Pandora: A Platform for Worm Simulations in Mobile Ad-Hoc Networks

Rahul Potharaju *rpothara@purdue.edu*
Cristina Nita-Rotaru *crisn@cs.purdue.edu*
Dept. Of Computer Science, Purdue University,
West Lafayette, IN, USA

I. Introduction

There has been significant research on the topic of malware with a majority of the research focusing on propagation modeling, detection, and application characterization. Malware spreads through computer networks by searching, attacking, and infecting remote computers automatically. Malware outbreaks such as the Slammer [10] and the Code Red [12] worms in the wired Internet have not only induced expenses in billions but also in a wealth of research [8, 10–12]. In order to understand the seriousness of future malware threats, there has been significant work into studying various Internet worm models. In epidemiological research, for instance, a number of deterministic and stochastic models have been explored that capture worm spreading dynamics [2, 3], the most popular being the SIR model [5].

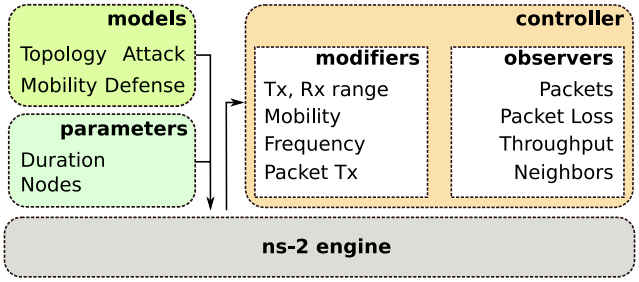
Previous approaches [11, 12] used a custom simulator to observe the effects of a worm on a wired network. However, such an approach is not very useful in a mobile environment because of a number of extra factors such as wireless interference, mobility, power constraints, coverage limitations etc. Motivated by this challenge, we present a new platform, **Pandora**, for worm simulations in mobile networks. The platform leverages ns-2's capabilities to provide: (i) ability to use various worm models, (ii) configurable node mobility patterns and runtime behavior, and (iii) addition of *healer* agents to aid in infection recovery.

II. Wired Vs. Mobile Worms

Although there are many works on how to deal with attacks on traditional "wired" networks, in this section, we highlight a few key differences that motivate the requirement for a platform exclusive to mobile networks.

II.A. Dependence on Proximity

Consider a collection of nodes distributed in a two dimensional plane which communicate using short-range radio transmission. The received radio signal strength at a device q resulting from a transmission by a device p decays with the distance between the sender and the receiver due to a combination of attenuation and shadowing effects. In addition, mobile devices may use different transmit powers to save energy. In such a scenario, existence of a wireless link from p to q does not imply that a link from q to p also exists, making the resulting communication link *asymmetric*. Thus, attacks can spread contagiously over wireless links based on proximity - analogous to real world diseases - in contrast to the any-to-any communication possible over the Internet. This renders previous models and analyses of Internet-based worm propagation not applicable as they cannot be directly mapped to mobile networks.



(a)

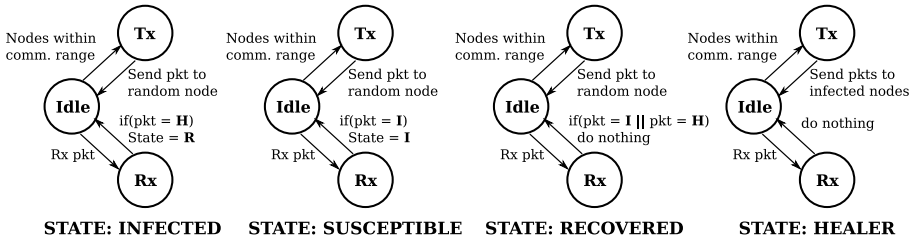
Figure 1: (a) Pandora's Design Overview

II.B. Presence of the MAC layer

In a wireless network, access to the available frequency is controlled by a coordination mechanism called the Medium Access Control [13]. The function of MAC is to ensure collision-free wireless transmissions of data packets in the network which is achieved by scheduling in time the transmissions of nearby devices in such a way that devices whose radio transmissions may interfere with each other do not get access to the wireless channel at the same time. This interference introduces spatio-temporal correlations in the dynamics of data communications in these networks which are absent in conventional wired networks.

II.C. Limitations on traffic control

Network traffic in mobile networks is difficult to control using conventional methods, in lack of "hard" enforcement points such as firewalls between the communicating nodes. Considering that today's devices



(a)

Figure 2: (a) Behavior of the different types of nodes as modeled in ns-2

connect to a wireless network on an ad-hoc basis, this could open up new opportunities for wireless worms. For instance, even if a firewall is running locally, with no open ports inbounds, a device may search outbound ports for a DHCP server in order to lease itself an IP address. Using a DHCP client exploit, the attacker can execute arbitrary code on the target device to place a worm. Machines that previously had adhoc associations become infected if they came back into the wireless range of this infected device.

III. Pandora Design Overview

The design of our simulator, Pandora, is mainly driven by the need for an exclusive platform for wireless scenarios. We use ns-2 [9] as a base engine for the simulations mainly due to its wide spread use in the research community.

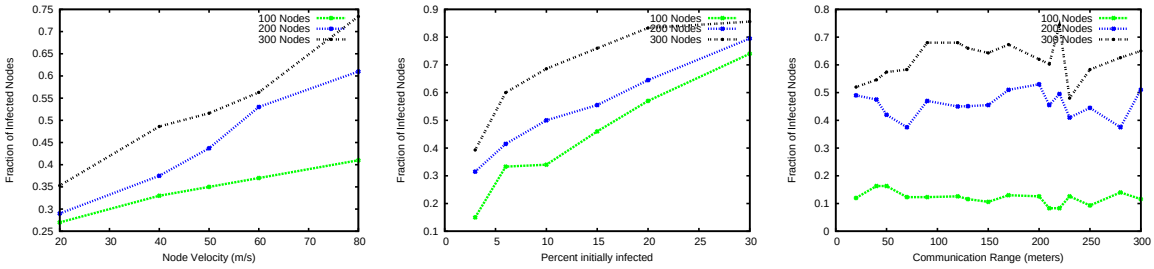
The *modeling component* allows selection of a topology (flatgrid, random etc.), mobility model (random waypoint [4] etc.), malware attack (random scanning, broadcast), and healer defense (random scanning, broadcast) strategies. In random scanning, at each simulation time step, the node first obtains a list of its current neighbors and sends a packet to a random node. In broadcast, at each simulation time step, the node obtains a list of its current neighbors and sends packets to all of them. Pandora enables the analysis of four different scenarios based on the above attack and defense strategies. For instance, one can observe the behavior of a random scanning worm with an optional addition of a healing agent that can recover nodes either through random scanning or broadcast.

The *controller* is a runtime component *i.e.*, its subcomponents are activated when the simulation starts. The *modifier sub-component* allows the user to define changes to the simulation during the runtime. For instance, one might want to evaluate the scenario where the communication range of a node changes at *t=50* in a simulation or perhaps to make a node stop moving at *t=60*. The *observers sub-component* provides an option to record various statistics about the packets being sent and received along with an option to record the instantaneous neighbors of a node during the simulation. For instance, using this metric, the average connectivity of the nodes can be calculated, which aids in predicting worm behavior in a simulation setting. Due to space constraints, we limit the description of our simulator.

IV. Pandora Evaluation Scenario

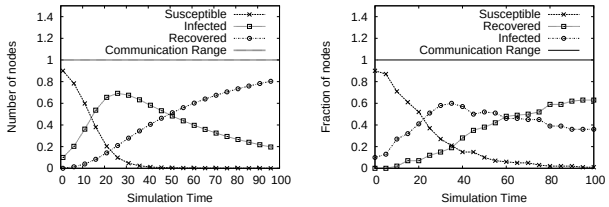
To explore the usefulness of Pandora, we are currently using it to design a solution to quarantining malware based on the SIR model. We explain briefly how we simulate the SIR model. Our simulation consists of N wireless hosts that can reach each other through a routing algorithm (AODV [1] in our initial implementation). Nodes are assumed to move in a limited region (of area A) and according to the *random waypoint mobility* model [4]. A node stays in one of the three states at any time: *susceptible*, *infectious*, or *recovered*. A node is in "recovered" state when it has been immunized against the infection which happens when the node comes into contact with a *security patch* deployed by *healer* nodes. Further, we also assume that *healer* nodes cannot be infected and once nodes have been recovered, they cannot be reinfected. Thus, the state transition of any host can be: *"susceptible - infectious - recovered"* or *"susceptible - infectious"*.

When a node becomes infected, it sends out a sequence of infection attempts during its lifetime. At each infection attempt, the *infected* node scans for neighbors within its communication range to infect and sends out a packet to a random neighbor. For the sake of brevity, the behavior of the different types of nodes is depicted in Figure 2(a). If this packet reaches a *susceptible* node, the node becomes infected. The set of neighbors will decrease if the transmission range is lower. Note that the receiver can reduce the sender's transmission range by lowering its carrier sense threshold and vice versa. To understand the effects of congestion, we allow *susceptible* nodes to select a destination (that is not necessarily one-hop away), and transmit packets to it. Because the packets may need to go through multiple-hops to reach the destination, increase in the infectious packets might result in legitimate packets getting delayed as well. In addition, observe that the recovery process depends



(a) (b) (c)

Figure 3: Time evolution of the fraction of infected nodes as obtained through Pandora for a simulation time of 100 seconds for varying number of nodes. (a) effect of node velocity, (b) effect of initial infection with nodes velocity set to 40 m/s, (c) effect of communication range with 10% of the nodes initially infected and node velocity set to 40 m/s



(a) (b)

Figure 4: (a) AMPL Simulation, (b) Pandora Simulation

on a number of other factors including the patch distribution frequency, total number of *healer* nodes and their transmission mode - unicast or broadcast.

V. Results

We compared the output of Pandora (see Fig. 4(b)) with an output obtained through modeling the SIR problem and solving it using AMPL [6] (which is an algebraic meta language for modeling optimization problems) equipped with the *snopt* solver [7] (see Fig. 4(a)). Because the basic SIR model does not consider network topology or node mobility, these results will act as a theoretical baseline. At the beginning of the wireless simulation, several hosts are initially infectious and the others are all susceptible. Whenever a node comes into the communication range of an *infected* node, the worm attempts to send out a sequence of infectious attempts. In addition, an infected host will not change its infection behavior if it is infected again by other copies of the worm. *Susceptible* nodes continue to communicate with other nodes within their communication range. *Healer* nodes remain unaffected by worm behavior and continue to patch infected nodes. We believe that the performance of Pandora bears close resemblance to the behavior predicted by the basic SIR model.

We are currently evaluating Pandora in a variety of other scenarios. Figure 3 shows the time evolution of the fraction of infected nodes as obtained through Pandora for a simulation time of 100 seconds in a field of area $1000m$ x $1000m$ for a scenario where the infected node infects a random neighbor. All values are averaged over 20 simulation runs. Figure 3(a), in particular, shows that increasing the node velocity has adverse effects with increasing number of nodes mainly due to the increased number of contacts made. Figure 3(b) shows a similar effect - with as low as 25% nodes initially infected, the infection spreads rapidly to more than 70% nodes. Figure 3(c) shows an interesting effect - that increasing the communication range has both positive and negative effects on the final fraction but results in the number saturating around a fixed percent. This is mainly due to increased packet collisions with increasing communication range. However, we are currently investigating this effect in more detail.

VI. Conclusion and Future Work

We presented an overview of our platform, Pandora, for flexible worm simulations in the context of mobile ad-hoc networks. Our preliminary performance studies reveal accurate replication of worm behavior observed from analytical models. This, combined with the wide spread popularity of ns-2, confirm the platform's potential. We see future work in the area of investigating how such a platform can be used for large-scale simulations, simulating other worm models and defense strategies.

References

[1] M. Ad, E. Royer, C. Perkins, and S. Das. Ad hoc on-demand distance vector routing. *Citeseer*, 2000.
[2] R. Anderson and R. May. *Infectious Diseases of Humans: Dynamics and Control*. Oxford University Press, 1992.
[3] H. Andersson and T. Britton. *Stochastic epidemic models and statistical analysis*. Springer Verlag, 2000.
[4] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc networks. *WCMC*, 2002.
[5] V. Capasso and G. Serio. A generalization of the kermack-mckendrick model. *Math. Biosci*, 42:41–61, 1978.
[6] R. Fourer et al. AMPL: A mathematical programming language. *CS Tech. Report, AT&T Bell Labs*, 1987.
[7] P. Gill, W. Murray, and M. Saunders. An SQP algorithm for large-scale constrained optimization. *SIAM Journal*, 2002.
[8] J. Kephart and S. White. Directed-graph epidemiological models of computer viruses. *Computation*, 1992.
[9] S. McCanne, S. Floyd et al. Network simulator ns-2, 1997.
[10] D. Moore et al. Inside the Slammer worm. *IEEE SnP*, 2003.
[11] S. Sellke, N. Shroff, and S. Bagchi. Automated Containment of worms. In *Procs. of DSN 2005*.
[12] C. Zou, W. Gong, and D. Towsley. Code Red Worm Modeling and Analysis. In *Procs. of CCS*, 2002.
[13] M.S. Gast. 802.11 Wireless Networks. *O'Reilly*, 2005.