Identifying Key Cyber-Physical Terrain*

Brian Thompson† ‡
The MITRE Corporation
7515 Colshire Drive
McLean, VA 22102
bthompson@mitre.org

Richard Harang
U.S. Army Research Lab
2800 Powder Mill Road
Adelphi, MD 20783
rich.harang@gmail.com

ABSTRACT

The high mobility of Army tactical networks, combined with their close proximity to hostile actors, elevates the risks associated with short-range network attacks. The connectivity model for such short range connections under active operations is extremely fluid, and highly dependent upon the physical space within which the element is operating, as well as the patterns of movement within that space. To handle these dependencies, we introduce the notion of "key cyber-physical terrain": locations within an area of operations that allow for effective control over the spread of proximity-dependent malware in a mobile tactical network, even as the elements of that network are in constant motion with an unpredictable pattern of node-to-node connectivity. We provide an analysis of movement models and approximation strategies for finding such critical nodes, and demonstrate via simulation that we can identify such key cyber-physical terrain quickly and effectively.

1. INTRODUCTION

1.1 Motivation

Army tactical networks in the field face a unique set of security considerations not found in either more conventional wireless networks or fixed infrastructure networks. While much previous work in analyzing the spread of malware in networks (including tactical networks) focuses on the *logical* connectivity of the graph over time, these logical connectivity paths are often dominated by long-range tactical links which introduce some degree of stability to the logical connectivity graph. However the close proximity of Army tactical networks to adversarial networks introduces new considerations in the form of *spatial* properties of the network: which units are in close proximity to each other and at what times. This form of connectivity is particularly relevant in the case of attacks that restrict themselves to short-range wireless communications – such as through 802.11 or Bluetooth network stacks – which may be more difficult to detect due to their failure to cross more conventional security boundaries or higher-resource nodes capable of fielding more sophisticated intrusion detection systems.

The short range of these attacks means that – at any given instant – the communications graph available to the malware is effectively disconnected, and it is only the mobility of the infected components over time that brings new victims into range and allows it to propagate. In addition, detection or remediation of such infections may be prohibitively difficult to perform in the field, perhaps involving detailed scans, or simply complete reimaging or replacement of any potentially compromised devices, and so only carried out at particular locations. Furthermore, while standard defensive measures are effective against known malware and minor variants, novel ("zero-day") attacks may be specifically developed for and deployed against military mobile networks. This malware may not be detectable, and so understanding how to bound the potential impact of such malware, even when not specifically alerted to its presence, is an important problem.

The notion of mobility over time, combined with the regularities in deployment and mobility of individual Army components (such as regular patrols, movement along roads and highways, and so forth), and limited capabilities to detect or remediate such attacks, leads us to our notion of *key cyber-physical terrain*: critical points in the spatio-temporal graph which can be exploited to limit the spread of short-range malware. Identifying such critical points turns out to be surprisingly difficult in practice and so we explore several methods – from simple graph-theoretic approaches to dynamical system approximations to full simulation – that capture different aspects of this problem.

1.2 Related Work

Mathematical models of virus spread were first developed in the context of biological epidemics, primarily compartmental models which assume homogeneous interaction rates within the population, such as the well-known SIR (Susceptible-Infected-Recovered) model [7] and its numerous variants. Kephart and White apply compartmental models to study the dynamics of malware spread in cyber networks, additionally using simulation to evaluate under which assumptions the compartmental models are most accurate [6]. They consider several network topologies, such as Erdos-Renyi random graphs, connected regular graphs, and sparse graphs with a high clustering coefficient. In all of these topological models they study the number of infected nodes in the population over time and how various factors affect convergence to a steady state, finding that in many cases there exists a sharp epidemic threshold.

*An extended version of this paper is available at https://arxiv.org/abs/1701.07331.

†This research was performed in part while the author was affiliated with the U.S. Army Research Laboratory.

‡The author's affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions or viewpoints expressed by the author.

IWSPA'17, March 24, 2017, Scottsdale, AZ, USA.

DOI: http://dx.doi.org/10.1145/3041008.3041015

Others have extended such models to additional network topologies and contexts. For example, Boguna et al. [1] and Dezső et al. [3] focus on epidemic models for power-law networks.

Marvel et al. propose a framework to evaluate cyber agility, but they focus on scenarios in which either a specific vulnerability or infected node is known to exist, and attempt to optimize the patching and isolation process in the network to preserve network integrity under various constraints including connectivity and power usage [8]. Huber et al. examine a similar problem using a decision support system in a small network of 10 active nodes [5]. Both cases assume malware with complete access to the network stack, which both allows longer-distance propagation than the local model we consider, and significantly increases the probability that the adversary will be detected.

Mickens et al. study device-to-device spreading of malicious software in mobile ad-hoc networks (MANETs) by explicitly modeling node mobility [9, 10]. Valler et al. develop a framework for analyzing malware spread in MANETs under the SIS (Susceptible-Infected-Susceptible) model [14]. Su et al. perform simulations using trace data drawn from real-life sampling of over 10,000 devices in a commuter train station to examine the propagation dynamics of Bluetooth worms, showing that Bluetooth worms can infect a large population of vulnerable devices relatively quickly in an urban environment [11]. On the other hand, Wang et al. model the spread of malware across networks of mobile phone users and observe that Bluetooth-based malware spreads slowly due to the short range of Bluetooth and therefore the relatively low contact rate between devices [15]. This highlights the fact that the dynamics of malware spread in MANETs varies significantly based on the properties of the underlying movement patterns. In particular, the highly-structured movement often seen in military contexts differentiates mobile tactical networks from civil MANETs and impacts the propagation of malware in such settings [13]. In this work, we explore how to leverage the structured mobility patterns of mobile tactical networks to develop more effective defense strategies, modeling tactical operations over a geographical region containing towns connected by a road network, and proposing computational methods to determine how to best allocate defensive resources.

1.3 Contributions and Outline

The main contributions of this work are:

- Model and problem formulation highlighting the need for improved security in cyber-physical tactical operations
- Three computational approaches for deciding where to place remediation stations to best control the spread of malware
- Evaluation and comparison of the three approaches

In Section 2 we describe our tactical model and propose three computational approaches to determine the optimal defender strategy. In Section 3 we perform experiments to evaluate and compare the effectiveness of the approaches. We conclude with some discussion and directions for future work in Section 4.

2. METHODS

2.1 Model and Problem Statement

We consider a scenario in which tactical units of soldiers are deployed to towns in the same geographical region, connected by a road network. As time goes on, a unit may get redeployed to another town, at which point it travels from its current town to the designated town through the road network.

Each soldier is equipped with a mobile device that facilitates short-range wireless communication, such as Bluetooth, on the battlefield. Each device regularly scans the environment for nearby friendly devices. When friendly devices come within communication range, they automatically connect, enabling data transmission.

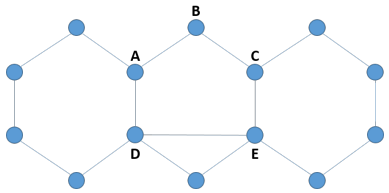

Figure 1: An example road network. Towns D and E may be the most central according to many metrics, but the best pair of towns would likely include one of $\{A, B, C\}$ and one of $\{D, E\}$.

Enemy forces may attempt to infiltrate the allied cyber network by infecting allied devices with self-propagating malware, for example by infecting the device of a captured soldier or by deploying cyber hacking teams that can infect allied devices remotely. When a soldier with an infected device comes within range of a friendly soldier with an uninfected device, the malware spreads. The malware could, for example, give the enemy access to sensitive information, or the capability to corrupt data on infected devices.

To protect their cyber network from attack, allied forces may establish some towns as *remediation zones*; any allied units entering such towns pass through a checkpoint where their devices are reset, replaced, or otherwise cleaned of malware. However, resources are limited, so judiciously choosing locations at which to establish remediation zones is critical.

Objective: Given knowledge of the road network, situational awareness of the location of enemy strongholds, and an assessment of remediation resources currently available, determine the optimal placement of remediation zones to minimize the fraction of devices that are infected with malware.

Below, we explore three approaches to addressing this problem: centrality analysis, dynamical systems, and agent-based modeling.

2.2 Centrality Analysis

In the centrality-based approach, we represent the road network as a graph and use network centrality analysis to identify the towns at which to establish remediation zones. The intuition is that the most central vertices are the most important, either visited most frequently or located at important junctures. Let G be an undirected graph with vertex set $V(G) = \{v_1, \ldots, v_n\}$ corresponding to the towns and edge set $E(G) \subseteq \binom{V}{2}$ corresponding to the roads. A centrality metric assigns weights to the vertices in a graph based on how central they are. For a given centrality metric μ, we let $\mu_G : V(G) \to \mathbb{R}$ denote the mapping from the vertices of G to their corresponding values under the centrality metric.

We consider two common centrality metrics:

- PageRank centrality [2] - favors vertices with connections to other well-connected vertices
- Betweenness centrality [4] - favors vertices that lie on shortest paths between many other pairs of vertices

The choice of metric may be context-specific. For example, PageRank centrality has a natural correspondence with the frequency of vertices being visited under a mobility model where units perform a random walk on the road network, i.e. choosing the next town to visit uniformly at random from the set of neighboring towns. On the other hand, Betweenness centrality naturally corresponds with vertex frequency under a random waypoint mobility model, i.e. where units choose a town uniformly at random from the set of all towns and then traverse a shortest path to get there.

Algorithm 1 Centrality-based algorithm

Input: A graph G, a centrality metric μ, and an integer $k \geq 1$.

Output: A subset $V_R^* \subseteq V(G)$ of size k corresponding to the towns at which to establish remediation zones.

Initialize $G_0 := G$

For $1 \leq i \leq k$:

- Compute $\mu_{i-1} = \mu_{G_{i-1}}$
- Set $v_i := \max_{v \in V(G_{i-1})} \mu_{G_{i-1}}(v)$
- Set $G_i := G_{i-1} - v_i$

Return $V_R^* = \{v_i : 1 \leq i \leq k\}$

If there are only resources for a single remediation zone, centrality metrics offer a straight-forward way to choose where to place it: at the town corresponding to the vertex with the highest centrality score. If there are resources for $k > 1$ remediation zones, however, the natural solution of choosing the towns corresponding to the vertices with the k highest centrality values may not be a very good strategy. For example, consider the graph in Figure 1 with $k = 2$. Vertices D and E have the top two centrality scores for PageRank and Betweenness centrality, yet a better strategy would likely be to choose one vertex in $\{A, B, C\}$ and one vertex in $\{D, E\}$ because that would cut the graph into two similarly-sized subgraphs between which malware could not propagate.

To address this problem, we present an iterative algorithm, described in Algorithm 1. The algorithm computes centrality scores, deletes the vertex with the highest score, and repeats until k vertices have been deleted. The remediation zones should be placed at the towns corresponding to the deleted vertices.

However, there are still times when this does not produce the desired behavior. For example, there is no clear way of incorporating situational awareness of which towns are controlled by the enemy and therefore most likely that allied devices will get infected with malware. This is a severe drawback of any centrality-based algorithm, since they are based solely on the network topology and are not sensitive to the locations of enemy strongholds. Next we consider an approach from the field of dynamical systems that addresses this problem.

2.3 Dynamical Systems

In the dynamical systems approach, we begin by modeling the movement of each unit as a continuous-time Markov chain, where states correspond to towns and roads, and transitions correspond to changes in location in response to new tactical orders. When deployed at a town, a unit stays there for some *deployment time* until it receives new orders to travel to a neighboring town. When it receives the travel order, it transitions to the road between the two towns, and remains there for the duration of the *travel time*, which may depend on the distance, terrain, weather conditions, etc.

Let S_i denote the state corresponding to town i, and let $S_{i,j}$ denote the state corresponding to traversing a road from town i to town j. We define the average wait time w_i for state S_i to be equal to the average deployment time for town i. We define the average wait time $w_{i,j}$ for state $S_{i,j}$ to be equal to the average travel time from town i to town j.

There are two types of transitions: from a state S_i to a state $S_{i,j}$, corresponding to departure from town i along a road to town j; and from a state $S_{i,j}$ to a state S_j, corresponding to arrival at town j along a road from town i. Assuming that units leaving a town have the same likelihood of traveling to each of the neighboring towns, the transition rates are as follows:

$$(\forall\, i,j : (v_i, v_j) \in E(G)) \qquad T(S_i, S_{i,j}) = \frac{1}{w_i \cdot d_i}$$

$$(\forall\, i,j : (v_i, v_j) \in E(G)) \qquad T(S_{i,j}, S_j) = \frac{1}{w_{i,j}}$$

The Markov chain for an example scenario is illustrated in Figure 2.

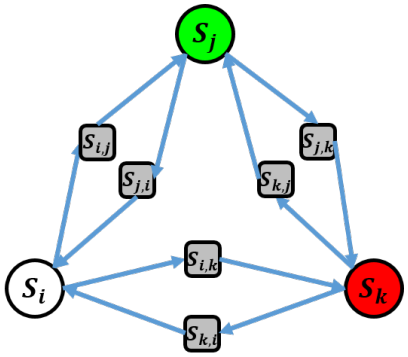

Figure 2: Markov chain for simple example scenario

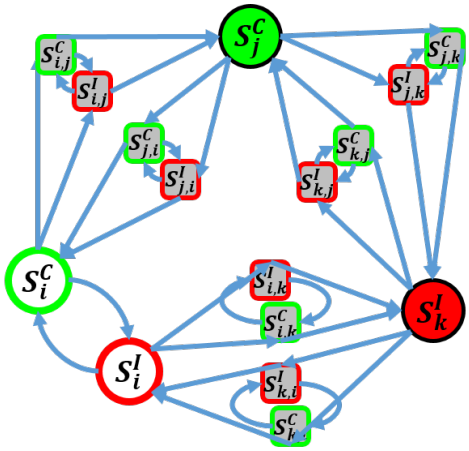

Figure 3: Modified Markov chain for simple example scenario

Next, we describe the movement of all units collectively using a compartmental model corresponding to the Markov chain described above, capturing the fraction of units in each state and the flows between them with a set of differential equations. These equations can then be used to solve for the fraction of units in each state at equilibrium, indicating which towns will be most frequently visited, which could be good candidates for remediation zones. Due to space constraints, we refer the reader to [12] for a more detailed technical description of this approach.

However, the same problems encountered under the centrality-based approach above still remain: choosing a set of towns based on each town's individual value may not yield the best results collectively; and we have not leveraged knowledge of the location of enemy strongholds.

To address these problems, we consider a modified Markov model that splits each previous state into two dual states, corresponding to whether the unit is clean or infected. We denote this by states S_i^C, S_i^I, $S_{i,j}^C$, and $S_{i,j}^I$. Let V_I denote the set of vertices corresponding to enemy strongholds, and let V_R denote the set of vertices corresponding to towns with remediation stations, with $V_R \cap V_I = \varnothing$.

Algorithm 2 Monte Carlo algorithm

Input: A graph G representing towns connected by a road network, a subset of vertices $V_I \subseteq V(G)$ corresponding to enemy strongholds, a function $f : 2^V(G) \to \mathbb{R}$ mapping vertex subsets V_R to the resulting fraction of infected units if remediation zones were established at the corresponding towns, an integer s indicating how many random samples to take at the corresponding point in the algorithm, and an integer $k \geq 1$ indicating the number of remediation zones for which resources are available.

Output: A subset $V_R^* \subseteq V(G)$ of size k corresponding to the towns at which to establish remediation zones.

1. Initialize $V_R^{(0)} := \varnothing$
2. Initialize $V_R^* := \varnothing$
3. Initialize $f^* := 0$
4. For $1 \leq i \leq k$:
   - For $v \in V(G) - V_R^{(i-1)}$:
     - Initialize $F^{\text{sum}}[v] = 0$
     - Initialize $F^{\text{count}}[v] = 0$
   - for $1 \leq j \leq s$
     - Randomly select a subset $V' \subseteq V(G) - V_R^{(i-1)}$ of size $k - (i-1)$
     - Set $V'' := V_R^{(i-1)} \cup V'$
     - For $v \in V'$:
       - Update $F^{\text{sum}}[v] := F^{\text{sum}}[v] + f(V'')$
       - Update $F^{\text{count}}[v] := F^{\text{count}}[v] + 1$
     - If $f(V'') < f^*$:
       - Set $V_R^* := V''$
       - Set $f^* := f(V'')$
   - Set $v_i := \min_{v \in V_R^* - V_R^{(i-1)}} \frac{F^{\text{sum}}[v]}{F^{\text{count}}[v]}$
   - Set $V_R^{(i)} := V_R^{(i-1)} \cup \{v_i\}$
5. Return $V_R^* = V_R^{(k)} = \{v_i : 1 \leq i \leq k\}$

We assume that any unit entering a town in V_I will become infected with malware, and any unit entering a town in V_R will become clean. In addition, we assume that a clean unit entering a town with at least one infected unit will become infected, and also that a clean unit traversing a road with at least one infected unit traveling in the opposite direction will become infected. The goal is to determine the optimal set V_R of size k, given V_I. The modified Markov chain for the example scenario is illustrated in Figure 3.

Similarly to above, the movement of all units collectively can be captured by a set of differential equations, which, given V_I and V_R, can be solved efficiently for the equilibrium fraction of units in each state (see [12] for details). In this modified model, however, we have a way of quantifying the effectiveness of a proposed solution: the total fraction of infected units at equilibrium. The remaining challenge is in finding the set V_R that minimizes that value.

If n, the number of towns, and k, the desired number of remediation stations, are small, then an exhaustive search may be feasible. Otherwise, we propose two algorithms: one which simply samples from the space of possible solutions and chooses whichever solution gives the best result; and one which is based on more sophisticated Monte Carlo methods, described in Algorithm 2.

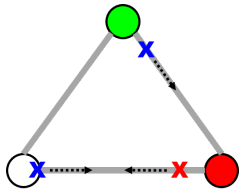

Figure 4: Agent-based model for simple example scenario

The dynamical systems approach addresses some of the problems with the centrality-based approach; in particular, it explicitly represents the presence of enemy strongholds. However, it is less flexible than the centrality-based approach in accommodating different mobility models; the Markov property is fine for modeling a random walk on the road network, but cannot easily represent multi-hop paths such as traversing the shortest path between two towns. In addition, the model makes several simplifying assumptions that could compromise the accuracy of the results.

Next, we present an approach that gives greater flexibility in modeling and also permits a higher degree of realism.

2.4 Agent-based Modeling

In this approach, we develop an agent-based model to represent the movement of and interactions between tactical units. The agents are the tactical units, each represented by a `Unit` object. The environment consists of `Town` objects, represented by circular regions, and `Road` objects, each connecting two `Towns`. `Towns` can be ally-controlled, enemy-controlled, or neutral. This approach can accommodate many different mobility models, including both the random walk and the random waypoint models for traversing the road network. An agent-based model for a simple example scenario is illustrated in Figure 4.

When a `Unit` is deployed at or passes through an enemy-controlled `Town`, we make the worst-case assumption that the enemy will be able to infect at least one of the soldiers' devices, and that relatively soon thereafter the malware will spread to the whole `Unit` as the soldiers interact with one another. In addition, we assume that if two `Units` are deployed to the same `Town` simultaneously, or if one `Unit` passes through the `Town` where the other is deployed, or if two `Units` pass each other on a `Road`, there will be at least some contact between the `Units`; therefore, if one of them is infected, the other will also become infected.

As before, our goal is to determine the set of vertices V_R at which to place remediation zones so as to minimize the fraction of infected units. An obvious way to evaluate a proposed solution, then, is to run the simulation for a period of time and then count how many of the units are infected. Because of random variation, the result should be averaged over multiple trials. As with the dynamical systems approach, the remaining challenge is in finding the set V_R that minimizes that value. For this, we propose using either the simple random sampling method or the same Monte Carlo algorithm proposed above, Algorithm 2, substituting the results of the agent-based simulation for the solution to the dynamical system when defining the function f.

The agent-based modeling approach has higher fidelity and expressiveness than the other approaches, but can also be more computationally expensive. In the following section, we evaluate both the effectiveness and computational efficiency of the three methods in determining the placement of remediation stations to best limit the spread of malware.

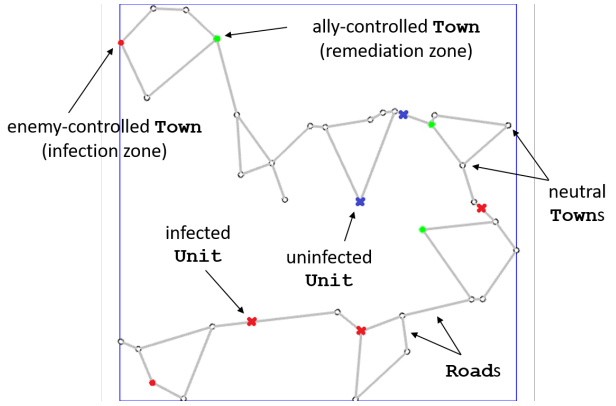

Figure 5: Labeled screenshot of simulation in Repast Simphony

3. EVALUATION

We now perform experiments to evaluate and compare the performance of the three approaches. Since the agent-based model has the highest fidelity of the three approaches, we use it as an evaluative metric to compare different recommended placement strategies. Given that, one might expect that the agent-based modeling approach would trivially yield the best results. However, as we will see, due to computational limitations this is not necessarily true.

Before we proceed with the experiments, we provide details of our implementation.

3.1 Implementation and Experimental Setup

All three of the approaches are implemented in Java. Solving systems of equations for the dynamical systems approach was done using the JAMA linear algebra package. Simulations of the agent-based model can be visualized using Repast Simphony, a Java-based agent-based modeling and simulation environment. Experiments are conducted on an Intel Core i7 processor operating at 2.40 GHz with 16 GB of memory running Windows 10.

Figure 5 gives a screenshot of an example run of the agent-based simulation. The black circles represent neutral `Towns`, the red circles represent `Towns` under enemy control, and the green circles represent `Towns` under allied control. `Units` are depicted by a red 'X' when infected and a black 'X' when uninfected.

For the experiments presented here, we consider five tactical units operating in a geographical area consisting of 35 towns connected by a road network. Units move at a speed of 10 m/s, and deployments last 2 hours. We vary the number of infected towns and remediation zones. Simulations were run for 10,000 time steps. Results were averaged over 20 independent trials.

3.2 Results

The results of our experiments are shown in Table 1. For a baseline, we also record the average fraction of infected units when remediation zones are chosen uniformly at random. Due to space constraints, only the results for Random Waypoint are shown.

The best performers under the Random Walk mobility model were the Iterative PageRank Centrality method and the Dynamical Systems methods (either using simple random sampling or the more sophisticated Monte Carlo algorithm). The other methods performed significantly worse than those, and comparably to one another, sometimes not even matching the results of the uniform random baseline.

Under the more realistic Random Waypoint mobility model, Iterative PageRank was the clear winner, performing even better than under Random Walk. This was surprising, and ran counter to our intuition that PageRank would perform best under Random Walk because it has a natural correspondence to walks on graphs. Similarly, we were surprised that Betweenness centrality did not perform better under the Random Waypoint model, given its natural correspondence to graph paths. With the exception of the Iterative Betweenness method, all methods out-performed the baseline.

We note that we configured the ABM method to use fewer MC samples than the Dynamical Systems method (10 instead of 100) to keep its runtime comparable, since it does an evaluation over 10 sample trials for each candidate strategy rather than just solving a system of equations once. We suspect that the small sample size resulted in a high variance across trials, which could explain why the ABM approach performed so poorly.

Runtimes are shown in Table 2. For this setting of the parameters, the centrality algorithms each ran in about 12 seconds, Dynamical Systems ran in about 25 seconds, ABM with MC in about 60 seconds, and ABM with random sampling in about 75 seconds.

4. CONCLUSIONS

We have proposed the notion of "key cyber-physical terrain" to describe the risk posed by short-range wireless attacks under the dynamic connectivity graphs of field operations: specific physical locations at which mobile devices can be examined and remediated to minimize the ability of an adversary to maintain a presence on the mobile network. As the exact solution to this problem is computationally intractable, we have also proposed three approximate methods of solving the associated minimization problem – centrality metrics, dynamical systems, and agent-based modeling – under two different models of unit mobility. Some of their pros and cons are listed in Table 3.

Our results suggest that the problem of malware propagating through short-range wireless communications is potentially quite significant, with a high prevalence of malware persisting on the network, even when the remediation zones are placed strategically in response to the locations of the infection zones. It is also worth noting that simple algorithms based on network centrality metrics, in particular PageRank centrality, can match and even outperform more complex approximations, even under the more realistic Random Waypoint mobility model. In either case, we obtain solutions reasonably quickly, with average runtimes of about 1 minute even for our most computationally intensive approach. We note, however, that the variance for the agent-based modeling is relatively high, as the total number of potential trajectories through the combinatorial number of remediation zones is prohibitively large; results could be improved by increasing sample sizes in the Monte Carlo algorithm, at the cost of longer runtimes, which would be further exacerbated as the problem scales up. Methods to mitigate this variance will be explored in future work.

Our current results show that both our centrality and dynamical systems methods can approach the accuracy of the more computationally intensive agent-based modeling approach under the mobility models used. On the other hand, the agent-based approach provides much greater flexibility for representing more sophisticated and realistic movement patterns and higher-fidelity models. For example, instead of random deployments and shortest-path traversals, simulations could be performed using real-world maps and scenarios, and paths may intentionally avoid locations of enemy strongholds. An alternative problem formulation could allow strategies to simultaneously define the traversal paths between pairs of towns as well as the locations of the remediation zones. This will be explored in future work, as well as extensions to our tactical model

Table 1: Experimental results under the Random Waypoint mobility model, in terms of the fraction of units infected, averaged over 20 trials

# Inf Zones	# Rmd Zones	Betweenness Top-k	Betweenness Iter	PageRank Top-k	PageRank Iter	Dynam Sys Basic	Dynam Sys MC	Agent-based Basic	Agent-based MC	Uniform Random
5	0	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
5	1	0.743	0.793	0.852	0.743	0.779	0.779	0.878	0.878	0.940
5	3	0.543	0.673	0.640	0.446	0.526	0.556	0.717	0.621	0.758
5	5	0.461	0.581	0.520	0.345	0.403	0.395	0.561	0.489	0.624
3	5	0.321	0.474	0.385	0.203	0.289	0.238	0.373	0.312	0.455
1	5	0.112	0.252	0.160	0.068	0.122	0.096	0.148	0.088	0.214
0	5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 2: Runtimes for the different approaches, in seconds, averaged over 20 trials

	Betweenness Top-k	Betweenness Iter	PageRank Top-k	PageRank Iter	Dynam Sys Basic	Dynam Sys MC	Agent-based Basic	Agent-based MC	Uniform Random
Runtimes	0.209	0.209	0.209	0.208	0.414	0.418	1.247	1.040	1.247

Approach	Pros	Cons
Centrality metrics	can be efficient, choice of metric can accommodate different contexts or mobility patterns	does not capture travel times, cannot specify enemy towns, not good for multi-site selection
Dynamical systems	efficient, good for multi-site selection	assumes Random Walk mobility pattern because of Markov property
Agent-based modeling	very flexible and expressive, most realistic, good for multi-site selection	not as efficient as other approaches

Table 3: Pros and cons of the three approaches

in which enemy infection regions as well as remediation zones may be dynamic or increase in number.

5. REFERENCES

[1] M. Boguná, R. Pastor-Satorras, A. Vespignani, et al. Epidemic spreading in complex networks with degree correlations. In *Proceedings of the XVIII Sitges Conference on Statistical Mechanics, Lecture Notes in Physics, Springer, Berlin*, 2003.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1):107 – 117, 1998.

[3] Z. Dezső and A.-L. Barabási. Halting viruses in scale-free networks. *Physical Review E*, 65(5):055103, 2002.

[4] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.

[5] C. Huber, P. McDaniel, S. E. Brown, and L. Marvel. Cyber fighter associate: A decision support system for cyber agility. In *2016 Annual Conference on Information Science and Systems (CISS)*, pages 198–203. IEEE, 2016.

[6] J. Kephart and S. White. Directed-graph epidemiological models of computer viruses. In *Research in Security and Privacy, 1991. Proceedings., 1991 IEEE Computer Society Symposium on*, pages 343–359, May 1991.

[7] W. O. Kermack and A. G. McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 115(772):700–721, 1927.

[8] L. M. Marvel, S. Brown, I. Neamtiu, R. Harang, D. Harman, and B. Henz. A framework to evaluate cyber agility. In *Military Communications Conference, MILCOM 2015-2015 IEEE*, pages 31–36. IEEE, 2015.

[9] J. W. Mickens and B. D. Noble. Modeling epidemic spreading in mobile environments. In *Proceedings of the 4th ACM Workshop on Wireless Security*, WiSe '05, pages 77–86, New York, NY, USA, 2005. ACM.

[10] J. W. Mickens and B. D. Noble. Analytical models for epidemics in mobile networks. In *Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2007)*, pages 77–77. IEEE, 2007.

[11] J. Su, K. K. Chan, A. G. Miklas, K. Po, A. Akhavan, S. Saroiu, E. de Lara, and A. Goel. A preliminary investigation of worm infections in a bluetooth environment. In *Proceedings of the 4th ACM Workshop on Recurring Malcode*, pages 9–16. ACM, 2006.

[12] B. Thompson and R. Harang. Identifying key cyber-physical terrain (extended version). https://arxiv.org/abs/1701.07331, January 2017.

[13] B. Thompson and J. Morris-King. The impact of hierarchy on bluetooth-based malware spread in mobile tactical networks. In *Proceedings of the Summer Computer Simulation Conference*, SCSC '16, pages 34:1–34:7, San Diego, CA, USA, 2016. Society for Computer Simulation International.

[14] N. C. Valler, B. A. Prakash, H. Tong, M. Faloutsos, and C. Faloutsos. Epidemic spread in mobile ad hoc networks: Determining the tipping point. In *Proceedings of the 10th International IFIP TC 6 Conference on Networking - Volume Part I*, NETWORKING'11, pages 266–280, Berlin, Heidelberg, 2011. Springer-Verlag.

[15] P. Wang, M. C. González, C. A. Hidalgo, and A.-L. Barabási. Understanding the spreading patterns of mobile phone viruses. *Science*, 324(5930):1071–1076, 2009.