Wireless Pers Commun
DOI 10.1007/s11277-012-0528-3

On Survivability of Mobile Cyber Physical Systems with Intrusion Detection

Robert Mitchell · Ing-Ray Chen

Abstract In this paper we address the survivability issue of a mobile cyber physical system (MCPS) comprising sensor-carried human actors, vehicles, or robots assembled together for executing a specific mission in battlefield or emergency response situations. We develop a mathematical model to assess the survivability property of a MCPS subject to energy exhaustion and security failure. Our model-based analysis reveals the optimal design setting for invoking intrusion detection to best balance energy conservation versus intrusion tolerance for achieving high survivability. We test the effectiveness of our approach with a dynamic voting-based intrusion detection technique leveraging sensing and ranging capabilities of mobile nodes in the MCPS and demonstrate its validity with simulation validation.

Keywords Survivability · Intrusion detection · Security attacks · Mobile cyber physical systems · Model-based analysis

1 Introduction

The increased presence of cyber physical systems (CPSs) in the world motivates in-depth research on survivability to sustain potential malicious attacks. In this paper we address the survivability issue of a mobile cyber physical system (MCPS) comprising sensor-carried human actors, vehicles, or robots assembled together for executing a specific mission in battlefield or emergency response situations.

MCPSs pose unique challenges to intrusion detection due to mobility, resource constraints (which discourage excessive energy use), scale, and heterogeneity. System designers use *distributed* intrusion detection applications because they are fault tolerant. Distributed *location-based* intrusion detection applications benefit from a richer data set from geometrically

R. Mitchell · I.-R. Chen (✉)
Department of Computer Science, Virginia Tech, Falls Church, VA, USA
e-mail: irchen@vt.edu

R. Mitchell
e-mail: rrmitche@vt.edu

Published online: 04 February 2012

diverse nodes. A MCPS often operates in a rough environment wherein energy replenishment is not possible and nodes may be compromised (or captured) at times. Thus, high survivability to sustain malicious attacks and energy consumption is of the utmost importance.

In the literature, [1] provides an excellent survey of security challenges of CPSs. [3] provides a theoretical treatment of anomaly detection for discrete sequences. [10] surveys survivability in mobile ad hoc networks designs. The authors advocate integrating intrusion prevention, intrusion detection and intrusion tolerance. They also promote decentralization. They distinguish two types of dynamic/responsive intrusion tolerance, namely, recovery and adaptation, and conclude that intrusion tolerance, particularly adaptation, comprises a significant gap in the literature. The intrusion tolerance literature falls into two categories: static/structural and dynamic/responsive. Examples of static/structural intrusion tolerance approaches are component redundancy, path redundancy, data redundancy, decentralization and threshold cryptography. Examples of dynamic/responsive intrusion tolerance approaches are self-organization, dynamic routing, backward recovery and forward recovery.

Common themes in the CPS intrusion detection/tolerance literature are: application-specific intrusion detection, dynamic/responsive intrusion tolerance, self-organization and decentralization. [2] advocates application-specific anomaly-based intrusion detection for CPSs and proposes dynamic/responsive versus static/structural intrusion tolerance. [15] discusses challenges of constructing intrusion detection systems for mobile ad hoc networks and wireless sensor networks and surveys existing intrusion detection techniques. [8] discusses feature selection and efficiency of wireless intrusion detection systems. [17] discusses intrusion detection techniques specifically for wireless sensor networks. [13] identifies power management as one critical CPS concern and proposes leveraging mobility patterns to address it. To the best of our knowledge no approach has been proposed to best balance power management and intrusion tolerance for achieving high survivability.

We adopt a top down approach to maximize the survivability of a MCPS. We consider two failure conditions for a MCPS: energy exhaustion and security fault. Maximizing the lifetime until energy exhaustion is equivalent to minimizing energy use. We consider distributed intrusion detection in the form of dynamic majority voting with the detection interval and the number of detectors being dynamically adjusted to adapt to environment changes. The use of dynamic majority voting also can cope with collusion for achieving a certain degree of intrusion intrusion. Four events induce security faults: A per-node false negative means that a single intrusion detector misidentifies a bad node as a good node. On the other hand, if a single intrusion detector misidentifies a good node as a bad node, this is a per-node false positive. A system-wide false negative occurs when a pool of intrusion detectors reaches an incorrect majority decision that a bad node is good. On the other hand, if a pool of intrusion detectors reaches an incorrect majority decision that a good node is bad, this is a system-wide false positive. Applying more resources to intrusion detection will hasten energy exhaustion but will delay security fault. Withholding resources from intrusion detection will extend a MCPS lifetime at the expense of less security.

Our methodology for MCPS survivability assessment is model-based analysis with simulation validation. Specifically, we develop a mathematical model to assess the survivability property of a MCPS equipped with dynamic voting-based intrusion detection capabilities subject to energy exhaustion and security failure. The mathematical model reveals optimal design settings for invoking dynamic voting-based intrusion detection to best balance energy conservation versus intrusion tolerance for achieving high survivability, when given a set of parameter values characterizing the operational environment and network conditions. We conduct extensive simulation to validate the analytical results obtained.

The rest of the paper is organized as follows: Sect. 2 gives the system model and reference MCPS configuration. Section 3 develops a mathematical model based on Stochastic Petri Nets [4,14] for theoretical analysis. Section 4 discusses the parameterization process for the reference MCPS and presents numerical data with physical interpretations given. Section 5 discusses the simulation tool and environment, parameterization, specifics regarding data collection and simulation results for the purpose of simulation validation of analytical results. Finally, Sect. 6 summarizes the paper and outlines some future research areas.

2 System Model/Reference Configuration

2.1 Reference MCPS with Sensing and Ranging Capabilities

Our reference MCPS model is based on the real-world architecture described in [16] comprising human-portable nodes each containing: 600 MHz Analog Devices Blackfin DSP, 8 MB Flash, 64 MB SDRAM, 902–928 MHz CDMA/TDMA radio, GPS receiver, 7.5 V / 35 Wh battery, inertial sensor, barometric sensor, physiological sensors monitoring the user and a radiological/environmental sensor. The primary functions of this MCPS are localization (with periodic ranging) and remote sensing.

The size of our reference MCPS is 128 sensor-carried mobile nodes. Each node ranges its neighbors periodically. Each node uses its sensor to measure any detectable phenomena nearby and transmits a CDMA waveform. Neighbors receiving that waveform transform the timing of the PN code (1023 symbols) and RF carrier (915 MHz) into distance. Specifically, the ranging algorithm used by our reference MCPS for localization has four key functions:

1. process data from inertial and barometric sensors for navigation and multipath mitigation;
2. calculate phase shift between code and carrier of a CDMA waveform to range;
3. distinguish multipath reflections from line of sight in RF input to improve ranging;
4. blend inertial, barometric and range inputs using a Kalman filter.

2.2 Attack Model

The first step in investigating network security is to define the attack model. We consider two forms of attack: node capture and bad data injection. Captured nodes defeat authentication. This creates an insider threat which enables insider attacks. Injecting bad data defeats integrity. This attack is prosecuted by an insider. These attacks can be organized by three criteria: attribution (self or peer), data type (location, telemetry or environmental) and collusion (isolated attacker or a group). It is impossible to prevent all attacks; therefore, intrusion detection is necessary. It is impossible to detect all attacks without false negatives; therefore, intrusion tolerance is necessary. While our theoretical model is general enough to take any security failure definition, in this paper without loss of generality we use a Byzantine fault model [9] to define a security fault, that is, if one-third or more of the nodes are compromised, then the system fails. The reason is that once the system contains more than 1/3 compromised nodes, it is impossible to reach a consensus, hence inducing a security failure. As we will see later our theoretical model can easily accommodate other security failure definitions such as data leaking, single-compromise or majority-compromise failures.

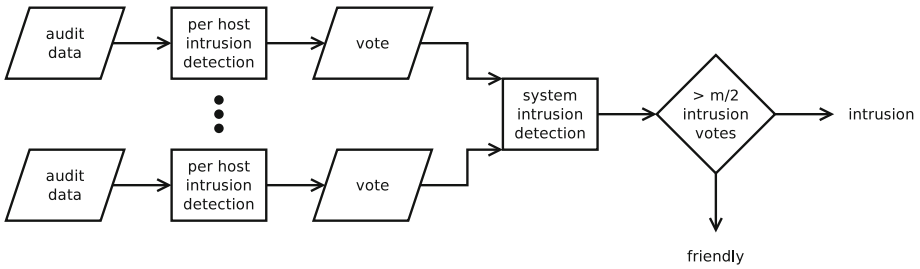

Fig. 1 Combined intrusion detection flowchart

2.3 Intrusion Detection Techniques for MCPSs

Our dynamic voting-based intrusion detection technique builds upon per-host intrusion detection [5–7]. Per-host intrusion detection can apply anomaly detection techniques based on event sequence matching [3]. Without loss of generality, we assume that positional discontinuity [12] is being employed by which a node periodically determines a sequence of locations of a neighbor node within radio range through ranging and detects if the location sequence violates the law of physical continuity, or deviates from the expected location sequence if known. Such anomaly intrusion detection technique is characterized by per-node false negative and false positive probabilities, denoted by p_{fn} and p_{fp}, respectively. This knowledge can be obtained after thoroughly testing the anomaly detection technique.

Our dynamic voting-based intrusion detection technique involves the selection of m detectors as well as the invocation interval T_{IDS} to best balance energy conservation versus intrusion tolerance for achieving high survivability. Each node periodically exchanges its routing information, location, and identifier with its neighbor nodes. A coordinator is selected randomly among neighbors so that the adversaries will not have specific targets. We add randomness to the coordinator selection process by introducing a hashing function that takes in the identifier of a node concatenated with the current location of the node as the hash key. The node with the smallest returned hash value would then become the coordinator. Because candidate nodes know each other's identifier and location, they can independently execute the hash function to determine which node would be the coordinator. The coordinator then selects m vote participants randomly (including itself), and let all voters know each others' identities so that each voter can disseminate its yes/no vote to other voters. Vote authenticity is achieved via preloaded public keys. At the end of the voting process, all voters will know the same result, that is, the node is diagnosed as good, or as bad based on the majority vote. Voting-based intrusion detection is also characterized by system false negative and false positive probabilities, denoted by $\mathcal{P}_{\text{fn}}$ and $\mathcal{P}_{\text{fp}}$, respectively. These two false alarm probabilities are not constant but vary dynamically, depending on the percentage of bad nodes in the system when majority voting is performed.

Figure 1 illustrates how our dynamic voting-based intrusion detection technique and our per-host intrusion detection technique interoperate.

3 Theoretical Analysis

The core of our theory is to predict the number of bad nodes and good nodes in the system as a result of compromising events happening in the system, and voting-based intrusion detection

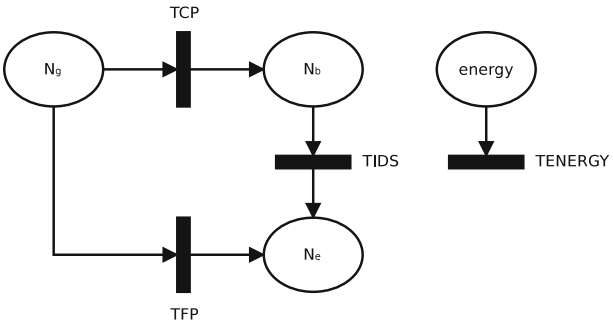

Fig. 2 The SPN model for describing a mobile cyber physical system

events catching and evicting bad nodes in the system. Our theoretical model requires the knowledge of per-node false negative and false positive probabilities, p_{fn} and p_{fp}, as input. It also requires the knowledge of per-node compromise rate, denoted by λ, which may be obtained by analyzing historical data if available and may be estimated dynamically based on the percentage of bad nodes evicted by voting-based intrusion detection.

3.1 Model

Our theoretical model utilizes Stochastic Petri Nets (SPN) techniques [14]. Figure 2 shows the SPN model describing the ecosystem of a MCPS. The underlying model of the SPN model is a continuous-time semi-Markov process with a state representation $(N_{\text{g}}, N_{\text{b}}, N_{\text{e}}, energy)$ where N_{g} is the number of good nodes, N_{b} is the number of bad nodes undetected, N_{e} is the number of nodes evicted (as they are considered as bad nodes by intrusion detection), and $energy$ is a binary variable with 1 indicating energy availability and 0 indicating energy exhaustion. Figure 3 shows the corresponding semi-Markov model of the SPN model for the case in which the number of nodes is 128 in our reference MCPS. Note that only a part of the underlying semi-Markov model is shown in Fig. 3 to illustrate the concept of states and state transitions. The SPN model is constructed as follows:

- We use places to hold tokens each representing a node. Initially, all n nodes are good nodes (e.g., 128 in our reference MCPS) and put in place N_{g} as tokens. The initial state thus is (128, 0, 0, 1) in the underlying semi-Markov model shown in Fig. 3.
- We use transitions to model events. Specifically, TCP models good nodes being compromised; TFP models a good node being falsely identified as compromised; TIDS models a bad node being detected correctly.
- Good nodes may become compromised because of insider attacks with per-node compromising rate λ. This is modeled by associating transition TCP with an aggregate rate $\lambda \times N_{\text{g}}$. Firing TCP will move tokens one at a time (if it exists) from place N_{g} to place N_{b}. Tokens in place N_{b} represent compromised but undetected nodes. For example, if in state (128, 0, 0, 1) a good node is compromised, a token will flow from N_{g} to N_{b} and the resulting state is (127, 1, 0, 1).
- When a bad node is detected as compromised, the number of compromised nodes evicted will be incremented by 1, so place N_{e} will hold one more token. On the other hand, the number of undetected compromised nodes will be decremented by 1, i.e., place N_{b} will hold one less token. This event is modeled by associating transition TIDS with a rate of $\frac{N_{\text{b}} \times (1-\mathcal{P}_{\text{fn}})}{T_{\text{IDS}}}$ accounting for the false negative probability of voting-based intrusion

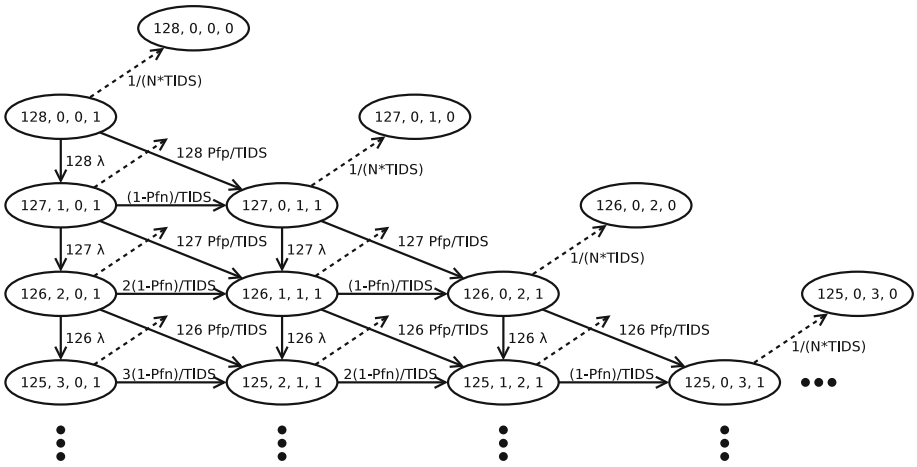

Fig. 3 The underlying semi-Markov model of the SPN model

detection. For example, if in state (127, 1, 0, 1) a bad node is detected and evicted, a token will flow from N_b to N_e and the resulting state is (127, 0, 1, 1).

- Voting-based intrusion detection can also incorrectly identify a good node as compromised. This is modeled by moving a good node in place N_g to place N_e from firing transition TFP with a rate of $\frac{N_g \times \mathcal{P}_{fp}}{T_{IDS}}$ accounting for the false positive probability of voting-based intrusion detection. For example, if in state (127, 1, 0, 1) a good node is misdiagnosed as a bad node and evicted, a token will flow from N_g to N_e and the resulting state is (126, 1, 1, 1).
- The system energy is exhausted after $N \times T_{IDS}$ intervals where N is the maximum number of intrusion detection intervals the MCPS can possibly perform before it exhausts its energy due to performing ranging, sensing, and intrusion detection functions. It can be estimated by considering the amount of energy consumed in each T_{IDS} interval. This energy exhaustion event is modeled by placing a token in place $energy$ initially and firing transition TENERGY with rate $\frac{1}{N \times T_{IDS}}$. When the energy exhaustion event occurs, the token in place $energy$ will be vanished. For example, if in state (128, 0, 0, 1) the energy exhaustion event occurs, the token in place $energy$ will be vanished and the resulting state is (128, 0, 0, 0). We use a dashed-line arrow in Fig. 3 to indicate the energy exhaustion transition. Note that the energy exhaustion event can possibly occur in any state in which the last state component's value is 1, that is, when energy is still available. To avoid clutter, we pick only 4 states in Fig. 3 to illustrate the energy exhaustion transition.

Given p_{fn}, p_{fp} and λ as input the underlying semi-Markov model of our SPN model can be solved utilizing solution techniques such as SOR, Gauss Seidel, or Uniformization [14] to yield the probability of the MCPS staying at a state at time t, as well as the expected values of N_g, N_b, N_e and $energy$ at time t. Once we have knowledge about N_g, N_b and N_e at time t we can calculate system false negative and false positive probabilities, $\mathcal{P}_{fn}$ and $\mathcal{P}_{fp}$, at time t as a result of applying voting-based intrusion detection based on Eqs. 1 and 2 (discussed below). Once $\mathcal{P}_{fn}$ and $\mathcal{P}_{fp}$ at time t are obtained, we dynamically adjust the transition rates to TIDS and TFP, thus modeling changes to $\mathcal{P}_{fn}$ and $\mathcal{P}_{fp}$ as a result of executing dynamic voting-based intrusion detection in response to changing environments.

We make use of Eqs. 1 and 2 to dynamically calculate $\mathcal{P}_{\text{fn}}$ and $\mathcal{P}_{\text{fp}}$. Initially there is no bad node in the system, so at $t = 0$, $N_{\text{g}} = N = 128$, $N_{\text{b}} = 0$ and $N_{\text{e}} = 0$ and the initial values of $\mathcal{P}_{\text{fn}}$ and $\mathcal{P}_{\text{fp}}$ are calculated based on Eqs. 1 and 2, respectively, with $N_{\text{g}} = 128$, $N_{\text{b}} = 0$ and $N_{\text{e}} = 0$ as input. As time progresses, $N_{\text{b}} > 0$ because some good nodes may be compromised, and $N_{\text{e}} > 0$ because bad nodes may be detected, and good nodes may be misdiagnosed as bad nodes and evicted. The values of N_{g}, N_{b} and N_{e} at time t again can be obtained by solving the underlying semi-Markov model of our SPN model using Uniformization solution techniques [14]. The average values of N_{g}, N_{b}, and N_{e} at time t are then plugged into Eqs. 1 and 2

$$\mathcal{P}_{\text{fn}} = \sum_{i=0}^{m-N_{\text{maj}}} \left[\frac{\binom{N_{\text{b}}}{N_{\text{maj}}+i}\binom{N_{\text{g}}}{m-(N_{\text{maj}}+i)}}{\binom{N_{\text{g}}+N_{\text{b}}}{m}} \right] + \sum_{j=0}^{m-N_{\text{maj}}} \left[\frac{\binom{N_{\text{b}}}{j} \sum_{k=N_{\text{maj}}-j}^{m-j} \left[\binom{N_{\text{g}}}{k} (p_{\text{fn}})^k \binom{N_{\text{g}}-k}{m-j-k} (1-p_{\text{fn}})^{(m-j-k)} \right]}{\binom{N_{\text{g}}+N_{\text{b}}}{m}} \right] \tag{1}$$

$$\mathcal{P}_{\text{fp}} = \sum_{i=0}^{m-N_{\text{maj}}} \left[\frac{\binom{N_{\text{b}}}{N_{\text{maj}}+i}\binom{N_{\text{g}}}{m-(N_{\text{maj}}+i)}}{\binom{N_{\text{g}}+N_{\text{b}}}{m}} \right] + \sum_{j=0}^{m-N_{\text{maj}}} \left[\frac{\binom{N_{\text{b}}}{j} \sum_{k=N_{\text{maj}}-j}^{m-j} \left[\binom{N_{\text{g}}}{k} (p_{\text{fp}})^k \binom{N_{\text{g}}-k}{m-j-k} (1-p_{\text{fp}})^{(m-j-k)} \right]}{\binom{N_{\text{g}}+N_{\text{b}}}{m}} \right] \tag{2}$$

to calculate $\mathcal{P}_{\text{fn}}$ and $\mathcal{P}_{\text{fp}}$ at time t.

3.2 False Alarm Probability

We explain Eq. 1 for obtaining $\mathcal{P}_{\text{fn}}$ in detail below. The explanation of Eq. 2 follows the same logic. In Eq. 1, m this is the number of voters and N_{maj} is the majority of m. The first summation is the special case; it aggregates the probability of a false negative stemming from selecting a majority of bad nodes. That is, it is equal to the number of ways to choose a majority of bad nodes from the set of all bad nodes times the number of ways to choose a minority of good nodes from the set of all good nodes divided by the number of ways to choose m nodes from the set of all good and bad nodes. The second summation is the general case; it aggregates the probability of a false negative stemming from selecting a majority of good nodes, some of which cast incorrect votes, coupled with selecting some number of bad nodes. That is, it is equal to the number of ways to choose a minority of bad nodes from the set of all bad nodes times the aggregate probability of a sufficient number of good nodes casting incorrect votes also divided by the number of ways to choose m nodes from the set of all good and bad nodes. The aggregate probability is a nested summation of the number of ways to choose a sufficient number of good nodes which cast incorrect votes and the remaining

good nodes which cast correct votes. Algorithm 1 below illustrates Eq. 1 programmatically. The algorithm for $\mathcal{P}_{\text{fp}}$ substitutes p_{fp} for p_{fn}.

Algorithm 1 $\mathcal{P}_{\text{fn}}$ Calculation Step by Step.

$\Sigma_0 = 0$
$\Sigma_1 = 0$
{Case 1: a majority of voters are intruders}
for all i such that $0 \leq i \leq m - N_m$ **do**
  $\Sigma_0 = \Sigma_0 + C(N_b, N_m + i) \times C(N_g, m - (N_m + i))/C(N_g + N_b, m)$
end for
{Case 2: a minority of voters are intruders}
for all j such that $0 \leq j \leq m - N_m$ **do**
  $\sigma = 0$
  for all k such that $N_m - j \leq k$, $k \leq m - j$ and $k \leq N_g$ **do**
    $\sigma = \sigma + C(N_g, k) \times p_{fn}^k \times C(N_g - k, m - j - k) \times (1 - p_{fn})^{m-j-k}$
  end for
  $\Sigma_1 = \Sigma_1 + C(N_b, j) \times \sigma/C(N_g + N_b, m)$
end for
return $\Sigma_0 + \Sigma_1$

3.3 Survivability Assessment

A MCPS fails due to either energy exhaustion or security failure. The survivability of a MCPS is measured by its expected lifetime before it fails, or equivalently, its mean time to failure (MTTF). For ease of analysis, let the time interval for performing voting-based intrusion detection, T_{IDS}, be the same as that for determining the location sequence of a neighbor node through ranging and sensing. Naturally as T_{IDS} decreases, the energy consumption rate increases because the system has to invoke intrusion detection more often. The MTTF of the MCPS can be calculated as the accumulated "reward" of the underlying semi-Markov reward model by assigning a reward of 1 to states in which the system is alive and 0 otherwise. Specifically, let r_i be the reward (representing the contribution to system lifetime) assigned to state i. Then,

$$r_i = \begin{cases} 1 \text{ if the system is alive in state } i \\ 0 \text{ otherwise} \end{cases} \tag{3}$$

Intuitively this reward assignment has the effect of accumulating a unit of 1 to the MCPS lifetime in states in which the system is still alive but accumulating nothing to the MCPS lifetime in states in which the system fails. Here state i in Eq. 3 refers to a particular state in the semi-Markov model shown in Fig. 3. It could be (128, 0, 0, 1), (127, 1, 0, 1), or any state in the semi-Markov model. The probability of the MCPS stays at a particular state at time t again can be obtained by solving the underlying semi-Markov model of our SPN model utilizing Uniformization solution techniques [14]. Also, we know exactly whether the MCPS fails or not in a particular state. That is, the MCPS fails when the energy is exhausted, i.e., when place *energy* does not have a token, or when the bad node population (given by the number of tokens in place N_{b}) is 1/3 or more of the total population (given by $N_{\text{b}} + N_{\text{g}}$). In the latter case the system fails from a Byzantine failure. With this knowledge, we can calculate the MTTF of the MCPS numerically.

Here we note that the way we calculate MTTF is based on assigning a reward of 1 to states in which the system is alive and 0 to states in which the system fails, with the "cumulative

reward till absorption" being the MTTF of the MCPS. The false probabilities, i.e., $\mathcal{P}_{\text{fn}}$ and $\mathcal{P}_{\text{fp}}$, and T_{IDS} affect the probability of the system staying at a particular state because they affect the transition rates, and consequently, they affect MTTF. There is no closed-form solution expressing MTTF as a function of $\mathcal{P}_{\text{fn}}$, $\mathcal{P}_{\text{fp}}$ and T_{IDS}.

4 Numerical Data

In this section we present numerical data for survivability assessment as a result of executing voting-based intrusion detection in a MCPS. Our objective is to identify optimal design settings in terms of the optimal values of T_{IDS} and m under which we can best trade off energy consumption versus intrusion detection to maximize the system MTTF, when given a set of parameter values characterizing the operational and networking conditions.

4.1 Parameterization

We consider the reference MCPS model introduced in Sect. 2 operating in a 2 $\times$ 2 area with a network size (n) of 128 nodes. Hence, the number of neighbors within radio range, denoted by $\bar{n}$, initially is about 128/4 = 32 nodes. A node in our reference MCPS uses a 35 Wh battery, so its energy is 126,000 J. The system energy initially, denoted by E_{o}, is therefore 126,000 J $\times$ 128 = 1,612,8000 J. Table 1 lists the set of parameters and their values for the reference MCPS. We vary m, T_{IDS} and λ over a range of perceivable values to test their effects on survivability. Here we note that $\mathcal{P}_{\text{fn}}$ and $\mathcal{P}_{\text{fp}}$ are calculated by Eqs. 1 and 2. MTTF is calculated from the SPN model by means of reward assignments in accordance with Eq. 3. The maximum number of intrusion detection cycles the system can possibly perform before energy exhaustion, denoted by N, is calculated as:

$$N = \frac{E_{\text{o}}}{E_{T_{\text{IDS}}}} \tag{4}$$

where E_{o} is the initial energy of the reference MCPS and $E_{T_{\text{IDS}}}$ is the energy consumed per T_{IDS} interval due to ranging, sensing, and intrusion detection functions, calculated as:

$$E_{T_{\text{IDS}}} = n \times (E_{\text{ranging}} + E_{\text{sensing}} + E_{\text{detection}}) \tag{5}$$

where E_{ranging}, E_{sensing}, $E_{\text{detection}}$ stand for energy spent for ranging, sensing, and intrusion detection in a T_{IDS} interval, respectively. Here the energy spend per node is multiplied with the node population in the MCPS to get the total energy spent by all nodes per cycle.

In Eq. 5, E_{ranging} stands for the energy spent for periodic ranging. It is calculated as:

$$E_{\text{ranging}} = \alpha \times [E_{\text{t}} + \bar{n} \times (E_{\text{r}} + E_{\text{a}})] \tag{6}$$

Here a node spends E_{t} energy to transmit a CDMA waveform. Its $\bar{n}$ neighbors each spend E_{r} energy to receive the waveform and each spend E_{a} energy to transform it into distance. This operation is repeated for α times for determining a sequence of locations. In Eq. 5, E_{sensing} stands for the amount of energy consumed due to periodic sensing. It is computed as:

$$E_{\text{sensing}} = \bar{n} \times (E_{\text{s}} + E_{\text{a}}). \tag{7}$$

Here a node spends E_{s} energy for sensing navigation and multipath mitigation data and E_{a} energy for analyzing sensed data for each of its $\bar{n}$ neighbors. Finally, $E_{\text{detection}}$ stands for the energy used for performing intrusion detection on a target node. It can be calculated by:

Table 1 Parameters and their values

Parameter	Meaning	Default value
n	Network size	128
$\bar{n}$	Number of neighbors within radio range	32
p_{fn}	Per-host false negative probability	[1–5]%
p_{fp}	Per-host false positive probability	[1–5]%
λ	Per-node capture rate	[1–24]/day
T_{IDS}	Intrusion detection interval	[0–700] s
m	Number of intrusion detectors per node	[3,11]
α	Number of ranging operations	5
E_{t}	Energy for transmission per node	0.000125 J
E_{r}	Energy for reception per node	0.00005 J
E_{a}	Energy for analyzing data per node	0.00174 J
E_{s}	Energy for sensing per node	0.0005 J
E_{o}	Initial system energy	16,128 kJ
$\mathcal{P}_{\text{fn}}$	System false negative probability	Eq. 1
$\mathcal{P}_{\text{fp}}$	System false positive probability	Eq. 2
MTTF	Mean time to failure	Eq. 3
N	Maximum cycles before energy exhaustion	Eq. 4
$E_{\text{T}_{\text{IDS}}}$	Energy consumed per T_{IDS}	Eq. 5

$$E_{\text{detection}} = m \times (E_{\text{t}} + \bar{n} \cdot E_{\text{r}}) + m \times (E_{\text{t}} + (m-1) \cdot (E_{\text{r}} + E_{\text{a}})). \tag{8}$$

Here we consider the energy required to choose m intrusion detectors to evaluate a target node (the first term) and the energy required for m intrusion detectors to vote (the second term). Specifically, the first term is the number of intrusion detectors times the cost of transmitting plus the number of nodes in radio range times the cost of receiving. The second term is the number of intrusion detectors times the cost of transmitting plus the number of peer intrusion detectors times the cost of receiving plus the cost of analyzing the vote.

4.2 Results

Figure 4 shows theoretical MTTF versus T_{IDS} as the number of detectors (m) in voting-based intrusion detection varies over the range of [3,11] in increments of 2. We see that there exists an optimal T_{IDS} value at which the system lifetime is maximized to best tradeoff energy consumption versus intrusion tolerance. Initially when T_{IDS} is too small, the system performs ranging, sensing and intrusion detection too frequently and quickly exhausts its energy, resulting in a small lifetime. As T_{IDS} increases, the system saves more energy and its lifetime increases. Finally when T_{IDS} is too large, although the system can save even more energy, it fails to catch bad nodes often enough, resulting in the system having many bad nodes. When the system has 1/3 or more bad nodes out of the total population, a Byzantine failure ensues. Furthermore, the optimal T_{IDS} value is sensitive to the m value. We observe a general trend that as m decreases, the optimal T_{IDS} value decreases. The reason is that as m decreases, the system has to compensate less vigorous intrusion detection (i.e., a smaller m) by a higher invocation frequency (i.e., a smaller T_{IDS}) to prevent security failures. For the

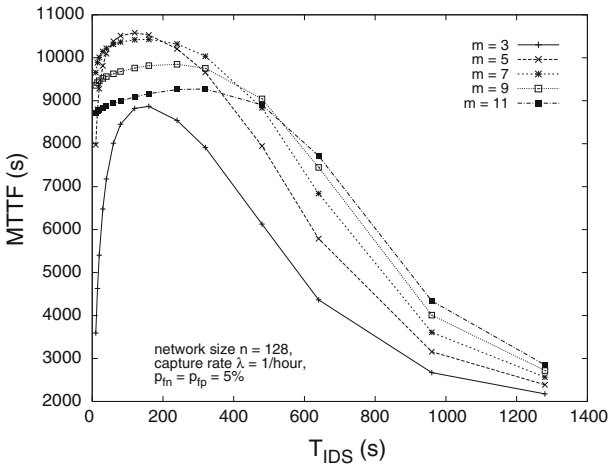

Fig. 4 MTTF versus T_{IDS} and m

reference MCPS, we also observe that $m = 5$ is optimal because too many intrusion detectors would induce energy exhaustion failure, while too few intrusion detectors would induce security failure. Using $m = 5$ can best balance energy exhaustion failure versus security failure for high survivability.

Figure 5 shows MTTF versus T_{IDS} as the compromising rate λ varies over the range of once per hour to once per day to test the sensitivity of MTTF with respect to λ (with m fixed at 7 to isolate its effect). We first observe that as λ increases, MTTF decreases because a higher λ will cause more compromised nodes to be present in the system. We also observe that the optimal T_{IDS} decreases as λ increases. This is because when more compromised nodes exist, the system needs to execute intrusion detection more frequently to maximize MTTF. Figure 5 identifies the best T_{IDS} to be used to maximize the lifetime of the reference MCPS to balance energy exhaustion versus security failure, when given p_{fn}, p_{fp} and λ characterizing the operational and networking conditions of the system.

Figure 6 shows the sensitivity of MTTF with respect to p_{f} under different m. For this dataset, we use a T_{IDS} of 160 s because our earlier results identified this as a near optimal configuration. As we can see, MTTF is highly sensitive to p_{f} and this affects the optimal m value at which MTTF is maximized. For highly reliable IDS with low p_{f}, a lower m benefits the MTTF. For less reliable IDS with high p_{f}, a higher m benefits the MTTF. The crossover occurs in $p_{\text{f}} \in [0.005, 0.09]$.

5 Simulation

5.1 Simulation Tool and Environment Setup

We instrumented a simulation using SMPL [11]. The simulation tracks node state with components for membership, goodness, time of last move, current location and current cell. The simulation schedules events for node capture, intrusion detection audits and energy exhaustion. A simulation run ends for one of three reasons: there is a security failure, the system exhausts its energy or all of the nodes have been evicted.

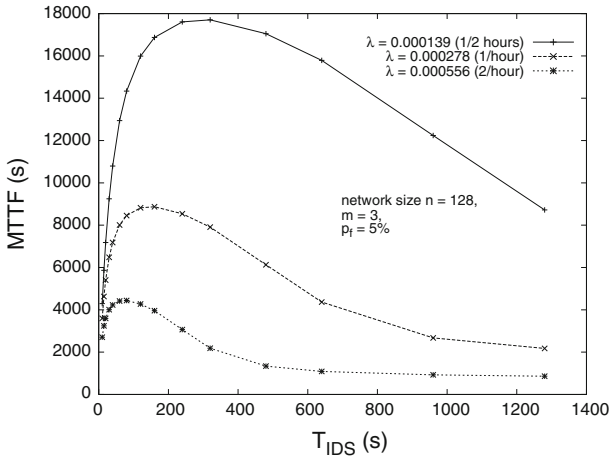

Fig. 5 MTTF versus T_{IDS} and λ

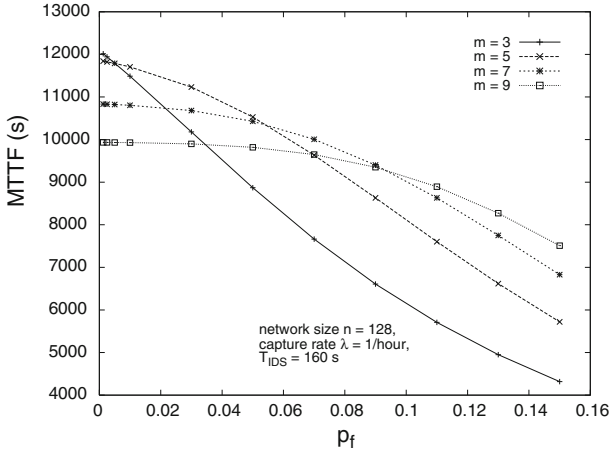

Fig. 6 MTTF versus p_{f} and m

5.2 Parameter Values

We configure the simulation with the values described in Table 1. We run the simulation over a range of values for capture rate (λ), number of intrusion detectors (m) and intrusion detection interval (T_{IDS}). We exercise λ over [1/day, 1/10 min], m over [3, 11] and T_{IDS} over [10, 1280] s.

5.3 Data Collection

We collect a MTTF observation by recording the starting simulator time, running a simulation, recording the ending simulator time and calculating the difference between simulator times. The average MTTF value reported represents a grand mean out of a large number of MTTF observations with statistical significance.

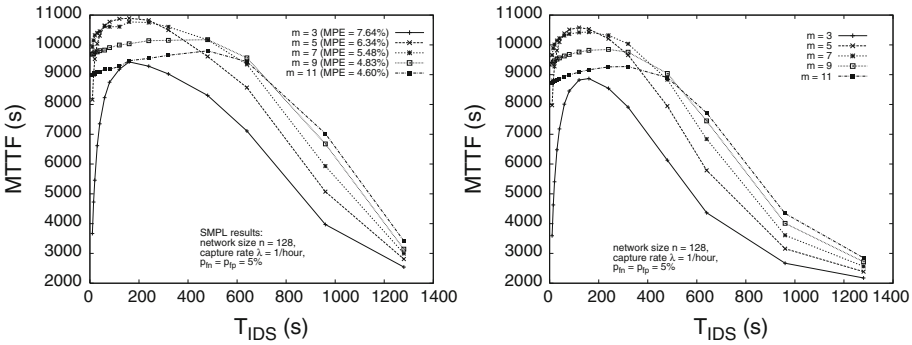

Fig. 7 Simulation and theoretical MTTF versus T_{IDS} and m

Specifically, we apply *batch means analysis* [11] to satisfy 95% confidence level and 10% accuracy requirements. We use a batch size of 100 MTTF observations to compute a batch mean out of 100 MTTF observations. The general idea is that for each given configuration, we run a number of batches to get a number of batch means. Then we calculate the grand mean out of the batch means and determine if the grand mean falls within 10% of the true mean with 95% confidence. If it does not, we increase our sample data by one more batch to collect another batch mean and then compute the grand mean again. We repeat this until the grand mean satisfies the 95% confidence level and 10% accuracy requirements.

5.4 Simulation Results

Figure 7 shows MTTF versus T_{IDS} simulation results (the left graph) with respect to analytical results (the right graph) shown earlier in Fig. 4. The shapes of both plots are remarkably similar: unimodal with similar kurtosis, a left/positive skew and a pronounced right tail. In both plots, MTTF peaks near $T_{\text{IDS}} = 160$ s between 9,000 and 11,000 s and $m = 5$ is the optimal value. The mean percentage error (MPE) separating the analysis and simulation results are between 4.60 and 7.64% curve by curve, as shown in Fig. 7. We conclude that simulation results match up with analytical results very well, thereby validating our survivability analysis methodology.

6 Conclusions

In this paper, we developed a mathematical model to analyze survivability of a mobile cyber physical system (MCPS) comprising sensor-carried mobile nodes with voting-based intrusion detection capabilities. Given the system failure definition caused by either energy exhaustion or security failure, we identified the optimal design settings for executing voting-based intrusion detection such that the system lifetime is maximized. We demonstrated the feasibility of our model-based analysis methodology by means of a reference MCPS for executing emergency rescuing missions leveraging a ranging-based locational discontinuity intrusion detection technique. Our results with simulation validation showed that, given knowledge of per-node false alarm probabilities and pre-node compromise rates, the system can dynamically select the best intrusion detection interval and the best number of detectors to balance

energy consumption due to intrusion detection versus security failure due to security attacks to maximize the MCPS lifetime.

In the future, we plan to develop a set of survivability-directed design principles to guide intrusion detection protocol design for both homogeneous and heterogeneous MCPSs. We also plan to investigate how our model-based analysis methodology can be applied to a wider range of MCPS applications operating under such intrusion detection techniques for survivability assessment.

References

1. Anand, M., Cronin, E., Sherr, M., Blaze, M., Ives, Z., & Lee, I. (2006). Security challenges in next generation cyber physical systems. In *Beyond SCADA: Networked Embedded Control for Cyber Physical Systems*. Pittsburgh, Pennsylvania, USA.
2. Cárdenas, A. A., Amin, S., Sinopoli, B., Giani, A., Perrig, A., & Sastry, S. (2009). Challenges for securing cyber physical systems. In: *Workshop on future directions in cyber-physical systems security*. Newark, NJ, USA.
3. Chandola, V., Banerjee, A., & Kumar, V. (2010). Anomaly detection for discrete sequences: A survey. *IEEE Transactions on Knowledge and Data Engineering*. doi:10.1109/TKDE.2010.235.
4. Chen, I. R., & Wang, D. C. (1996). Analyzing dynamic voting using petri nets. In: *15th IEEE symposium on reliable distributed systems*. Niagara Falls, ON, Canada.
5. Cho, J. H., & Chen, I. R. (2010). Modeling and analysis of intrusion detection integrated with batch rekeying for dynamic group communication systems in mobile ad hoc networks. *Wireless Networks, 16*(4), 1157–1173.
6. Cho, J. H., & Chen, I. R. (2011). Performance analysis of hierarchical group key management integrated with adaptive intrusion detection in mobile ad hoc networks. *Performance Evaluation, 68*(1), 58–75.
7. Cho, J. H., Chen, I. R., & Feng, P. G. (2010). Effect of intrusion detection on reliability of mission-oriented mobile group systems in mobile ad hoc networks. *IEEE Transactions on Reliability, 59*(1), 231–241.
8. El-Khatib, K. (2010). Impact of feature reduction on the efficiency of wireless intrusion detection systems. *IEEE Transactions on Knowledge and Data Engineering, 21*(8), 1143–1149.
9. Lamport, L., Shostak, R., & Pease, M. (1982). The byzantine generals problem. *ACM Transactions on Programming Languages and Systems, 4*(3), 382–401.
10. Lima, M., dos Santos, A., & Pujolle, G. (2009). A survey of survivability in mobile ad hoc networks. *IEEE Communications Surveys and Tutorials, 11*(1), 66–77.
11. MacDougall, M. H. (1987). *Simulating computer systems, techniques and tools*. Cambridge: The MIT Press.
12. Mitchell, R., & Chen, I. R. (2011). A hierarchical performance model for intrusion detection in cyber-physical systems. In *IEEE wireless communication and networking conference*. Cancun, Mexico.
13. Noble, B. D., & Flinn, J. (2006). Wireless, self-organizing cyber-physical systems. In: *NSF workshop on cyber-physical systems*. Austin, Texas, USA.
14. Sahner, R. A., Trivedi, K. S., & Puliafito, A. (1996). *Performance and reliability analysis of computer systems*. Dordrecht: Kluwer Academic Publishers.
15. Sun, B., Osborne, L., Xiao, Y., & Guizani, S. (2007). Intrusion detection techniques in mobile ad hoc and wireless sensor networks. *IEEE Wireless Communications, 14*(5), 56–63.
16. US Department of Homeland Security. (2009). *BAA-09-02 Geospatial Location Accountability and Navigation System for Emergency Responders (GLANSER)*.
17. Wang, Y., Wang, X., Xie, B., Wang, D., & Agrawal, D. P. (2008). Intrusion detection in homogenerous and heterogeneous wireless sensor networks. *IEEE Transactions on Mobile Computing, 7*(6), 698–711.

Author Biographies

Robert Mitchell received the B.S. and M.S. degrees in Computer Science from Virginia Polytechnic Institute and State University in 1997 and 1998, respectively. Currently he is a Ph.D. student in the Department of Computer Science at Virginia Tech. His research interests include security, mobile computing, sensor networks, embedded systems, and coding and information theory.

Ing-Ray Chen received the B.S. degree from the National Taiwan University, Taipei, Taiwan, and the M.S. and Ph.D. degrees in Computer Science from the University of Houston. He is a professor in the Department of Computer Science at Virginia Tech. His research interests include mobile computing, wireless networks, security, multimedia, distributed systems, real-time intelligent systems, and reliability and performance analysis. Dr. Chen currently serves as an editor for Wireless Personal Communications, IEEE Communications Letters, IEEE Transactions on Network and Service Management, Wireless Communications and Mobile Computing, The Computer Journal, Security and Communication Networks, and International Journal on Artificial Intelligence Tools. He is a member of the IEEE/CS and ACM.