0

Towards a Probabilistic Complexity-theoretic Modeling of Biological Cyanide Poisoning as Service Attack in Self-organizing Networks

Jiejun Kong, Dapeng Wu
University of Florida
USA

Xiaoyan Hong
University of Alabama
USA

Mario Gerla
University of California, Los Angeles
USA

1. Introduction

In this chapter we propose to use probabilistic complexity theory, which is also known as the foundation of modern cryptography, to study very complex systems in the form of large-scale self-organizing networks. A very complex system is typically probabilistic and stochastic in its nature. In such a system, it is impossible to describe every system behavior in a deterministic manner. We have to adopt a *probabilistic* framework. Like what is advocated by modern cryptographers, in regard to an input metric x measuring the operational algorithms being studied (Figure 1), such as security attacks and countermeasures, we speak of the "feasibility or infeasibility" of breaking the system rather than the "possibility or impossibility" of breaking the same system.

As the first step to initiate our course, the circulatory system in human body is selected to be the very complex system defying researcher's deterministic descriptions to understand its complex and probabilistic behaviors. In bioinformatics, analyzing mobility related problems is a new challenge. Typical research efforts in sequence alignment, gene finding, genome assembly, protein structure alignment, protein structure prediction, and the modeling of evolution do *not* study molecule level mobility and related security threats. For instance, many biological threats use the circulatory system as their entrance to disable the life sustaining functions. Related analysis must study a large amount of molecules moving in the circulatory system. Due to the probabilistic nature of each molecule's mobility pattern and the intractable complexity caused by the sheer amount of molecules, it is a non-trivial challenge to deliver a meaningful analysis to answer the following questions: How do we quantitatively measure the impact of a simple algorithmic attacking strategy in an environment with probabilistic mobility and huge amount of nodes? What types of metrics can we use to quantify the highly complex behavior in the system (in the example, the circulatory system)? Does the system have any stable states or equilibriums by any chance?

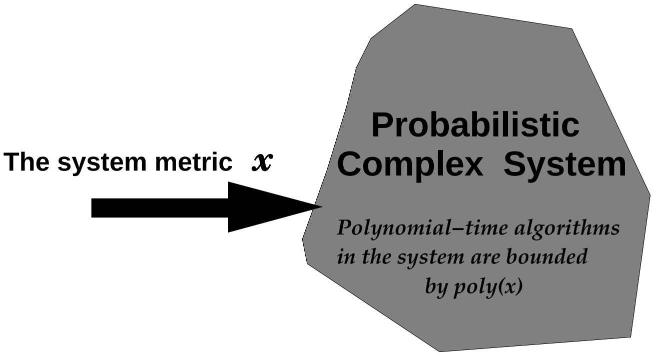

The system metric x
Probabilistic Complex System
Polynomial–time algorithms in the system are bounded by poly(x)

Figure 1. **A probabilistic complex system with the input metric** x

In this work we answer these questions by identifying the connection between the biological threats and probabilistic complexity theory. Our study shows that a generic class of self-organizing network algorithms can be modeled by a variant of computational *complexity theory* if commonly used computational metrics are replaced with network metrics. We show that biological *cyanide poisoning*(Vick & Froehlich, 1985)(Sykes, 1981) is a real-world example which belongs to the network-metric complexity class. This network-metric complexity class has well-defined properties, such as some asymptotic invariants, similar to the "amplification" property of the $\mathcal{B}PP$ class used in modern cryptography.

1.1 Notation

For the ease of formal presentation, we list the notions used in this paper below:

N	network scale (number of nodes in the network)
θ	percentage of compromised nodes, i.e., $\theta \cdot N$ nodes is compromised
$\lvert x \rvert$	the cardnality of a set x
τ	least network time granularity (e.g., 1 nano-second)
$\alpha = poly(N)$	α is a polynomial of N
$\Sigma < O(poly(N))$	Σ is asymptotically less than $poly(N)$
S	the size of the entire network space
s	the size of an average node "position"
l	the size of the largest mobile node's Turing tape storage

1.2 Complexity-theoretic Overview

First, we adopt a formal approach to characterize a general category of random algorithms, which belongs to the family of Monte Carlo algorithms with 1-sided or 2-sided errors.

We seek to prove that the success (or failure) probability of a network operation is *negligible* in regard to a *network metric* x, which in this chapter is *the network scale* N (the total number of network nodes) in the finite network space.

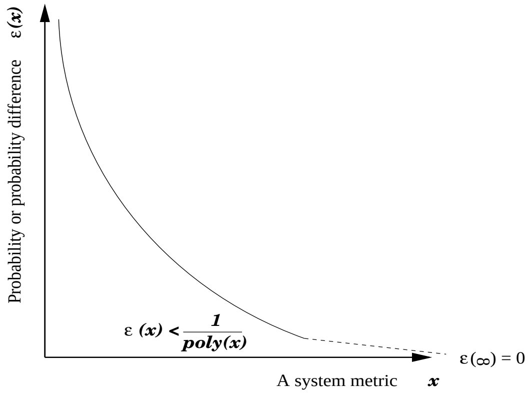
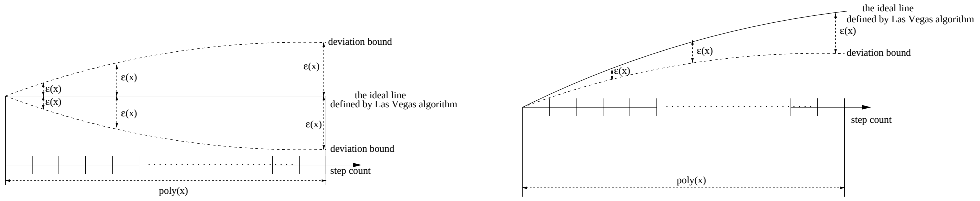


Figure 2. **The negligible function with respect to an input system metric x, which is the key size in modern cryptography or the network size in our research**



(a) 2-sided errors depicted for $\mathcal{B}PP$ class

(b) 1-sided errors depicted for $\mathcal{R}P$ class

Figure 3. Polynomial-time Monte Carlo algorithm family with *asymptotically stable* deviation $\epsilon(x)$

Second, not all Monte Carlo algorithms are qualified in the new formal model. As depicted in Figure 3(a) and 3(b), we explore a constraint similar to the polynomial-time "amplification" approach in modern cryptography, which is about the probability difference between polynomial-time Las Vegas algorithms and Monte Carlo algorithms. If an algorithm belongs to the Las Vegas random algorithm family, then the randomized algorithm always produces correct result but the algorithm execution is probabilistically efficient, sometimes inefficient or impractical. If an algorithm belongs to the Monte Carlo algorithm family, then the algorithm execution is always efficient but with probabilistic 1-sided or 2-sided errors/deviations from the ideal Las Vegas results. In particular, we use a special Monte Carlo algorithm family with clearly defined metric bounds:

1. The special class of Monte Carlo algorithm used in this work belongs to the complexity class P. It ends in *polynomial-time* (or polynomial-step) and with *negligible deviation* $\epsilon(x)$ (Figure 2) from the ideal result defined by a counterpart Las Vegas algorithm.

2. Moreover, as depicted in Figure 3, the deviation stays as negligible in polynomial steps $poly(x)$. That is, the negligible deviation $\epsilon(x)$ is an asymptotic invariant in terms of an input metric x. We show that it meets the definition of Lyapunov's *asymptotic stability* and *exponential stability*.

3. The input metric x of any polynomial discussed in this paper is defined as N, the total number of network members in the finite network space (or equivalently, *network density* if we treat the finite network space as a single unit).

Unlike formal cryptology based on the $\mathcal{B}PP$ or $\mathcal{P}/Poly$ class with 2-sided errors, in this work we use the $\mathcal{R}P$ class with 1-sided errors. We prove the depicted *polynomial-time asymptotically stable negligibility property* in the 1-sided error model to illustrate *how a node-wise simple local behavior affects global behaviors of the entire system with $N < poly(N)$ peer nodes*. Therefore, we seek to explain simpler node-wise local behaviors in a complex probabilistic system, then the complex global behaviors can be assessed safely due to the important stable property.

Third, we define a concept of "$\mathcal{GVG}$ *polynomial-time algorithm*" (or "$\mathcal{GVG}$ polynomial-step protocol" in network term) by introducing a $\mathcal{GVG}$ oracle in network complexity theory. Given a "global virtual god" ($\mathcal{GVG}$) that virtually oversees the network, we show that the number of steps in a protocol is indeed polynomially bounded in regard to the number of nodes N. This includes the following modeling aspects:

- $\mathcal{R}P$ **(n-runs) model**: Like $\mathcal{B}PP$ class used in modern cryptography, our $\mathcal{R}P$ (n-runs) class characterizes probabilistic polynomial-time algorithms. In cyanide poisoning, the circulatory oxygen-transport function is reduced into *negative* $\mathcal{GVG} - \mathcal{R}P$ class, which has negligible success probability $\epsilon(N)$ at every step and globally.

Table 1. Probabilistic behaviors of various algorithm classes

Las Vegas Answer	Monte Carlo Answer					
	$\mathcal{R}P$ (1-run) class		$\mathcal{GVG} - \mathcal{R}P$ & $\mathcal{R}P$ (n-runs) class		Negative $\mathcal{GVG} - \mathcal{R}P$ (n-runs) class	
	SUCCESS/YES	FAILURE/NO	SUCCESS/YES	FAILURE/NO	SUCCESS/YES	FAILURE/NO
SUCCESS/YES	$> \frac{1}{2} + \epsilon(x)$	$\leq \frac{1}{2} - \epsilon(x)$	$> 1 - \epsilon(x)$	$\leq \epsilon(x)$	$\leq \epsilon(x)$	$> 1 - \epsilon(x)$
FAILURE/NO	0	1	0	1	0	1

- **Polynomially-bounded adversary**: The adversary is allowed to compromise a fraction θ of N (since $\theta \cdot N$ is a polynomial of N) network members. In cyanide poisoning, a hemoglobin binding with cyanide ion but not oxygen is a compromised node, otherwise it is uncompromised. The cyanide ions do *not* (directly) kill biological cells or organs like the heart (the centralized server), but rather disable the fully distributed oxygen-transport function.

1.3 Biological Overview

Hemoglobin (Hb) is the oxygen-transport metalloprotein in the red cells of the blood in mammals and other animals. For example, hemoglobin in human's circulatory system transports oxygen from the lungs to the rest of the body, such as to the muscles, where it releases the oxygen load. If this function is blocked, the host will quickly die from hypoxia.

The hemoglobin's binding of oxygen is affected by molecules such as cyanide ion (CN-), carbon monoxide (CO), sulfur monoxide (SO), etc. For example, hemoglobin's binding affinity for CO is 200 times greater than its affinity for oxygen, and for cyanide the affinity is thousands of times greater (Kindwall, 1977)(Vick & Froehlich, 1985)(Sykes, 1981). This means that small amounts of cyanide (or CO, SO, etc.) dramatically inhibits oxygen-binding, reduces hemoglobin's ability to transport oxygen, hence causes grave toxicity and eventually death[1].

[1] Another basis for cyanide poisoning is by binding cyanide to the active site of cytochrome oxidase, there by stopping aerobic cell metabolism so that the cell can no longer aerobically produce ATP for energy. But this biological effect is beyond the scope of this paper.

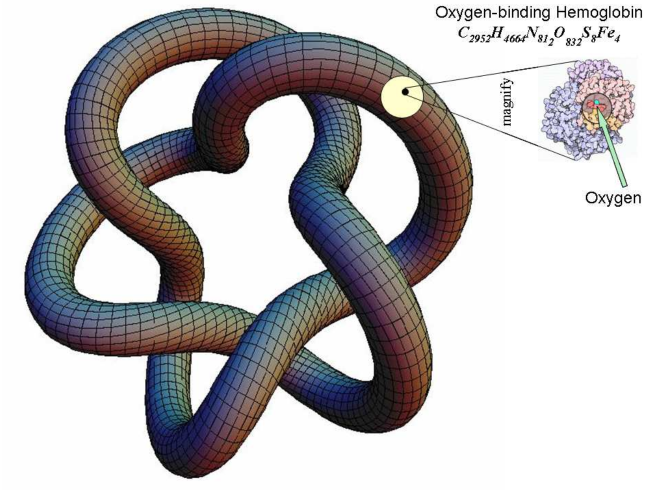

Oxygen-binding Hemoglobin
$C_{2952}H_{4664}N_{812}O_{832}S_8Fe_4$
magnify
Oxygen

Figure 4. **Abstraction of circulatory system as an enclosed finite network space S holding the mobile hemoglobin nodes**

As Figure 4 shows, at a highly abstract level we can treat the circulatory system as an enclosed (3-dimensional) space. Each hemoglobin is a molecular agent node who carries a message, which is oxygen, from one region in the space to another. Any attack that successfully blocks the message-transport function also destroys the circulatory system's function.

1.4 Researchers that inspired our approach

Since 1940s, the foundation of cryptology has seen two developing phases: (1) the information-theoretic notion developed by Shannon in (Shannon, 1949) and (2) the complexity-theoretic notion developed in 1980s by Blum-Micali (Blum & Micali, 1982)(Blum & Micali, 1984), Yao (Yao, 1982), et al.

Information-theoretic security In (Shannon, 1949), Shannon proposed the concept of *perfect secrecy*, which is a Zero-error Probabilistic Polynomial-time (ZPP, i.e., the trivial Las Vegas case of encryption) algorithm producing truly random ciphertext bits from the same length truly random key bits by applying a Latin Square cipher[2]. Shannon's algorithm ends exactly in n steps (which $< poly(n)$ steps). Unfortunately, this ZPP algorithm is impractical because it cannot process any message longer than n.

Complexity-theoretic security Formal cryptology developed since the Diffie-Hellman (Diffie & Hellman, 1976) and RSA (Rivest et al., 1978) era is founded on a different base. In early 1980s, Yao (Yao, 1982), Blum and Micali (Blum & Micali, 1982)(Blum & Micali, 1984) were the earliest complexity theorists who formally defined security using complexity classes like Bounded-error Probabilistic Polynomial-time ($\mathcal{B}PP$) class and non-uniform polynomial-size circuits ($\mathcal{P}/Poly$). The concept of being "*negligible*" is central in the definitions of these complexity classes.

Definition 1. ***(Negligible function)*** *A continuous function $\epsilon(x)$ is* negligible *if for all sufficiently large x's, for every positive polynomial $poly(x) > 0$ such that $|\epsilon(x)| < \frac{1}{poly(x)}$.* □

[2] The *exclusive-OR* operation used in Vernam cipher is an instance of Latin Square cipher.

In complexity-theoretic cryptography, a security scheme is *provably secure* if the probability of security failure (e.g., inverting a one-way function, distinguishing cryptographically strong pseudorandom bits from truly random bits) is negligible $\epsilon(x)$ in terms of $x = n$ the cryptographic key length.

In general, provably secure algorithms are Monte-Carlo algorithms belonging to certain well-defined complexity classes (e.g., BPP, P/Poly) with the input metric x as the key length n. A provably secure algorithm must ensure that an adversary cannot break its security guarantee with non-negligible probability. For example, a sequence of provably secure pseudo-random ciphertext bits is polynomially *indistinguishable* (IND) from a sequence of truly random bits (coin-flips, coin-tosses), where the term "indistinguishability" is defined as the polynomial adversary's incapability of distinguishing pseudorandom ciphertext bits (of $poly(x)$ length) from truly random bits (of the same length) with non-negligible probability. Given adversaries with various capabilities, a provably secure crypto-scheme features various levels of ciphertext indistinguishability shown below (IND-CPA, IND-CCA1 and IND-CCA2).

Polynomial-time adversary	Security guarantee
Chosen plaintext attacker (CPA)	IND-CPA (aka. semantic security (Goldwasser & Micali, 1984))
Chosen ciphertext attacker (CCA1)	IND-CCA1 (aka. non-malleable security (Naor & Yung, 1990))
Adaptive chosen ciphertext attacker (CCA2)	IND-CCA2 (aka. non-malleable security (Dolev et al., 1991))

Similar to complexity-theoretic cryptographers, we pursue the same direction when studying network security problems. In many complexity-theoretic problems associated with networks and graphs, e.g., the Hamilton Circuit problem, the problem's complexity is measured by the number of the nodes N in a network/graph. Here the term "*polynomial-time algorithm/protocol*" refers to an algorithm/protocol which must end in polynomially bounded stops/steps at those nodes.

In complexity-based modern cryptography, a security scheme is *provably secure* if the probability of security failure (e.g., inverting a one-way function, distinguishing cryptographically strong pseudorandom bits from truly random bits) is negligible in terms of the cryptographic key length $x = n$. Nevertheless, *the general notion of negligibility has never said that the system input parameter x must be the key length n*. Indeed, x can be any predetermined system metric and corresponding mathematic analysis would illustrate some hidden analytical behaviors of the system. Hence in our analysis, the input parameter x for any negligibility notion is changed from the cryptographic key length n to certain network metric, in particular, network density ρ (the number of network nodes per unit area) or network scale N (the number of network nodes if we treat the finite network area as the unit).

2. Complexity-theoretic model of cyanide poisoning

In below we propose a concept of "*$\mathcal{GVG}$-polynomial time*" protocol/algorithm as the formal model of network security and the explanation of the probabilistic nature of cyanide poisoning. As in typical randomized complexity-theoretic analysis, our concepts are defined on top of probabilistic Turing Machines with polynomially-bounded tapes.

Given a probabilistic Turing Machine controlled by a virtual oracle "global virtual god" ($\mathcal{GVG}$) who oversees the entire network, the number of protocol steps is polynomially bounded by $x = N$, the number of network nodes.

2.1 Required Turing Machines

At first, we use the same probabilistic Turing Machines used in formal cryptology. A Turing machine consists of a tape, a head, a state register, and an action table. According to the number of used tapes

Turing machine is classified into two classes, namely 1-tape and k-tape Turing machine. We define now formally Turing machine.

Definition 2. *A Turing machine is a septuple $M = (Q, \Gamma, \Sigma, q_I, \#, F, \delta)$, where*

- *Q is a finite set of states.*
- *Γ is a finite set of the tape alphabet.*
- *$\Sigma \subseteq \Gamma$ is a finite set of the input alphabet.*
- *$q_I \in Q$ is the initial state.*
- *$\# \in (\Gamma - \Sigma)$ is the blank symbol.*
- *$F \subseteq Q$ is the set of final or accepting states.*
- *δ is the transition set. For 1-tape Turing Machine, δ is*

$$\delta : Q \times \Gamma \leftarrow Q \times \Gamma \times \{Le, Ri\},$$

while for k-tape Turing Machine, δ is

$$\delta : Q \times \Gamma^k \leftarrow Q \times (\Gamma \times \{Le, Ri, St\})^k$$

Here Le is left shift, Ri is right shift, and St is stationary without shift. □

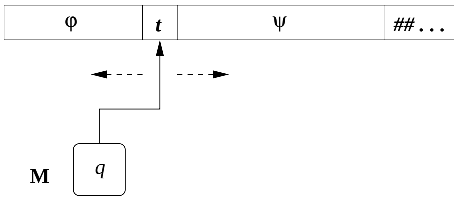


Figure 5. **1-tape Turing Machine M in configuration** (q, φ, t, ψ)

Using 1-tape Turing Machine as an example, as depicted in Figure 5, a *configuration*, or *instantaneous description*, of M is a quadruple

$$(q, \varphi, t, \psi), \quad \varphi\psi \in \Gamma*, \ t \in \Gamma, \ q \in Q$$

in which the rightmost symbol of ψ is not #. The string of symbols $\varphi t \psi$ is called the *tape* of the configuration. If $\varphi = \lambda$ and $q = q_I$, the configuration is an *initial configuration* of M.

Upon each left (or right) *move*, the current symbol t under the tape head is replaced by t', and the tape head is moved to the immediate left (or right) of the replaced symbol. Then M's current state q is replaced by q'. If a machine enters a state $q' \in F$ or has no moves from a given configuration, the configuration is *dead*. Otherwise, we say that

$$(\lambda, q_I, t, \psi) \Longrightarrow (\varphi', q', t', \psi')$$

is a *computation* of M, if M has a sequence of moves leading from the initial configuration (λ, q_I, t, ψ) to the final configuration $(\varphi', q', t', \psi')$, and call the computation *halted* if the final configuration is dead.

Definition 3. *A Turing Machine is* deterministic Turing Machine (DTM) *if at most one move is possible from each configuration in the machine's transition set δ.*
A Turing Machine is non-deterministic Turing Machine (NDTM) *if more than one move is possible from each configuration in the machine's transition set δ.*
A Turing Machine is probabilistic Turing Machine (PTM) *if it is NDTM and the different moves are taken with certain probabilistic distributions.* □

A probabilistic Turing machine is a non-deterministic Turing machine which randomly chooses between the available transitions at each point with certain probability. As a consequence, a probabilistic Turing machine can (unlike a deterministic Turing Machine) have stochastic results; on a given input and instruction state machine, it may have different run times, or it may not halt at all; further, it may accept an input in one execution and reject the same input in another execution.
A common reformulation of PTM is a DTM with an added *random tape* full of random bits, which are pre-determined by an oracle's coin-flips and placed on the tape to replace the DTM's own coin-flips in decision. The DTM with added random tape is equivalent to the PTM if the oracle's coin-flips and the DTM's (assumed-to-be) coin-flips follow the same probabilistic distribution.

2.2 Required complexity classes

Then we define the ideal Las Vegas protocol for mobile ad hoc message/oxygen transportation:

Definition 4. ***(The ideal Las Vegas message/oxygen transporting)*** *The ideal Las Vegas case of message/oxygen transporting is characterized by a pair of probability quantities P_{lv} and P'_{lv}, when there are no other molecules blocking an oxygen molecule combining with the nearest hemoglobin. The probability P_{lv} is the success probability an oxygen molecule can combine with the nearest hemoglobin, while $P'_{lv} = 1 - P_{lv}$ is the failure probability the same oxygen molecule can combine with the nearest hemoglobin.* □

If such Las Vegas algorithm/protocol returns FAILURE/NO, then any Monte Carlo algorithm/protocol also returns FAILURE/NO (since there are some other things, such as cyanide ions or carbon monoxide molecules, interfering the combination of oxygen-hemoglobin combination). There is no error/deviation when the protocols return NO. Thus only 1-sided Monte-Carlo error/deviation is possible when the Las Vegas protocol returns YES. This is the reason why we use 1-sided error $\mathcal{R}P$ algorithm class in our modeling.
We then define the $\mathcal{R}P$ protocol/algorithm class with 1-sided errors.
Let x be the input in the polynomial size of a system parameter N, let $M(x)$ be the random variable denoting the output of a PTM M. Let

$$\mathsf{Pr}\left[M(x) = y\right] = \frac{|\{d \in \{0,1\}^{t_M(x)} : \; M_d(x) = y\}|}{r^{t_M(x)}}$$

where d is a truly random coin-flip, $t_M(x)$ is the polynomial number of coin-flips made by M on input x, and $M_d(x)$ denotes the output of M on input x, when d is the outcome of its coin-flips (i.e., the random tape of an equivalent DTM).

Definition 5. ***(Randomized Polynomial-time, $\mathcal{R}P$ class):*** *We say that L is recognized by the probabilistic polynomial-time Turing Machine M with biased single-sided errors if*

- *for every $x \in L$ it holds that*[3] $\textsf{Pr}[M\ accepts\ x] \geq \frac{1}{2} + \frac{1}{poly(n)}$ *for every polynomial $poly(n)$.*
- *for every $x \notin L$ it holds that $\textsf{Pr}[M\ accepts\ x] = 0$.*

$\mathcal{R}P$ is the class of languages that can be recognized by such a probabilistic polynomial time Turing Machine. □

Definition 6. ***($\mathcal{R}P$ n-runs class):*** *We say that L is recognized by the probabilistic polynomial-time Turing Machine M with negligible single-sided errors if*

- *for every $x \in L$ it holds that $\textsf{Pr}[M\ accepts\ x] \geq 1 - \frac{1}{poly(n)}$ for every polynomial $poly(n)$.*
- *for every $x \notin L$ it holds that $\textsf{Pr}[M\ accepts\ x] = 0$.*

$\mathcal{R}P$ n-runs class is the class of languages that can be recognized by such a probabilistic polynomial time Turing Machine. □

Definition 7. ***(Negative $\mathcal{R}P$ n-runs class):*** *We say that L is recognized by the probabilistic polynomial-time Turing Machine M with negligible single-sided success if*

- *for every $x \in L$ it holds that $\textsf{Pr}[M\ accepts\ x] \leq \frac{1}{poly(n)}$ for every polynomial $poly(n)$.*
- *for every $x \notin L$ it holds that $\textsf{Pr}[M\ accepts\ x] = 0$.*

Negative $\mathcal{R}P$ n-runs class is the class of languages that can be recognized by such a probabilistic polynomial time Turing Machine. □

The procedure to obtain $\mathcal{R}P$ n-runs class from $\mathcal{R}P$ 1-run class is called $\mathcal{R}P$ *amplification*, which trivially runs an $\mathcal{R}P$ 1-run class algorithm n times, then the failure probability of an $\mathcal{R}P$ n-runs algorithm (i.e., not returning YES after running the $\mathcal{R}P$ 1-run algorithm n times) exponentially decreases, and becomes negligible.

2.3 $\epsilon(x)$ as the polynomial-time asymptotically stable equilibrium

Now we prove the property of $\mathcal{R}P$ (n-runs) depicted in Figure 3(b)[4]. This property illustrates that how a node-wise local behavior at each individual peer node affects the network-wise global behavior of the entire probabilistic system, which is comprised of $N < poly(N)$ independent peer nodes.

Theorem 1. *If an $\mathcal{R}P$ (n-runs) protocol X's failure probability is negligible, then the failure probability stays as negligible when the same protocol X is independently executed polynomial times.*

Proof: *By assumption, X will be repeated $p(N)$ steps, where $p(N)$ is a positive polynomial. Given that per-step security success probability is $P_{onetime}$, the probability of success of the entire execution $P_{polytime}$ is*

$$P_{polytime} \ = \ 1 - (1 - P_{onetime})^{p(N)}.$$

[3] In the definition $\frac{1}{2}$ can be replaced by any constant fraction number in the open range (0..1), not necessarily the value $\frac{1}{2}$.

[4] The same property of $\mathcal{B}PP$ in Figure 3(a) is known to be true by applying Chernoff's bound in its proof.

By assumption, $P_{onetime}$ is negligible, thus is asymptotically less than any given $\frac{1}{p(N)\cdot q(N)}$, where $q(N)$ is a positive polynomial and so $p(N)\cdot q(N)$ is also a positive polynomial. In other words, there exists a positive integer $N_c > 0$, such that $P_{onetime} < \frac{1}{p(N)\cdot q(N)}$ for all $x > N_c$. Then we have

$$(1 - P_{onetime})^{p(N)} > \left(1 - \frac{1}{p(N)\cdot q(N)}\right)^{p(N)} > e^{-\frac{1}{q(N)}}$$

since $(1 - \frac{1}{x})^x > e^{-1}$ for all $x > 1$.
According to Lagrange mean value theorem, for a function $f(x)$ continuous on $[a, b]$, there exists a $c \in (a, b)$ such that $f(b) = f(a) + f'(c) \cdot (b - a)$ for $0 < a < b$. Then let $f(x) = e^{-x}$, there exists a $\xi \in (0, z)$, such that $e^{-z} = 1 + (-e^{-\xi})\cdot z > 1 - z$. Thus we have

$$(1 - P_{onetime})^{p(N)} > e^{-\frac{1}{q(N)}} > 1 - \frac{1}{q(N)}.$$

Therefore, for any polynomial $q(N)$ and sufficiently large N,

$$P_{polytime} \ = \ 1 - (1 - P_{onetime})^{p(N)} < \frac{1}{q(N)}. \ \square$$

According to our model using the input metric $x = N$, any Monte Carlo network protocol belongs to one of the four categories (Figure 6) in regards to the stability property:

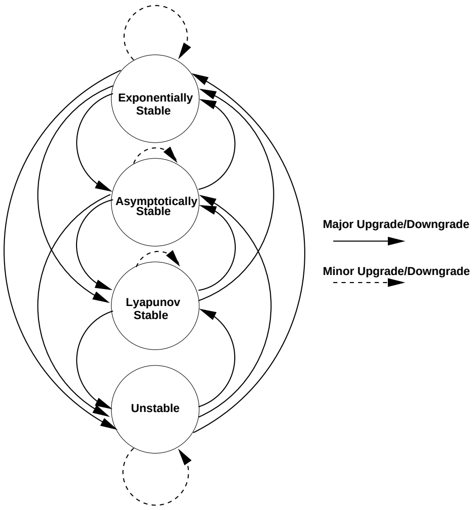


Figure 6. **The classification of network protocols with respect to the input metric $x = N$. Inside each category it is only possible to do minor upgrade or downgrade. Major upgrade or downgrade happens when the network protocol is optimized or damaged from one category to another.**

1. The Monte Carlo protocol is *unstable*: The probability difference between the Monte Carlo protocol and the ideal Las Vegas baseline is *not* bounded in the open interval (0,1). For example, a Monte Carlo protocol in negative $\mathcal{R}P$ class has a FAILURE/NO probility $1 - \epsilon(x)$, which is bounded in (0,1], but not in (0,1).
2. The Monte Carlo protocol is *Lyapunov stable*: The probability difference between the Monte Carlo protocol and the ideal Las Vegas case is bounded in (0,1), but the bound is *not* an invariant if the protocol is repeated independently polynomial times.
3. The Monte Carlo protocol is *asymptotically stable*: The probability difference between the Monte Carlo protocol and the ideal Las Vegas case is bounded in (0,1), and the bound is an invariant if the protocol is repeated independently polynomial times. In our study, this invariant is the negligible quantity $\epsilon(N)$.
4. The Monte Carlo protocol is *exponentially stable*: The probability difference between the Monte Carlo protocol and the ideal Las Vegas case is bounded in (0,1), but the bound is an exponential invariant if the protocol is repeated independently polynomial times.

It is possible to upgrade an unstable protocol to be Lyapunov stable, then asymptotically stable, finally exponentially stable. This is called *major upgrade* or *major optimization*. Afterwards, it is only possible to do *minor upgrade/optimization* inside the exponentially stable category. That is, it is only possible to adjust the exponent of the exponentially negligible quantity inside this category.

Likewise, it is possible to downgrade an exponentially stable protocol on the reverse direction to be unstable eventually. Afterwards, it is only possible to do *minor degrade/damage* inside the unstable category.

2.4 Modeling mobile networks: a PTM approach with a $\mathcal{GVG}$ oracle

We propose to use a special form of PTM to model the probabilistic stochastic behaviors of a mobile network. The fundamental idea is to use a *global virtual god* ($\mathcal{GVG}$) oracle to handle the PTM's control states, while each mobile node is only treated as a tape carrier.

As depicted in Figure 7, the entire network space is of finite size S. The finite network space S is divided into large number of tiles (or cubes for 3-D space) of tiny size s, and each tile/cube is smaller than the physical size of any single mobile node. In other words, each tile/cube is virtually a node "position" to place on. The number of node "positions" $\eta = \frac{S}{s}$ is quite large. It is nevertheless a finite number. In a nutshell, $\eta = \frac{S}{s}$ is a large constant, but is always asymptotically less than $poly(N)$, that is, $\eta < O(poly(N))$.

Tape Each mobile node functions as a carrier of a *moving tape* of polynomial size of the network scale N. That is, each mobile node carries a tape of $O(poly(N))$ bits. A moving tape is intuitively the computer memory snapshot of the corresponding mobile node. Let $l < O(poly(N))$ be the size of the largest moving tape. An empty node "position" is occupied by a blank tape of l blank symbols. This blank tape is replaced with a node's moving tape once the corresponding position is taken by the node, or the tape goes back to the blank tape upon the node's leaving of the position. If the largest tape length of each mobile node can carry is $l < O(poly(N))$, then the $\mathcal{GVG}$ PTM's *consummate tape* length is $\eta{\cdot}L$, which is $< O(poly(\eta)){\cdot}O(poly(N))$, thus $< O(poly(N))$.

Control state operations Each mobile node's decision of network operation (e.g., packet transmission), though autonomous in nature, can be translated into an equivalent form *as if all the decisions are made by the $\mathcal{GVG}$ using coin-flips*. Along the timeline, there exists a minimal time granularity τ such

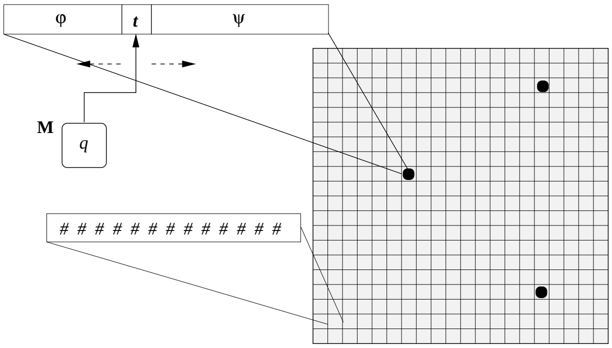


Figure 7. **A $\mathcal{GVG}$ Probabilistic Turing Machine ($\mathcal{GVG}$ PTM) to model mobile nodes in a finite cubic space (only 2-D depicted) with a large number of node "positions". The figure shows that $N = 3$ of η $(N \ll \eta < O(poly(N)))$ such "positions" have been taken by $N = 3$ mobile nodes. Each empty "position" is filled with a tape of $poly(N)$ blank symbols, and the blank tape is replaced with a mobile node's tape once the corresponding position is taken, or the tape goes back to the blank tape upon the node's leaving of the position. If the largest tape length of each mobile node can carry is $l < O(poly(N))$, then the $\mathcal{GVG}$ PTM's consummate tape length is $\eta{\cdot}l$. The $\mathcal{GVG}$ PTM's tape head always parks at the place corresponding to the current symbol of the first mobile node (i.e., the node with least node index). The mobile node's mobility patterns are "as if" decided by the $\mathcal{GVG}$ using coin-flips. In theory, the $\mathcal{GVG}$ does all symbol processing and coin-flipping operations and its operation speed is fast enough to process all symbols on its tape within the least network time granularity τ**

that any Turing Machine operation latency less than τ will make *no* difference in network protocol execution. We model that the $\mathcal{GVG}$ can make decisions for all mobile nodes and emulate all the decisions globally within the granularity τ (e.g., 1 nano-second).

The mobile nodes are indexed from 1 to N. At the beginning/end of each τ time granularity, the PTM's tape head always parks at the place corresponding to the current symbol of the first mobile node (with node index 1). During a τ interval, the PTM processes every mobile node's tape one by one (treating the corresponding node as a puppet Turing Machine of the $\mathcal{GVG}$).

Environmental randomness As to environmental conditions, for each network operation (e.g., message delivery), the $\mathcal{GVG}$ emulates the physical condition (e.g., blood condition and obstacles that affect blood flowing) in a perfect manner, and precisely moves each message from one node to another. That is, the message content is deleted from the sending node's moving tape, and the received message content is added to the proper place of the receiving node's moving tape. In the eyes of the $\mathcal{GVG}$, any message transportation is simply a movement of a set of tape symbol from one place of its consummate tape to another place.

PTM as DTM with random tape If we use DTM rather than PTM to model the network protocol execution, the $\mathcal{GVG}$ can pre-cast many coin-flips to emulate the probabilistic events in the network, and place the result of the coin-flips to an added consummate random tape. These probabilistic events include mobile node's probabilistic moving pattern, probabilistic message delivery requests at the message sources and destinations, and so on. The total number of coin-flips (or the length of the consummate random tape) is bounded by network space and network scale $< O(poly(\eta))\cdot O(poly(N))$, thus $< O(poly(N))$.

Definition of $\mathcal{GVG}$ PTM We formally define $\mathcal{GVG}$ Probabilistic Turing Machine and $\mathcal{GVG}$ polynomial-time protocols in below.

Definition 8. *A $\mathcal{GVG}$* Polynomial-time Probabilistic Turing Machine ($\mathcal{GVG}$-PPTM) *is an octuple*

$$M = (N, \mathcal{GVG}(Q,r), \Gamma, \Sigma, q_I, \#, F, \delta),$$

- *N is a pre-defined system parameter. N quantifies the size of the $\mathcal{GVG}$-PPTM's input and output. For any configuration (q, φ, t, ψ), $\quad \varphi\psi \in \Gamma*,\ t{\in}\Gamma,\ q{\in}Q$ on any single tape of the machine, $|\varphi|, |\psi| < O(poly(N))$.*
- *$\mathcal{GVG}(Q,r)$ is a global virtual god oracle with finite set of states Q and a probabilistic coin-flip sequence r (i.e., the random tape input of an equivalent DTM). $|Q|$ and $|r|$ are $< O(poly(N))$.*
- *Γ is a finite set of the tape alphabet.*
- *$\Sigma \subseteq \Gamma$ is a finite set of the input alphabet.*
- *$q_I{\in}Q$ is the initial state.*
- *$\#{\in}(\Gamma - \Sigma)$ is the blank symbol.*
- *$F{\subseteq}Q$ is the set of final or accepting states.*
- *δ is the transition set. For 1-tape $\mathcal{GVG}$-PPTM, δ is*

  $$\delta : Q \times \Gamma {\leftarrow} Q \times \Gamma \times \{Le, Ri\},$$

  while for k-tape $\mathcal{GVG}$-PPTM, δ is

  $$\delta : Q \times \Gamma^k {\leftarrow} Q \times (\Gamma \times \{Le, Ri, St\})^k$$

  Here Le is left shift, Ri is right shift, and St is stationary without shift.

We say that L is recognized by the $\mathcal{GVG}$-PPTM M with negligible errors if

- *for every $x{\in}L$ it holds that $\mathsf{Pr}[M\ accepts\ x] \geq 1 - \frac{1}{poly(N)}$ for every polynomial $poly(N)$;*
- *for every $x{\notin}L$ it holds that $\mathsf{Pr}[M\ accepts\ x] = 0$.*

$\mathcal{GVG} - \mathcal{R}P$ (n-runs) is the class of languages that can be recognized by such a $\mathcal{GVG}$-PPTM.
We say that L is recognized by the $\mathcal{GVG}$-PPTM M with negligible success if

- *for every $x{\in}L$ it holds that $\mathsf{Pr}[M\ accepts\ x] \leq \frac{1}{poly(N)}$ for every polynomial $poly(N)$;*
- *for every $x{\notin}L$ it holds that $\mathsf{Pr}[M\ accepts\ x] = 0$.*

Negative $\mathcal{GVG} - \mathcal{R}P$ (n-runs) is the class of languages that can be recognized by such a $\mathcal{GVG}$-PPTM.
□

For every $x{\in}L$, $\mathsf{Pr}[M\ accepts\ x]$ means "probability of protocol success", while its complement $\mathsf{Pr}[M\ rejects\ x]$ means "probability of protocol failure". In $\mathcal{GVG} - \mathcal{R}P$[5], the former one must be $1 - \epsilon(N)$ and the latter one must be $\epsilon(N)$ in terms of network scale N.

[5] In this paper, the notion "$\mathcal{GVG}$-$\mathcal{R}P$" denotes "$\mathcal{GVG}$-$\mathcal{R}P$ (n-runs)" class for the ease of presentation.

Example 1. ***(Modeling mobile message transportation using Figure 7)*** *In a mobile ad hoc network, peer nodes can be viewed as controlled proxy agents of the $\mathcal{GVG}$. Based on the random coin-flips (or the random tape of an equivalent DTM) that simulate the probabilistic environment, $\mathcal{GVG}$ initiates a message/oxygen on a source node. When intended destination node successfully accepts the message/oxygen, $\mathcal{GVG}$ enters a final acceptance state to finish the mobile message delivery. For a poisoned oxygen transportation process which is in the negative $\mathcal{GVG} - \mathcal{RP}$ class, the probability of hemoglobin's oxygen-transportation being FAILURE/NO* $\mathsf{Pr}[destination\ cannot\ receive\ oxygen]$ *must be* $1 - \epsilon(N)$*, while the probability of transportation SUCCESS/YES* $\mathsf{Pr}[destination\ receives\ oxygen]$ *must be* $\epsilon(N)$. □

2.5 Mobility model

In an enclosed network space S, we divide the network space S into a large amount of small virtual tiles (cubes) of area (volume) s, so that the tile area (cube volume) is even smaller than the physical size of the smallest network node. This way, each tile (cube) is either empty, or is occupied by a single node. Also because the network space is much larger than the sum of all mobile nodes' physical size, the probability that a tile (cube) is occupied by a mobile node is very small.

Now a binomial distribution $B(\eta, p)$ defines the probabilistic distribution of how these tiles (cubes) are occupied by each mobile ad hoc node. Here $\eta = \frac{S}{s}$, the total number of "positions", is very large but $< O(poly(N))$; and p, the probability that a cube is occupied by the single node, is very small. When η is large and p is small, it is well-known that a binomial distribution $B(\eta, p)$ approaches Poisson distribution with parameter $\rho_1 = \eta \cdot p$. Hence this binomial spatial distribution is translated into a *spatial Poisson point process* (Cressie, 1993) to model the random presence of the network nodes. In other words, ρ_1 can be treated as a mobile node's arrival rate of each presence "position". Moreover, suppose that N events occur in space $\mathcal{S}$ (here an event is a mobile node's physical presence), $\rho_{_N} = \frac{N}{\mathcal{S}}$ (where $\rho_{_N} = N \cdot \rho_1$ by treating ρ_1 as the average node PDF amongst the N nodes) is equivalent to a random sampling of $\mathcal{S}$ with rate $\rho_{_N}$.

Eulerian and Lagrangian motion models In kinematics, a given flow's motion depends not only upon position but upon time as well. Consider any scalar quantity σ which is a continuous function of the four independent variables representing position and time (x, y, z) and t, with t being time, for which the space and time derivatives exists. The total rate of change of σ with time is in general defined by an operator $\frac{D}{Dt}$:

$$\frac{D}{Dt}\sigma \;=\; \frac{\partial\sigma}{\partial t} + \frac{\partial\sigma}{\partial x} \cdot \frac{dx}{dt} + \frac{\partial\sigma}{\partial y} \cdot \frac{dy}{dt} + \frac{\partial\sigma}{\partial z} \cdot \frac{dz}{dt} \;=\; \frac{\partial\sigma}{\partial t} + \vec{V} \cdot \bigtriangledown\sigma,$$

where the differential displacements dx, dy, dz are specified for the elapsed time dt. Here $-\bigtriangledown$ is the *gradient* of a scalar:

$$\bigtriangledown = \vec{x} \cdot \frac{\partial}{\partial x} + \vec{y} \cdot \frac{\partial}{\partial y} + \vec{z} \cdot \frac{\partial}{\partial z}$$

and $\vec{V}$ is the *flow vector*:

$$\vec{V} \;=\; \vec{x} \cdot \frac{dx}{dt} + \vec{y} \cdot \frac{dy}{dt} + \vec{z} \cdot \frac{dz}{dt},$$

where $\vec{x}$,$\vec{y}$,$\vec{z}$ are unit vectors in the x, y, z directions, respectively. Clearly, the term $\frac{\partial\sigma}{\partial t}$ represents the local time rate of change of the quantity σ at a fixed position point. The term $\vec{V} \cdot \bigtriangledown\sigma$ is a scalar representing the advectional or field changes in the flow associated with the motion of the flow.

For a network of many mobile nodes flowing through a finite area, we can specify either the field of $\vec{V}$ or the paths (trajectories) of the mobile nodes. The former is normally referred to as the *Eulerian description of motion* while the latter is endowed with the title of *Lagrangian description of motion*.
In below, we will adopt an Eulerian description in our stochastic mobility analysis. The scalar quantity σ is the *arrival rate* of an average node on a position, that is, the probability of an average node's presence at a position.

The stochastic mobility PDF Let ρ_1 denote the mobility probability distribution function of a single node in the bounded network space S. For a network deployed in a bounded system space, let the random variable $\Omega = (X, Y, Z)$ denote the Cartesian location of a mobile node in the 3-dimensional network space at an arbitrary time instant t.
The spatial distribution of a node is expressed in terms of the probability density function

$$\rho_1 \ = \ f_{XYZ}(x, y, z) \ = \\ \lim_{\delta \to 0} \frac{\mathsf{Pr}\left[(x - \frac{\delta}{2} < X \leq x + \frac{\delta}{2}) \wedge (y - \frac{\delta}{2} < Y \leq y + \frac{\delta}{2}) \wedge (z - \frac{\delta}{2} < Z \leq z + \frac{\delta}{2})\right]}{\delta^3}$$

The probability that a given node is located in a subspace $\mathcal{S}'$ of the system space $\mathcal{S}$ can be computed by integrating ρ_1 over this subspace

$$\mathsf{Pr}\left[\text{node in } \mathcal{S}'\right] \ = \ \mathsf{Pr}\left[(X, Y, Z) \in \mathcal{S}'\right] \ = \ \iiint_{\mathcal{S}'} f_{XYZ}(x, y, z) \ \mathrm{d}\mathcal{S}$$

where $f_{XYZ}(x, y, z)$ can be computed by a stochastic analysis of an arbitrary mobility model.
Let x denote the random variable of number of independent mobile nodes in any network space concerned.

- (*Uniform* ρ_1) the probability that there are exactly k nodes in a specific space $\mathcal{S}'$ following a uniform distribution model is

$$\mathsf{Pr}\left[x = k\right] \ = \ \frac{(N \cdot \rho_1 \cdot \mathcal{S}')^k}{k!} \cdot e^{-N \cdot \rho_1 \cdot \mathcal{S}'}. \tag{1}$$

- (*Non-uniform* ρ_1) More generally, in arbitrary distribution models including non-uniform models, the arrival rate is *location dependent*. The probability that there are exactly k nodes in a specific space $\mathcal{S}'$ is

$$\mathsf{Pr}\left[x = k\right] \ = \ \iiint_{\mathcal{S}'} \left(\frac{(N \cdot \rho_1)^k}{k!} \cdot e^{-N \cdot \rho_1} \right) \ \mathrm{d}\mathcal{S}. \tag{2}$$

The choice of ρ_1 depends on the underlying mobility model. Some stochastic mobility models which directly choose a destination direction rather than a destination point and allow a bound back or wrap-around behavior at the border of the system area, including the random walk model on a 2-D torus surface, are able to achieve a uniform spatial distribution (Bettstetter et al., 2004)(Bettstetter & Wagner, 2002)(Bettstetter, 2001). However, ρ_1 is typically non-uniform. Fortunately, our $\mathcal{GVG} - \mathcal{RP}$ and negative $\mathcal{GVG} - \mathcal{RP}$ models do *not* assume any specific mobility model and mobile node presence PDF. As depicted in Figure 8, the stochastic PDF can be an arbitrary but continuous function over the network area/space.

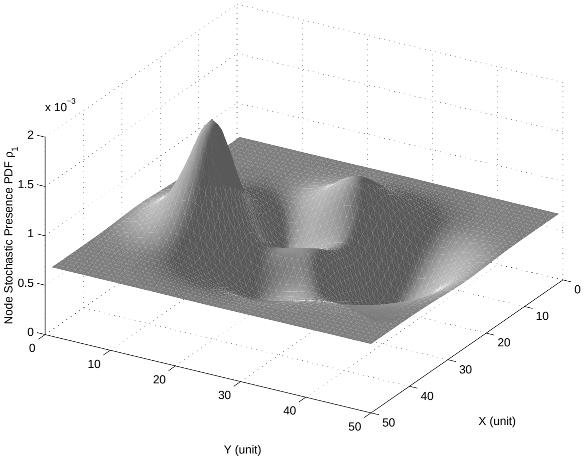


Figure 8. **Stochastic node presence PDF in an arbitrary mobility model by Eulerian description (in a 50 unit x 50 unit network space)**

2.6 Cyanide poisoning: formal specification

In this section, we use the negligibility-based model to explain a theoretic reason why cyanide poisoning is fatal. We show that the *success probability of carrying message at a single node* is negligible under cyanide poisoning.

As specified previously, there are N nodes in the bounded network area, amongst them there are $\theta{\cdot}N$ compromised (i.e.,binding with $\theta{\cdot}N$ cyanide ions) and $(1-\theta){\cdot}N$ uncompromised nodes. The random variable y denotes the number of uncompromised nodes in an arbitrary space $\mathcal{S}'$. The probability that there are k uncompromised nodes in the space $\mathcal{S}'$ is

$$\mathsf{Pr}[y=k] \;=\; \iiint_{\mathcal{S}'} \frac{((1-\theta)\cdot N\cdot\rho_1)^k}{k!}\cdot e^{-(1-\theta)\cdot N\cdot\rho_1}\,\mathrm{d}\mathcal{S}$$

Let z denote the random variable of number of compromised nodes in the same space $\mathcal{S}'$. The probability that there are k compromised nodes in the space $\mathcal{S}'$ is

$$\mathsf{Pr}[z=k] \;=\; \iiint_{\mathcal{S}'} \frac{(\theta{\cdot}N\cdot\rho_1)^k}{k!}\cdot e^{-\theta\cdot N\cdot\rho_1}\,\mathrm{d}\mathcal{S}$$

We assume that oxygen molecules are always available in the circulatory system. When there is no poisoning agents like cyanide ion or carbon monoxide in the enclosed finite space, given a hemoglobin in binding mode, its oxygen binding success ratio is

$$\begin{aligned} P_{regular} &= \mathsf{Pr}[y{\geq}1] \;=\; 1-Pr[y=0] \\ &= 1-\iiint_{\mathcal{S}'} e^{-(1-\theta)\cdot N\cdot\rho_1}\,\mathrm{d}\mathcal{S} \\ &= 1-\iiint_{\mathcal{S}'} \epsilon(N)\,\mathrm{d}\mathcal{S} \\ &= 1-\epsilon(N) \end{aligned}$$

This $P_{regular}$ is the characteristic probability P_{lv} defined in Definition 4.

Given a hemoglobin in binding mode, one cyanide ion within the binding area S' will deprive the chance of nearby oxygen molecules' chance to bind with the hemoglobin. The hemoglobin node's oxygen binding success ratio *at the presence of cyanide ions* is reduced to be

$$\begin{aligned}
P_{success} &= \mathsf{Pr}[y \geq 1] \cdot \mathsf{Pr}[z = 0] \\
&= \iiint_{\mathcal{S}'} \left((1 - e^{-(1-\theta)\cdot N\cdot\rho_1})\cdot e^{-\theta\cdot N\cdot\rho_1} \right) \, \mathrm{d}\mathcal{S} \\
&= \iiint_{\mathcal{S}'} ((1 - \epsilon(N))\cdot\epsilon(N)) \, \mathrm{d}\mathcal{S} \\
&< \iiint_{\mathcal{S}'} \epsilon(N) \, \mathrm{d}\mathcal{S} \\
&= \epsilon(N).
\end{aligned}$$

where $\mathcal{S}'$ denotes the nominal size of the biochemical binding range and $\epsilon(N)$ denotes a negligible quantity with respect to N. This $P_{success}$ is the hemoglobin node's oxygen binding success probability as in the Monte Carlo case with 1-sided error. The probability difference $P_{regular} - P_{success}$ is of our concern. In $\mathcal{R}P$ (n-runs) case, this difference should be negligible $\epsilon(N)$. In negative $\mathcal{R}P$ (n-runs) case, this difference is however $1 - \epsilon(N)$.

The mobility PDF ρ is arbitrary in our study as long as it is continuous in the space S, thus could be location dependent and becomes a function of the location space $\mathcal{S}$. Therefore, triple integrals must be used here. Fortunately, because e^x is a fixed point in differential and integral calculus, that is, $\frac{\mathrm{d}e^x}{\mathrm{d}x} = e^x$ and $\int e^x \, \mathrm{d}x = e^x + C = O(e^x)$, *such integrals or differentials do not change the magnitude of order*. In a nutshell, exponential orders $O(e^N)$ and polynomial orders $O(poly(N))$ are unchanged in magnitude through these integrals or differentials. And this concludes that the last step $= \epsilon(N)$ holds.

Hence we have proved that cyanide poisoning reduces step-wise success probability of oxygen-transport from $1 - \epsilon(N)$ to $\epsilon(N)$ for every single hemoglobin node. This single binding step is unchanged (identically distributed) and repeated independently $poly(N)$ times. Then by the asymptotic stability proved in Theorem 1, the network-wise global success probability of oxygen-transport counting all $N < poly(N)$ nodes stays as $\epsilon(N)$. In a nutshell, cyanide poisoning reduces the hemoglobin's oxygen-transport protocol into the negative $\mathcal{GVG} - \mathcal{R}P$ class. The protocol under poisoning belongs to the *unstable* protocol category, the worst category in Figure 6.

2.7 Countermeasuring treatments

Nevertheless, it is easy to restore an algorithm/protocol from the negative $\mathcal{R}P$ class to the $\mathcal{R}P$ class. This can be done by introducing treatment agents that have a much greater binding affinity for cyanide ion than hemoglobin(Vick & Froehlich, 1988)(Heijst & Meredith, 1990)(Mushett, 1952). For example, cyanide preferentially bonds to methemoglobin rather than the cytochrome oxidase, and hydroxycobalamin (a form of vitamin B_{12}) can be used to bind cyanide to form the harmless vitamin B_{12a} cyanocobalamin. A treatment like this can be analyzed below.

Suppose we can introduce $\gamma\cdot N$ treatment agent nodes to bind with cyanide ion. The probability that there are k such treatment agents in the space $\mathcal{S}'$ is

$$\mathsf{Pr}[w = k] = \iiint_{\mathcal{S}'} \frac{(\gamma \cdot N \cdot \rho_1)^k}{k!} \cdot e^{-\gamma\cdot N\cdot\rho_1} \, \mathrm{d}\mathcal{S}$$

Whenever $\mathit{Pr}[w \geq 1]$, a cyanide ion will bind toward the treatment agent rather than a hemoglobin node. A hemoglobin node's oxygen binding success probability is changed to be

$$\begin{aligned}
P_{success} &= \mathit{Pr}[y \geq 1] \cdot \mathit{Pr}[w \geq 1] \\
&= \iiint_{\mathcal{S}'} \Big((1 - e^{-(1-\theta)\cdot N \cdot \rho_1}) \cdot (1 - e^{-(1-\gamma)\cdot N \cdot \rho_1} \Big) \ \mathrm{d}\mathcal{S} \\
&= \iiint_{\mathcal{S}'} \left((1 - \epsilon(N)) \cdot (1 - \epsilon(N)) \right) \ \mathrm{d}\mathcal{S} \\
&> \iiint_{\mathcal{S}'} (1 - 2\epsilon(N)) \ \mathrm{d}\mathcal{S}.
\end{aligned}$$

This is a $1 - \epsilon(N)$ quantity. In contrast, the failure probability becomes $\epsilon(N)$. This way, the hemoglobin node's oxygen-transport scheme is converted from the negative $\mathcal{R}P$ class back to the $\mathcal{R}P$ class. The protocol under countermeasuring healing belongs to the *exponentially stable* protocol category, the best category in Figure 6.

3. Summary

In this work we have formally described the behavior of cyanide poisoning following a complexity-theoretic approach. Like modern cryptography, security threat in our network-centric model is defined on the complexity-theoretic concept of "negligible" $\epsilon(x)$, which is asymptotically sub-polynomial with respect to a pre-defined system parameter x. The parameter x is the key length n in modern cryptography, but is changed to the total number of network nodes N in our model.

When the victim circulatory system is treated as an enclosed network space, a hemoglobin is treated as a mobile node, and a hemoglobin binding with cyanide ion is treated as a compromised node, we define a randomized complexity class $\mathcal{GVG} - \mathcal{R}P$ to show how cyanide poisoning can reduce the probability of oxygen/message delivery to a $\epsilon(N)$ quantity. This is accomplished in two steps: (1) We prove that the negligibility property $\epsilon(N)$ is an asymptotic invariant in terms of the input parameter N for any polynomial-time algorithm; (2) We also prove that the life-sustaining node-wise oxygen-transport function succeeds (or fails in case of poisoning treatment) with $\epsilon(N)$ probability, then by the invariant property the network-wise oxygen-transport function also succeeds (or fails in case of poisoning treatment) with the invariant $\epsilon(N)$ probability. This leads to a new analysis of biological threats based on network and complexity theoretic study.

Within this complexity-theoretic model, we classify the evaluation result of any self-organizing network protocol into four categories: *unstable*, *Lyapunov stable*, *asymptotically stable* and *exponentially stable*. Research efforts could produce optimizing designs to upgrade the target protocol to be exponentially stable eventually, or attacking designs to downgrade the target protocol to be unstable eventually.

4. References

Bettstetter, C. (2001). Mobility Modeling in Wireless Networks: Categorization, Smooth Movement, and Border Effects, *ACM Mobile Computing and Communication Review* **5**(3): 55–67.

Bettstetter, C., Hartenstein, H. & Perez-Costa, X. (2004). Stochastic Properties of the Random Waypoint Mobility Model, *ACM/Kluwer Wireless Networks, Special Issue on Modeling and Analysis of Mobile Networks* **10**(5): 555–567.

Bettstetter, C. & Wagner, C. (2002). The Spatial Node Distribution of the Random Waypoint Mobility Model, *German Workshop on Mobile Ad Hoc Networks (WMAN)*, pp. 41–58.

Blum, M. & Micali, S. (1982). How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits, *Symposium on Foundations of Computer Science (FOCS)*, pp. 112–117.

Blum, M. & Micali, S. (1984). How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits, *Society for Industrial and Applied Mathematics (SIAM) Journal on Computing* **13**(4): 850–864.

Cressie, N. (1993). *Statistics for Spatial Data*, John Wiley and Sons.

Diffie, W. & Hellman, M. E. (1976). New Directions in Cryptography, *IEEE Transactions on Information Theory* **22**(6): 644–654.

Dolev, D., Dwork, C. & Naor, M. (1991). Non-malleable cryptography, *23th Symposium on the Theory of Computation (STOC)*, pp. 542–552.

Goldwasser, S. & Micali, S. (1984). Probabilistic Encryption, *Journal of Computer and System Sciences* **28**(2): 270–299.

Heijst, A. V. & Meredith, J. (1990). Antidotes for Cyanide Poisoning, *in* G. Volanis, J. Sims, F. Sullivan & P. Turner (eds), *Basic Science in Toxicology (Brighton, Taylor and Francis Publishers)*, pp. 558–566.

Kindwall, E. (1977). Carbon monoxide and cyanide poisoning, *in* J. Davis & T. Hunt (eds), *Hyperbaric Oxygen Therapy, Bethesda MD: Undersea Medical Society*, pp. 177–190.

Mushett, C. (1952). Antidotal Efficacy of Vitamin B12 a (hydroxocobalamin) in Experimental Cyanide Poisoning, *Proceedings of The Society for Experimental Biology and Medicine* **81**: 234–237.

Naor, M. & Yung, M. (1990). Public-key Cryptosystems Provably Secure Against Chosen Ciphertext Attacks, *22nd Symposium on the Theory of Computation (STOC)*, pp. 427–437.

Rivest, R. L., Shamir, A. & Adleman, L. M. (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, *CACM* **21**(2): 120–126.

Shannon, C. E. (1949). Communication Theory of Secrecy Systems, *Bell System Technical Journal* **28**(4): 656–715.

Sykes, A. (1981). Early studies on the toxicology of cyanide, *in* B. Vennesland, E. Conn, C. Knowles, J. Westly & F. Wissing (eds), *Cyanide in Biology*, pp. 1–9.

Vick, J. & Froehlich, H. (1985). Studies of Cyanide Poisoning, *Arch Int Pharmacodyn Ther* **273**: 314–322.

Vick, J. & Froehlich, H. (1988). Treatment of Cyanide Poisoning, *J Toxicol Clin Exp* **25**: 125–138.

Yao, A. C.-C. (1982). Theory and Applications of Trapdoor Functions (Extended Abstract), *Symposium on Foundations of Computer Science (FOCS)*, pp. 80–91.