Behavioral Detection and Containment of Proximity Malware in Delay Tolerant Networks

Wei Peng*, Feng Li†, Xukai Zou* and Jie Wu‡
*Department of Computer and Information Science
†Department of Computer, Information, and Leadership Technology
Indiana University-Purdue University, Indianapolis, Indianapolis, IN, U.S.A.
‡Department of Computer and Information Science
Temple University, Philadelphia, PA, U.S.A.

***Abstract*—With the universal presence of short-range connectivity technologies (e.g., Bluetooth and, more recently, Wi-Fi Direct) in the consumer electronics market, the delay-tolerant-network (DTN) model is becoming a viable alternative to the traditional infrastructural model. Proximity malware, which exploits the temporal dimension and distributed nature of DTNs in self-propagation, poses threats to users of new technologies. In this paper, we address the proximity malware detection and containment problem with explicit consideration for the unique characteristics of DTNs. We formulate the malware detection process as a decision problem under a general behavioral malware characterization framework. We analyze the risk associated with the decision problem and design a simple yet effective malware containment strategy, *look-ahead*, which is distributed by nature and reflects an individual node's intrinsic trade-off between staying connected (with other nodes) and staying safe (from malware). Furthermore, we consider the benefits of sharing assessments among directly connected nodes and address the challenges derived from the DTN model to such sharing in the presence of liars (i.e., malicious nodes sharing false assessments) and defectors (i.e., good nodes that have turned malicious due to malware infection). Real mobile network traces are used to verify our analysis.**

***Keywords*-delay-tolerant networks (DTNs); malware behavioral characterization; proximity malware; look-ahead λ; λ-robustness; dogmatism δ**

I. Introduction

Mobile consumer electronics permeate our lives. Laptop computers, PDAs, and more recently and prominently, smart-phones, are becoming indispensable tools for our academic, professional, and entertainment needs. These new devices are often equipped with a diverse set of non-infrastructural connectivity technologies, e.g., Infra-red, Bluetooth, and more recently, Wi-Fi Direct. With the universal presence of these short-range connectivity technologies, the communication paradigm, identified by the networking research community under the umbrella term Delay-tolerant Networks (DTNs), is becoming a viable alternative to the traditional infrastructural paradigm. Because of users' natural mobility, new information distribution applications, based on peer-to-peer contact opportunities instead of persistent connection channels among nodes, are considered to be the game changer for future network applications.

The popularity of new mobile devices (e.g., smart-phones), the adoption of common platforms (e.g., Android), and the economic incentive to spread malware (e.g., spam) combinedly exacerbate the malware problem in DTNs. Malware is a piece of malicious code which disrupts the host node's functionality and duplicates and propagates itself to other nodes via contact opportunities.

In the traditional infrastructural model, the carrier serves as a gatekeeper who can centrally monitor network abnormalities and inhibit malware propagation; moreover, the resource bottleneck for individual nodes naturally limits the impact of the malware. However, the central gatekeeper and natural limitations are absent in the DTN model. Proximity malware, which exploits the temporal dimension and distributed nature of DTNs in self-propagation, poses serious threats to users of new technologies and challenges to the networking and security research community.

A common malware detection method currently in practice is pattern matching. More concretely, a sample of malware is first reported by an infected user. The sample is analyzed by security specialists, and a pattern which (hopefully) uniquely identifies the malware is extracted; the pattern can be either code or data, binary or textual. The pattern is then used for the detection of malware[1]. The analysis and extraction often involve extensive manual labor and expertise. The overhead, the lack of generality, and high false positive rate in one round of analysis make it unsuitable for promising DTN applications on smart devices.

The quest for a better malware detection method comes to the very question of how to characterize proximity malware in DTNs. In this paper, we consider an approach to characterize proximity malware by *the behaviors of an infected node observed by other nodes in multiple rounds*. The individual observation can be *imperfect* for one round, but

[1] More sophisticated techniques exist to cope with metamorphic or compressed malware, which does not exhibit a single fixed pattern; however, the pattern matching model still applies conceptually.

infected nodes' abnormal behavior will be *distinguishable* in the long-run. Methods like pattern matching can be used in one round of observation for the behavioral characterization of proximity malware.

Instead of assuming a sophisticated malware containment capability, such as patching or self-healing [1], [2], we consider the simple capability of "cutting off communication". In other words, if a node i suspects another node j of being infected with the malware, i may cease to connect with j in the future. We want to explore how far such a simple technique can take us. Our focus is on how individual nodes make such cut-off decisions based on direct and indirect observations.

Due to the temporal dimension and distributed nature of DTNs, the major challenge faced by the proximity malware behavioral detection and containment mechanism is a decision problem: when to cut-off? This challenge can be compared with a motivating example in real life. When a person smells something burning, he or she is facing with two choices. One is to call the fire emergency service immediately; the other is to collect more evidence and to make a more informed decision later. The first choice is associated with a high cost for a possible false fire alarm, while the second choice is associated with the risk of losing the early opportunity of containing the fire.

We are facing a similar dilemma in the context of proximity malware in DTNs. Hyper-sensitivity leads to high false positive while hypo-sensitivity leads to high false negative. In this paper, we present a simple yet effective solution which reflects an individual node's intrinsic trade-off between staying connected with other nodes and staying safe from the malware. We also consider the benefits of sharing observations among nodes and address the challenges of liars and defectors derived from the DTN model.

We summarize our contributions below:

- We give a general behavioral characterization of proximity malware, which allows for functional but imperfect assessments on malware presence.
- Under the behavioral malware characterization, and with a simple cut-off malware containment strategy, we formulate the malware detection process as a decision problem. We analyze the risk associated with the decision and design a simple yet effective malware containment strategy, *lookahead*, which is distributed by nature and reflects an individual node's intrinsic trade-off between staying connected with other nodes and staying safe from malware (Section III-A).
- We consider the benefits of sharing assessments among directly connected nodes and address the challenges derived from the DTN model in the presence of liars (i.e., malicious nodes sharing false assessments) and defectors (i.e., good nodes that have turned malicious due to malware infection) (Section III-B). Real mobile network traces are used to verify our analysis and design (Section IV).

II. Problem Formulation

Consider a DTN consisting of n nodes. The *neighbors* of a node are the nodes it has contact opportunities with. Each node keeps a log, chronologically recording the neighbors it had contact with, and uses this log to estimate its contact pattern with them.

A *proximity malware* is a piece of malicious code which disrupts the host node's functionality and has a chance of duplicating itself to other nodes during inter-nodal communication; when duplication occurs, we say the other node is *infected* by the malware.

Suppose each node is capable of assessing the other party for *suspicious actions* after each encounter, resulting in a binary assessment of either *suspicious* or *non-suspicious*. An example of a suspicious action is sending a self-signed program which modifies system configurations.

At any particular time, we say a node's *nature* is either *evil* or *good* based on if it *is* or *is not* infected by the malware. We assume that the suspicious-action assessment is an *imperfect*, but *functional* indicator of malware infection: it may assess an evil node's actions as "non-suspicious" (or a good node's actions as "suspicious") at times, but most suspicious actions are correctly attributed to evil nodes.

By the *functional* assumption on the suspicious-action assessment, we characterize a node's nature by its frequency of suspicious actions. More specifically, if node i has N (pair-wise) encounters with its neighbors and s_N of them are assessed as suspicious by the other party, its *suspiciousness* S_i is defined as:

$$S_i = \lim_{N \to \infty} \frac{s_N}{N}; \tag{1}$$

we assume the existence of such a limit and therefore have $S_i \in [0, 1]$. A number $L_e \in (0, 1)$ is chosen[2] as the *line between good and evil*. Node i is deemed good if $S_i \leq L_e$ or evil if $S_i > L_e$. In other words, *we draw a fine line between good and evil, and judge a node by its deeds*.

Based on the *past* assessments, a node i can decide whether to refuse connecting with a neighbor j in the *future* (we say that "i cuts j off" if i ceases connecting with j). The reason i might refrain from cutting j off immediately upon observing a single suspicious assessment (against j) is the cost of losing a good neighbor (and the service it can provide) based on *insufficient* indicting evidence (the assessments are imperfect by our assumption).

Our questions are:

1) How shall a node make its cut-off decision?
2) Will the evil nodes eventually be cut off from the network as a *collective* result derived from nodes' *individual* decisions?

[2] We simply assume that L_e is given here. In reality, if the suspicious-action assessment is a functional discriminant feature of malware, L_e can be chosen as the (Bayesian) decision boundary which minimizes the false positive rate.

3) Will the network disintegrate from nodes' (wrong) decisions about cutting off good nodes?

III. Mechanism Design

In the following discussion, we investigate the decision process of a node i, which has k neighbors $\{n_1, n_2, \ldots, n_k\}$, against a neighbor j (with no loss of generality, we let j be n_1).

A. *Household Watch*

Let us first consider the case in which all of the evidence node i uses to make the cut-off decision against j is i's own assessment of j. Since only direct observations are used in this model, we call it *household watch.*

Let $\mathcal{A} = (a_1, a_2, \ldots, a_A)$ be the assessment sequence (a_i is either 0 for "non-suspicious" or 1 for "suspicious") in chronological order, i.e., a_1 is the oldest assessment, and a_A is the newest one.

Bayes' theorem tells us:

$$P(S_j|\mathcal{A}) \propto P(\mathcal{A}|S_j) \times P(S_j). \tag{2}$$

$P(S_j)$ encodes our *prior* belief on j's suspiciousness S_j; $P(\mathcal{A}|S_j)$ is the *likelihood* of observing the assessment sequence $\mathcal{A}$ given S_j; $P(S_j|\mathcal{A})$ is the *posterior* probability, representing the plausibility of j having a suspiciousness of S_j *given the observed assessment sequence* $\mathcal{A}$. Since the *evidence* $P(\mathcal{A})$ does not involve S_j and serves as a normalization factor in the computation, we omit it and write the quantitative relationship in the less cluttered proportional form.[3]

By adopting the Bayesian interpretation of these probabilities, we shall note that all of these quantities are conditioning on some *background knowledge* B, which, in our case, is the problem formulation in Section II, i.e., $S_j \in [0, 1]$, the existence of the threshold L_e, the infection cost C_i^e, and the cut-off cost C_i^g. Furthermore, we assume that *assessments are mutually independent*; more specifically, for any two distinct assessments a_l and a_m, we have $P(a_l|a_m, B) = P(a_l|B)$ and $P(a_m|a_l, B) = P(a_m|B)$. Due to the universal presence of B as a condition, to simplify the notation, we remove B from the symbols and write, for instance, $P(S_j|B)$ as $P(S_j)$, with the implicit understanding that S_j is conditioned on B.

By the derivation in the Appendix, we have:

$$P(S_j|\mathcal{A}) \propto S_j^{s_\mathcal{A}} (1 - S_j)^{A - s_\mathcal{A}} \tag{3}$$

and

$$\underset{S_j \in [0,1], \mathcal{A} \neq \emptyset}{\arg\max} P(S_j|\mathcal{A}) = \frac{s_\mathcal{A}}{A}, \tag{4}$$

in which $s_\mathcal{A}$ is the number of suspicious assessments in $\mathcal{A}$ (i.e., the assessments equal to 1), and $A = |\mathcal{A}|$ is the number of assessments collected so far.

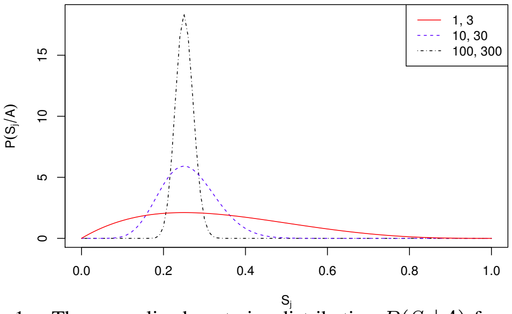

Figure 1. The normalized posterior distribution $P(S_j|\mathcal{A})$ for assessment samples with different sizes. The two numbers for each line of the legend show the number of suspicious and non-suspicious assessments, respectively. In each case, the ratio between suspicious and non-suspicious assessments is $1 : 3$. All distributions have a maximal value at $S_j = \frac{1}{1+3} = 0.25$. However, the distribution becomes more sharp with a larger sample, which corresponds to a sense of increasing certainty of the suspiciousness S_j.

Figure 1 shows the normalized posterior distributions $P(S_j|\mathcal{A})$ for assessment samples with different sizes, given by Equation 3. In each case, the ratio between suspicious and non-suspicious assessments is the same, i.e., 1:3; by Equation 4, $S_j = \frac{1}{1+3} = 0.25$ is the maximizer of $P(S_j|\mathcal{A})$, which is clearly shown in Figure 1. The distribution becomes sharper with a larger sample; this gives an intuitive sense of increasing certainty of the suspiciousness S_j.

However, what holds i back from "cutting off j immediately upon observing the first few suspicious assessments" is *not* the *exact* value of S_j, but the *risk of misjudging j's true nature and hence making the wrong cut-off decision.*

From i's perspective, after observing an assessment sequence $\mathcal{A}$, the probability that j is good is:

$$P_g(\mathcal{A}) = \int_0^{L_e} P(S_j|\mathcal{A}) \, \mathrm{d}S_j, \tag{5}$$

and the probability that j is evil is:

$$P_e(\mathcal{A}) = 1 - P_g(\mathcal{A}) = \int_{L_e}^1 P(S_j|\mathcal{A}) \, \mathrm{d}S_j. \tag{6}$$

Let $\mathcal{C} = (\int_0^1 S_j^{s_\mathcal{A}} (1 - S_j)^{A - s_\mathcal{A}})^{-1} \, \mathrm{d}S_j$ be the (probability) normalization factor in Equation 3, we have:

$$P_g(\mathcal{A}) = \mathcal{C} \int_0^{L_e} S_j^{s_\mathcal{A}} (1 - S_j)^{A - s_\mathcal{A}} \, \mathrm{d}S_j \tag{7}$$

and

$$P_e(\mathcal{A}) = \mathcal{C} \int_{L_e}^1 S_j^{s_\mathcal{A}} (1 - S_j)^{A - s_\mathcal{A}} \, \mathrm{d}S_j. \tag{8}$$

When $P_g(\mathcal{A}) \geq P_e(\mathcal{A})$, the evidence collected so far (i.e., $\mathcal{A}$) is favorable to j and leads i to believe that j is good. In contrast, when $P_g(\mathcal{A}) < P_e(\mathcal{A})$, the evidence is unfavorable to j and i needs to *decide whether to cut j off.*

The cut-off decision problem has an *asymmetric* structure in the sense that cutting j off will immediately terminate the decision process (i.e., i will cease connecting with j; no further evidence will be collected) while the opposite

[3] When we use proportional form in this paper, we have implicitly done the same thing.

decision will not. Thus, we only need to consider the decision problem when $P_g(\mathcal{A}) < P_e(\mathcal{A})$, i.e., i considers to cut j off due to unfavorable evidence against j.

To see whether i should cut j off when $P_g(\mathcal{A}) < P_e(\mathcal{A})$, we need to understand the *risk* (for i) carried with the decision of "cutting j off". To estimate the risk, i needs to *look ahead*.

More concretely, given the current assessment sequence $\mathcal{A} = (a_1, \ldots, a_A)$, the next assessment a_{A+1} (which has not been taken yet) might be either 0 (non-suspicious) or 1 (suspicious).

Let $\mathcal{A}' = (\mathcal{A}, a_{A+1})$. If $a_{A+1} = 1$, the evidence becomes even more unfavorable to j, so $P_g(\mathcal{A}') < P_g(\mathcal{A}) < P_e(\mathcal{A}) < P_e(\mathcal{A}')$[4].

If $a_{A+1} = 0$, however, by Equation 7 and 8, it might turns out that either $P_g(\mathcal{A}') < P_e(\mathcal{A}')$ or $P_g(\mathcal{A}') \geq P_e(\mathcal{A}')$.

If $P_g(\mathcal{A}') < P_e(\mathcal{A}')$, we say that i's decision of cutting j off is *one-step-ahead robust*. Otherwise, the decision is not one-step-ahead robust. If the cut-off decision is one-step-ahead robust, i knows that exposing itself to the potential danger of infection by collecting *one further assessment* on j will not change the outlook that j is evil.

Similarly, i can look *multiple* steps ahead. In fact, the number of steps i is willing to look ahead is a *parameter* of the decision process rather than a *result* of it. This parameter shows i's willingness to expose to a higher infection risk in exchange for a (potentially) lower risk of cutting off a good neighbor; in other words, it reflects i's *intrinsic* trade-off between staying connected (and hence receiving service) and keeping itself safe (from malware infection).

Definition 1 (Look-ahead λ): The *look-ahead* λ is the number of steps i is willing to look ahead before making a cut-off decision.

We can make a similar decision robustness definition for look-ahead λ.

Definition 2 (λ-robustness): At a particular point in i's cut-off decision process against j (with assessment sequence $\mathcal{A} = (a_1, \ldots, a_A)$), i's decision of cutting j off is said to be *λ-step-ahead robust*, or simply *λ-robust*, if the estimated probability of j being good $P_g(\mathcal{A}')$ is still less than that of j being evil $P_e(\mathcal{A}')$ for $\mathcal{A}' = (\mathcal{A}, a_{A+1}, \ldots, a_{A+\lambda})$, even if the next λ assessments $(a_{A+1}, \ldots, a_{A+\lambda})$ all turn out to be non-suspicious (i.e., 0).

Given the look-ahead λ, the proposed malware containment strategy is to proceed with cutting off if the decision is λ-robust and refrain from cutting off otherwise.

We wrap up the discussion on the household-watch model by illustrating how the look-ahead λ relates to i's intrinsic trade-off between staying connected and staying safe.

Since the only way to be infected by the malware is to contact an already infected node, i.e., an evil node, it is natrual to relate the risk of infection with the times of contact with a suspect evil node. Suppose the risk of infection is $R(n)$ in which n is the contact times between the node pair. One possible instantiation of $R(n)$ is $R(n) = 1 - (1-p)^n$, in which p is the (fixed) infection probability for a single encounter.

Suppose i's cost of cutting j off (and hence losing j's service) is $C_i(j)$. To be comparable with the instantiation $R(n) = 1 - (1-p)^n$, let $0 < C_i(j) < 1$. One possible instantiation of $C_i(j)$ is j's frequency in i's contact history.

If i suspects that j is evil from the observation $P_g(\mathcal{A}) < P_e(\mathcal{A}')$, λ can be chosen by $\lambda = \max\{n | R(n) \leq C_i(j)\} = \max\{n | 1 - (1-p)^n \leq C_i(j)\}$. In plain words, λ is chosen to be the maximal number of steps i is willing to look ahead in which the infection risk is less than the cost of cutting off.

B. Neighborhood Watch

Besides the direct evidence which i could use to judge j's nature, i could ask for its other neighbors' assessments on j.

This extension of evidence collection is inspired by the real-life neighborhood (crime) watch program, which encourages residents to report suspicious criminal activities in their neighborhood and to alert their neighbors. Similarly, i shares its assessments on j with its neighbors and receives their assessments in return. By Equation 3, the suspiciousness estimation does not depend on the temporal order of assessments. Therefore, neighbors may share an aggregate of their assessments (i.e., the number of suspicious and non-suspicious assessments), which is lightweight in comparison with the whole assessment sequence.

Before proceeding to further discussions, we make explicit the assumptions in the neighborhood-watch model: an evil node's behavior is *consistent* and *non-targeted.*

- **Consistency.** This rephrases the functional assumption in characterizing a node's nature by the suspiciousness (Equation 1). Only those nodes with suspiciousness higher than the threshold L_e are capable of transmitting the malware. In other words, a node cannot do the evil (transmitting the malware) and pretend to be good (maintaining a low suspiciousness).
- **Non-targetedness.** An evil node j's suspicious actions should be observable to all of its neighbors rather than a specific few. Otherwise, if j targets at i, i's other neighbors' opinions, even faithful ones, only confuse i. This assumption is vital to all evidence collecting methods which incorporate neighbors' observations.

The benefit of collecting and applying the additional evidence from neighbors is the *expanded scope*. The notion of suspiciousness, as defined in Equation 1, which characterizes the nature of j, is essentially *global*. However, the

[4] We prove this inequality in the Appendix. But the explanation here appeals to intuition and therefore serves our purpose better.

assessments i has on j are *local*. The possible divergence in the global and the local assessment statistics might lead to i's misjudgment on j's nature. Sharing assessments among a group of neighbors will help to expand the scopes of all parties and therefore lead to better estimation on the assessment target's true suspiciousness.

1) Challenges: All of the benefits in incorporating neighbors' assessments hinge upon the *faithfulness* assumption, which states that the neighbors will report their assessments without exaggeration or understatement. However, in the context of malware containment, there are two cases in which the faithfulness assumption fails to hold which we call the *liars* and the *defectors*.

Liars are those evil nodes whose purpose is to confuse other nodes by sharing false assessments. A false assessment is either a *false praise* or a *false accusation*. False praises understate evil nodes' suspiciousness while false accusations exaggerate good nodes' suspiciousness. Furthermore, a liar can fake assessments about nodes that it has never had contact with. To hide their true nature, liars may refrain from spreading the malware to its neighbors and appear to have a low suspiciousness.

Defectors are those nodes which change their nature due to malware infection. They start out as good nodes and faithfully share assessments with their neighbors; however, due to malware infection, they become evil nodes after a while. Their behaviors after the infection are under the control of the malware.

The break-down of the faithfulness assumption brings up the issue of *evidence filtering*. Two extreme, but naive evidence-filtering strategies are: 1) to trust no one and 2) to trust everyone. The former degenerates to the household-watch model with the twist of the defectors (defectors change their nature and hence their behavioral pattern); the latter leads to confusion among good nodes, which might induce undesirable cut-off decisions.

In this section, we propose an evidence-filtering strategy which, in the neighborhood-watch model, limits the negative influence of the liars on evidence fusion and detects the defectors more promptly.

Again, before proceeding to the detailed analysis below, we explicitly state the assumptions in addition to the *consistency* and *non-targetedness* assumptions,

- **Majority.** We assume the majority of the neighbors of the node under consideration are good. Otherwise, there is simply not enough evidence to estimate a node's suspiciousness in the presence of liars.

2) Evidence: We first specify the *concrete* form of *evidence* that we have been *abstractly* referring to so far.

At each encounter, a pair of nodes (which are neighbors themselves, by definition) exchange *their own aggregated* assessments on their neighbors (with the exception of the assessments on the other party) with each other. More concretely, for a pair of nodes i and j (which are neighbors themselves), let $\mathcal{N}_i$ and $\mathcal{N}_j$ be the neighbors of i and j, respectively. At each encounter, i shares with j its aggregated assessments (i.e., the number of suspicious vs. non-suspicious assessments) on n for each $n \in \mathcal{N}_i - \{j\}$; similarly, j shares with i its aggregated assessments on each neighbor $n \in \mathcal{N}_j - \{i\}$.

Since the cut-off decision only needs to be made against a neighbor, i only considers the aggregated assessments of its own neighbors $\mathcal{N}_i \cap (\mathcal{N}_j - \{i\})$ from the evidence provided by j. Moreover, since there is no superimposed trust relationship among the nodes, i and j only share *their own* aggregated assessments instead of also forwarding the ones provided by their neighbors.

3) Initialization: In our evidence filtering strategy, there is an *initialization* phase, during which the nodes acquaint themselves with their neighbors (by exchanging evidence) *without* using their neighbors' evidence in the cut-off decision process. The motivation for the initialization phase is to prevent the liars (which are the minority among the neighbors by our assumption) to dominate the cut-off decision process in the early phase.

To understand this, let us consider the following (somewhat extremal) scenario.

Suppose the first two nodes i meets are j and k. i makes a "non-suspicious" assessment on j after the (first) encounter. Then, i meets with k, during which k claims that it has 1000 suspicious and 0 non-suspicious encounters with j. If i believes k, it will reach the conclusion that j has a suspiciousness very close to 1 (since j has 1000 suspicious and 1 non-suspicious encounters with its neighbors, as far as i knows), which is very likely to lead i to cut j off.

However, without further evidence, it is equally plausible that k is lying and cutting j off is a wrong decision for i. The initialization phase provides i with the opportunity to collect further evidence without hastening to make the cut-off decision against j.

4) Evidence Aging: The presence of defectors breaks yet another assumption we have been using up to this point. Namely, a node's nature, as characterized by the suspiciousness in Equation 1, can change over *time*. More specifically, a defector starts as a good node but turns evil due to malware infection; the assessments collected before the defector's change of nature, even faithful, are misleading.

What we need here is to explicitly introduce a temporal dimension into the evidence filtering strategy. More concretely, we need to discard those assessments which are too old; in other words, the evidence is temporally decaying, which we simply call *evidence aging*.

To implement evidence aging, a node can associate a *timestamp* with each aggregated assessment it receives from its neighbor. Suppose the current time is T; only those assessments with a timestamp after $T - T_E$ for a certain temporal interval T_E, i.e., the *difference* between the aggregated assessment at the moments T and $T - T_E$, are considered relevant in the cut-off decision. We call T_E the *(evidence) aging window*.

The aging window T_E alleviates the defector problem. To see this, suppose a good node infects the malware at the moment T. Without evidence aging, all evidence before T mounts to testify that the node is good; if the amount of the prior evidence is large, it may take a long time for its neighbors to find out about its change in nature. In comparison, with evidence aging, at the moment $T + T_E$, all prior evidence expires and only those assessments after the change of nature are considered relevant.

However, we shall note that, in practice, the choice of the aging window T_E is highly context-dependent. More explicitly, though small T_E may speed up the detection of defectors, it is imperative that T_E is large enough to accommodate *enough assessments to make a sound cut-off decision*. If T_E is too small, a node simply does not have enough assessments for a λ-robust cut-off decision (Definition 2).

5) Evidence Filtering: Though evidence aging alleviates the defector problem, it does not address the liar problem. In this section, we consider the problem of *evidence filtering* which limits (if not eliminates) the negative impact of liars on the evidence quality.

In the following exposition, we consdier a node i's evidence-filtering problem on the evidence provided by its neighbor k concerning another neighbor j, i.e., whether i should use the aggregated assessment provided by k (within the evidence aging window T_E) in i's cut-off decision against j.

- During the initialization phase, i *accumulates but does not use* the aggregated assessments on j provided by its other neighbors (in particular, k). The only evidence it uses during this phase is its own assessments on j.
- After the initialization phase, i starts to incorporate filtered evidence provided by its neighbors. Now, during their encounter, i receives from k its (alleged) aggregated assessment on j. Let $\mathcal{A}$ be all of the aggregated assessments (on j) i collects so far (including its own and those from k), and let $\mathcal{A}_k$ be the aggregated assessment (on j) that i receives from k *within the evidence aging window* T_E. By Equation 7, i computes the probabilities $P_g(\mathcal{A}_k)$ and $P_g(\mathcal{A} - \mathcal{A}_k)$ that j is good with the evidence $\mathcal{A}_k$ provided by k and all of the evidence $\mathcal{A} - \mathcal{A}_k$ *not* provided by k.
- If the difference between $P_g(\mathcal{A}_k)$ and $P_g(\mathcal{A} - \mathcal{A}_k)$ is greater than a pre-specified number δ, i will *not* use the evidence provided by k in its cut-off decision against j, and the evidence provided by k within T_E will be discarded; otherwise, i will use the evidence provided by k in *this particular encounter*.

We call the number δ the *dogmatism* of i. It reflects i's cautiousness against incorporating the evidence (provided by a single neighbor) which drastically differs from the evidence provided by others.

Definition 3 (Dogmatism): The *dogmatism* δ of a node i is the evidence filtering threshold in the neighborhood-watch model. i will use the evidence $\mathcal{A}_k$ provided by its neighbor k within the evidence aging window T_E *only if* $|P_g(\mathcal{A} - \mathcal{A}_k) - P_g(\mathcal{A}_k)| \leq \delta$, in which $\mathcal{A}$ is all of the evidence that i has (including its own assessments) within T_E.

Now, we give an intuitive explanation on why, by using the dogmatism δ defined in Definition 3, the evidence-filtering strategy outlined above alleviates the liar problem.

By assumption, the majority of a node's neighbors are good. Thus, we may consider the case of a single liar. Again, by assumption, within the evidence aging window T_E, the good neighbors have enough assessments for suspiciousness estimation. Therefore, if we exclude the liar, the evidence collected from other (good) neighbors will lead to a good enough suspiciousness estimation.

By this observation, the only way that the single liar can confuse its neighbors is to claim to have sizable (fake) evidence which drastically differs from the evidence provided by the other (good) nodes. This is *exactly* the situation that the dogmatism δ is designed to resolve.

With multiple liars, by a similar analysis as above, the evidence-filtering strategy using the dogmatism δ will exclude the evidence provided by those influential liars which, if incorporated, will lead to erroneous suspiciousness estimations.

6) Summary: In the neighborhood-watch model, there is an initialization phase during which each node accumulates but does not use the evidence (aggregated assessment) provided by its neighbors. During this phase, a node only uses its own assessments in making its cut-off decision.

After the initialization phase, each node starts to incorporate filtered evidence provided by its neighbors. For a particular encounter, only if the evidence provided by the neighbor (within the evidence aging window T_E) passes the dogmatism test (Definition 3) will the evidence provided *in this particular encounter* be used in making a cut-off decision. Otherwise, all of the evidence provided by this neighbor within T_E will be ignored.

IV. Simulation

A. Datasets

We verify our design with two real mobile network traces: Haggle[3] and MIT reality[4].

The raw datasets are rich in information, many of which are irrelevant to our study, e.g., call logs and cell tower IDs for MIT reality. Therefore, we remove the irrelevant fields and only retain the nodal IDs and time-stamp for each pair-wise nodal encounter. Since the Haggle dataset has only $22,459$ entries and spans 3 days, we repeat it another 3 times

Table I
DATASET STATISTICS.

	nodes	entries	time span	avg. interval
Haggle	41	$89,836$	12 days	12 secs
MIT reality	96	$114,046$	490 days	371 secs

Table II
NEIGHBOR NATURE AND CUT-OFF DECISION COMBINATION.

	cut-off	no cut-off
evil neighbor	true positive	false negative
good neighbor	false positive	true negative

to make it into a dataset with $89,836$ entries and spanning 12 days. The statistics for the clean-up datasets are summarized in Table I.

B. Setup

Without loss of generality, we choose $L_e = 0.5$ to be the line between good and evil. For each dataset, we randomly pick 10% of the nodes to be the evil nodes and assign them with suspiciousness greater than 0.5; the rest of the nodes are deemed as good nodes and are assigned suspiciousness less than 0.5.

For a particular pairwise encounter, a random number is generated for each node; a node receives a "suspicious" assessment if its random number is greater than its suspiciousness and receives a "non-suspicious" assessment otherwise. Thus, each assessment is binary, while the frequency of "suspicious" assessments for a particular node reflects its suspiciousness.

As noted at the end of Section III-B4, the choice of the aging window T_E is context-dependent. We choose an aging window of size of 20 minutes for Haggle and 20 days for MIT reality.

C. Performance Metric

The performance comparison is based on two metrics: *detection rate* and *false positive rate*. More concretely, for each good node, by the "neighbor nature" and "cut-off decision" combination, the eventual cut-off decision made by the good nodes on its neighbors can be categorized as follows: true positive (evil neighbor gets cut off), false negative (evil neighbor stays connected), true negative (good neighbor stay connected), and false positive (good neighbor gets cut off), as shown in Table II.

For each category, we sum up all of the corresponding decisions made by the *good* nodes (it is not useful to consider the cut-off decision for the evil nodes) and obtain four counts: TP (true positive), FN (false negative), TN (true negative), and FP (false positive). Then, the detection rate DR is defined as:

$$DR = \frac{TP}{TP + FN} \times 100\%,$$

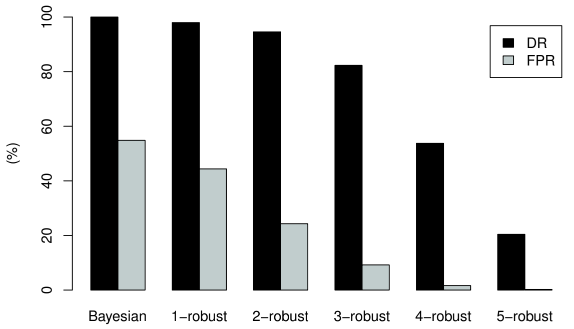

Figure 2. Bayesian decision with and without the look-ahead extension for Haggle. "Bayesian" shows the vanilla Bayesian decision; "λ-robust" shows λ-robust decision.

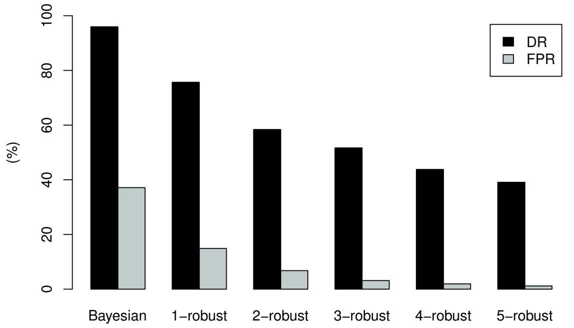

Figure 3. Bayesian decision with and without the look-ahead extension for MIT reality. "Bayesian" shows the vanilla Bayesian decision; "λ-robust" shows λ-robust decision.

and false positive rate FPR is defined as:

$$FPR = \frac{FP}{FP + TN} \times 100\%.$$

In plain words, the detection rate is the percentage of all fo the links between a good node and a bad one that rightly get cut off; false positive rate is the percentage of all of the links between a pair of good nodes that wrongly get cut off.

High detection rate and low false positive rate are desirable; depending on the context, we might put more emphasis on either. For instance, if the result of a particular round of simulation produces $TP = 121$, $FN = 26$, $FP = 118$ and $TN = 1162$, then we have $DR = 121/(121+26) \times 100\% = 82.31\%$ and $FPR = 118/(118 + 1162) \times 100\% = 9.22\%$.

D. Results

1) Look-ahead λ: We first evaluate the performance of the Bayesian decision with and without the look-ahead λ extension (i.e., λ-robust decision) in the household-watch model (i.e., no assessment sharing). The vanilla Bayesian decision does not look ahead and proceed with cutting off once $P_g(\mathcal{A})$ (Equation 7) becomes less than $P_e(\mathcal{A})$ (Equation 8).

Figures 2 and 3 show the performance comparison for the two data sets.

Both datasets show the same trend. The vanilla Bayesian decision shows both the highest detection rate and the

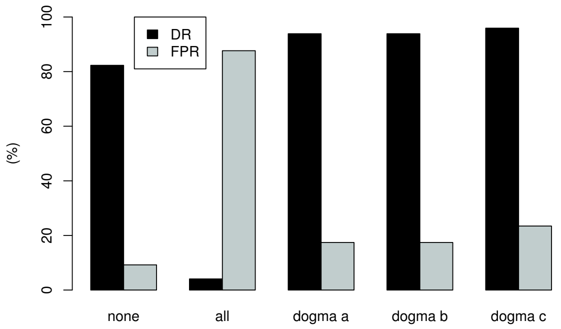

Figure 4. Effect of dogmatism δ on Haggle. Look-ahead is 3. "none" takes no indirect evidence; "all" takes all indirect evidence; "dogma" a, b, and c takes a dogmatism of 0.0001, 0.0010, and 0.0100, respectively.

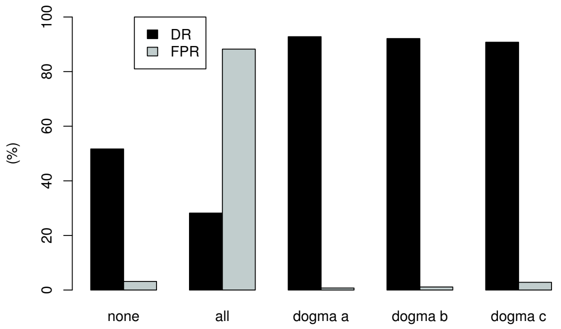

Figure 5. Effect of dogmatism δ on MIT reality. Look-ahead is 3. "none" takes no indirect evidence; "all" takes all indirect evidence; "dogma" a, b, and c takes a dogmatism of 0.0001, 0.0010, and 0.0100, respectively.

highest false positive rate; both rates drop with increasing look-ahead. However, the false detection rate drops much faster than detection rate. Indeed, for the Haggle dataset, though the vanilla Bayesian decision shows 100% detection rate, it also has a high false positive rate of 54.84%. In contrast, a 2-robust decision has a detection rate of 94.56% while having a relatively low false positive rate of 24.30%.

The results confirm the intuition that leads to the look-ahead extension to the Bayesian decision: being conservative in the cut-off decision by looking into the future helps in maintaining connections while not compromising much safety (from the malware).

In certain scenarios, trading a small decrease in detection rate for a large decrease in false positive rate is worthwhile. In those scenarios, the λ-robust decision process provides a simple yet effective method to stay connected while cutting off most connections with malware-infected nodes.

2) Dogmatism δ: We also evaluate the effect of dogmatism δ on filtering false evidence in the neighborhood-watch model. We use a look-ahead $\lambda = 3$ for this purpose.

In our study, 10% of the evil nodes play the dual role of evil-doers and liars. There are many possible liar strategies. We adopt an *exaggerated false praise/accusation* strategy. More specifically, a liar (falsely) accuses good nodes for suspicious actions and (falsely) praises other evil nodes for non-suspicious actions; besides, to exert major influence, they exaggerate the false praise/accusation by 10 times (since they are only 10% of the whole population).

We simulate the 3-robust decision with dogmatism δ taking the values 0.0001, 0.0010, and 0.0100. To see how it affects the evidence filtering, we also compare it with two other (naive) evidence filtering methods: 1) taking *none* indirect evidence, i.e., filtering all evidence; 2) taking *all* indirect evidence, i.e., filtering no evidence. The results are shown in Figures 4 and 5.

A comparison of "none" and "all" in both Figures 4 and 5 shows how the liar strategy drastically affects the performance. The "all" is rendered completely useless by taking all indirect evidence indiscriminately. In contrast, by filtering the evidence with the dogmatism test (Definition 3), the detection rate is increased (compared to "none") with a modest increase in the false positive rate. The detection rate is almost doubled in MIT reality, which is in plain sight by comparing "none" and "dogma a".

We note that, in both datasets, the one with the smallest dogmatism (0.0001) shows the best overall performance. However, we should *not* make an induction from this observation that smaller dogmatism always leads to a better performance. This is because dogmatism $\delta = 0$ basically degenerates to "none", which has inferior detection rate than the dogmatism shown here. Therefore, we conjecture that there is a dogmatism below which the performance begins to degrade. We plan to look into this in the future.

V. Related Work

Proximity malware and existing prevention schemes. A number of studies demonstrate the severe threat of proximity malware propagation. Su *et al.* collected Bluetooth scanner traces and used simulations to show that malware can effectively propagate via Bluetooth [5]. Yan *et al.* developed a Bluetooth malware model [6]. Bose and Shin showed that malware that uses both SMS/MMS and Bluetooth can propagate faster than by messaging alone [7]. Rather than assuming a sophisticated malware containment capability, such as patching or self-healing in previous works[1], [2], we base our design on quarantine and develop a decision mechanism using direct and indirect observations to deal with proximity malware.

Packet forwarding in mobile networks. In mobile networks, one cost-effective way to route packets is via short-range communication capabilities of intermittently connected smartphones [8], [9], [10]. While early work in mobile networks used a variety of simplistic random *i.i.d.* models, such as random waypoint, recent findings [11] show that these models may not be realistic. Moreover, many recent studies [11], [12], [13], based on real mobile traces, revealed that nodes' mobility showed certain social network properties. We use two real mobile network traces in our study.

Trust evaluation schemes. We base our design on the observation that trust evaluations can link past experiences with future predictions. Various frameworks[14] have been designed to model trust relationships. Three schools of thoughts emerge from studies. The first one uses a central authority, which by convention is called the trusted third party. In the second school, one global trust value is drawn and published for each node, based on other nodes' opinions of it. EigenTrust[15] is an example. The last school includes the trust management systems that allow each node to have its own view of other nodes [16]. Unlike these works, we evaluate trustworthiness on pieces of evidence rather than on individual nodes; this allows us to promptly cope with changing nature of nodes with minimum overhead.

VI. Concluding Remarks

In this paper, we address the proximity malware detection and containment problem with explicit consideration for the characteristics of DTNs. Rather than relying on a particular malware detection technique (e.g., viral pattern matching), we propose a general behavioral characterization of malware infection, which allows for functional but imperfect assessments on malware presence. Under this framework, we formulate the malware detection process as a decision problem, analyze the risk associated with the decision problem, and design a simple yet effective malware containment strategy, *lookahead*, which is distributed by nature and reflects an individual node's intrinsic trade-off between staying connected (with other nodes) and staying safe (from malware). Furthermore, we consider the benefits of sharing assessments among directly connected nodes and address the challenges derived from the DTN model in the presence of liars (i.e., malicious nodes sharing false assessments) and defectors (i.e., good nodes that have turned malicious due to malware infection). Real mobile network traces are used to verify our analysis. In prospect, the proposed behavioral malware characterization and the presented malware detection and containment method provide clearer understanding on the prevention of proximity malware in DTNs and serve as a foundation for future work along this line.

Appendix

A. Posterior $P(S_j|\mathcal{A})$

The assumption on the background knowledge B in Section III-A leads to the following observations:

- By the *principle of maximal entropy*[17] (which states that, subject to known constraints, or *testable information*, the probability assignment best representing our state of knowledge is the one which maximizes the *entropy* as defined by Shannon[18]), before obtaining any assessment, a node i which *holds no prejudice against another node* j should assign a *uniform* distribution to the prior $P(S_j)$, which is
$$P(S_j) = 1, \tag{9}$$
since by definition $S_j \in [0,1]$. Any other assignment of $P(S_j)$ reflects prejudice that i holds against j, which is *not* warranted by our assumption on the background knowledge B.
- The independence between pairs of assessments implies the *equivalence* of *batch* and *sequential* computation for $P(S_j|\mathcal{A})$. More precisely, Equation 2 shows the batch computation for $P(S_j|\mathcal{A})$. If we apply the assessment sequentially by using the posterior of the previous round as the prior of this round, we have:
$$\begin{aligned} P(S_j|\mathcal{A}) &= P(S_j|a_1,\ldots,a_A) \\ &\propto P(a_D|S_j,a_1,\ldots,a_{D-1}) \\ &\quad \times P(S_j|a_1,\ldots,a_{A-1}) \\ &= P(a_D|S_j) \times P(Sj|a_1,\ldots,a_{A-1}) \qquad (10) \\ &\cdots \\ &\propto P(S_j)\prod_{k=1}^{D} P(a_k|S_j). \end{aligned}$$

By the definition of suspiciousness S_j in Equation 1 and the independence among assessments, we have:

$$P(a_k|S_j) = \left\{ \begin{array}{rcl} S_j & \text{for} & a_k = 1 \\ 1 - S_j & \text{for} & a_k = 0 \end{array} \right. . \tag{11}$$

By Equations 9, 10, and 11, we obtain Equation 3:

$$P(S_j|\mathcal{A}) \propto S_j^{s_{\mathcal{A}}}(1-S_j)^{A-s_{\mathcal{A}}},$$

in which $s_{\mathcal{A}}$ is the number of suspicious assessments in $\mathcal{A}$ (i.e., the assessments equal to 1), and $A = |\mathcal{A}|$ is the number of assessment collected so far.

B. Posterior Maximizer

By Equation 3, we can calculate the $S_j \in [0,1]$ which maximizes $P(S_j|\mathcal{A})$. Let $a = s_{\mathcal{A}}$ and $b = A - s_{\mathcal{A}}$. If $a = 0$ and $b \neq 0$, $S_j = 0$ is the maximizer; conversely, if $a \neq 0$ and $b = 0$, $S_j = 1$ is the maximizer. If both a and b are both non-zero, let $\mathcal{C}$ be the normalization constant in Equation 3 (which is a constant for S_j), we have:

$$\begin{aligned} \frac{\mathrm{d}P(S_j|A)}{\mathrm{d}S_j} &= \frac{\mathrm{d}}{\mathrm{d}S_j}\left(\mathcal{C}S_j^a \sum_{k=0}^{b}\binom{b}{k}(-S_j)^k\right) \\ &= \mathcal{C}aS_j^{a-1}\sum_{k=0}^{b}\binom{b}{k}(-S_j)^k \\ &\quad - \mathcal{C}bS_j^{a}\sum_{k=0}^{b-1}\binom{b-1}{k}(-S_j)^k \\ &= \mathcal{C}S_j^{a-1}(1-S_j)^{b-1}\left(a(1-S_j)-bS_j\right). \end{aligned}$$

The unique $S \in (0,1)$ which makes $\frac{\mathrm{d}}{\mathrm{d}S_j}P(S_j|A) = 0$ is the S_j which satisfies $a(1-S_j) - bS_j = 0$, i.e, $S_j = \frac{a}{a+b}$.

Moreover, it maximizes $P(S_j|\mathcal{A})$, even when either a or b (but not both) is zero. Therefore, we have:

$$\operatorname*{arg\,max}_{S_j \in [0,1], \mathcal{A} \neq \emptyset} P(S_j|\mathcal{A}) = \frac{a}{a+b} = \frac{s_{\mathcal{A}}}{A},$$

which is exactly Equation 4.

C. Monotonicity of $P_g(\mathcal{A})$ and $P_e(\mathcal{A})$ on $s_{\mathcal{A}}$

By Equation 5 and 6, $P_g(\mathcal{A}) = 1 - P_e(\mathcal{A})$. Thus, we only need to prove the monotonicity of any one of them; the other follows naturally.

Here, we prove that $P_g(\mathcal{A})$ is a monotonically decreasing function on $s_{\mathcal{A}}$.

By Equation 7, let $a = s_{\mathcal{A}}$ and $b = A - s_{\mathcal{A}}$, we only need to prove:

$$\begin{aligned}
&(\int_0^1 S_j^a (1-S_j)^{b+1} \, \mathrm{d}S_j)^{-1} \int_0^{L_e} S_j^a (1-S_j)^{b+1} \, \mathrm{d}S_j \\
\geq &(\int_0^1 S_j^{a+1} (1-S_j)^{b} \, \mathrm{d}S_j)^{-1} \int_0^{L_e} S_j^{a+1} (1-S_j)^{b} \, \mathrm{d}S_j,
\end{aligned}$$

or, equivalently,

$$\begin{aligned}
&\int_0^1 S_j^{a+1} (1-S_j)^{b} \, \mathrm{d}S_j \int_0^{L_e} S_j^a (1-S_j)^{b+1} \, \mathrm{d}S_j \\
\geq &\int_0^1 S_j^a (1-S_j)^{b+1} \, \mathrm{d}S_j \int_0^{L_e} S_j^{a+1} (1-S_j)^{b} \, \mathrm{d}S_j.
\end{aligned}$$

Subtract $\int_0^{L_e} S_j^{a+1}(1-S_j)^b \, \mathrm{d}S_j \int_0^{L_e} S_j^a (1-S_j)^{b+1} \, \mathrm{d}S_j$ from both sides, we get

$$\int_{L_e}^1 S_j^{a+1} (1-S_j)^{b} \, \mathrm{d}S_j \int_0^{L_e} S_j^a (1-S_j)^{b+1} \, \mathrm{d}S_j$$

for the left side and

$$\int_0^{L_e} S_j^{a+1} (1-S_j)^{b} \, \mathrm{d}S_j \int_{L_e}^1 S_j^a (1-S_j)^{b+1} \, \mathrm{d}S_j$$

for the right side.

Finally, we have:

$$\begin{aligned}
\text{left} &= \int_{L_e}^1 S_j^{a+1} (1-S_j)^{b} \, \mathrm{d}S_j \int_0^{L_e} S_j^a (1-S_j)^{b+1} \, \mathrm{d}S_j \\
&\geq \int_{L_e}^1 L_e S_j^{a} (1-S_j)^{b} \, \mathrm{d}S_j \int_0^{L_e} (1-L_e) S_j^a (1-S_j)^{b} \, \mathrm{d}S_j \\
&= \int_0^{L_e} L_e S_j^{a} (1-S_j)^{b} \, \mathrm{d}S_j \int_{L_e}^1 (1-L_e) S_j^a (1-S_j)^{b} \, \mathrm{d}S_j \\
&\geq \int_0^{L_e} S_j^{a+1} (1-S_j)^{b} \, \mathrm{d}S_j \int_{L_e}^1 S_j^a (1-S_j)^{b+1} \, \mathrm{d}S_j = \text{right}.
\end{aligned}$$

Thus, we have proved that "$P_g(\mathcal{A})$ is a monotonically decreasing function on $s_{\mathcal{A}}$" and "$P_e(\mathcal{A})$ is a monotonically increasing function on $s_{\mathcal{A}}$".

References

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

[2] F. Li, Y. Yang, and J. Wu, "Cpmc: an efficient proximity malware coping scheme in smartphone-based mobile networks," in *Proc. of INFOCOM*. IEEE, 2010.

[3] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD data set cambridge/haggle (v. 2006-09-15)," http://crawdad.cs.dartmouth.edu/cambridge/haggle, Sep. 2006.

[4] N. Eagle and A. Pentland, "CRAWDAD data set MIT/reality (v. 2005-07-01)," http://crawdad.cs.dartmouth.edu/mit/reality, Jul. 2005.

[5] J. Su, K. Chan, A. Miklas, K. Po, A. Akhavan, S. Saroiu, E. de Lara, and A. Goel, "A preliminary investigation of worm infections in a bluetooth environment," in *Proc. of the workshop on Rapid malcode (WORM)*. ACM, 2006.

[6] G. Yan, H. Flores, L. Cuellar, N. Hengartner, S. Eidenbenz, and V. Vu, "Bluetooth worm propagation: mobility pattern matters!" in *Proc. of the Symposium on Information, Computer and Communications Security (ASIACCS)*. ACM, 2007.

[7] A. Bose and K. Shin, "On mobile viruses exploiting messaging and bluetooth services," in *Proc. of the ICST Securecomm*, 2006.

[8] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," in *Technical Report CS-200006, Duke University*, 2000.

[9] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks," in *Proc. of INFOCOM*. IEEE, 2006.

[10] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, "Delegation forwarding," in *Proc. of MobiHoc*. ACM, 2008.

[11] W. Hsu, T. Spyropoulos, K. Psounis, and A. Helmy, "Modeling time-variant user mobility in wireless mobile networks," in *Proc. of INFOCOM*. IEEE, 2007.

[12] E. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *Proc. of MobiHoc*. ACM, 2007.

[13] N. Djukic, M. Piorkowski, and M. Grossglauser, "Island hopping: Efficient mobility-assisted forwarding in partitioned networks," in *Proc. of SECON*. IEEE, 2006.

[14] A. Josang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, 2007.

[15] S. Kamvar, M. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in P2P networks." in *Proc. of WWW*. ACM, 2003.

[16] S. Buchegger and J. Boudec, "Performance analysis of the confidant protocol." in *Proc. of MobiHoc*. ACM, 2002.

[17] E. Jaynes, "Information theory and statistical mechanics. ii," *Physical Review*, vol. 108, no. 2, pp. 171–190, 1957.

[18] C. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, 1949.