North Carolina Agricultural and Technical State University

Aggie Digital Collections and Scholarship

Theses

Electronic Theses and Dissertations

2012

Modeling And Analysis Of Cascading Effects Of Weapons Of Mass Destruction (Wmd) Events On Critical Infrastructure Systems

Sliman Amrani Joutei

North Carolina Agricultural and Technical State University

Follow this and additional works at: https://digital.library.ncat.edu/theses

Recommended Citation

Joutei, Sliman Amrani, "Modeling And Analysis Of Cascading Effects Of Weapons Of Mass Destruction (Wmd) Events On Critical Infrastructure Systems" (2012). *Theses*. 72.
https://digital.library.ncat.edu/theses/72

This Thesis is brought to you for free and open access by the Electronic Theses and Dissertations at Aggie Digital Collections and Scholarship. It has been accepted for inclusion in Theses by an authorized administrator of Aggie Digital Collections and Scholarship. For more information, please contact iyanna@ncat.edu.

MODELING AND ANALYSIS OF CASCADING EFFECTS OF WEAPON OF MASS DESTRUCTION (WMD) EVENTS ON CRITICAL INFRASTRUCTURE SYSTEMS

by

Sliman Amrani Joutei

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Department: Electrical and Computer Engineering
Major: Electrical and Computer Engineering
Major Professor: Dr. Marwan Bikdash

North Carolina A&T State University
Greensboro, North Carolina
2012

School of Graduate Studies
North Carolina Agricultural and Technical State University

This is to certify that the Master's Thesis of

Sliman Amrani Joutei

has met the thesis requirements of
North Carolina Agricultural and Technical State University

Greensboro, North Carolina
2012

Approved by:

Dr. Marwan Bikdash Major Professor	Dr. Numan S. Dogan Committee Member
Dr. Robert Li Committee Member	Dr. John C. Kelly Department Chairperson

Dr. Sanjiv Sarin
Associate Vice Chancellor of Research and Graduate Dean

DEDICATION

This work is dedicated to my wife and children, my dad, my mom, my mother-in-law and my brother’s-in-law family for their prayers and unflinching support through my years of education.

BIOGRAPHICAL SKETCH

Sliman Amrani Joutei was born on September 17, 1968, in Casablanca, Morocco. He received the Bachelor of Science degree in Chemistry from Mohamed V University of Morocco in 1997. He is a candidate for the Master in Electrical Engineering at the North Carolina Agricultural and Technical State University.

ACKNOWLEDGMENTS

It is a great pleasure to express many thanks to my advisor, Dr. Marwan Bikdash, for his support and enthusiastic advice. The past two and half years at North Carolina Agricultural and Technical State University have been some of the happiest times in my life. I also would like to thank Dr. Numan S. Dogan and Dr. Robert Li for their service as my committee members and for providing valuable advice.

Additionally, I would like to thank the Advanced Systems and Concepts Office (ASCO) at the Defense Threat Reduction Agency (DTRA) for their support of the Student Research Associate Program (SRAP) under the University Strategic Partnership (USP) and for the support of Pennsylvania State University (PSU) and DTRA under contract DTRA01-03-0010-0016, which provided partial funding for my work. Also, special thanks are due to Dr. Bharat B. Madan, PSU for collaborating and helping with this work.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES **PAGE**

LIST OF TABLES

TABLES **PAGE**

LIST OF ACRONYMS

BSCC	Bottom Strongly Connected Components
CIS	Critical Infrastructure Systems
IIS	Information Infrastructure Systems
HFM	Hierarchical fault model
HIS	Health Infrastructure Systems
TIS	Transportation Infrastructure Systems
MMR	Markov Model Representation

ABSTRACT

Amrani Joutei, Sliman. MODELING AND ANALYSIS OF CASCADING EFFECTS OF WEAPONS OF MASS DESTRUCTION (WMD) EVENTS ON CRITICAL INFRASTRUCTURE SYSTEMS. **(Major Professor: Marwan Bikdash)**, North Carolina Agricultural and Technical State University.

This research studies how the global network behaves after a Weapon of Mass Destruction (WMD) attack. The goal is to find a reliable model that will help capture the behavior of the network in the event of a WMD attack and then proceed to a systematic analysis of that model. We discuss a hierarchical model that visualizes how a WMD attack will impact different infrastructure systems. The second level of the Hierarchical Fault Model (HFM) illustrates the failure propagation from one Critical Infrastructure System (CIS) to another CIS. Next, we construct a stochastic Petri Net for HFM of which we compute a Markov model representation that is subsequently developed and analyzed.

In the developed models, both the repair and the failure are modeled to be random, and the repair process for one system depends on the actual system as well as on the damage severity in the other systems as well. For example, if the repair process of the Information System requires one day normally, the repair process will be further delayed by damages in the Transportation System or in the Hospital System.

We develop algorithms and methods that allow the analysis of different aspects of the HFM such as understanding state trajectory dynamics and the statistics of the transient and steady-state behavior.

CHAPTER 1

INTRODUCTION

1.1 WMD attacks and other threats under investigation

The south tower of the world trade center in New York City was slammed by a plane at 9:03 AM, and at 9:21 AM another plane hit the North tower. Additionally, more attacks were carried out in Washington DC and New Pittsburg. This act left 2819 dead [1]. The rescue mission was impaired by secondary damages. For instance, the Emergency Operation Center, a $15 million facility opened in 1999 and located in the 23d floor of the World Trade Center building was also destroyed [1]. In this particular case, terrorists used knives to hijack planes but what if the terrorists decide to use more sophisticated weapons in order to conduct and carry out their operations? What if terrorists used chemical or biological weapons?

Terror events are not equal, therefore they should be classified. Furthermore, terrorist acts can be modeled as consisting of successive steps. The probability of a terrorist event is in fact the product of the conditional probabilities associated with the steps. Hence, in order to reduce the probability of terror events, we should aim at reducing conditional probabilities to the minimum [2].

An example of terrorist attacks could be a cyber attack, which can be described as security failures (as opposed to reliability failures). Security failures are often by reliability failures [3]. An example is that of a Distributed Denial of Service (DDoS) attack when a group of synchronized requests to a resource are sent to a network. The goal of the attack is to prevent legitimate users access the resources and to push the system to the brink of reliability failures. The simulation tools that can detect a cascading failure independently of security failure must be developed [4].

The entire natural resources can be thought of as linked together through "Cyber care", thus leading to a superior response to a terrorist attack [5]. Attacks may also be decentralized on integrated networks [6].

The Idaho National Laboratory (INL) is in the process of becoming an all hazard WMD technology evaluation and training range. The benefits of consolidating all types of training related to WMD in one facility are apparent [7].

To increase the efficiency of a WMD attack response, an integrated Hierarchical Fault Model (HFM) such the one considered in this thesis is greatly beneficial. The modeling effort is complemented by designing sophisticated alarm systems to combat terrorism. A micromachined differential mobility spectrometer (DMS), for instance can detect chemical and biological agents simultaneously on a time scale of seconds [8].

A full model may also have to include government policies such as those enforcing antiterrorist regulations recommendations such as Biological Weapon Convention (BWC) and Chemical Weapon Convention (CWC) [9].

One substructure of the network is the transportation network. Terrorists may use the trucking network to conduct an attack or try to destroy parts of it. Many command and control security systems can be used to prevent such attacks, such as, the use of GPS technology [10].

Another substructure of interest is the communication network. Its survivability is assessed using simulations and analytical model [11]. Although WMD deals primarily with the physical destruction of the network units, intrusion is possible after a WMD event to further weaken the network.

Network intrusion can be detected by using Markov Chains and the flag field in TCP/IP protocol [12].

One measure of the survivability counts the number of connected nodes in the largest communicating section of the network after attack. Algorithms are available to identify vulnerable nodes in the network [13]. Some nodes are more central than others and hence are crucial to survivability [14].

Survivability can also be defined as the ability of the system to meet the minimum performance in the presence of undesirable events. Stochastic reward net and continuous Markov chains can be used to model the network survivability [15].

1.2 Petri Nets

Adam Petri introduced Petri Nets in 1962. Petri Nets are not only a graphical representation of dynamic systems but also a mathematical interpretation. This combination made Petri Nets successful in many different applications. For example, communication systems and survivability are few areas among many that benefited a lot from Petri Nets [16].

A Petri Net consists of 3 objects: places, transitions, and directed arcs. A Petri Net marking denotes the number of tokens in each place. Transitions and arcs dictate how the distributions of the tokens evolve over time [17].

The flow of tokens and their distribution over the places is governed by two rules. First, the enabling rule in which a transition is enabled if it contains at least k tokens where k is the multiplicity of the directed arc connecting a place P to a transition T. Second, the firing rule in which a number of k tokens equal to the multiplicity of the directed arc TP are deposited in the output places [18].

Transient analysis for prioritized failure recovery in communication network was studied using Petri Nets. However, the focus was on the transient behavior of a queuing system with prioritized recovery [19].

Petri nets were also used to assess the survivability of object oriented software in design phase [20]. The distribution functions were defined to be the probability for the transition to be a failure. The coverage testing and fault density analysis were used to compute those distributions.

Formal verification of a network survivability model was also presented using Petri Nets [21] where formal verification enabled the extraction of certain properties like reachability, boundedness, and fairness.

The firing times can be stochastic, thus leading to Stochastic Petri Nets (STPN) where the exponential distribution is often used. The STPN can often be simplified to Continuous-Time Markov Chains (CTMC). If the resulting Markov chain is ergodic, the steady state probability could readily be obtained by using standard linear algebra techniques [22].

1.3 Markov Chains

Andrei A. Markov (1856-1922) proposed the concept of Markov Chains. Since its creation, Markov Chains found many applications as survivability and statistical models of real-world processes. Markov models have been used to study queuing systems with propagated breakdowns and thus survivability analysis was provided. In fact, the system returns to the normal state via the threshold in the queue length [23].

A survivability quantitative evaluation model was created based on Markov models, and was used to develop a methodology for choosing proper strategies and optimization of the system design [24].

Markov models were also used to evaluate the survivability for intrusion-tolerant real-time database systems (ITRDB) [25] and to predict with high accuracy the behavior of a real intrusion-tolerant database system. Experimental results show the validity of the model used [26].

In Markov Chains, the state space of possible values of the Markov Chain is finite or countable. Moreover, all the information needed to predict the future is contained in the present. At equally- spaced time points, the process evolves from one state to another and the one step transition matrix contains the probabilities of transitioning among the Markov states [27]. When the states are hidden and observations are generated according to some random rule, Hidden Markov models are used. They can represent complex Markov processes where the states emit the observations. More information could be found in [28].

For a continuous time Markov Chains, a system in state O, can transition after a random time with an exponential distribution of parameter λ to another state. When a state has more than one state to transition to, the waiting time is exponentially distributed with parameter $\lambda_1 + \lambda_2 + ... + \lambda_n$. The probability to jump to a particular state is therefore simply the proportion of that rate [27].

1.4 Synopsis

Critical Infrastructure systems (CIS) contain many infrastructure systems (e.g., Military, financial, transportation, health, etc.). When a Weapon of Mass Destruction (e.g. dirty terrorist bombs, massive nuclear explosion, etc.) affects one or many infrastructure systems, the rest of the CIS will be likely affected as well. For example: if a WMD destroys a power grid, then the process of repair of the power grid will be impaired or delayed because many processes (i.e. Information, Transportation… etc.) depends on the power grid.

In this work, we limit our attention to three systems; namely, the Information System (IS), the Transportation System (TS), and the Health System (HS). The WMD may damage the different sub-networks differently, It might destroy the IS, TS, and HS by 30%-25%-40% respectively. A nuclear bomb attack might destroy the IS, TS, and HS by 30%-70%-10% respectively.

In chapter 2, we review some of the background necessary to understand this research. In chapter 3, we explore the Markov Model Representation of Petri Nets. In chapter 4, a Hierarchical Fault Model (HFM) for Weapon of Mass Destruction is constructed. In chapter 5, the HFM is subjected to various analyses, such as extensive Monte Carlo simulations, and deriving a representative Markov Model (MM) which is subsequently analyzed and develop the methodology and algorithms needed. We conclude in Chapter 6.

CHAPTER 2

BACKGROUND

2.1 Petri Nets review

Petri nets are primarily used for studying the concurrent dynamic behavior of network-based systems where there is a discrete flow represented by a number of tokens. A Petri net is a bipartite directed graph formed by three types of objects: transitions, places, and directed arcs. Directed arcs connect either places to transitions or transitions to places. A Petri net consists of two types of nodes: places and transitions. An arc exists only from a place to a transition or from a transition to a place. A place may have zero or more tokens. Graphically, places, transitions, arcs, and tokens are represented respectively by circles, bars, arrows, and dots respectively.

Figure 2.1 is an example of a Petri net with two places and one transaction. The transition node is ready to fire if and only if there is at least one token at each of its input places. As shown in Figure 2.1, the Petri Net has one token in the input place P_1. The place P_2 receives one token after transition T_1 fires. Figure 2.2 represents the resulting Petri Net after T_1 fires. The Petri Nets shown in Figure 2.1 and Figure 2.2 can be represented by the markings $[1,0]$ and $[0,1]$. Therefore, the marking transition has the form $\left[1,0\right] \rightarrow \left[0,1\right]$.

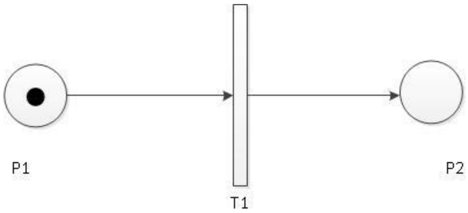


Figure 2.1. A Petri Net with one enabled transition, two places and one token

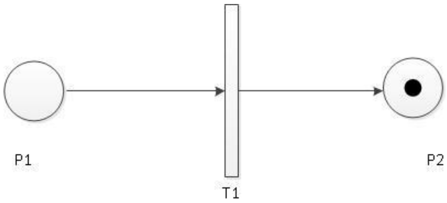


Figure 2.2. Place P_1 looses one token and Place P_2 wins one token

Mathematically we write $C = (P, T, I, O)$ defined with the following sets and mappings.

$$\begin{array}{ll} \text{Places:} & P = \{p_1, p_1, p_1, \ldots. p_n\}. \\ \text{Transitions:} & T = \{t_1, t_1, t_1, \ldots t_n\}. \\ \text{Input:} & I : T \to P^r (r = \text{number of places}). \\ \text{Output:} & \text{O} : T \to P^q (q = \text{number of places}). \\ \text{Marking} & \mu\text{: assignment of tokens into places.} \\ \text{Marking} & \mu\text{: } \mu{=}\mu_1, \mu_2, \mu_3, \ldots\ldots.\mu_n. \end{array} \qquad (2.1)$$

The simulation of a stochastic Petri Net follows a simple scheme. In the case of one unique transition, the transition will fire if:

$$\text{rand} \le 1 - \exp(-\lambda t) = P(X \le t), \qquad (2.2)$$

Here rand is a random number $\in [0,1]$ generated from a uniform distribution.

In the case of many transitions, one transition will fire if:

$$\text{rand} \leq 1 - \exp(-(\lambda_1 + \lambda_2 + \lambda_3 + ... + \lambda_n)t). \tag{2.3}$$

But any transition i will fire if $\text{rand} \leq \dfrac{\lambda_i}{\lambda_1 + \lambda_2 + ... + \lambda_n}$.

2.1.1 Synchronization, Concurrency, and Conflict in Petri Nets

In Figure 2.3 transition T_2 can fire only after the firing of T_1. This imposes the precedence of constraints "T_2 after T_1". With this property, a Petri net is able to model processes executing sequentially in time.

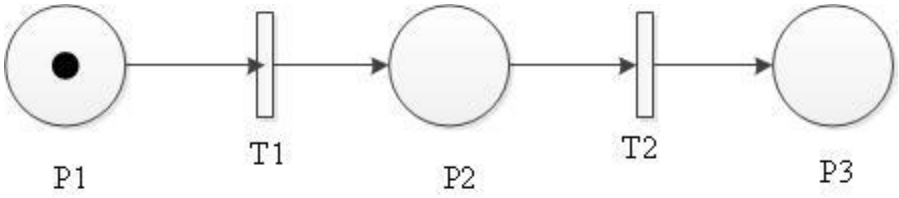


Figure 2.3. Example of sequential execution: T_2 fires only when T_1 has already fired

In Figure 2.4 transition T_1 will be enabled only when there are at least one token at each of its input places. With this property, a Petri net is able to model multiple processes executing with synchronization in time.

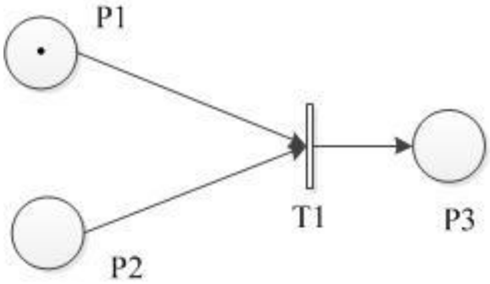


Figure 2.4. Synchronization example: Place P1 cannot fire until P2 receives a token

In Figure 2.5 merging happens when tokens from several places arrive for service at the same transition. When transition T_1 fires, the resources located in the p1aces P_1 and P_2 are consolidated into the place P_3.

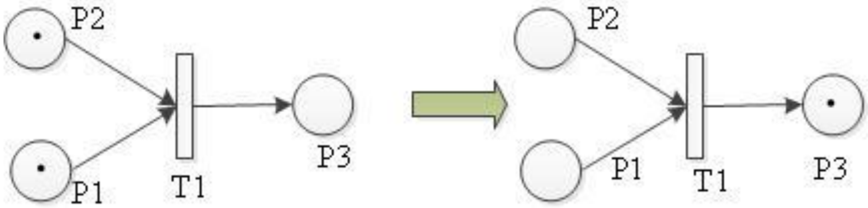


Figure 2.5. Example of merging proprieties

Two transitions are concurrent if they can happen simultaneously. In Figure 2.6 the Transitions T_3 and T_2 are concurrent. In fact, the Transitions T_3 and T_2 can happen simultaneously (or not). Note that a token in place P_2 has been duplicated into 2 tokens, one in P_2 and the other in P_3.

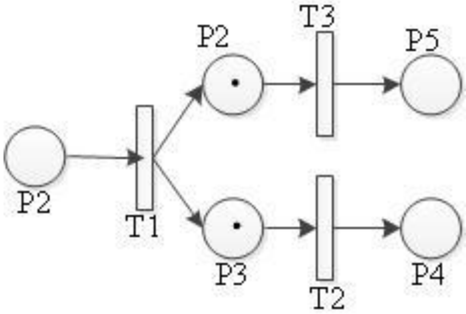


Figure 2.6. Example of concurrent transitions

Conflict occurs when two transitions T_1 and T_2 are both ready to fire but the firing of one transition leads to the inhibition of the other transition. In Figure 2.7, the transition T_1 cannot fire if the transition T_2 fires. Similarly, the transitions T_2 cannot fire if the transition T_1 fires.

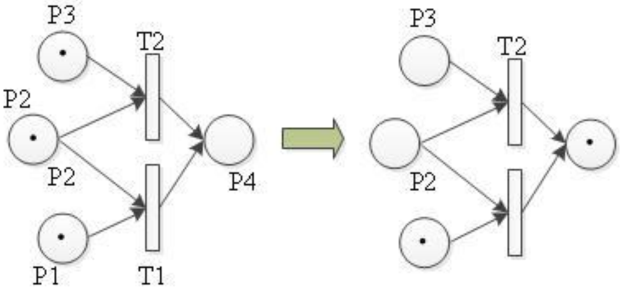


Figure 2.7. Example of a conflict

2.1.2 Stochastic Petri Nets

The continuous-time Stochastic Petri net $SPN = (PN, \Lambda)$ is formed from the Place-Transition net $PN = (P, T, I_{-}, I_{+}, M_0)$ by adding the set of rates $\Lambda = (\lambda_1, \lambda_2, \lambda_3, ..\lambda_m)$ to the definition.

In [29], conflicts and fairness in Petri Nets were studied. There are many issues involved such as detecting conflicts, and determining whether a given Petri Net is prone to conflicts. In this thesis, we will use the method outlined in Figure 2.8 to resolve conflicts:

```
Conflict resolution Pseudo code
Input: The incidence matrix and the initial marking
If several n transitions are competing for the tokens at a place  over sampling period TS, do as follows:
    1. Generate r1 = Uniform [0, 1]. Compare r1 with Pr(<1 transition), Pr(<2 transitions), etc..
       Determine the largest number m1 of transitions allowed at this period
    2. Generate a random permutation using MATLAB randperm of [1, 2…n], say
       [3 2 5 4 1]. This is the order in which transitions will be checked for being enabled.
    3. Check whether the transitions are enabled in the order in 2, fire them with the underlying
       probability, then remove the tokens.
        a) Generate r2 from uniform [0, 1] and check if r2< lambda (3)/ (sum lambdas).
           If yes, fire T3 and remove the necessary tokens. Keep doing this (proceeding to
           T2, T5, etc.) until
             i.   We get more than the number of transitions m1 allowed in 1
             ii.  We run out of tokens
           Note: you can cycle through [3 2 5 4 1] until a or b occurs
```

Figure 2.8. Conflict resolution pseudo-code

2.2 Markov chains

2.2.1 Discrete-Time Markov Chains (DTMC)

The state of a Markov chain at time t is denoted X_t. The state space S of a Markov chain is the set of values that X_t can take. For example, $S = \{0, 1, 2, 3\}$ is a state space.

The Markov property of a Markov chain is that only the most recent point in the trajectory affects the future. Mathematically, the Markov property for all times t and states s could be stated as follow:

$$P(X_{t+1} = s \mid X_0 = s_0, X_1 = s_1, ..., X_t = s_t) = P(X_{t+1} = s \mid X_t = s_t). \tag{2.4}$$

The one-step state transition matrix P for the Markov model has the elements

$$P_{ij} = P(X_{t+1} = j \mid X_t = i) \text{ for } i, j \in \mathrm{S} \text{ and } t = 0,1,2,.. \tag{2.5}$$

The tth step transition matrix has elements

$$P(X_t = j \mid X_0 = i) = (P^t)_{ij}. \tag{2.6}$$

Note that a probability transition matrix is irreducible if the state space is a single communicating class.

Typically, the initial probability distribution of X_0 is denoted $\pi(0)$.

$$P(X_0 = i) = \pi_i(0). \tag{2.7}$$

And the probability distribution of X_t at time t with initial distribution π is given by:

$$P(X_t) = \pi_i(t). \tag{2.8}$$

Equilibrium is obtained when there is no change in the distribution:

$$\pi^T P = \pi^T. \tag{2.9}$$

2.2.2 Continuous-time Markov chains (CTMC)

Let $\{Z(t),\ t \geq 0\}$ represent a homogenous finite-state continuous time Markov chain with state space Ω. Let τ_{ij} represent the random variable time for a transition from state i to state j. We assume that τ_{ij} is exponentially distributed with parameter r_{ij}. The matrix $\left[r_{ij} \right]$ is called the rate matrix. Note that for all i we have $r_{ii} = 0$. From the rate matrix, one defines the infinitesimal generator Q matrix as follows:

$$\begin{cases} q_{ij} = r_{ij} \quad \text{if } \ i \neq j \\ q_{ii} = -\sum_{j \neq i} r_{ij} \end{cases} \tag{2.10}$$

The infinitesimal generator of a stochastic process contains a lot of information about the process. The unconditional probability of the CTMC being in state i at time t is:

$$\pi_i(t) = P(Z(t) = i). \tag{2.11}$$

The transient state probability vector of the CTMC is

$$\pi(t) = [\pi_1(t), \pi_2(t), \pi_3(t), \ldots]. \tag{2.12}$$

The behavior of the CTMC can be described by the following Kolmogorov differential equation:

$$\frac{d\pi(t)}{dt} = \pi(t)Q, \tag{2.13}$$

Whose solution is

$$\pi(t) = \pi(0)e^{Qt}$$

where the matrix Q is the infinitesimal generator; additionally, the steady state probabilities can be computed by:

$$\pi Q = 0, \ \sum \pi_i = 1. \tag{2.14}$$

The vector π_e is the left eigenvector of the matrix Q corresponding to the zero eigenvalue. The probability distribution is normalized (i.e. $\sum \pi_i = 1$). More information could be found in [30].

The one step transition matrix P of the discrete time Markov Chain equivalent to a continuous time Markov Chain with rate matrix R and infinitesimal generator Q is:

$$P = e^{QT_S} \tag{2.15}$$

where T_S is the sampling time. If T_S is small then:

$$e^{QT_S} = I + QT_S + \frac{\left(QT_S\right)}{2!} + ... \approx I + QT_S \tag{2.16}$$

Combining the above two equations we get

$$Q \cong \frac{P - I}{T_S}.$$

Thus, the matrix Q can be computed using the one step transition matrix and the time.

2.3 Hidden Markov Chains

In the Markov models the states are visible. In hidden Markov Chains the states are visible only through a random emission process, in the sense that observations can be non trivial functions of the states. Consider the Markov process in Figure 2.9 where

$X_0X_1X_2X_3$ is the Markov process and $O_0O_1O_2O_3$ are the observations. The matrix A is the transition matrix; the matrix B is the observation matrix which contains the state distribution of the observations given a Markov process X. The vector π is the initial Markov state distribution. Consider the example in which we have a system that has two states and three observations.

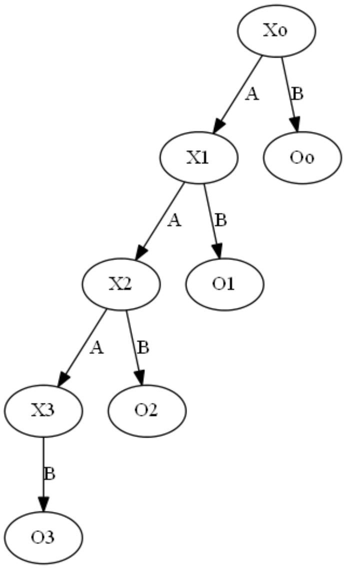


Figure 2.9. Hidden Markov Model (HMM) example

We wish to know the probability of a state sequence of length four:

$$X = (x_0, x_1, x_2, x_3). \qquad (2.17)$$

Given the observations:

$$A = \begin{bmatrix} .7 & .3 \\ .2 & .8 \end{bmatrix},\ B = \begin{bmatrix} .1 & .4 & .5 \\ .7 & .2 & .1 \end{bmatrix},\ \pi = \begin{bmatrix} .5 & .5 \end{bmatrix}, \text{ and } O = (o_0, o_1, o_2, o_3). \qquad (2.18)$$

Therefore:

$$P(X) = \pi_{x_0} b_{x_0}(o_0) a_{x_0,x_1} b_{x_1}(o_1) a_{x_1,x_2} b_{x_2}(o_2) a_{x_2,x_3} b_{x_3}(o_3). \qquad (2.19)$$

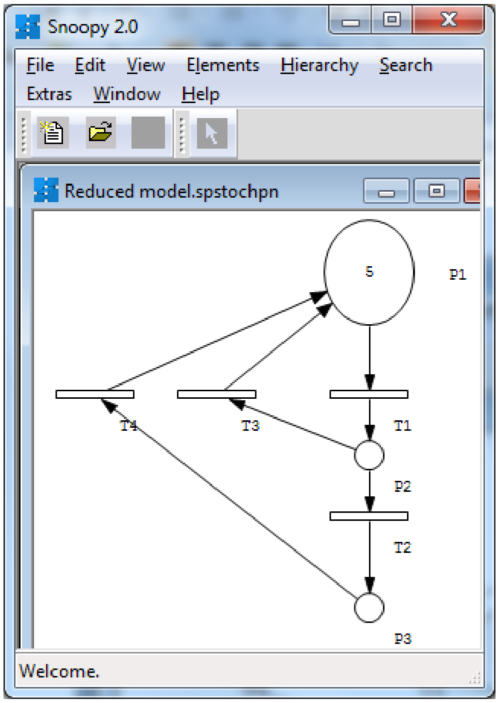
where O and X are given. Therefore, it seems that if we are given an observation it is possible to get the transition matrix. The MatLab command h_{mmesti} is able to estimate the transition matrix A and the observation matrix B given a collection of observations.

CHAPTER 3

MARKOV MODEL REPRESENTATION OF PETRI NETS

3.1 A motivating example

Figure 3.1 is a simple model of the network. When a WMD attack strikes the network (transition T_1 is in play), the working units in place P_1 go through a failure transition to become failed units in place P_2. Some failed units undertake repair transition T_3 to become working units again. However, other units stay failed units because of the failure propagation and go through transition T_2 to place P_3; from there they can be repaired through transition T_4 to become working units again.



Welcome.

Figure 3.1. The simple model snapshot from Snoopy 2.0

The elements of the pre-incidence matrix are $(D_-)_{ij}$ = multiplicity of the arc connecting place j to the input of transition i, and it is zero otherwise.

The pre-incidence matrix D_- and the post-incidence matrix D_+ are $m \times n$ matrices with m is the number of transitions and n is the number of places. The matrix D is simply the difference between the matrix D_+ and the matrix D_-:

$$D = D_+ - D_- \tag{3.1}$$

For example, for the transition T_1 there is only one input place P_1 with multiplicity one. Then $D_-(1,1) = 1$ but $D_-\left(1,2\right) = D_-\left(1,3\right) = 0$, and so forth. The resulting matrix is therefore:

$$D_- = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

In order to find D_+, we look to each transition output places and we report the multiplicity of that place to the corresponding transition in the post-incidence matrix. For example, for the transition T_1 there is only one output place P_2 with the multiplicity one then $D_+(1,2) = 1$ but $D_+\left(1,1\right) = D_+\left(1,3\right) = 0$, and so forth. The resulting matrix is therefore:

$$D_{+} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{3.3}$$

Finally, the resulting incidence matrix D is the following:

$$D = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \tag{3.4}$$

Now assume that T_1 fires which is represented by $F = [1, 0, 0, 0]$. the resulting marking is $M1 = [4, 2, 1]$. however, the same result could be obtained using the state's equation $FA + M_0 = M$ illustrated below:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} + \begin{bmatrix} 5 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 4 & 1 & 0 \end{bmatrix}. \tag{3.5}$$

3.2 Use of the Incidence Matrices

We further illustrate the use of the incidence matrices using the example in Figure 3.2, where some arcs have multiplicity larger than 1.

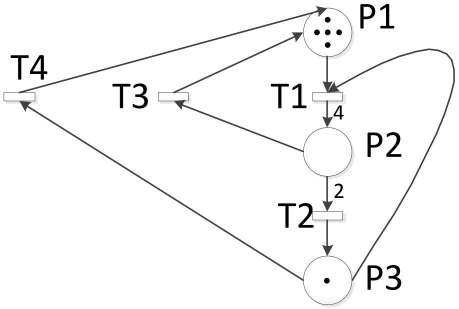


Figure 3.2. Petri Net example with 3 places and 4 transitions

The incidence matrices are

$$D^{-}=\begin{bmatrix}1 & 0 & 1\\ 0 & 2 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}, D_{+}=\begin{bmatrix}0 & 4 & 0\\ 0 & 0 & 1\\ 1 & 0 & 0\\ 1 & 0 & 0\end{bmatrix}.$$

And therefore,

$$D=D_{+}-D^{-}=\begin{bmatrix}0 & 4 & 0\\ 0 & 0 & 1\\ 1 & 0 & 0\\ 1 & 0 & 0\end{bmatrix}-\begin{bmatrix}1 & 0 & 1\\ 0 & 2 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}=\begin{bmatrix}-1 & 4 & -1\\ 0 & -2 & 1\\ 1 & -1 & 0\\ 1 & 0 & -1\end{bmatrix}$$

Transition T_i is enabled at a marking M if and only if: $D_{ij}^{-}\leq M\left(p_j\right), j=1, 2,\cdots, m.$

The marking update equation, often called the state equation for a Petri net, is

$$M_k = M_{k-1}+D^T u_k\ ,\ k=1,2,... \tag{3.6}$$

u_k is the firing vector

In the example above, the initial marking is $M_0 = [5\ 0\ 1]$ and T_1 is enabled because $\left[5\ 0\ 1\right] \geq \left[1\ 0\ 1\right]$.Using the state equation, one obtain the updated marking

$$M = [5\ 0\ 1] + [1\ 0\ 0\ 0]\begin{bmatrix} -1 & 4 & -1 \\ 0 & -2 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \rightarrow M = [4\ 4\ 0] \qquad (3.7)$$

3.3 Markov Model Representation (MMR) of a Petri Net

If all transitions in a Petri Net are conservative (the number of tokens in the net never increases), or if token-increasing transitions do not occur in a cycle, and the initial number of markings is less than a pre-specified number, then there is a finite number of possible markings that can be produced by the simulation. Not all the possible markings will actually occur, and some may be extremely improbable (in STPN). A Markov model representation can then be found where the nodes are the occurring markings (called the states of the Markov chain) and the arcs represents the direct transitions from one marking to another.

A Petri Net example is shown in Figure 3.3 allowing only 3 tokens or less in the Petri Net.

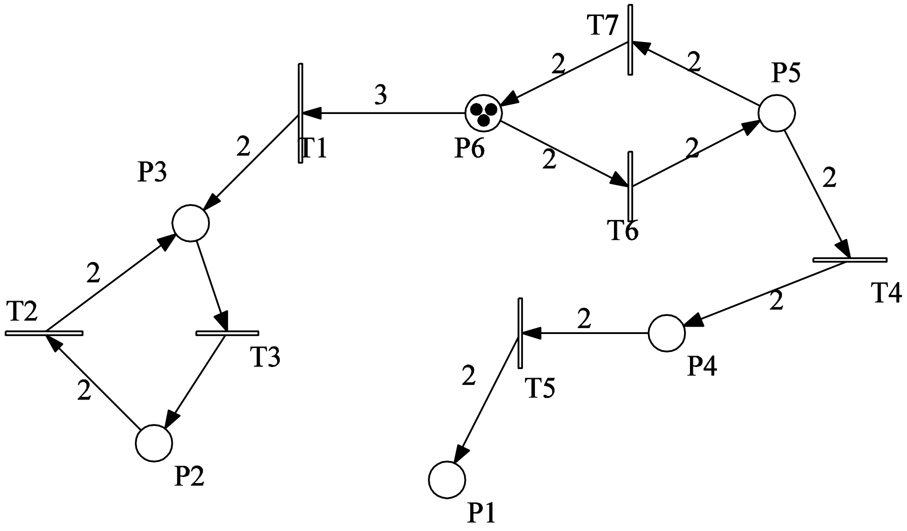


Figure 3.3. A Petri Net example constructed in Snoopy

It is obvious that the three tokens will be distributed according to transitions. In fact, each time a transition is in play a different distribution is reached. Table 3.1 lists all the possible markings that can result.

Table 3.1. The different markings reached in the Petri Net example

Marking	Tokens
1	[0 0 0 0 0 3]
2	[0 0 2 0 0 0]
3	[0 0 0 0 2 1]
4	[0 1 1 0 0 0]
5	[0 0 0 2 0 1]
6	[0 2 0 0 0 0]
7	[2 0 0 0 0 1]

Careful consideration will show that the transition diagram containing the different markings can be summarized in Figure 3.4. The number on the arcs denotes the transition needed. For example, starting with the marking $M_1 = [0,0,0,0,0,3]$ and following transition 6 (moving two tokens from P_6 to P_5), will lead to marking $M_3 = [0,0,0,0,2,1]$. Moreover, one can assign a rate λ_i to transition T_i, thus leading to a stochastic Petri Net. The stochastic Petri Net transitions in Figure 3.3 can be then represented by a CTMC in Figure 3.4

$$R = \begin{bmatrix} 0 & \lambda_1 & \lambda_6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_3 & 0 & 0 & 0 \\ \lambda_7 & 0 & 0 & 0 & \lambda_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda_5 \\ 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.8}$$

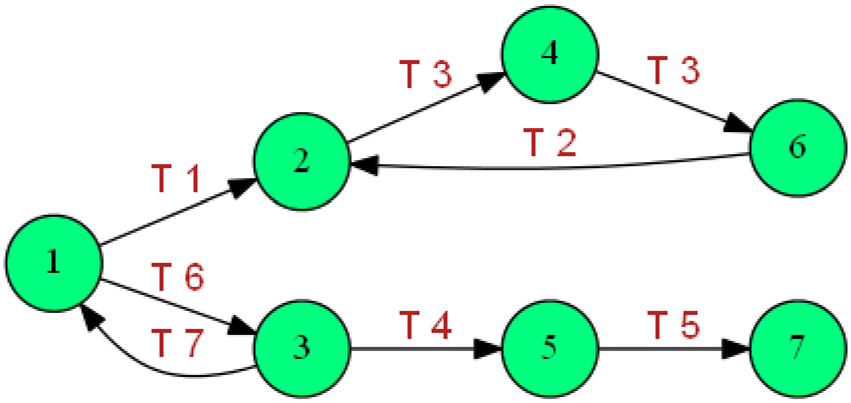


Figure 3.4. The Markov Chain representation of the Petri Net in Figure 3.3

A Markov model transition matrix Z can be defined in terms of the rate matrix as follows

$$\begin{aligned} R_{ij} > 0 &\Leftrightarrow Z_{ij} = 1 \\ R_{ij} = 0 &\Leftrightarrow Z_{ij} = 0 \end{aligned} \tag{3.9}$$

The transition on Zero-pattern matrix Z for Figure 3.4 is:

$$Z = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The reachability captures the system's ability to get from one marking to another in one or more steps. The reachability graph is a digraph that contains all the possible markings of Petri Net as nodes. An arrow from node i to node j explained earlier indicates that there is a sequence of feasible transitions that allow marking i to reach marking j.

The reachability graph can be represented by the matrix:

$$W_{ij} = \begin{Bmatrix} 1 \text{ if } \; Z^t(i,j) > 0 \;\text{ for some t} \geq 0 \\ 0 \;\text{ otherwise} \end{Bmatrix}$$

It can be shown [31] that the nonzero elements of W are the nonzero elements of: $I + Z + Z^2 + ... + Z^{n-1}$ with n is the dimention of Z. Hence one can use the MATLAB code: $W = (\text{eye}(\text{size}(Z,1) + Z)^{n-1} > 0$.

If a state i is reachable from state j and vice versa, then they are said to belong to a communicating class. Such a class is the set of all states that communicate with each other or with a class representative. If there are m communicating classes then there are m representative markings: $U_1,...,U_m$. This is summarized in the matrix C where:

$$C_{ij} = \begin{Bmatrix} 1 \text{ if state } j \text{ is in class } i \text{ includes} \\ 0 \text{ otherwise} \end{Bmatrix} \tag{3.10}$$

For the Markov model in Figure 3.4 there are 4 classes:

$$\{2,4,6\},\{1,3\},\{5\}, \text{ and } \{7\} \tag{3.11}$$

which can be represented by the list of class representatives:

$$U = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

3.4 Algorithm to Find the MMR for Petri Net Incidence Matrices

The idea behind an algorithm that computes the Markov representation (or state transition matrix Z is to count all the states reachable from the initial state M_0 in the current system in study. We start from an initial state M_0, then we record all the unique states obtained by firing enabled transitions. For each new state we repeat the same process. This algorithm makes the assumption that the Petri Net is bounded, i.e. the state space is finite.

Figure 3.5 details the steps followed to obtain the transition matrix. At every iteration, the algorithm has a current marking list and a list of considered markings L_c. If the current marking has not been considered before, it is then considered and appended to

L_c after updating the adjacency or transition matrix Z. The algorithm searches for new markings by finding enabled transitions. The markings found are appended to the list L_N of markings to be considered. Then the algorithm iterates until no new marking is found and L_N is empty. Arcs are added to the graph when an enabled transition takes marking i to marking j. The flow chart in Figure 3.5 offers more details concerning this algorithm.

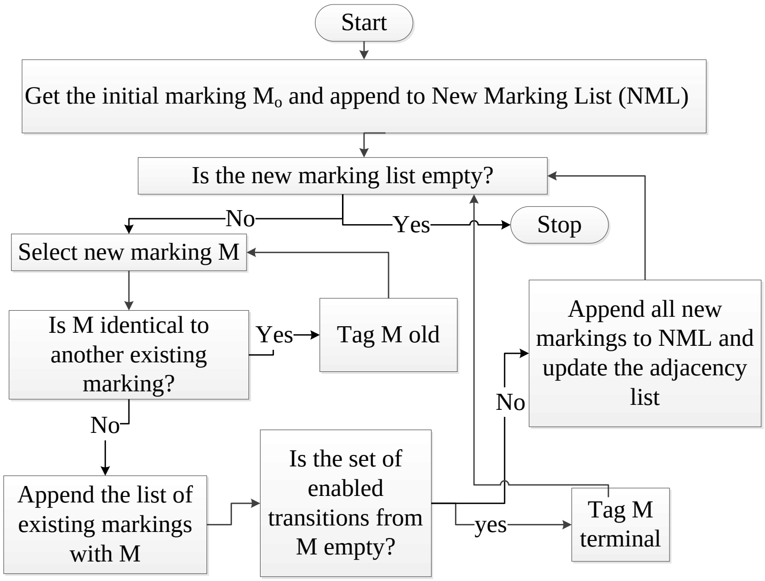

Start
Get the initial marking M$_o$ and append to New Marking List (NML)
Is the new marking list empty?
No
Yes
Stop
Select new marking M
Is M identical to another existing marking?
Yes
Tag M old
Append all new markings to NML and update the adjacency list
No
No
Append the list of existing markings with M
Is the set of enabled transitions from M empty?
yes
Tag M terminal

Figure 3.5. Algorithm for the representation of the Petri Net by a Markov model

The program resulting from the above algorithm works fine with small nets like the one shown in Figure 3.3. However; when the net starts to be bigger like the Fault model, the program is not efficient in the sense that we got a huge data. In fact, the data obtained is 2020 markings in 3hours the time we left the program running.

3.5 Steady-State of the MMR

The rate matrix $R,$ is defined in Equation (3.8) and corresponding to Figure 3.3 is

$$R = \begin{bmatrix} 0 & 1 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 7 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, E = \begin{bmatrix} 7 \\ 3 \\ 11 \\ 3 \\ 5 \\ 2 \\ 0 \end{bmatrix}$$

Where E(i)=sum(R(i,:)) therefore, from equation (2.10) Q becomes:

$$Q = \begin{bmatrix} -7 & 1 & 6 & 0 & 0 & 0 & 0 \\ 0 & -3 & 0 & 3 & 0 & 0 & 0 \\ 7 & 0 & -11 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & -3 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & -5 & 0 & 5 \\ 0 & 2 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Since $\pi(t) = \pi(0)e^{Qt}$, the steady state probability can be written as:

$$\pi_\infty = \lim_{t\to\infty} \pi(t) = \left[0, 0.089, 0, 0.089, 0, 0.134, 0.685\right].$$

Another way to compute the steady state probability is to simulate the dynamics of the CTMC model. In fact, the impulse response of a single-input state-space model can be used:

$$\rho = \pi^T$$
$$\frac{d\rho}{dt} = Q^T \rho(t) + \pi^T(0)u(t) \qquad (3.12)$$
$$y(t) = \rho(t)$$

The above equation is equivalent to the following unforced response with initial state $\pi(0)$:

$$\frac{d\pi}{dx} = \pi Q, \text{ and y=}\pi \qquad (3.13)$$

Figure 3.6 illustrates the history of the state probabilities. Note that the steady state probability vector is found to be the same as π_∞ .

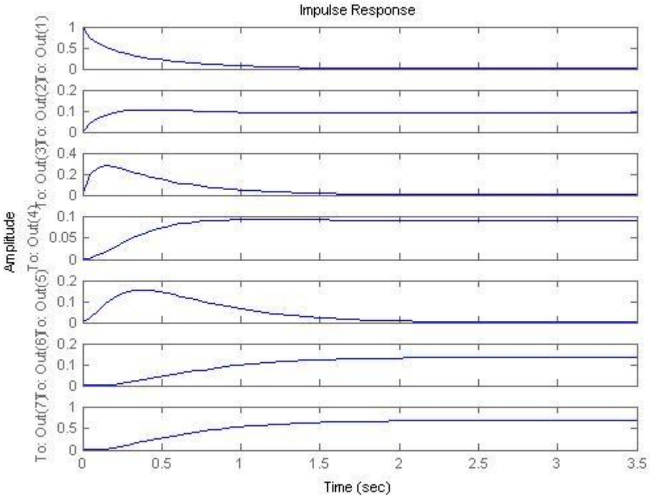


Figure 3.6. State trajectory as an impulse response

3.6 Estimating the Probability Transition Matrix from Monte Carlo Simulations

The method in Section finds a Markov model containing all the possible states no matter how improbable they are. The method proposed here focuses on the probable states and is based on an intuitive understanding of the probability transition matrix. Essentially, one can estimate the probability P_{ij} by counting how many times state j occurred immediately after i in a set of simulations or Monte Carlo runs, and then dividing by the total number that state i occurred. The implementation is summarized in Figure 3.7.

We simulate the Petri Net example in Figure 3.3 using Snoopy using a step size $T_S = 0.01 \text{ for } 10000 \text{ steps}$ .We used two different methods in order to compute the transition matrix P and the infinitesimal generator Q from the simulation files.

```
Pseudo code to compute the Q matrix
Input: data from Snoopy software
1. Simulate the Petri net for a given number of Monte Carlo runs
2. Collect the history of the markings and determine the matrix M of the unique rows
3. Initialize a zero square matrix P that has the same size as the number of rows in M
4. For each Monte Carlo run collect the marking trajectory in matrix G and do the following:
    4.1 For all m,n find the number of times where a marking n follows the marking m
        immediately. This is achievable using the MATLAB command K =FINDSTR (G', [m n])
    4.2 Add K to the corresponding entry in P(m,n)
5. Normalize every row of P by the sum of the elements on the row
6. Q=(P-I)/Δ   where Δ is the sampling period and I is the identity matrix
7. Return P and Q
```

Figure 3.7. Algorithm pseudo-code to compute P and Q

Alternatively, we use MATLAB’s command hmmestimate which computes the same probability using a more sophisticated method. The implementation is shown in Figure 3.8.

```
Alternative Pseudo code to compute the Q matrix
 Input: data from Snoopy software
        Given M (labels of distinct markings)
         Given G^1, G^2… the collection of marking trajectories for MC run 1,2,….
   1.  For the j^th MC run,  j=1,2,…
            1.1 Transform G^j into g^i history of labels as named in M
            1.2 Let g^i be the i^th row of F
   2.  [P,O]=hmmestimate(F,F);
   3.  Q=(P-I)/Δ   where Δ is the time step and I the identity matrix
   4.  Return P and Q
```

Figure 3.8. Alternative algorithm to compute P and Q

Using the algorithm in Figure 3.7 one obtains the matrices $\hat{P}$ and $\hat{Q}$ using a sampling period $T_S = 0.01$. The estimated probability transition matrix is:

$$\hat{P} = \begin{bmatrix} 0.90 & 0.04 & 0.039 & 0.003 & 0 & 0 & 0 \\ 0 & 0.95 & 0 & 0.04 & 0 & 0 & 0 \\ 0.01 & 0 & 0.94 & 0 & 0.03 & 0 & 0 \\ 0 & 0 & 0 & 0.95 & 0 & 0.04 & 0 \\ 0 & 0 & 0 & 0 & 0.96 & 0 & 0.03 \\ 0 & 0.05 & 0 & 0 & 0 & 0.94 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.99 \end{bmatrix}, \qquad (3.14)$$

And the corresponding semi-infinite generator is

$$\hat{Q} = \begin{bmatrix} -9.09 & 4.74 & 3.95 & 0.39 & 0 & 0 & 0 \\ 0.01 & -4.66 & 0 & 4.51 & 0 & 0.13 & 0 \\ 1.54 & 0 & -5.15 & 0 & 3.60 & 0 & 0 \\ 0 & 0.17 & 0 & -4.87 & 0 & 4.68 & 0 \\ 0 & 0 & 0 & 0 & -3.15 & 0 & 3.15 \\ 0.01 & 5.61 & 0 & 0.14 & 0 & -5.76 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (3.15)$$

where $Q \cong (\hat{P} - I) / T_S$.

The above algorithm tends to shuffle the states. The transition matrix is close to the identity matrix. This is explained by the fact that the sampling period T_S is quite small compared to the transition times.

Once we have the transition matrix, it is possible to trace back the transition diagram by simply drawing an arc from one state to another where the probability of transition is greater than zero. As shown in Figure 3.9, the transition diagram the states were shuffled compared to the transition diagram in Figure 3.4.

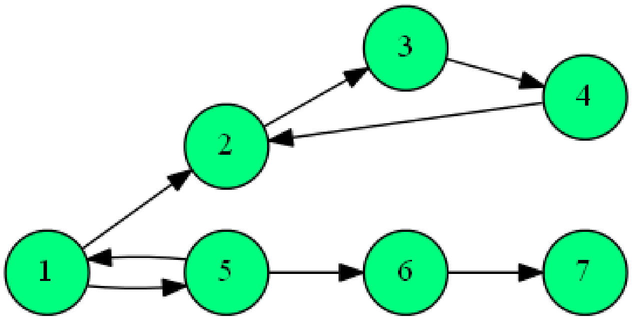


Figure 3.9. Markov Model transition diagram for the Petri Net example

3.7 Analysis of Communicating Classes

Given the zero-pattern matrix Z of a Markov model, one can compute the communicating classes defined in Equation(3.10). Moreover, one can determine the class-transition diagram which is a directed tree showing transitions from the so-called top strongly-connected components (or classes) of the Markov transition graph to the Bottom-Strongly Connected Components (BSCC). A BSCC is a “sink” or “absorbing class” and a state entering a BSCC never leaves it. The class transition can be described by the class transition matrix H

$$H_{ij} = \begin{cases} 1 \text{ if class } i \text{ transits to class } j \\ 0 \text{ otherwise} \end{cases} \tag{3.16}$$

Note that the matrix H is similar to the matrix Z , and is a reduced version of it. A BSCC is recognized as a class that has no outgoing transitions. This corresponds to a zero row in H . The algorithm is shown in Figure 3.10.

```
Communicating classes Pseudo code
Input: The transition matrix of size n×n
    1. Compute the adjacency matrix (Z=double(p>0))
    2. Compute the reachability matrix (W=(eye(n) +Z)^(n-1))
    3. Compute the communicating classes (C= unique(W &W', 'rows'))
    4. Compute the image matrix (H=C*Z*C’) and replace H by H=H& ~eye(size(H,1))
    5. Eh=(sum(H,2)==0) returns the indices of the BSCCs
```

Figure 3.10. Pseudo-code to compute Communicating classes

The class transition diagram for Figure 3.3 is shown in Figure 3.11. Note that the numbering represents classes and not states.

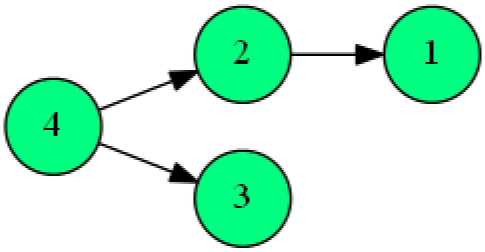


Figure 3.11. The Class transition diagram for the petri Net example in Figure 3.3

The following matrix relates class to states:

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (3.17)$$

Here $C_{ij} = 1$ if state j is in communicating class i and 0 otherwise.

That is class 1= {state 7}, class 2= {state 5}, class 3= {state 2, state 4, state 6}, and finally class 4= {state 1, state 3}.

CHAPTER 4

A HIERARCHICAL FAULT MODEL FOR WMD ATTACKS

We limit our research to the three interaction infrastructure systems; namely the Information systems, the Transportation systems, and the Health systems.

4.1 A Hierarchical Model

The hierarchical model shown in Figure 4.1 illustrates how different systems are individually affected by WMD events (e.g. dirty terrorist bomb, nuclear incidents of different scales, massive nuclear explosion, high altitude nuclear explosion leading EMP, etc.). The second level of this hierarchical model captures WMD effects on the inter-dependence of different CISs and the propagation of failures among different systems subjected to WMD events.

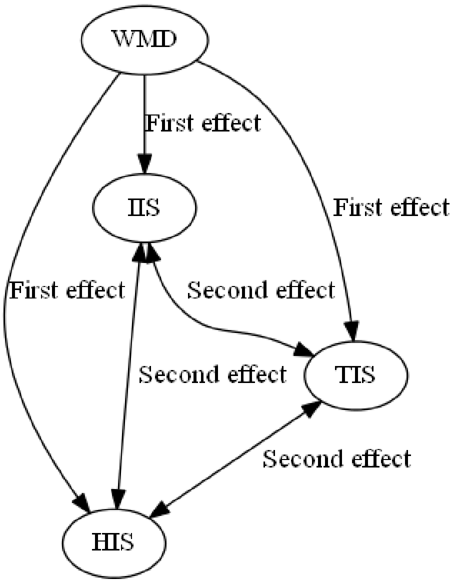


Figure 4.1. A Hierarchical model with first-order and second-order effects

The directed graph model shown in Figure 4.2 captures the interdependence among CISs. In more details, this also identifies different variables (e.g. Repair needed alert, Travel time etc.) We reported in the graph just few variables in order for the graph to be more readable. The Inter-Infrastructure Dependence graph focuses on the second effect caused by the WMD attack. We are able to identify many variables. For example, Traffic management, Worker's transportation, Access to records, Worker's health. Those variables are weakened when the WMD attack targeted the corresponding substructure.

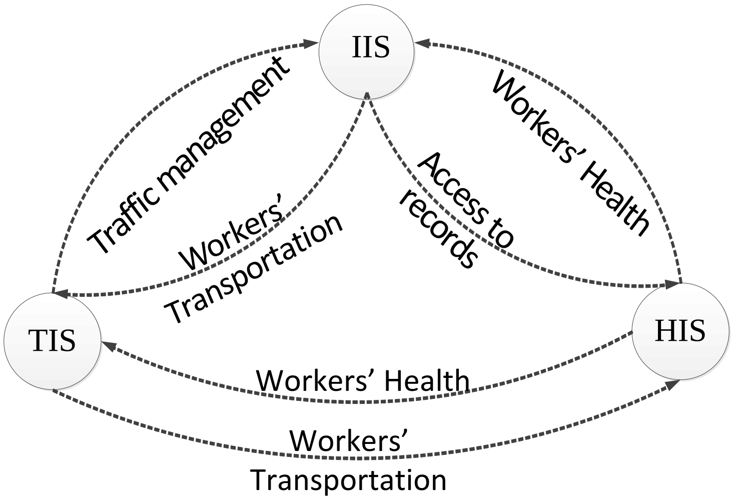


Figure 4.2. Inter-Infrastructure dependence graph

4.1.1 The Health Infrastructure Subsystem (HIS)

The Health Infrastructure Subsystem as shown in Figure 4.3 consists of healthy people that get injured, get transported to the hospital, get proper medication, and finally become healthy people again. For simplification we considered that dead people are going to be replaced by healthy people. Also, we considered Doctors, Ambulances, and Nurses resources as invulnerable.

The HIS has 40 people. When one person gets injured, he or she became disabled. The disabled person has to wait an exponentially-distributed random time with rate λ_H. After an exponentially-distributed random time, he or she will receive medication to become healthy again. This process is done indefinitely and it is done for one token at a time. This is a simplifying assumption.

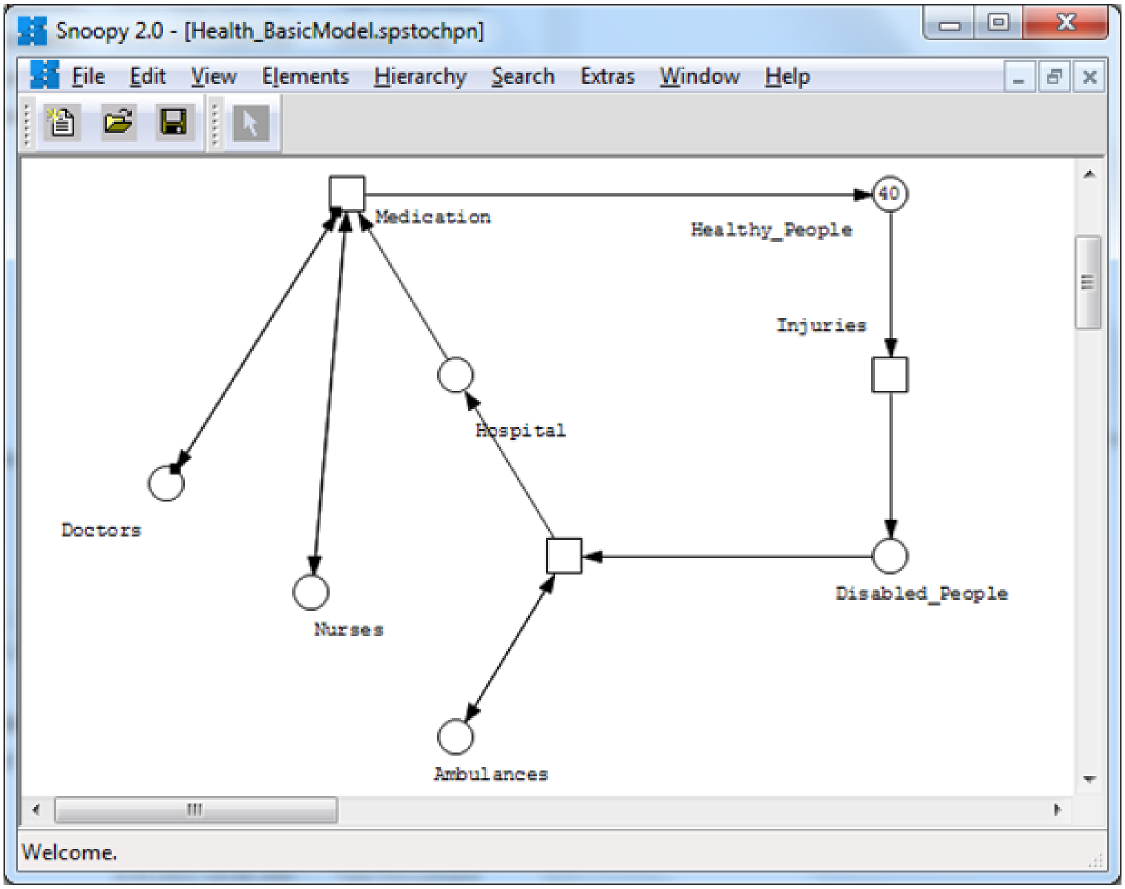


Figure 4.3. Health Infrastructure Model

4.1.2 The Information Infrastructure Sub-system (IIS)

The Information Infrastructure Subsystem is shown in Figure 4.4. It has many working information units. Examples of units which can fail and be repaired are computers, instructions, stored facts, and procedures. The transitions in this model are stochastic transitions with rates λ_F and λ_R.

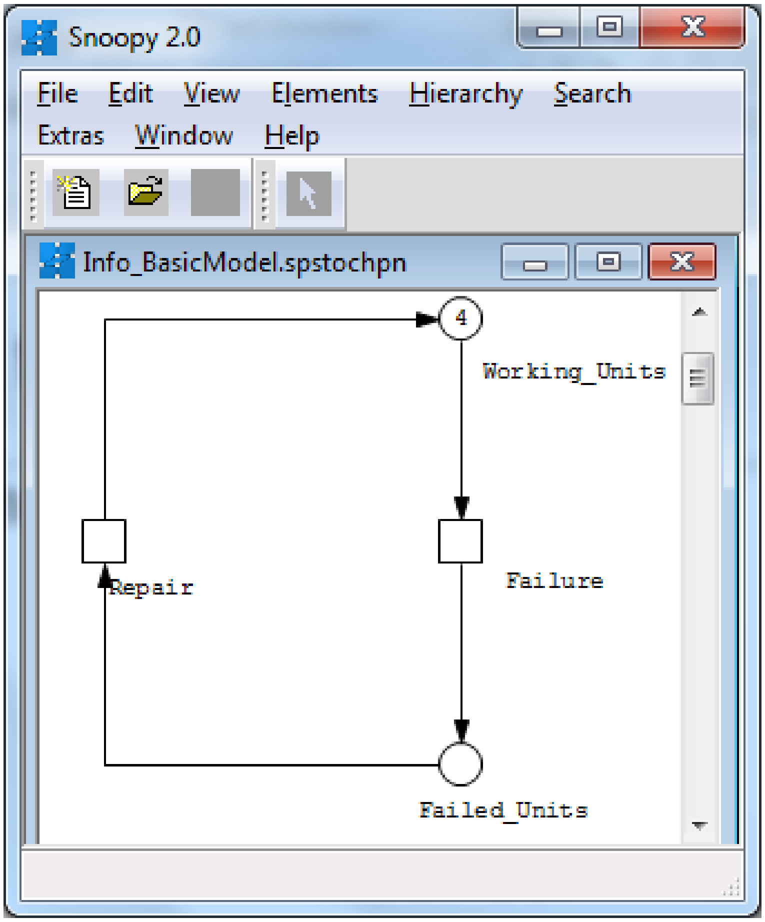


Figure 4.4. Information Infrastructure Model

4.1.3 The Transportation Infrastructure Subsystem (TIS)

The Transportation Infrastructure Subsystem as shown in Figure 4.5 has drivable roads that get damaged, get repaired using materials and workers, after which the roads become drivable again. In this model, we consider that the shipment transition is always enabled and hence the Materials are always available. We also considered that workers are always available.

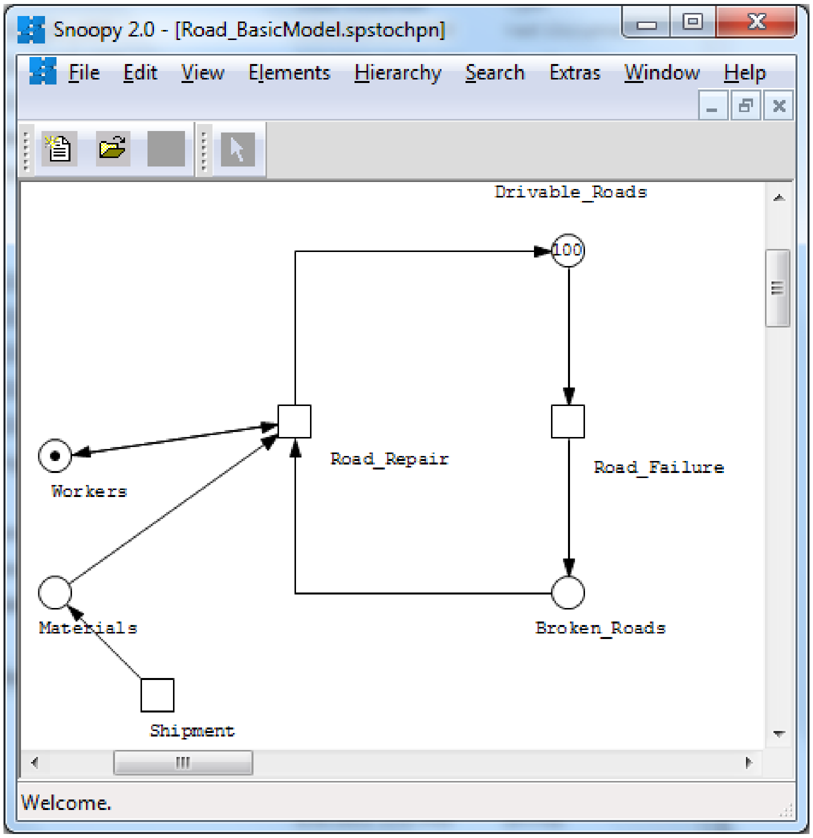


Figure 4.5. The Transportation Infrastructure Model

4.2 The Weapon of Mass Destruction Event

A Weapon of Mass Destruction (WMD) is assumed to damage all the subsystems at the same time. When WMD attack hits the IIS, TIS, and HIS are first affected separately. Figure 4.6 shows graphically how the three sub-systems will be affected by a WMD attack. This representation ignores the cascading effect that might result because of the attack. Note that the place WMD_transition will replace one token with 3. But this duplication is a onetime event only, and the Petri Net is bounded.

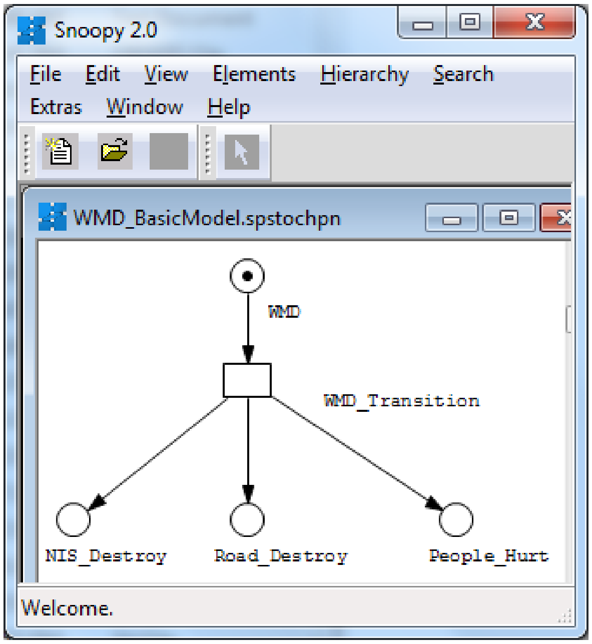


Figure 4.6. WMD Event Model

Table 4.1 lists the software names of the places in the Petri Net used in the simulations, explains them, and lists their corresponding initial markings. Working_Units denotes the number of working units in the IIS. These could be hubs, routers, etc. The Failed_Units are the information failed units. They could be as before hubs, routers, computers, etc. The place Hospitals represents the number of hospitals but here only one hospital is considered. Healthy_People represents the number of people available. The other variables follow the same logic.

Table 4.1. The different places in the hierarchical fault model (HFM) and their explanations

Place	Id	Explanation	Initial marking
Failed_Units	1	IIS failed units	0
Workings_Units	2	IIS working units	5
WMD	3	The WMD attack	1
NIS_Destroy	4	The IIS receive the WMD attack	0
People_Hurt	5	The number of people hurt	0
Road_Destroy	6	The number of roads destroyed	0
Ambulances	7	The number of ambulances available	1
Hospitals	8	Contains the disabled people	0
Healty_People	9	The number of healthy people	10
Disabled_People	10	The number of disabled people	0
Drivable_Roads	11	The number of drivable roads	10
Broken_Roads	12	The number of broken roads	0
H_Server_Down	13	Hospital server down	0
T_Server_Down	14	Transportation server down	0
Wait_Room	15	The waiting room	0
T_Grid_Broken	16	Transportation grid broken	0
T_Grid_Online	17	Transportation grid is online	0
H_Server_Online	18	The hospital server is online	0

Table 4.2 explains the stochastic transitions involved. Those transitions are stochastic and follow the exponential distribution.

Table 4.2. Stochastic transitions and their explanations

Stochastic transition	**Id**	**Explanation**	**Rate**	**average time between transitions**	**Input places (multiplicity)**	**output places (multiplicity**
Unit_Repair	1	IIS unit repair	1/(4*7)	4 weeks	9(5),1,11	9(5),2,17,18,11
Unit_Failure	2	IIS unit failure	1/(365*2)	2 years	2	1,13,14
WMD _Transition	3	Enable the WMD attack	1	1 day	3	4,6,5
Trans_to_hosp	4	transit disabled people to Hospital	1/(1/24)	1 hour	10,11,7	8,11,7
Injuries	5	people get injured	1/(7*4*6)	6 months	9	10
Medication	6	People get medication	1/(14)	14 days	8	9
Road_Failure	7	Road not drivable	1/365	1 year	11	12
Road_Repair	8	Road become drivable	1/(2*4*7)	2 months	12,9(5)	11,9(5)
T_Grid_Repair	9	Transportation server get repaired	1/(1/24)	1 hour	16,17	11
H_Server_Repair	10	Hospital server get repaired	1/(1/24)	1 hour	15,18	8
T_Grid_Failure	11	2nd effect of WMD over TS	240	6 minutes	11,14	16
TS_failure	12	1st effect of WMD over TS	240	6 minutes	6,11	12
H_Server_Failure	13	2ndeffect of WMD over HS	240	6 minutes	8,13	15

Table 4.2 *(cont.)*

IS_failure	14	1st effect of WMD over IS	240	6 minutes	2,4	1,13,14
HS_failure	15	1st effect of WMD over HS	240	6 minutes	5,9	10

4.3 The overall Model in Snoopy

Now we have enough information to build the aggregated model shown in Figure 4.7.

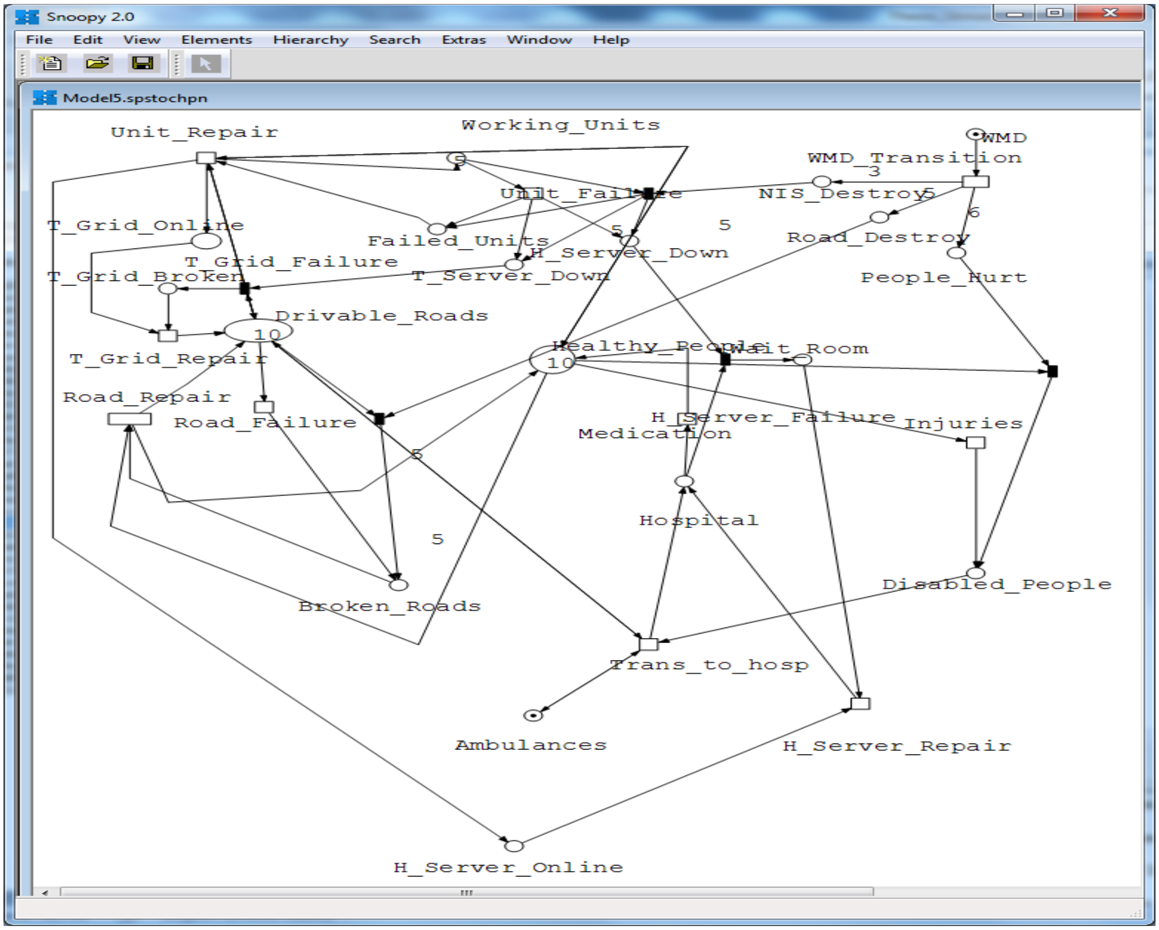


Figure 4.7. Stochastic Petri Net Model for the hierarchical Fault Model

CHAPTER 5

ANALYSIS USING MONTE CARLO SIMULATIONS

5.1 Monte Carlo simulation description

Monte Carlo simulation is a problem solving technique that approximates the probability of certain outcomes by running multiple simulations. We have performed 100 Monte Carlo simulations using the Snoopy software, each for 300 days. The sampling period was one hour.

5.2 The output data from Snoopy software

Each output from Snoopy is a csv file. Each of the 100 files contains 7202 rows, and each row except the header row contain the number of tokens in every place (column). Table 5.1 shows 12 rows extracted from the first simulation file. Table 5.2 lists the headers of the columns which they represent Time, Failed_Units…etc.

Table 5.1. Sample data (12) from simulation file 1

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	0	5	1	0	10	0	10	0	0	0	0	0	0	0
1	3	2	0	0	4	6	2	5	3	0	0	3	0	0
2	3	2	0	3	4	0	2	5	0	0	3	3	0	0
3	3	2	0	3	4	0	2	5	0	0	3	3	0	0
4	3	2	0	3	4	0	2	5	0	0	3	3	0	0
5	3	2	0	3	4	0	2	5	0	0	3	3	0	0
10	3	2	0	3	4	0	2	5	0	0	3	3	0	0
50	3	2	0	0	7	0	2	5	0	0	3	3	0	0
100	3	2	0	0	7	0	7	0	0	0	3	3	0	0
200	0	5	0	0	10	0	10	0	0	0	0	0	0	0

Table 5.1 *(cont.)*

250	0	5	0	0	10	0	10	0	0	0	0	0	0	0
300	0	5	0	0	10	0	10	0	1	0	0	0	0	1

Table 5.2. The list of the headers in csv files (The places in the STPN)

Time	Failed_Units	Working_Units
WMD	Hospital	Healthy_People
Disabled_People	Drivable_Roads	Broken_Roads
H_Server_Down	T_Server_Down	Wait_Room
T_Grid_Broken	T_Grid_Online	H_Server_Online

5.3 Statistical Analysis of the Monte Carlo Simulation

5.3.1 The Tokens distribution

We concentrate our analysis effort on three main places: Working Units, Healthy People, and Drivable Roads. The histograms shown in Figure 5.2 suggest that the IIS has recovered as well as HIS. However, the TIS do not seem to have recovered. This is explained by the fact that roads need a long time to recover.

Figure 5.1 shows the average (over 100 Monte Carlo runs) time history of the number of tokens in the places Working_Units, Healthy_People, and Drivable _Roads. The top graph is an average over 50 simulations, and the subsequent graphs are for 75 and 100 Monte Carlo runs, respectively. Since there is little difference between using 75 and 100 simulations, we conclude that 75 Monte Carlo simulations are enough.

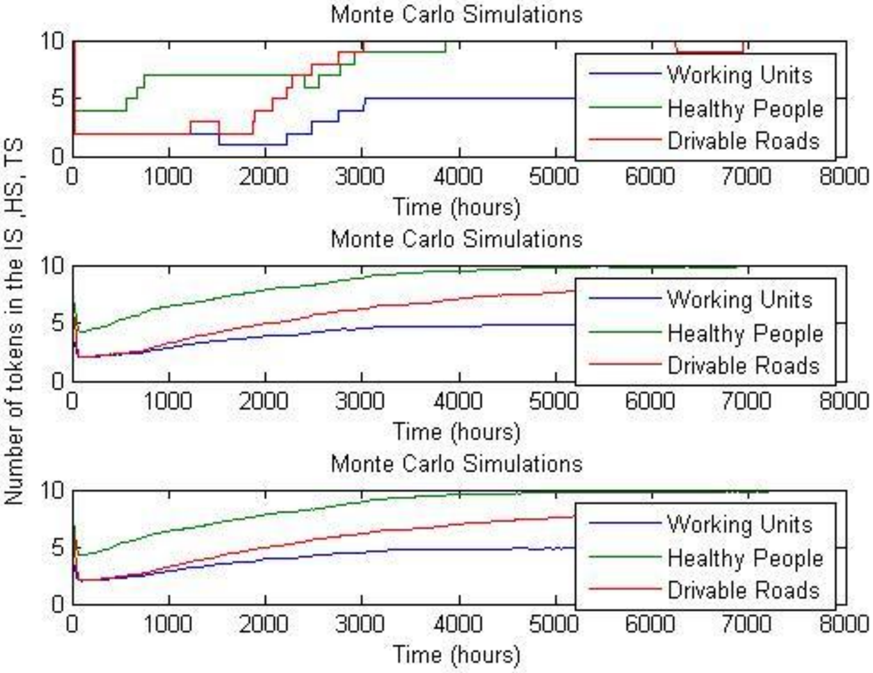


Figure 5.1. Monte Carlo simulations averaged

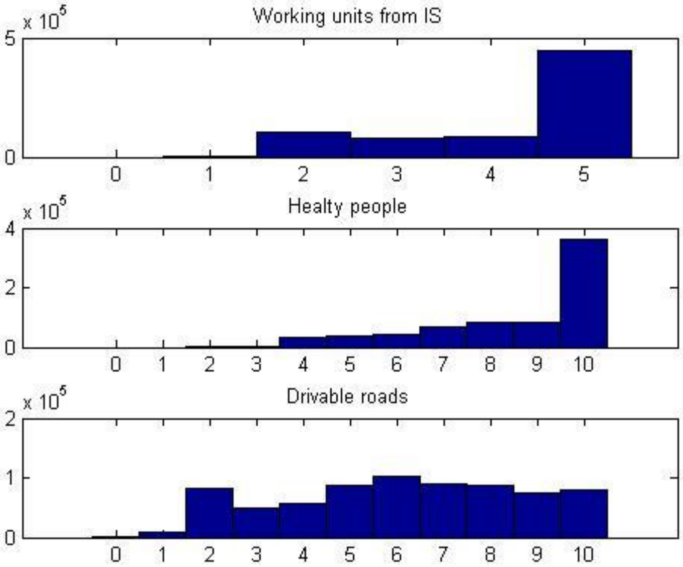


Figure 5.2. Token distributions for the main 3 places

Next we show the correlation between the numbers of tokens in all the places. The images of the resulting 18×18 matrices are shown in Figure 5.3. This Figure shows four plots, which correspond to four different simulations. The four plots are quite similar. For each plot we observe that the locations of the white squares are almost the same. This tells us that there is a strong correlation between the variables of those squares. However, a dark square suggest a weak correlation between those variables. Figure 5.4 shows the correlation image of the resulting average of all 100 simulation files.

The correlations were computed as follows. Let $x_i(t)$ represent the number of tokens in place i at time t. Then the data matrix X has $X_{ij} = x_i(jT_S)$ and the correlation matrix is computed using the MATLAB's command corrcoef.

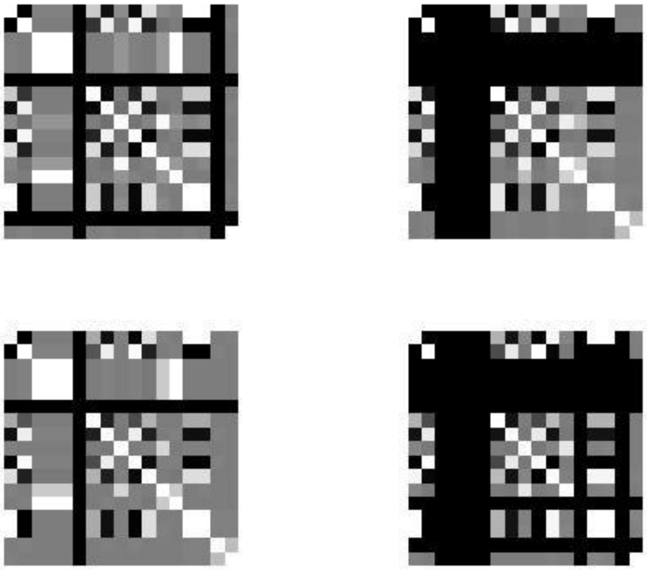


Figure 5.3. Images of correlation matrices based on 4 Monte Carlo runs

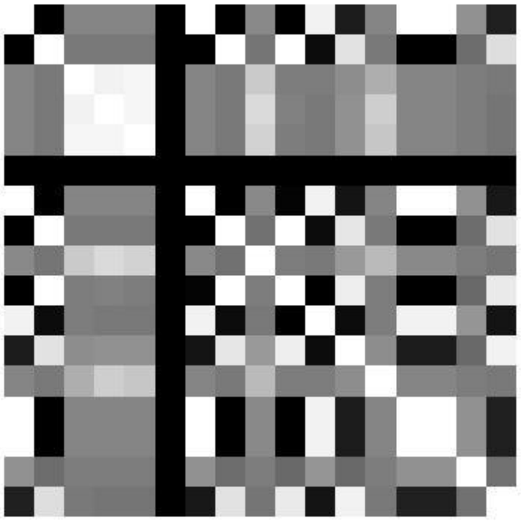


Figure 5.4. Image of the average correlation matrix

5.3.2 The Communicating Classes

Next, we apply the algorithm shown in Figure 3.7 to obtain the transition matrix P . From there we get the Zero-pattern matrix Z by applying estimate Equation(3.9). From there we obtain the transition graph of the Markov Chain Representation of the Hierarchical Fault Model (HFM) using the algorithm shown in Figure 3.10. Figure 5.5 shows the resulting representation.

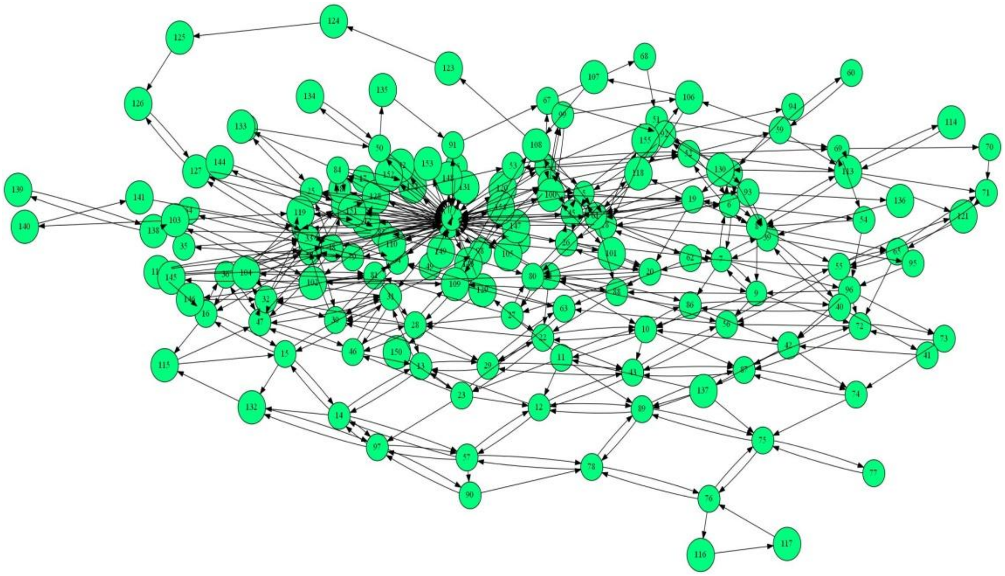


Figure 5.5. Transition graph of the Markov Chain Representation of the HFM

Once the zero-pattern matrix Z is obtained, one can compute the communicating classes by applying the algorithms shown in Figure 3.10. The result is the reduced graph shown in Figure 5.6. This figure shows that class 72 is the only BSCC. The normal working condition states (states 35 and 220) are members of class 72. Figure 5.7 is the image of $P-I$ matrix. We note from the image that many states have low probability or never occur.

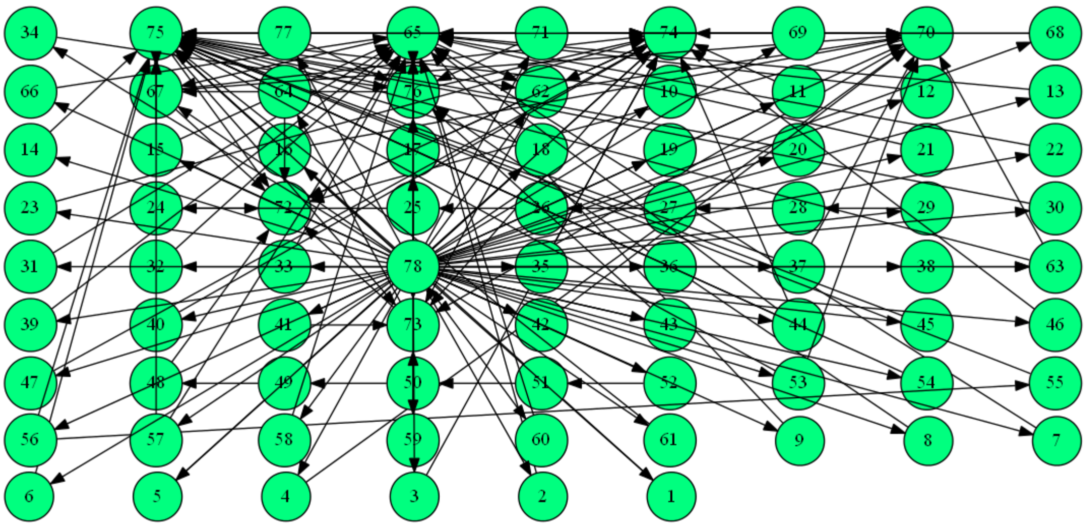


Figure 5.6. Class representation of the hierarchical fault model

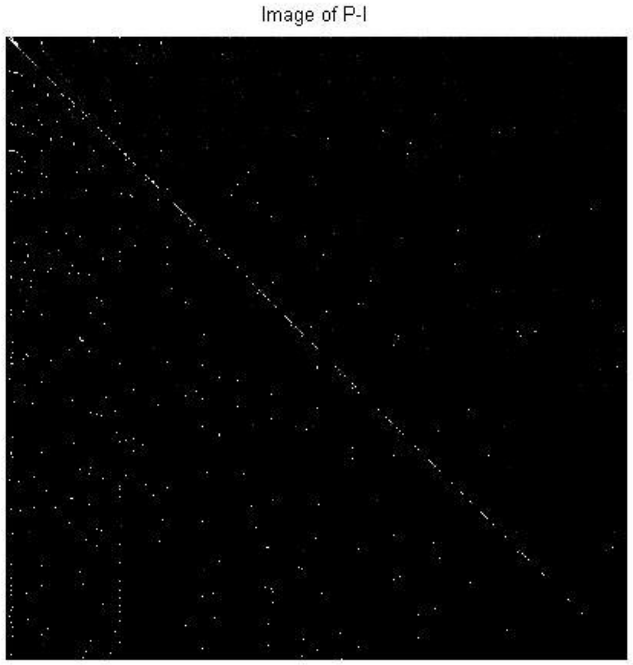


Figure 5.7. Image of P-I matrix

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

We have developed a methodology to capture the interdependency between the different subsystems of the critical infrastructure systems using stochastic Petri Nets. The modeling combines common sense understanding of what happen while specific data are available.

We have subsequently developed a methodology that can extract the most probable marking from Monte Carlo Simulations of the resulting Petri Net and to obtain a far simpler and more Markov model. The one-step transition matrix, the probability transition matrix, the reachability graph, the class transition matrix and the BSCC classes were successfully obtained using reliable algorithms. Simulations shows that the HFM model developed recovers in a reasonable amount of time and that the normal states are attracting in the long term.

In future work, we will extract more information from the infinitesimal generator, such as time needed to move from one state to another. Also, future work will compare different designs of the network in order to determine which configuration is more robust.

REFERENCES

[1] Z. Glenn, "A New World of Terror," *IEEE SPECTRUM*, pp. 24-25, October 2001.

[2] G. James and H. Paul, "An engineering model for managing counter-terrorism," in *Engineering Management Conference*, vol. 1, pp. 362 - 367, Flint, 2002.

[3] S. Norman, "Reliability – Security Model," in *Engineering of Complex Computer Systems, 2006. ICECCS 2006. 11th IEEE International Conference on*, Monterey, 2006.

[4] P. Jayashree, K. Easwarakumar, V. Anandharaman and K. Aswin, "A Proactive Statistical Defense Solution for DDOS Attacks in Active Networks," *First International Conference on Emerging Trends in Engineering and Technology, 2008. ICETET '08.*, pp. 878 - 88, Chennai, 2008.

[5] J. Rosen, "Cybercare: Responding to a Mass Casualty Event in the 21st century," in *Information Technology Applications in Biomedicine, 2000. Proceedings. 2000 IEEE EMBS International Conference on*, pp. 189 - 189, Lebanon, 2000.

[6] J. Kumagai, "The Web As Weapon," *IEEE Spectrum*, vol. 38, no. 1, pp. 118 - 121, 2001.

[7] K. Young, "Weapons of Mass Destruction Technology Evaluation and Training Range," *IEEE Conference on* in *Technologies for Homeland Security, 2009. HST '09.*, pp. 591 - 598, Idaho, 2009.

[8] K. Melissa and Z. Angela, "Detection of Biological and Chemical Agents," *IEEE*

SENSORS, vol. 5, no. 4, pp. 696-703, 2005.

[9] A. Smithson, "International Mechanisms for Threat Reduction if Chemical and Biological Weapons," *Engineering in Medicine and Biology,* vol. 21, no. 5, pp. 116 - 120, Sept.-Oct. 2002.

[10] J. Harvey, "The secure networked truck: protecting America's transportation infrastructure," *IEEE 60th* in *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004,* vol.7, pp. 5281- 528, San Diego, 2004.

[11] P. Heegaard and K. Trivedi, "Survivability Modeling With Stochastic Reward Nets," in *Simulation Conference (WSC), Proceedings of the 2009 Winter,* pp. 807-818, Trondheim, 2009.

[12] R. Hixon and D. Gruenbacher, "Markov Chains in Network Intrusion Detection," in *Proceedings of the Fifth Annual IEEE SMC Information Assurance Workshop, 2004.,* pp. 432- 433, Wichita, 2004.

[13] H. Frank, "Survivability Analysis of Command and Control Communications Networks-Part I," *IEEE Transactions on Communications*, vol. 22, no. 5, pp. 589 - 595, 1974.

[14] T. Feyessa and M. Bikdash, "Measuring Nodal Contribution to Global Network Robustness," in *Southeastcon, 2011 Proceedings of IEEE,* pp. 131-135, Greensboro, 2011.

[15] P. Heegaard and K. Trivedi, "Survivability Quantification of Communication Services," *IEEE International Conference on Dependable Systems and Networks*

With FTCS and DCC. DSN 2008, pp. 462-471, Trondheim, 2008.

[16] R. Zurawski and M. Zhou, "Petri Nets and Industrial Applications: A Tutorial," *IEEE Transactions on Industrial Electronics,* vol. 41, no. 6, pp. 567 - 583, 1994.

[17] H.-Q. Cui, "Tree Petri Nets: Properties and Applications in Logical Problems," in *2010 Second WRI Global Congress on Intelligent Systems (GCIS),* pp.*295-298*, Qingdao, 2010.

[18] A. Tzes, "Applications of Petri networks to transportation network modeling," *IEEE Transactions on Vehicular Technology,* vol. 45, no. 2, pp. 391 - 400, 1996.

[19] S. Hairong, J. Han and H. Levendel, "Transient Analysis for Prioritized Failure Recovery in Communication networks," *21st IEEE International Communications Conference* in *Performance, and Computing, 2002,* pp. *213-219*, Elk Grove Village, 2002.

[20] H. Jueliang, D. Zuohua, L. Jing and Y. Ling, "Measuring the survivability of Object Oriented Software," *Third IEEE International Symposium on* in *Theoretical Aspects of Software Engineering, 2009. TASE 2009*, pp. *329-330*, Hangzhou, 2009.

[21] F. Pei and L. Pan, "Formal Verification of a Network Survivability Validate Model," *2010 International Conference on Biomedical Engineering and Computer Science (ICBECS),*, pp. *1-4*, Zhengzhou, 2010.

[22] W. Jiacun, "Petri Nets for Dynamic Event-Driven System Modeling," in *Handbook of Dynamic System Modeling*, Boca Raton, Chapman and hall/CRC, 2007, pp. 1-16.

[23] A.-B. Khalid, D. Alexander and K. Valentina, "Survivability of the MAP/PH/N

Queue with Propagated Failures," *2010 International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT),* pp. *518-524,* Pontypridd, 2010.

[24] M. He, H.-P. Qiu, A.-q. Hu and J.-C. Quan, "Quantification and Evaluation of Survivability on Information Systems," *International Conference on Computer Engineering and Technology, 2009. ICCET '09,* vol.2, pp. *385-389,* Nanjing, 2009.

[25] C. Changqing, W. Weimin, Z. Heng and S. Gang, "A Semi-Markov Survivability Evaluation Model for Intrusion Tolerant Real-time Database Systems," *2011 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM),* pp. *1-4*, Wuhan, 2011.

[26] W. Alex, Y. Su and L. Peng, "A Semi Markov Survivability Evaluation Model for Intrusion Tolerant Database Systems," *ARES '10 International Conference on Availability, Reliability, and Security, 2010.* pp. *104-111*, Dunmore, 2010.

[27] E. P. Kao, An Introduction to Stochastic Processes, Belmont: Duxbury Press, 1996.

[28] Stamp and Mark, "A Revealing Introduction to Hidden Markov Models," 22 February 2012. [Online]. Available: http://www.cs.sjsu.edu/~stamp/RUA/HMM.pdf. [Accessed 9 April 2012].

[29] R. Sreenivas and B. Krogh, "On Fairness and Conflicts in Petri Nets," *Proceedings of the 32nd Midwest Symposium on Circuits and Systems, 1989,* pp. *406-409* vol.1, Pittsburgh, 1989.

[30] H. Holger, K. Joost-Pieter, M.-K. Joachim and S. Markus, "A tool for model-

checking Markov chains," Springer-Verlag, Erlangen, 2002.

[31] J. Montgomery, "Communication Classes," 26 February 2009. [Online]. Available: http://www.ssc.wisc.edu/~jmontgom/commclasses.pdf. [Accessed 23 April 2012].

[32] H. Goldstrein, "Respond to Terror Like a Terrorist," *IEEE Spectrum,* p. 25, October 2001.

[33] B. Stauffer and B. Christie, "Terror: What's Next," *IEEE Spectrum,* vol. 43, no. 9, pp. 36 - 45, 2006.

[34] D. Torrieri, "Algorithms for Finding an Optimal Set of Short Disjoint Paths in a Communication Network," *Military Communications Conference, 1991. MILCOM '91, 'Military Communications in a Changing World', IEEE,* pp. 11-15, vol.1, Orlando, 1991.