Then, the average time needed for a successful probing process is $\frac{S_{prb}}{\eta(t)}$. Therefore, in order for the probing process to finish successfully, the following proposition must hold:

$$Q_2 : L(t) + T_{gap}(t) \ge \tau_s^{(k)}(t) + T_{conn}^{good}(t) + \frac{S_{prb}}{\eta(t)}. \tag{37}$$

The prior knowledge for device 0 to probe the kth neighbor is not only that the neighbor is discovered in its inquiry phase, but also that device 0 successfully establishes a connection to it. That is to say, $(B_1 \vee B_2) \wedge Q_1$ 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}. Furthermore, the probability that device 0 attempts to probe the kth 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. It is easy to see that $Q_1 \wedge Q_2 = Q_2$. Similarly, some logic computation leads to the following: $Q_2 \wedge (B_1 \vee B_2) = Q_0 \wedge Q_2$. Then, by applying Bayes' rule, we have

$$\begin{aligned}
P_{prb}^{succ}\Big(t, \tau_s^{(k)}(t)\Big) &= P_{page}^{pos}(t) \cdot \mathbb{P}\{Q_1 \mid B_1 \vee B_2\} \\
&\quad \cdot \mathbb{P}\{Q_2 \mid (B_1 \vee B_2) \wedge Q_1\} \\
&= P_{page}^{pos}(t) \cdot \frac{\mathbb{P}\{Q_0 \wedge Q_2\}}{\mathbb{P}\{B_1 \vee B_2\}}.
\end{aligned} \tag{38}$$

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

$$\begin{aligned}
P_{prb}^{fail}\Big(t, \tau_s^{(k)}(t)\Big) &= P_{page}^{pos}(t) \cdot \mathbb{P}\{Q_1 \mid B_1 \vee B_2\} \\
&\quad \cdot (1 - \mathbb{P}\{Q_2 \mid (B_1 \vee B_2) \wedge Q_1\}) \\
&= P_{page}^{pos}(t) \cdot \frac{\mathbb{P}\{Q_0 \wedge Q_1\} - \mathbb{P}\{Q_0 \wedge Q_2\}}{\mathbb{P}\{B_1 \vee B_2\}}.
\end{aligned} \tag{39}$$

In (38) and (39), $\mathbb{P}\{B_1 \vee B_2\}$ is equal to $\mathbb{P}\{B_1\} + \mathbb{P}\{B_2\}$, and both $\mathbb{P}\{Q_0 \wedge Q_1\}$ and $\mathbb{P}\{Q_0 \wedge Q_2\}$ can be computed using (36).

6.3 Step of Replicating the Worm Code

After the probing step, if device 0 finds that neighbor k is not yet 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 the following: 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 yet. The probability that all these three conditions are satisfied, denoted by $P_{rep}^{prior}(t, \tau_s^{(k)}(t))$, is

$$\begin{aligned}
P_{rep}^{prior}\Big(t, \tau_s^{(k)}(t)\Big) &= P_{page}^{pos}(t) \cdot \mathbb{P}\{Q_1 \mid B_1 \vee B_2\} \\
&\quad \cdot \mathbb{P}\{Q_2 \mid (B_1 \vee B_2) \wedge Q_1\} \cdot P_u^{page}(t) \\
&= P_{page}^{pos}(t) \cdot \frac{\mathbb{P}\{Q_0 \wedge Q_2\}}{\mathbb{P}\{B_1 \vee B_2\}} \cdot P_u^{page}(t).
\end{aligned} \tag{40}$$

On the other hand, let $P_{inf}^{prb}(t, \tau_s^{(k)}(t))$ be the probability that device 0 finds that neighbor k has already been infected. It is actually

$$\begin{aligned}
P_{inf}^{prb}\Big(t, \tau_s^{(k)}(t)\Big) &= P_{page}^{pos}(t) \cdot \mathbb{P}\{Q_1 \mid B_1 \vee B_2\} \\
&\quad \cdot \mathbb{P}\{Q_2 \mid (B_1 \vee B_2) \wedge Q_1\} \cdot P_i^{page}(t) \\
&= P_{page}^{pos}(t) \cdot \frac{\mathbb{P}\{Q_0 \wedge Q_2\}}{\mathbb{P}\{B_1 \vee B_2\}} \cdot P_i^{page}(t).
\end{aligned} \tag{41}$$

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

$$Q_3 : L(t) + T_{gap}(t) \ge \tau_s^{(k)}(t) + T_{conn}^{good}(t) + \frac{S_{prb}}{\eta(t)} + \frac{S_{worm}}{\eta(t)}. \tag{42}$$

By some logic computation, we have $Q_3 \wedge (B_1 \vee B_2) = Q_0 \wedge Q_3$. Moreover, 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 can be established as follows:

$$\begin{aligned}
P_{rep}^{succ}\Big(t, \tau_s^{(k)}(t)\Big) &= P_{rep}^{prior}\Big(t, \tau_s^{(k)}(t)\Big) \cdot \mathbb{P}\{Q_3 \mid Q_2 \wedge (B_1 \vee B_2)\} \\
&= P_{page}^{pos}(t) \cdot P_u^{page}(t) \cdot \frac{\mathbb{P}\{Q_0 \wedge Q_3\}}{\mathbb{P}\{B_1 \vee B_2\}}.
\end{aligned} \tag{43}$$

On the other hand, if the worm code is not successfully replicated before the code replication timer expires, the failed worm code replication process takes T_{rep}^{to} time units. Let $P_{rep}^{fail}(t, \tau_s^{(k)}(t))$ denote the probability that device 0 fails to deliver the worm code successfully onto the victim. We then have

$$\begin{aligned} P_{rep}^{fail}\Big(t, \tau_s^{(k)}(t)\Big) &= P_{rep}^{prior}\Big(t, \tau_s^{(k)}(t)\Big) \\ &\quad \cdot (1 - \mathbb{P}\{Q_3 \mid Q_2 \wedge (B_1 \vee B_2)\}) \\ &= P_{page}^{pos}(t) \cdot P_u^{page}(t) \\ &\quad \cdot \frac{\mathbb{P}\{Q_0 \wedge Q_2\} - \mathbb{P}\{Q_0 \wedge Q_3\}}{\mathbb{P}\{B_1 \vee B_2\}}. \end{aligned} \tag{44}$$

In (40), (41), (43), and (44), $\mathbb{P}\{B_1 \vee B_2\}$ is equal to $\mathbb{P}\{B_1\} + \mathbb{P}\{B_2\}$, and both $\mathbb{P}\{Q_0 \wedge Q_2\}$ and $\mathbb{P}\{Q_0 \wedge Q_3\}$ can be computed using (36). For simplicity, we ignore the time needed to destroy the connection because it is much shorter than the other phases, although technically there is no difficulty in incorporating it into our model.

6.4 Total Time Spent on Processing All the Neighbors Discovered

The analysis presented in the previous sections suggests that 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.

Suppose that $\vec{V}$ is a vector of five elements. We define function $\Omega(t, k, \tau_s^{(k)}(t), \vec{V})$ recursively as follows:

$$\Omega\Big(t, k, \tau_s^{(k)}(t), \vec{V}\Big) = \begin{cases} 0, & \text{if } k > R(t), \\ \omega, & \text{if } k \le R(t), \end{cases} \tag{45}$$

where $\omega =$

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

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

$$T_{proc}(t) = \Omega\Big(t, 1, T_{inq}(t), \vec{V}_{proc}(t)\Big), \tag{46}$$

where

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

We can further establish the following equation on $T_{cycle}(t)$, the total duration of an infection cycle starting at time t:

$$T_{cycle}(t) = T_{inq}(t) + T_{proc}(t) + T_{idle}^{to}. \tag{47}$$

Similarly, the total time spent on the paging process, i.e., $T_{page}(t)$ in (14), is equal to

$$\Omega\big(t, 1, T_{inq}(t), \langle T_{conn}^{to}, T_{conn}^{good}(t), T_{conn}^{good}(t), T_{conn}^{good}(t), T_{conn}^{good}(t)\rangle\big).$$

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

$$\alpha(t) = \Omega\big(t, 1, T_{inq}(t), \langle 0, 0, 0, 1, 0\rangle\big). \tag{48}$$

7 Modeling the Packet Loss Probability and the Data Throughput

In this section, we present how to compute $P_{loss}^{page}(t)$ and $\eta(t)$ discussed in Sections 6.1, 6.2, and 6.3. We model packet losses due to cochannel interference in our framework. Cochannel interference occurs when devices on different channels use the same frequency to transmit packets at the same time. A detailed packet loss probability model calls for not only a physical model that characterizes signal attenuation on the propagation path, but also a statistical model that captures the distribution of the distance between any two interacting devices. To avoid a complicated model that is difficult to analyze or even solve numerically, we resort to a simple solution that only specifies an interference range, denoted by r_{it}. When a listening device hops on a frequency and receives a packet from a sender in its radio range, if during the reception period another device within its interference range is sending a packet *at the same frequency*, we assume that the received packet is corrupted because of cochannel interference.

It is easy to see that the probability of cochannel interference is related to the size of a packet because it decides the transmission time in the air. The physical transmission rate of a Bluetooth device is 1 Mbps. Fig. 3 depicts the packet transmission times in three different cases. In the first case, an infected device is in either the inquiry phase or the paging phase. In this case, the infected device sends out an inquiry or paging packet in each 625-μs time slot. Because a neighbor only responds when its scanning frequency matches that of the inquiry or paging packet, there are much fewer response packets than the inquiry packets or paging packets in this case. Hence, we ignore these response packets from the neighbors when computing the packet loss rate. Both inquiry packets and paging packets are ID packets in the parlance of the Bluetooth protocol [5]. An ID packet only consists of the access code of the sender and has 68 bits in it. Hence, the

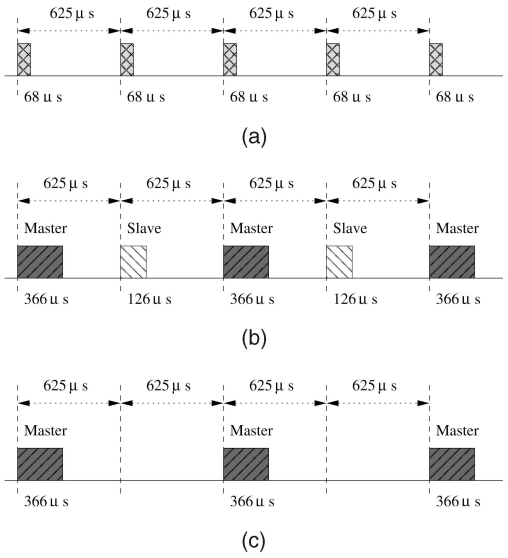

Fig. 3. Transmission times in three different cases. (a) Inquiry packets and paging packets. (b) Data packets and data response packets. (c) Data packets without response.

transmission time of an ID packet is 68 μs. In addition, the transmission frequency of an ID packet is selected from a sequence of 32 frequencies. In the second case, an infected device (the master) sends a data packet to a neighbor (the slave) and the neighbor sends back a reply packet. In our model, we only consider DH1 packets, the payload of which has 27 bits. Taking the packet header into consideration, a DH1 data packet has 366 bits and the reply packet has 126 bits in it. The data packet is transmitted in every other 625-μs time slot, as illustrated in Fig. 3b. The transmission time of the data packet is 366 μs and that of the reply packet is 126 μs. The transmission frequency of a data packet or a reply packet is selected from a sequence of 69 frequencies. We also notice that if the receiver moves out of the communication range of the sender, it cannot send back reply packets. This case is shown in Fig. 3c.

Let $N_1^{ts}(t)$, $N_2^{ts}(t)$, and $N_3^{ts}(t)$ denote the number of 625-μs time slots that an infected device is in these three cases, respectively. We first consider the packet loss probability due to a single infected neighbor X. Let $p_{loss}^{ID}(t)$ denote the loss probability of an ID packet because of a single infected neighbor. We assume that the starting transmission time of this ID packet is uniformly distributed in the whole infection cycle of X. Recall that the transmission time of an ID packet is 68 μs. If the transmission of the ID packet conflicts with that of another ID packet sent by X, the probability that they use the same frequency is $1/32$. On the other hand, if the transmission of the ID packet conflicts with that of a data packet or a data reply packet sent by X, the probability that they use the same frequency is $1/69$, because the 32 frequencies used to send ID packets are a subset of the 69 frequencies used to send data packets. As illustrated in Fig. 3, the idle time interval between two ID packets is 557 μs; the idle time interval between a data packet and the corresponding data reply packet is 259 μs; the idle time interval between a data reply packet and the next data packet is 499 μs; the idle time interval between two consecutive data packets is 884 μs. Suppose that the idle time interval is ω and the transmission time of a packet is ν. If ω is smaller than ν, then the packet cannot be transmitted during that idle time interval successfully. Otherwise, the time window into which the starting transmission time of the packet falls without packet corruption is $\omega - \nu$. Hence, we can compute $p_{loss}^{ID}(t)$ as follows:

$$\begin{aligned} p_{loss}^{ID}(t) &= \frac{1}{32}\cdot\frac{(625-(557-68))\cdot 10^{-6}\cdot N_1^{ts}(t)}{T_{cycle}(t)} \\ &\quad+\frac{1}{69}\cdot\frac{(1{,}250-(259-68)-(499-68))\cdot 10^{-6}\cdot N_2^{ts}(t)/2}{T_{cycle}(t)} \\ &\quad+\frac{1}{69}\cdot\frac{(1{,}250-(884-68))\cdot 10^{-6}\cdot N_3^{ts}(t)/2}{T_{cycle}(t)} \\ &\approx 10^{-6}\times\left(4.25\cdot\frac{N_1^{ts}(t)}{T_{cycle}(t)}+4.5507\cdot\frac{N_2^{ts}(t)}{T_{cycle}(t)}\right. \\ &\quad\left.+3.1449\cdot\frac{N_3^{ts}(t)}{T_{cycle}(t)}\right). \end{aligned}$$

Similarly, let $p_{loss}^{d}(t)$ and $p_{loss}^{r}(t)$ denote the loss probability of a data packet and a data reply packet, respectively, because of a single infected neighbor. Note that a data packet cannot be transmitted during the gap between another data packet sent by a neighbor and the corresponding data reply packet. We can establish the following:

$$\begin{aligned} p_{loss}^{d}(t) &= \frac{1}{69}\cdot\frac{(625-(557-366))\cdot 10^{-6}\cdot N_1^{ts}(t)}{T_{cycle}(t)} \\ &\quad+\frac{1}{69}\cdot\frac{(1{,}250-(499-366))\cdot 10^{-6}\cdot N_2^{ts}(t)/2}{T_{cycle}(t)} \\ &\quad+\frac{1}{69}\cdot\frac{(1{,}250-(884-366))\cdot 10^{-6}\cdot N_3^{ts}(t)/2}{T_{cycle}(t)} \\ &\approx 10^{-6}\times\left(6.2899\cdot\frac{N_1^{ts}(t)}{T_{cycle}(t)}+8.0942\cdot\frac{N_2^{ts}(t)}{T_{cycle}(t)}\right. \\ &\quad\left.+5.3043\cdot\frac{N_3^{ts}(t)}{T_{cycle}(t)}\right), \end{aligned}$$

$$\begin{aligned} p_{loss}^{r}(t) &= \frac{1}{69}\cdot\frac{(625-(557-126))\cdot 10^{-6}\cdot N_1^{ts}(t)}{T_{cycle}(t)} \\ &\quad+\frac{1}{69}\cdot\frac{(1{,}250-(259-126)-(499-126))\cdot 10^{-6}\cdot N_2^{ts}(t)/2}{T_{cycle}(t)} \\ &\quad+\frac{1}{69}\cdot\frac{(1{,}250-(884-126))\cdot 10^{-6}\cdot N_3^{ts}(t)/2}{T_{cycle}(t)} \\ &\approx 10^{-6}\times\left(2.8116\cdot\frac{N_1^{ts}(t)}{T_{cycle}(t)}+5.3913\cdot\frac{N_2^{ts}(t)}{T_{cycle}(t)}\right. \\ &\quad\left.+3.5652\cdot\frac{N_3^{ts}(t)}{T_{cycle}(t)}\right). \end{aligned}$$

Computation of $N_1^{ts}(t)$, $N_2^{ts}(t)$, and $N_3^{ts}(t)$ is similar to that of $T_{proc}(t)$ as explained in Section 6. Here, when an infected device fails to replicate the worm code onto a victim, we ignore the possible data reply packets if the victim leaves the infected device's radio range during the worm code replication process. We have the following:

$$N_1^{ts}(t)=\frac{T_{inq}(t)+\Omega\left(t,1,T_{inq}(t),\vec{V}^*\right)}{0.000625}, \qquad (49)$$

$$N_2^{ts}(t) = \frac{\Omega\Big(t, 1, T_{inq}(t), \Big\langle 0, 0, 0, \frac{S_{worm}}{\eta(t)}, 0\Big\rangle\Big)}{0.000625}, \quad (50)$$

$$N_3^{ts}(t) = \frac{\Omega\Big(t, 1, T_{inq}(t), \Big\langle 0, 0, 0, 0, T_{rep}^{to}\Big\rangle\Big)}{0.000625}, \quad (51)$$

where $\vec{V^*}$ is $\langle T_{conn}^{to}, T_{conn}^{good}(t), T_{conn}^{good}(t), T_{conn}^{good}(t), T_{conn}^{good}(t)\rangle$.

Given that there are $i(t)\cdot\pi r_{it}^2$ devices that have already been infected in the interference range, $P_{loss}^{page}(t)$, the loss probability of a paging packet, can be obtained from the following equation:

$$P_{loss}^{page}(t) = 1 - \big(1 - p_{loss}^{ID}(t)\big)^{i(t)\cdot\pi r_{it}^2}. \quad (52)$$

Let $P_{loss}^{data}(t)$ and $P_{loss}^{reply}(t)$ denote the loss probability of data packets and data reply packets at time t. Similarly, we can establish the following equations:

$$P_{loss}^{data}(t) = 1 - \big(1 - p_{loss}^{d}(t)\big)^{i(t)\cdot\pi r_{it}^2}, \quad (53)$$

$$P_{loss}^{reply}(t) = 1 - \big(1 - p_{loss}^{r}(t)\big)^{i(t)\cdot\pi r_{it}^2}. \quad (54)$$

Now, we can estimate the data throughput under packet losses due to cochannel interference. Let η_0 be the average data throughput in a loss-free environment. We also define $\zeta(t)$ as follows:

$$\begin{aligned}\zeta(t) &= \big(1 - P_{loss}^{data}(t)\big)\Big(1 - P_{loss}^{reply}(t)\Big)\\ &= \big(\big(1 - p_{loss}^{d}(t)\big)\big(1 - p_{loss}^{r}(t)\big)\big)^{i(t)\cdot\pi r_{it}^2}.\end{aligned}$$

The following table illustrates the computation of the average data throughput in a lossy environment:

Probability	Throughput
$\zeta(t)$	η_0
$(1-\zeta(t))\zeta(t)$	$\eta_0/2$
$(1-\zeta(t))^2\zeta(t)$	$\eta_0/3$
$\cdots$	$\cdots$
$(1-\zeta(t))^{z-1}\zeta(t)$	η_0/z
$\cdots$	$\cdots$

The average data throughput is

$$\eta(t) = \begin{cases}\eta_0, & P_{loss}(t) = 0,\\ \eta_0 \times \frac{\zeta(t)\ln\zeta(t)}{\zeta(t)-1}, & 0 < P_{loss}(t) < 1,\\ 0, & P_{loss}(t) = 1.\end{cases} \quad (55)$$

8 Modeling the Infection Curve

We model the Bluetooth worm infection curve using the logistic equation with variable pairwise infection rate. By assuming that individuals are homogeneously mixed, the model can be written as the differential equation given (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 following equation:

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

Then, by solving (1), the worm propagation curve can be characterized as follows:

$$i(t + \Delta t) = \frac{i(t)\cdot\rho(t)}{i(t) + (\rho(t) - i(t))e^{-\beta(t)\cdot\rho(t)\cdot\Delta t}}. \quad (56)$$

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

$$i\big(t + T_{cycle}(t)\big) = \frac{i(t)\cdot\rho(t)}{i(t) + (\rho(t) - i(t))e^{-\alpha(t)\cdot\rho(t)/(\rho(t)-i(t))}}. \quad (57)$$

Equation (57) directly leads to an approach to computing 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:

$$t_{k+1} = t_k + T_{cycle}(t_k), \quad (58)$$

$$i(t_{k+1}) = \frac{i(t_k)\cdot\rho(t_k)}{i(t_k) + (\rho(t_k) - i(t_k))e^{-\alpha(t_k)\cdot\rho(t_k)/(\rho(t_k)-i(t_k))}}. \quad (59)$$

However, we notice that 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 infected devices 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 $S_{inq}(t)$, can be obtained by

$$S_{inq}(t) = \pi r_{ra}^2 + 2\cdot r_{ra}\cdot v(t)\cdot T_{inq}(t), \quad (60)$$

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. We define $i'(t)$ as follows:

$$i'(t) = \max\left\{i(t), \frac{1}{S_{inq}(t)}\right\}. \quad (61)$$

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 (59):

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

The second problem with (58) and (59) is related with the assumption that new infections out of a single infection cycle is evenly distributed in the infection cycle. If there are a significant number of devices that 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, using (59) to predict the worm propagation at the

TABLE 1
Mobility Parameter Settings

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

early phase tends to underestimate the worm propagation speed. Furthermore, 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

$$t_x = t_k + T_{cycle}(t_k) - T_{proc}(t_k). \tag{62}$$

Actually, t_x is the latest time when an infected device finishes its inquiry phase such 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

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

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

$$\alpha' = \frac{\rho(t_k) - i(t_k)}{\rho(t_k)} \cdot \alpha(t_k) + \frac{i(t_k)}{\rho(t_k)} \cdot \alpha(t_x). \tag{63}$$

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

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

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.

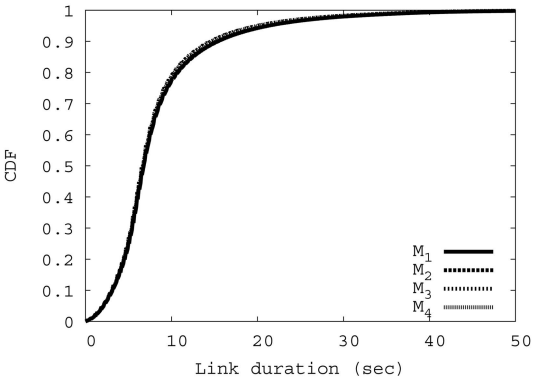

Fig. 4. Cumulative density functions of link durations.

9 Experiments

The system of equations that we have proposed to characterize the Bluetooth worm propagation is comprehensive, covering both dynamics of the Bluetooth protocol behavior and statistical properties of mobility patterns. However, as the number of equations used suggests, it is 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 or intractable. Under such circumstance, we resort to numerical solutions to our model. We have implemented our model as a system of equations in Octave [14], an open source numerical computation software. We then use function *fsolve*, a nonlinear equations solver provided in Octave, to derive the solutions numerically.

9.1 Model Validation

We use the ns-2 network simulator [2], extended with the UCBT Bluetooth simulation module [1], to validate the model. The UCBT Bluetooth simulation module offers a very detailed implementation of the Bluetooth protocol stack. We only modify the component that decides packet losses due to cochannel interference. We replace the original packet loss model with the one introduced in Section 7: During the reception of a packet, if there exists another device within the receiver's interference range transmitting a packet on the same frequency, we assume that the first packet is corrupted because of cochannel interference. In our experiments, we let the interference range, i.e., r_{it}, be 15 m.

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 m/s. The average device speed in our experiments is roughly the same of pedestrians. Table 1 presents N_{dev}, S_{dev}, λ_{ne}, and J_{in} used in our experiments, and Fig. 4 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, although they have different device densities. This can be explained as follows: device densities affect how often two Bluetooth devices "meet" each other, but once they move into each other's communication range, their mobility patterns decide the link duration, which

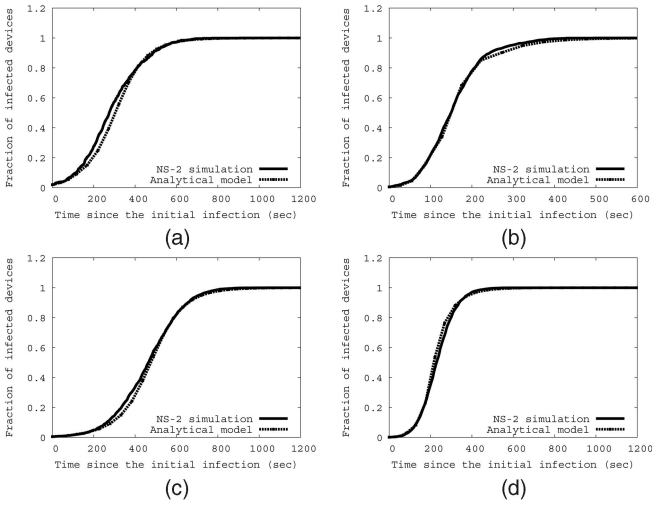

Fig. 5. Infection curves under different mobility models ($T^{to}_{inq} = 10.24$ seconds, $N^{to}_{inq} = 5$, $T^{to}_{idle} = 20$ seconds). (a) Mobility scenario M_1. (b) Mobility scenario M_2. (c) Mobility scenario M_3. (d) Mobility scenario M_4.

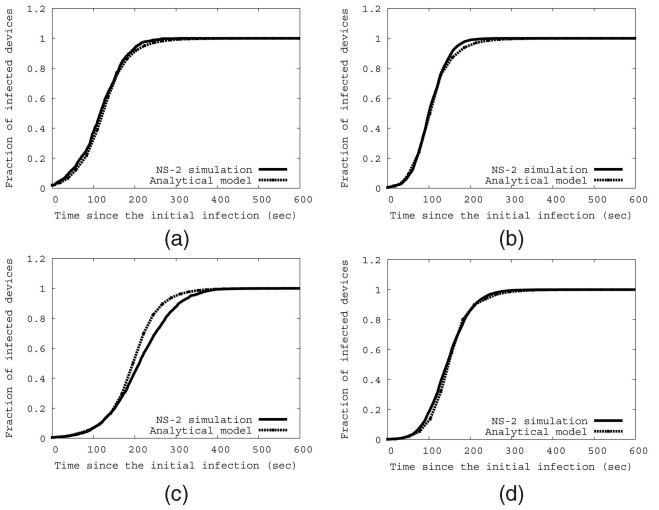

Fig. 6. Infection curves under different mobility models ($T^{to}_{inq} = 5.12$ seconds, $N^{to}_{inq} = 3$, $T^{to}_{idle} = 10$ seconds). (a) Mobility scenario M_1. (b) Mobility scenario M_2. (c) Mobility scenario M_3. (d) Mobility scenario M_4.

means how long it is before they depart from each other's radio range. Since we use the same parameterized random walk model in the four mobility scenarios, their CDFs of link durations bear a lot of similarity.

We use two sets of Bluetooth worm parameters, denoted by W_1 and W_2, respectively. In setting W_1, we have $T^{to}_{inq} = 10.24$ seconds, $N^{to}_{inq} = 5$, and $T^{to}_{idle} = 20$ seconds; in setting W_2, we have $T^{to}_{inq} = 5.12$ seconds, $N^{to}_{inq} = 3$, and $T^{to}_{idle} = 10$ seconds. Hence, we have eight scenarios in total. For each one of them, we use ns-2 to simulate 20 sample runs, in which an initial infection device is randomly chosen among the whole population. Some other parameters used in the experiments are configured as follows: $r_{ra} = 10$ m, $S_{worm} = 20{,}000$ bytes, $S_{prb} = 27$ bytes, $T^{to}_{conn} = 5.12$ seconds, $T^{to}_{prb} = 1$ second, $T^{to}_{rep} = 10$ seconds, $T^{to}_{disc} = 0.1$ second.

Figs. 5 and 6 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 further quantify the prediction errors from the model, we consider the times needed to infect 20 percent, 40 percent, 60 percent, and 80 percent of the whole population. The model, due to its variable time steps, may not produce infection states at these points. We simply use a linear model to predict the infection states between any two time steps resulting from the model.

We compute the relative errors on the times needed to infect 20 percent, 40 percent, 60 percent, and 80 percent of the whole population and the results are illustrated in Table 2. The table shows that in all the cases, the relative errors are smaller than 20 percent, and actually, in most of the cases, the relative errors are below 10 percent. Hence, the infection curves predicted by our analytical model matches well with those that are derived from a far more complicated simulation model with a detailed implementation of Bluetooth protocol.

TABLE 2
Relative Errors on the Times Needed to Infect 20 Percent, 40 Percent, 60 Percent, and 80 Percent of the Whole Population

Mobility scenario	Bluetooth worm parameter setting W_1				Bluetooth worm parameter setting W_2			
	20%	40%	60%	80%	20%	40%	60%	80%
M_1	17.14%	12.49%	8.41%	0.47%	13.60%	7.16%	4.01%	2.55%
M_2	0.84%	3.85%	0.34%	0.78%	1.78%	1.21%	2.31%	2.97%
M_3	7.94%	3.58%	1.75%	0.41%	1.48%	5.47%	9.00%	12.42%
M_4	0.20%	4.88%	5.88%	5.16%	8.71%	5.35%	1.59%	2.17%

9.2 Model-Based Prediction

As it is difficult to improve the performance of a discrete-event Bluetooth worm simulator by several orders of magnitude, our model offers hope for predicting the propagation curve of Bluetooth worms in a large population. We use the City of Los Angeles as an example. It has a population of approximately four million people and it spans about 500 square miles [10]. Suppose that every one in the City of Los Angeles carries a vulnerable Bluetooth-enabled cellular phone and walks in the city. Each person moves according to the same parameterized random walk model as in Section 9.1. To obtain the input parameters including λ_{ne}, J_{in}, and the CDF of link durations, we simulate the mobility of 200 devices moving in a 253×253 m^2 area according to the random walk model. The device density in this small area is the same as the population density in the City of Los Angeles. We then obtain $\lambda_{ne} = 0.2108$ and $J_{in} = 0.2372$. The CDF of link durations is presented in Fig. 7. We suppose that the Bluetooth worm sets its control parameters as setting W_2 in Section 9.1. We then feed these input parameters to our model and the derived propagation curve is shown in Fig. 8. We observe that the worm propagates very slowly at the initial stage, but once the density of the infected devices reaches 10 percent, the worm propagates much faster. Actually, such slow propagation at the initial stage has also been observed during the spread of Internet worms, such as Code Red [24]. The overall time taken to infect the majority of the devices is slightly less than 1 hour.

The computation of using the model to predict the Bluetooth worm propagation in the City of Los Angeles can be completed within half an hour on an ordinary desktop PC

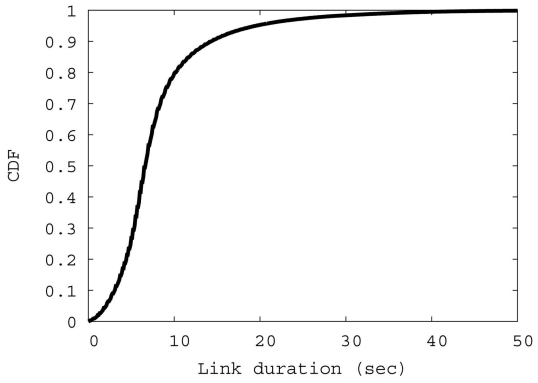

Fig. 7. CDF of link durations for the Los Angeles Case.

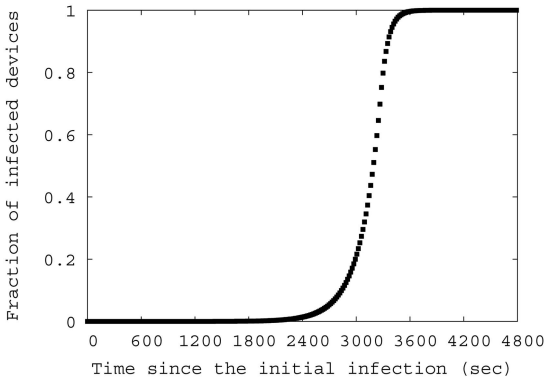

Fig. 8. Bluetooth worm propagation curve in Los Angeles.

equipped with 2-GHz CPU and 1.6-Gbyte memory. By contrast, from our experience, the current ns-2 simulator can simulate only a few thousand Bluetooth-enabled mobile devices within a reasonable amount of time. Hence, our mathematical model offers hope for predicting spreading curves of Bluetooth worm propagation in large areas without the high computational cost of discrete-event simulation.

As the mobility model we considered above is unrealistic, we are cautious to draw any practical conclusion from the results. In reality, people's mobility patterns in a big city such as Los Angeles are far more complicated than the simple random walk model used in our experiments. For instance, people tend to travel among a small set of places in their daily life, instead of randomly wandering in the whole city. To further improve the accuracy of our model, we can borrow some concepts from existing modeling work about human epidemic spreading. For example, we can divide a large area into a set of small sites or patches; it is assumed that people move according to a random walk model inside each patch, but meanwhile, people also traverse between different patches with some probabilities. Integrating the concept of multipatch mobility patterns into the Bluetooth propagation model proposed in this paper remains as our future work.

10 Further Discussions

Modeling propagation dynamics of Bluetooth worms is a challenging undertaking. Although the experimental results suggest that our Bluetooth worm model predicts the infection curves very accurately compared with a detailed Bluetooth worm simulator, our model still has a few limitations. *First*, Bluetooth is such a complicated communication protocol that our model has to rely on some simplifying assumptions. For instance, we assume that when more than one infected device contact the same victim simultaneously, only one can succeed. In reality, the victim may receive worm payloads from multiple sources in a round-robin fashion. Also, we resort to simulation to derive how long it takes for an infected device to find a new neighbor given that multiple Bluetooth devices scan for neighbors simultaneously. *Second*, it is also difficult to model human mobility precisely. In our work, we abstract the underneath mobility model into a few key statistical metrics, including average node degree, average node meeting rate, and link duration distribution. For a realistic mobility model, it is sometimes difficult to derive closed-form formula about these parameters. For instance, our experiments presented in Section 9 rely on simulations to obtain these input parameters, which can be tedious in some cases. *Third*, the Bluetooth worm propagation model presented in this paper also ignores many interesting details in a more realistic setting. When building our model, for example, we did not consider normal Bluetooth traffic, human involvement during the propagation process, and device states other than susceptible and infected (e.g., a Bluetooth device can recover from an infection). However, we believe our model, after some proper modifications, can be extended to incorporate these factors. Moreover, our model only tries to predict infection curves in an average sense. As suggested in [13], stochastic variance of the infection curves is also an important factor. Our model, for instance, does not consider the variance incurred due to the location of the initial infection.

An accurate worm propagation model can help us estimate the scale of an epidemic outbreak when it occurs in reality. Infection curves presented in Section 9 suggest that a Bluetooth worm without human involvement requires propagation times only on the order of minutes to spread itself widely. This poses a significant challenge to worm mitigation. Fortunately, existing Bluetooth worms that have been observed so far all require human intervention, which inevitably slows down their propagation. However, based on the history of Internet worms, such heydays may be over once mobile malware writers find a way to spread them automatically. From the infection curves shown in Fig. 6, we observe that the Bluetooth worm propagates faster under scenarios M_1 and M_3 than under M_2 and M_4. The difference is that the former two lead to higher device densities than the latter two. Hence, Bluetooth worms propagate more slowly in a sparse network than in a dense network. This suggests that a potential defense scheme is software diversity, which applies different implementations of Bluetooth applications to reduce the density of Bluetooth devices that are vulnerable to the same exploit.

The key focus of this paper is to build a mathematical model for predicting temporal infection curves of Bluetooth worms. From the quarantine point of view, it is also important to understand the spatial dynamics of Bluetooth worm propagation. For instance, suppose that we have identified the initial infection location and time. If we know geographically the front wave of the epidemic spreading after some propagation time, we can quarantine Bluetooth devices inside the propagation boundary. In [9], Hoh and

Gruteser have proposed a model that predicts quarantine boundary of mobile worms, but they did not consider spreading dynamics that are specific to Bluetooth worms. In the future, we plan to build a temporal-spatial Bluetooth worm propagation model that offer more insights on how we should quarantine Bluetooth worms given an outbreak.

11 Related Work

There have been substantial efforts in modeling the propagation dynamics of Internet worms in the last few years. In [17], Staniford et al. used the classical logistic function to fit the propagation curve of the Code Red I worm. In [24], Zou et al. proposed a two-factor worm model to characterize the epidemic spreading of Internet worms. Many models have also been brought forward for special types of Internet worms, such as e-mail worms [25], P2P worms [20], and so on. Our work is aimed at modeling the propagation dynamics of Bluetooth worms, the research on which is still in its infancy.

So far, there are only a few papers that study the behavior of mobile worms. Bose and Shin [4] gave a survey of existing Bluetooth viruses and worms. As a starting point of research on Bluetooth worms, simulations of the Bluetooth worm propagation have been pursued from different perspectives in [4], [22], and [18]. In [21], Yan et al. investigated the impact of mobility patterns on Bluetooth worm propagation. The work presented in this paper is extended from our earlier paper [23]; in this paper, we provide detailed analysis of packet loss rates in Bluetooth networks. Michens et al. proposed a probabilistic queueing model to model epidemic spreading in mobile environments [12]. Similar to our work in this paper, their model incorporated the notion of node mobility. Their work, however, is not specific to Bluetooth worms, and it is thus unclear how well their model characterizes the Bluetooth worm propagation.

12 Conclusions

Recent occurrences of Bluetooth worms have created growing security concerns over the data stored on mobile devices such as cellular phones and PDAs. This paper has proposed a detailed model that characterizes the propagation dynamics of Bluetooth worms. The results from the validation experiments show that our model predicts the infection curves of Bluetooth worms with high accuracy. In our future work, we plan to extend the model in this paper to a spatial-temporal model that estimates the propagation progress of Bluetooth worms in multiple patches.

References

[1] http://www.ececs.uc.edu/cdmc/ucbt/ucbt.html, 2008.
[2] http://www.isi.edu/nsnam/ns/index.html, 2008.
[3] *The Bluetooth Special Interest Group*, http://www.bluetooth.com/, 2008.
[4] A. Bose and K.G. Shin, "On Mobile Viruses Exploiting Messaging and Bluetooth Services," *Proc. Second Int'l Conf. Security and Privacy in Comm. Networks (SecureComm '06)*, Aug. 2006.
[5] J. Bray and C. Sturman, *Bluetooth: Connect without Cables.* Prentice Hall, Dec. 2000.
[6] A. Busboom, I. Herwono, M. Schuba, and G. Zavagli, "Unambiguous Device Identification and Fast Connection Setup in Bluetooth," *Proc. Int'l Conf. Next Generation Wireless Networks: Technologies, Protocols, Services and Applications (European Wireless)*, 2002.
[7] J.N. Daigle, *Queueing Theory with Applications to Packet Telecommunication.* Springer, 2005.
[8] J. Haartsen, "Bluetooth—The Universal Radio Interface for Ad Hoc, Wireless Connectivity," *Ericsson Rev.*, vol. 3, pp. 110-117, 1998.
[9] B. Hoh and M. Gruteser, "Computer Ecology: Responding to Mobile Worms with Location-Based Quarantine Boundaries," *Proc. Int'l Workshop Research Challenges in Security and Privacy for Mobile and Wireless Networks (WSPWN '06)*, Mar. 2006.
[10] *Los Angeles, California*, http://en.wikipedia.org/wiki/Los_Angeles,_California, 2008.
[11] J. Medlock and M. Kot, "Spreading Disease: Integro-Differential Equations Old and New," *Math. Biosciences*, vol. 184, no. 2, pp. 201-222, 2003.
[12] J.W. Mickens and B.D. Noble, "Modeling Epidemic Spreading in Mobile Environments," *Proc. Fourth ACM Workshop Wireless Security (WiSe '05)*, Sept. 2005.
[13] D.M. Nicol, "The Impact of Stochastic Variance on Worm Propagation and Detection," *Proc. Fourth ACM Workshop Recurring Malcode (WORM '06)*, Nov. 2006.
[14] http://www.gnu.org/software/octave/, 2008.
[15] B.S. Peterson, "Device Discovery in Frequency Hopping Wireless Ad Hoc Networks," PhD thesis, Air Force Inst. Technology, 2004.
[16] *Specification of the Bluetooth System: Core, Version 1.1*, Feb. 2001.
[17] S. Staniford, V. Paxson, and N. Weaver, "How to Own the Internet in Your Spare Time," *Proc. 11th USENIX Security Symp.*, 2002.
[18] 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," *Proc. Fourth ACM Workshop Recurring Malcode (WORM)*, 2006.
[19] C. Taylor and N. Mawston, *Bluetooth Market Doubles: CSR Still Gaining Momentum*, http://www.strategyanalytics.net/, 2005.
[20] R.W. Thommes and M.J. Coates, "Epidemiological Modelling of Peer-to-Peer Viruses and Pollution," *Proc. IEEE INFOCOM*, 2006.
[21] G. Yan, L. Cuellar, S. Eidenbenz, H.D. Flores, N. Hengartner, and V. Vu, "Bluetooth Worm Propagation: Mobility Pattern Matters," *Proc. ACM Symp. Information, Computer and Comm. Security (ASIACCS '07)*, Mar. 2007.
[22] G. Yan and S. Eidenbenz, "Bluetooth Worms: Models, Dynamics, and Defense Implications," *Proc. 22nd Ann. Computer Security Applications Conf. (ACSAC)*, 2006.
[23] G. Yan and S. Eidenbenz, "Modeling Propagation Dynamics of Bluetooth Worms," *Proc. 27th IEEE Int'l Conf. Distributed Computing Systems (ICDCS '07)*, June 2007.
[24] C.C. Zou, W. Gong, and D. Towsley, "Code Red Worm Propagation Modeling and Analysis," *Proc. Ninth ACM Conf. Computer and Comm. Security (CCS)*, 2002.
[25] C.C. Zou, D. Towsley, and W. Gong, "Email Worm Modeling and Defense," *Proc. 13th Int'l Conf. Computer Comm. and Networks (ICCCN)*, 2004.

Guanhua Yan received the PhD degree in computer science from Dartmouth College, Hanover, New Hampshire, in 2005. From 2003 to 2005, he was a visiting graduate student at the Coordinated Science Laboratory, University of Illinois, Urbana-Champaign. He is currently with the Information Sciences Group (CCS-3), Los Alamos National Laboratory. His research interests are cybersecurity, networking, and large-scale modeling and simulation techniques. He has contributed about 20 articles in these fields.

Stephan Eidenbenz received the PhD degree in computer science from the Swiss Federal Institute of Technology (ETH), Zurich, in 2000. He is currently a team leader in the Information Sciences Group (CCS-3), Los Alamos National Laboratory, where he leads the Multiscale Integrated Information and Telecommunications System (MIITS) project that models and simulates large-scale communication networks. His research interests are in wireline and wireless networking, sensor networks, selfish networking, infrastructure modeling, discrete event simulation, computational geometry, and algorithms. He has published about 50 articles in these fields.