Towards a Framework for Multilayer Computing of Survivability

Abolghasem Sadeghi[1], Mohammad Reza Valavi[2], Morteza Barari[3]
Department of Information and Computer Technology
Malek Ashtar University of Technology
Tehran, Iran

***Abstract*—The notion of survivability has an important position in today enterprise systems and critical functions. This notion has been defined in different ways. However, lacking a comprehensive and multilayer model for computing the survivability quantitatively, is the major gap happened in researches of this field; a model that is tally general and applicable in various applications. This research tries to design a comprehensive, multilayer as well as general model for modeling and computing the survivability. Considering that the Markov property is true in our proposed model, we used the Markov model. Using the proposed three layer architecture and designing a Markov structure, we could have been able to compute the survivability initially for each of infrastructure components separately and regardless of their functional dependency to each other. The computations were generalized to consider component dependencies as well as the upper layers entering dependencies in Markov model and could compute the survivability of each vital function for the highest architectural layer based on the underlying layers. Finally, a common and ordinary structure of crisis management has been studied and its results analyzed. We could examine the abilities of our model to compute the survivability of the whole crisis management system successfully.**

***Keywords*—Network survivability; survivability quantification; survivability computation; system survivability**

I. Introduction

Today, all social, cultural, political and economic life aspects of societies and states are dependent on the information technology infrastructures and this dependency is ever increasing. Due to this dependency, important concerns on the functional quality and serving those infrastructures have been emerged. This issue is becoming more important day by day that whether these infrastructures can tolerate different challenges -including natural ones from flood and earthquake to human errors or adversary invasions- and can provide their major and essential services. Therefore we need to compute the resistance of infrastructures against such challenges for better planning, implementing and utilizing them. This will enable us to find appropriate solutions for improving the resistance property. This is explained by the survivability metric.

A. Qualitative Definitions of Survivability

Like many other scientific subjects, there is no consensus and unanimous definition for the survivability. Table I summarizes definitions yet provided. The definitions have been ordered chronologically and their references have also been given. The definitions are dependent on the field within which are required and their own origins. There are multiple differences between these definitions, so there should be a deep understanding of intended problem to find the more suitable one. In our context, the dominant definition that many other researches have used, is the fourth definition. So we will use it too.

B. Quantitative Definition of Survivability

All definitions contained in Table I have a qualitative approach. ANSI has provided a quantitative definition for survivability [27] that models the survivability concept parametrically. Fig. 1 shows this definition. In this definition, the measure of interest M has the value m_0 just before a failure occurs. The survivability of this system is represented by the following attributes:

- m_a is the value of M immediately after the failure.
- m_u is the maximum possible difference between m_0 and m_a after failure.
- m_r is the restired value of M after time t_r.
- t_r is the time required for achieving the value of m_0 for M again or a reduced but acceptable value m_0 if m_0 is impossible to be fully restored.

The notion of survivability may seem similar or overlapping to certain notions of dependability field like reliability, availability, fault tolerance, maintainability, security and safety. These similarities and differences have been discussed in various important references such as [12,24,25,28-30]. Thus, we refer the reader to those references.

II. Related Works

Various researches have been performed on the survivability implicitly and explicitly. By the explicit researches, we mean those that have been clearly focused on the survivability. However, implicit researches are those dealing with related concepts like system recovery or intrusion tolerance. Moreover, some of them have just provided a qualitative model in this field and have not indicated a way to inferring the level of survivability from such models. Others have attempted to make the issue quantitative and compute the survivability.

TABLE I. DIFFERENT DEFINITIONS PROVIDED FOR THE SURVIVABILITY

No	Scientific area	Year	Definition	Reference
1	IT systems	1988	Survivability is the degree to which essential functions are still available even though some part of the system is down.	[2]
2	Telecommunication Systems	1996	Survivability is a property of a system, subsystem, equipment, process, or procedure that provides a defined degree of assurance that the named entity will continue to function during and after a natural or man-made disturbance.	[15]
3	Network Computing Systems	1997	Survivability is the ability of a network computing system to provide essential services in the presence of attacks and failures and recover full services in a timely manner.	[3]
4	Critical and Defense Systems	1999	Survivability is the capability of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures or accidents.	[4-7] [9,10]
5	Critical and Defense Systems	2000	Survivability is the ability [of a system] to continue to provide service, possibly degraded or different, in a given operating environment when various events cause major damage to the system or its operating environment.	[1,11]
6	Networking & Telecommunication	2005	Survivability is the ability of system to deliver the minimum expected service when defined threats are faced and the system must retain those properties wanted by the users.	[19]
7	Telecommunication Systems	2009	Survivability is the system's ability to continuously deliver services in compliance with the given requirements in the presence of failures and other undesired events.	[16]
8	Aerospace & Military	2009	Survivability is the ability of a system to minimize the impact of finite-duration environmental disturbances on value delivery.	[20]
9	Networking & Telecommunication	2015	Survivability is a concept that describes the capability of a system to achieve timely recovery after the occurrence of undesired events	[26]

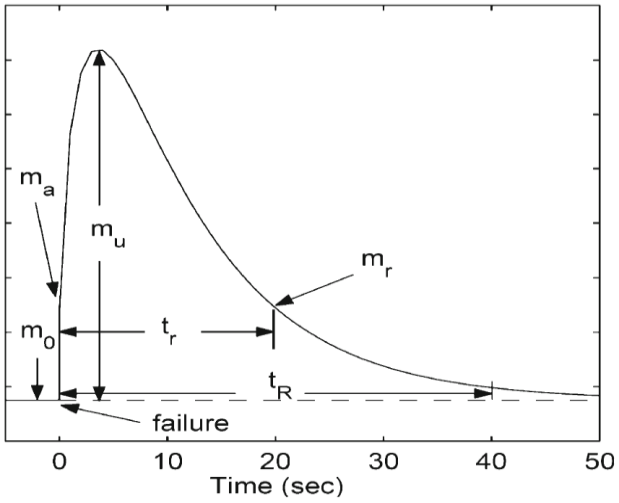

Fig. 1. ANSI Definition for Survivability.

SABER model in [13] dealt with providing an appropriate architecture for intrusion tolerance in the systems. This architecture has a conventional network security approach enabling it to continue the wanted services under an intrusion or attack using IDS sensors and a higher security level called SOS. This research includes only software attacks and has nothing covering other malicious and non-malicious undesired events.

ITDOS architecture [14] provided an intrusion-tolerant software structure for software systems using facilities based on CORBA firmware. Thereby, it is ensured that all CORBA-based softwares produced with proposed extensions are intrusion-tolerant. However, no review and implementation of mentioned architecture has been reported.

AWDRAT method [18] provides a self-adaptive method in software firmware to be able to detect the possibly compromised point comparing operating program behavior with the desired behavior. Then, a trust management system manages the restoration process changing the execution path from the previously compromised components to the unaffected ones to enable the system continue secure and trusted operation under attack conditions. Although this research had successful experimental results, it has focused merely on the software malicious threats and overlooked other threat aspects and is not fully comprehensive yet.

DPASA architecture [17] provides a model for system state recovery after a cyber attack. It uses a set of tools and methods for identifying, protecting and adaptive reactions. For assessing the model performance, it was entered in an applied example of JBI belonging to US Air Force laboratory and has modeled and represented accuracy of recovery and attack tolerance with several parameters. However, the parameters introduced in this research are not public and can not be generalized to other problems. Thereby, any new problem will need its own parameters to be extracted. This research emphasizes only on the cyber attacks and is also general and qualitative. Qualitative means it does not have any measureable parameters.

Willow architecture [8] claims creating survivability in wide critical and distributed systems. This architecture uses a combination of fault avoidance, fault elimination and fault tolerance. It disables vulnerable components under threatening conditions. Then replaces damaged components after a fault or intrusion. When an indispensable fault appears it will be able to strengthen the system against that using reconfiguration methods based on a control system feedback. This architecture reported an experimental study for the US air forces on the JBI basis whose results have not yet been published and has sufficed just the claim that the system has functioned successfully.

The previous works reviewed here were qualitative models that provide no accurate computations. Trivedi et al. in [16] and [26] have provided a model for computing survivability quantitatively that has been designed based on composition of availability and performability. They provided two Markov models for availability and performability in their research.

Then, they combined them and devised a complex Markov model for system survivability. For verification of the provided model, they modeled it on a given telecommunication system and computed the survivability supposing that the input variables of the model have been taken. This model has the advantage of being quantitative and computable. However, the research finally gives no accurate sense to the software service users on that how they can verify their service survivability. Moreover, the supposed problem in this research is a very well-known and already solved problem with clear solutions while imagination of such clarity for broader and more complex problems is generally difficult if not impossible. For this, we can say that the proposed method is slightly difficult to generalize to other problems in this field.

In Survivability Analysis of a Computer System under an Advanced Persistent Threat Attack, [31] has attempted to model and compute a software system survivability under APT attacks. For this purpose, they proposed an integrated model of process of an APT attack as well as different steps for defending it. Their final goal was to create a continuous time Markov model for this issue to compute the total survivability of the system under an APT attack. To create the model, different steps of the ATP attack have been modeled with Stochastic Reward Nets and its graph has been produced. Then, the reachability graph of this petri net is drawn as a continuous time Markov model for computing survivability. The graph introduces system recovery, system reachability, data confidentiality, and data accuracy as the four parameters of the survivability model to compute the survivability of this system. SRN net and Markov model are created here for computation of the aforementioned four parameters. Finally, to be able to verify the model, authors have obtained some of probable values required for the model from the valid references or supposed them and applying the values to the model. So, they computed survivability quadruplet probability measures. While an appropriate computation has been proposed in this model for survivability, the proposed model is allocated to APT cyber attacks and isn't suitable to be applied to other applications.

For computing the survivability generally in software layer, [21] attempted to provide measurable criteria for defining and assessing software survivability from the end user's viewpoint. Doing so, they have provided a framework for defining software survivability quantities and enabling the user to design and execute various policies for achieving survivability based on those quantities. A decision support model has also been proposed to realize the survivability quantities to ensure the minimum survivability for the software. The aforementioned quantities are classified into five groups as follow:

- Adaptability
- Recoverability
- Fault tolerance
- Reliability
- Performance degrading

Each group represents one of the characteristics of the survivability and each has several quantities that are survivability related quantities. Finally, survivability computations are classified into two groups: contribution-oriented and concern-oriented. The contribution-oriented functions compute those characteristics of the survivability that the user needs them essentially and must be met fully. In contrast, the concern-oriented functions deal computation of those characteristics and quantities of the survivability on which the user is concerned about but can tolerate violation of them up to a certain level.

The same author in [22] and [23] has attempted to use proof-carrying codes for survivability assessment. The general idea of this method is to enable the user to define his software survivability requirements and provide it to the software vendor. Then, the vendor will be able to provide the user with a system using proof-carrying code method that enables the user himself to assess his system survivability based on the initially proclaimed requirements. The main reference for introducing proof-carrying codes is [32] which is used in this method.

All works reported here from Dr. Zuo have computed and parameterized the survivability only based on outstanding characteristics of the software system itself. While the secure and correct execution of any software system is subjected to the security and correctness of infrastructure components performance that the system relies on them. Unfortunately, these researches have not discussed them and have not replied the ambiguity here.

III. Basic Architecture

As mentioned in section 2 the general and widespread weakness in all works of this field was that the system user can not compute the overall system survivability based on his information about different layers. Some of works have dealt with computing the survivability of the infrastructure layer without enabling the user to use it for computing service/software survivability. Some of them have performed it in software layer without taking into account a logical and working dependency between the software layers with underlying layers. Naturally, these computations are not comprehensive and do not have enough accuracy and integrity. A suitable model is required for survivability computation that connects layers to remove this challenge. Fig. 2 shows this model.

The provided model is a set of various and heterogeneous agents and components that are set up beside each other randomly and unpredictably and each component can be connected to others and there is no predefined limitation for services that is provided to other components. Of course it is clear that we do not mean the practical limits like memory, connection link capacity, etc. Each of those components participates in one or more application belonging to the software layer. The total system is depicted as a set of functions or services in the top layer namely operation level. In this layer we deal with organizational processes as functional components of that layer. Functions are executed using several applications. In other words, each function need some applications for operating. In the given model of Fig. 2, the system is supposed to have X functions that use n applications for fulfilling their functions and services. Application systems are executed on the basis of k components.

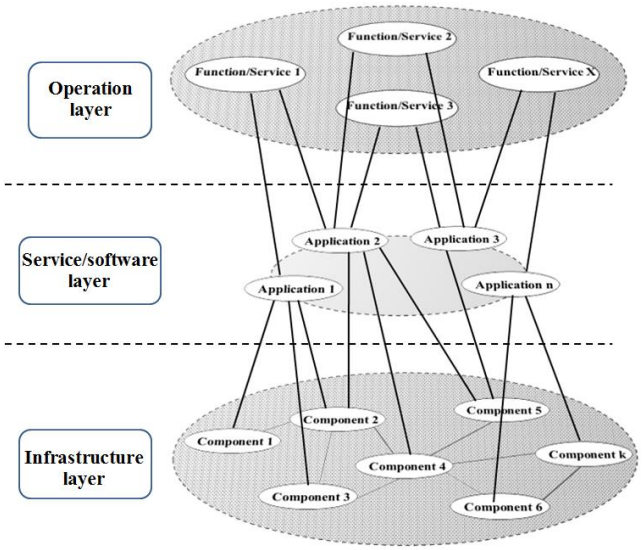

Fig. 2. Basic Model of Layers for Survivability Quantification.

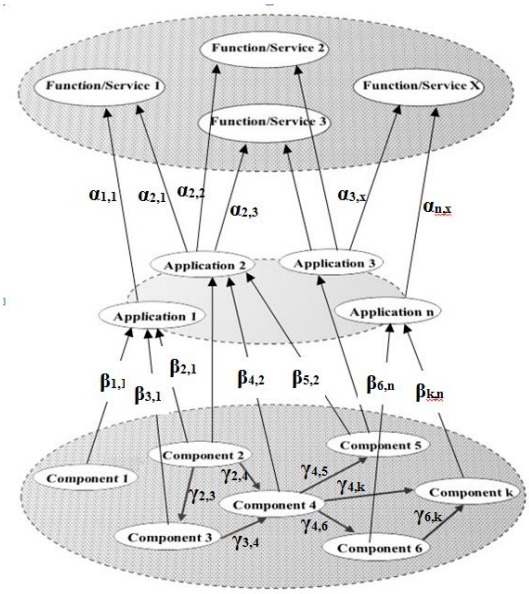

Fig. 3. Dependency Metrics in Survivability Model.

C. Relations in the Model

The relationship between components must be understood and analyzed accurately to make the model efficient and practically useable. The components relationships to each other in the infrastructure layer is transversal while the relationship of infrastructure layer components with software layer applications and between software layer applications with functions of the operation level is longitudinal. In a real environment the components can serve each other. Therefore, it is required that to suppose the relationship between components a directional relationship for demonstrating that which component is client and which one is service provider.

Given the directionality of the graph, it must be cleared that does the graph have a loop or would be a DAG? Although it is acceptable to suppose that this graph can involve a loop, it may be a DAG. In this regard, what is important here is our attitude resolution and granularity. For instance, if a smart building management system is taken as a component then this component can provide the inputs needed for other systems like ventilation, cooling, electricity, etc. where the supposed component has low granularity in this situation. If the BMS system is separated to its basic facilities and modules and each module is considered as a component, then that components will be single task that lead us achieving a loop free graph. Whether or not, we supposed the graph of components as DAG and provide our algorithm based on it. Although this decomposition process helps to achieve a DAG, it is obvious that appropriate algorithms can be developed in future works considering the graph a cyclic one.

In the upper layer, it is possible to consider no direct dependency between them because when system A serves system B it means that some of components in system A serve some of components in system B. Indeed, this concept is considered in relationship between components. Therefore, there is no explicit transversal dependency between applications and independent set of applications form a function in operation layer. Fig. 3 depicts this notion.

Now, we must analyze dependency of components to each other, dependency of applications to components and dependency of functions to applications separately and quantify them. Doing so, the model edges are named according to fig. 2. Moreover, for the sake of facilitation in representing topics, the applications are symbolized with AP_i, functions with FS_i and components with CMP_i. In this model, $\alpha_{i,j}$ represents the total dependency of FS_i to AP_i. Further, $\beta_{i,j}$ represents dependency of AP_j to CMP_i. $\gamma_{x,y}$ represents the dependency of component y to component x. α, β and γ coefficients are real numbers between 0 and 1. Now, we discuss properties of these coefficients in the graph of Fig. 3.

$$\forall j, \sum_i \alpha_{i,j} = 1 \tag{1}$$

Because, each function FS_j is consisted of its applications and regardless of user mistakes, the full execution of applications means that the function FS_j will be executed completely.

$$\forall j, \sum_i \beta_{i,j} = 1 \tag{2}$$

Meaning that the full operation of any application is subjected to the fact that all concerning components fulfill their tasks completely, because each system only is consisted of its components functioning well and no other components intervening correct application execution.

$$\forall cmp_j, \sum_i \gamma_{i,j} < 1 \tag{3}$$

Meaning that each component would be partially –and not fully- dependent on other components functionally. In fact, each component definitely has its own special and independent functionality that cause the above summation should be less than 1. If the required inputs for a client aren't provided from one of the service provider components, the function of client

component will be damaged proportionate to coefficient of dependency to the service providing component.

IV. Survivability basic Model

In this section we propose our basic conceptual model for the survivability of any system generally. As we saw in section 1, the survivability aims at enabling the system to continue its vital and essential services and operations under crisis until recovery of failed subsystems. Thus, for modeling the survivability of any system it is required to consider three basic states. The first state is where the system operates normally and naturally. Under such state, the crisis has no degrading effects on the system and operates normally that is called Healthy state. The other is loss of the important and critical subsystems that results in the total failure and break down and is called Fail state. However, the third state is one that some of non critical subsystems are failed but the system can continue its fundamental operation until the problem is removed. This state is called Survive state. Tri-state Markov model in Fig. 4 represents these definitions. In this model, μ and ρ parameters show the transmission rate between various states of Markov model.

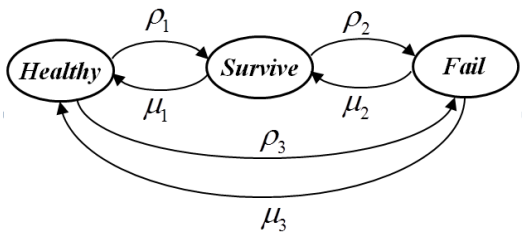

Fig. 4. Survivability basic Model.

V. Computing Survivability

For multilayer computing the survivability across the layers of Fig. 2, it is required to start from the lowest layer and compute it separately for each infrastructure component regardless of its dependency to other components. Then, the infrastructure layer components survivability is computed taking into account their dependency. In the next step, the applications survivability in software layer is computed given their dependency to the infrastructure layer components and computations in that layer. Finally, the functions survivability in the operation layer will be computed based on computations of the software level.

A. Computing the Survivability of a Single Infrastructure Component

The model depicted in Fig. 5 is Markov model for survivability of a single component of infrastructure layer. $attr_i$ is an attribute or subsystem of the component and αi is the probability of failing any $attr_i$. Some attributes or subsystems are critical for basic functioning of the component while others are not. The component could not tolerate failure of critical attributes and the component will enter the fail state. In the case of failing non critical attributes or subsystems, the component can continue its essential functions while entering the survive state. We show critical attributes with * mark in Fig. 5.

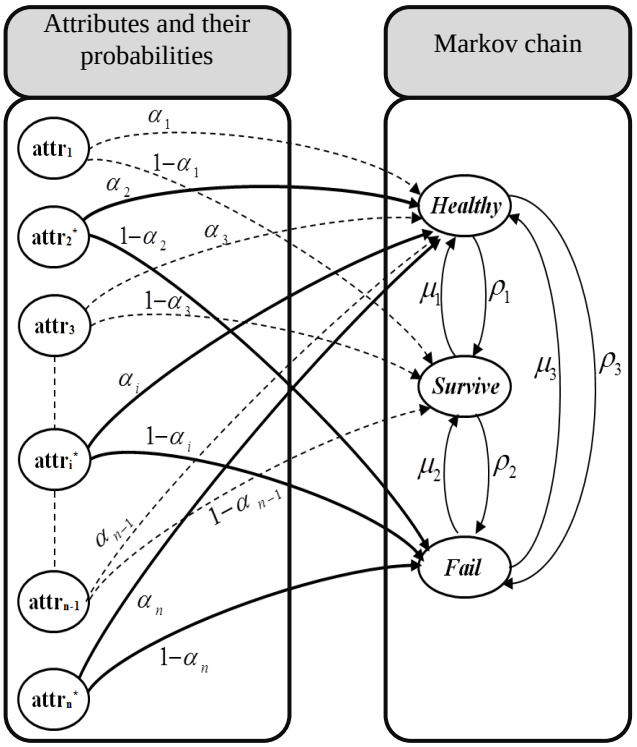

Fig. 5. Combined Markov Model of Survivability Quantification of a Single Component.

In the model shown in Fig. 5, each property of $attr_i$ has a bi-state Markov model as represented in Fig. 6.

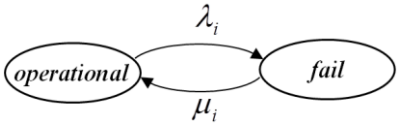

Fig. 6. Markov Model for Each Property.

In this model, λ_i and μ_i are failure and recovery rate of the property i. In transient state, the probability of healthy and fail states in Markov model is computed as follow:

$$\left.\begin{aligned} &\frac{d\,\pi_i^{op}}{d\,t} = -\pi_i^{op}.\lambda_i + \pi_i^{f}.\mu_i \\ &\pi_i^{op} + \pi_i^{f} = 1 \end{aligned}\right\} \Rightarrow \begin{cases} \pi_i^{op} = \dfrac{\mu_i}{\mu_i+\lambda_i} + c.e^{-(\lambda_i+\mu_i)t} \\ \pi_i^{f} = \dfrac{\lambda_i}{\mu_i+\lambda_i} - c.e^{-(\lambda_i+\mu_i)t} \end{cases} \quad (4)$$

In Eq. (4), π_i^{op} and π_i^{f} mean the probability of healthy and failure states for the property i that are symbolized as α_i and 1-α_i in Fig. 5 for the sake of facilitation in reading and writing. The number c is an arbitrary constant. Thus, we have:

$$\alpha_i = \frac{\mu_i}{\mu_i+\lambda_i} + c.e^{-(\lambda_i+\mu_i)t} \quad (5)$$

In the steady state, the probability of healthy and fail states is as follow:

$$\left.\begin{aligned}\pi_i^{op}.\lambda_i=\pi_i^{f}.\mu_i\\ \pi_i^{op}+\pi_i^{f}=1\end{aligned}\right\}\Rightarrow\begin{cases}\pi_i^{op}=\dfrac{\mu_i}{\mu_i+\lambda_i}\\ \pi_i^{f}=\dfrac{\lambda_i}{\mu_i+\lambda_i}\end{cases}\quad(6)$$

In model shown in Fig. 5, values assigned to α_i are probability type while values of ρ and μ are rate. On the other hand, values of α_i are given and known already. Therefore, probabilities of Markov model tri-states must be obtained first to compute rates of ρ and μ. Then, ρ and μ are computed based on the probabilities of three states. For this purpose, three sets are introduced for using in Eq. (7). *S* is a set including all properties of this component. The set *IC* is a subset of critical characteristics of *S* and the set *INC* includes non-criticals. Following section shows formulae for the survivability computation.

$$S=\{i\in \mathrm{N}\,|\,attr_i\ is\ valid\}$$
$$IC=\{i\in S\,|\,attr_i\ is\ critical\ \ attribute\}$$
$$INC=\{i\in S\,|\,attr_i\ is\ not\ critical\ \ attribute\}$$
$$IC\cap INC=\phi\ \ \&\&\ \ IC\cup INC=S$$

$$\pi_{Healthy}=\prod_{i\in S}\alpha_i \quad (a)$$

$$\pi_{Survive}=\prod_{i\in IC}\alpha_i\left(\sum_{\substack{A\subset INC\\ A\neq INC}}\left(\prod_{j\in A}\alpha_j\prod_{k\in INC-A}(1-\alpha_k)\right)\right)\quad(b)$$

$$\pi_{Fail}=\sum_{\substack{A\subset IC\\ A\neq IC}}\left(\prod_{i\in A}\alpha_i\prod_{j\in IC-A}(1-\alpha_j)\right)\sum_{B\subseteq INC}\left(\prod_{k\in B}\alpha_k\prod_{l\in INC-B}(1-\alpha_l)\right)\quad(c)\qquad(7)$$

Given the practical conditions in this model, it is possible to consider all properties independent. Even with some of properties depending on each other practically, the desired independency can be obtained through changing the system design. According to this assumption, the probability of a fully healthy state equals to multiplication of all properties healthy states probabilities that is shown in Eq. (7)(a). for computing the probability of survive state, the failure probability of non-critical properties are considered with their different permutations and multiply it by the probability of critical properties healthy probability. This is shown in Eq. (7)(b). However, the probability for the fail state equals the state within which some of critical properties are failed regardless of whether non-critical properties are healthy or not that is shown in Eq. (7)(c).

The probabilities related to states of Markov model of Fig. 5 have been computed in Eq. (7). Now, we should prove that summation of these three states equals 1 according to Markov model conditions. In other word, following equation must be true.

$$\pi_{Healthy}+\pi_{Survive}+\pi_{Fail}=1$$

Theorem: prove that the following equation is true in Markov model of Fig. 5:

$$\pi_{Healthy}+\pi_{Survive}+\pi_{Fail}=1$$

Proof: first, for simplification of notations we define:

$$\left.\begin{aligned}X=\sum_{\substack{M\subset INC\\ M\neq INC}}\left(\prod_{i\in M}\alpha_j\prod_{j\in INC-M}(1-\alpha_k)\right)\\ Y=\sum_{\substack{N\subset IC\\ N\neq IC}}\left(\prod_{i\in N}\alpha_i\prod_{j\in IC-N}(1-\alpha_j)\right)\end{aligned}\right\}\Rightarrow\begin{cases}\pi_{Healthy}=\prod_{i\in S}\alpha_i\\ \pi_{Survive}=\prod_{i\in IC}\alpha_i.X\\ \pi_{Fail}=Y.\left(X+\prod_{i\in INC}\alpha_i\right)\end{cases}$$

Now, given the sections (a) to (c) of Eq. (7):

$$\begin{aligned}&\pi_{Healthy}+\pi_{Survive}+\pi_{Fail}=\\ &\prod_{i\in S}\alpha_i+\prod_{i\in IC}\alpha_i.X+Y.\left(X+\prod_{i\in INC}\alpha_i\right)=\\ &\prod_{i\in S}\alpha_i+\prod_{i\in IC}\alpha_i.X+Y.X+Y.\prod_{i\in INC}\alpha_i=\\ &\prod_{i\in IC}\alpha_i.\prod_{i\in INC}\alpha_i+\prod_{i\in IC}\alpha_i.X+Y.X+Y.\prod_{i\in INC}\alpha_i=\\ &X.\left(\prod_{i\in IC}\alpha_i+Y\right)+\prod_{i\in INC}\alpha_i.\left(\prod_{i\in IC}\alpha_i+Y\right)=\\ &\left(X+\prod_{i\in INC}\alpha_i\right)\left(Y+\prod_{i\in IC}\alpha_i\right)=\\ &\sum_{A\subseteq INC}\left(\prod_{j\in A}\alpha_j\prod_{k\in INC-A}(1-\alpha_k)\right).\sum_{A\subseteq IC}\left(\prod_{j\in A}\alpha_j\prod_{k\in IC-A}(1-\alpha_k)\right)=\\ &\sum_{A\in P(S)}\left(\prod_{i\in A}\alpha_i\prod_{j\in S-A}(1-\alpha_j)\right)\end{aligned}\qquad(8)$$

The *P(S)* in the final result of Eq. (8) is the power set of *S*. Indeed, the final result in Eq. (8) contains all possible permutations of failure or healthy state probability for each of properties through a linear polynomial. Now, it must be proved that the last sentence of Eq. (8) equals 1. To do so, the mathematical induction method is applied. For base case S must have two members. We know that sum of two elements of *S* is 1. So:

$$\begin{aligned}&S=\{\alpha_1,\alpha_2\}\ ,\ \alpha_1+\alpha_2=1\\ &\sum_{A\in P(S)}\left(\prod_{i\in A}\alpha_i\prod_{j\in S-A}(1-\alpha_j)\right)=\alpha_1\alpha_2+\alpha_1(1-\alpha_2)+\alpha_2(1-\alpha_1)+(1-\alpha_1)(1-\alpha_2)=\\ &\alpha_1\alpha_2+\alpha_1-\alpha_1\alpha_2+\alpha_2-\alpha_1\alpha_2+1-\alpha_1-\alpha_2+\alpha_1\alpha_2=1\end{aligned}\qquad(9)$$

Therefore the theorem for S with two members is true. Now, suppose that for S with n members the desired sentence equals 1. It must be proved that the relation is also true for *S* with n+1 members. S^n represents the set S has n members. Thus, we have:

$$\sum_{A\in P(S^{n+1})}\left(\prod_{i\in A}\alpha_i\prod_{j\in S^{n+1}-A}(1-\alpha_j)\right)= \tag{10}$$

$$\alpha_{n+1}\cdot\sum_{A\in P(S^n)}\left(\prod_{i\in A}\alpha_i\prod_{j\in S^n-A}(1-\alpha_j)\right)+(1-\alpha_{n+1})\sum_{A\in P(S^n)}\left(\prod_{i\in A}\alpha_i\prod_{j\in S^n-A}(1-\alpha_j)\right)=$$

$$\sum_{A\in P(S^n)}\left(\prod_{i\in A}\alpha_i\prod_{j\in S^n-A}(1-\alpha_j)\right)=1$$

Therefore, the sum of probability of three states will be always equal to 1.

Now, combining formulae in Eq. (4) and Eq. (7) for computing probabilities related to the model of Fig. 5 in transient state, we have:

$$\pi_{Healthy}=\prod_{i\in S}\alpha_i=\prod_{i\in S}\left(\frac{\mu_i}{\mu_i+\lambda_i}+e^{-(\lambda_i+\mu_i)t}\right) \tag{a}$$

$$\pi_{Survive}=\prod_{i\in IC}\alpha_i\left(\sum_{\substack{A\subset INC\\A\neq INC}}\left(\prod_{j\in A}\alpha_j\prod_{k\in INC-A}(1-\alpha_k)\right)\right)= \tag{11}$$

$$=\prod_{i\in IC}\left(\frac{\mu_i}{\mu_i+\lambda_i}+e^{-(\lambda_i+\mu_i)t}\right)\left(\sum_{\substack{A\subset INC\\A\neq INC}}\left(\prod_{j\in A}\left(\frac{\mu_j}{\mu_j+\lambda_j}+e^{-(\lambda_j+\mu_j)t}\right)\prod_{k\in INC-A}\left(\frac{\lambda_k}{\mu_k+\lambda_k}-e^{-(\lambda_k+\mu_k)t}\right)\right)\right) \tag{b}$$

$$\pi_{Fail}=\sum_{\substack{A\subset IC\\A\neq IC}}\left(\prod_{i\in A}\alpha_i\prod_{j\in IC-A}(1-\alpha_j)\right)\sum_{B\subseteq INC}\left(\prod_{k\in B}\alpha_k\prod_{l\in INC-B}(1-\alpha_l)\right)=$$

$$=\sum_{\substack{A\subset IC\\A\neq IC}}\left(\prod_{i\in A}\left(\frac{\mu_i}{\mu_i+\lambda_i}+e^{-(\lambda_i+\mu_i)t}\right)\prod_{j\in IC-A}\left(\frac{\lambda_j}{\mu_j+\lambda_j}-e^{-(\lambda_j+\mu_j)t}\right)\right)\sum_{B\subseteq INC}\left(\prod_{k\in B}\left(\frac{\mu_k}{\mu_k+\lambda_k}+e^{-(\lambda_k+\mu_k)t}\right)\prod_{n\in INC-B}\left(\frac{\lambda_n}{\mu_n+\lambda_n}-e^{-(\lambda_n+\mu_n)t}\right)\right) \tag{c}$$

Probabilities of the steady state in Fig. 5 are as follows:

$$\pi_{Healthy}=\prod_{i\in S}\alpha_i=\prod_{i\in S}\left(\frac{\mu_i}{\mu_i+\lambda_i}\right) \tag{a}$$

$$\pi_{Survive}=\prod_{i\in IC}\alpha_i\left(\sum_{\substack{A\subset INC\\A\neq INC}}\left(\prod_{j\in A}\alpha_j\prod_{k\in INC-A}(1-\alpha_k)\right)\right)=\prod_{i\in IC}\left(\frac{\mu_i}{\mu_i+\lambda_i}\right)\left(\sum_{\substack{A\subset INC\\A\neq INC}}\left(\prod_{j\in A}\left(\frac{\mu_j}{\mu_j+\lambda_j}\right)\prod_{k\in INC-A}\left(\frac{\lambda_k}{\mu_k+\lambda_k}\right)\right)\right) \tag{b}$$

$$\pi_{Fail}=\sum_{\substack{A\subset IC\\A\neq IC}}\left(\prod_{i\in A}\alpha_i\prod_{j\in IC-A}(1-\alpha_j)\right)\sum_{B\subseteq INC}\left(\prod_{k\in B}\alpha_k\prod_{l\in INC-B}(1-\alpha_l)\right)=\sum_{\substack{A\subset IC\\A\neq IC}}\left(\prod_{i\in A}\left(\frac{\mu_i}{\mu_i+\lambda_i}\right)\prod_{j\in IC-A}\left(\frac{\lambda_j}{\mu_j+\lambda_j}\right)\right)\sum_{B\subseteq INC}\left(\prod_{k\in B}\left(\frac{\mu_k}{\mu_k+\lambda_k}\right)\prod_{n\in INC-B}\left(\frac{\lambda_n}{\mu_n+\lambda_n}\right)\right) \tag{c}$$

(12)

adad

In this model, the number of failures of the system over time t is obtained from Eq. (13):

$$N_{Failure}(t)=\int_0^t(\rho_2+\rho_3)\,\pi_{Fail}(x)dx=2\int_0^t\rho_2\,\pi_{Fail}(x)\,dx \tag{13}$$

We assume $\rho_2=\rho_3$ *in* Eq. (13) because they implicitly describe an equivalent rate. ρ_2 is the rate of transmission from survive state to failure state, but ρ_3 is the rate of transmission from healthy state to failure state. Actually, both ρ_2 and ρ_3 describe the rate of failure of critical subsystems of Fig. (5). So, assuming them to be equal can be correct.

B. Survivability Propagation Model of Dependent Components in Infrastructure Layer

At this step, we suppose that a technical component CMP_i is functionally dependent on components C_1 to C_n. Thus, while the CMP_i has its own independent survivability, its final survivability also depends on survivability of C_1 to C_n with coefficients δ and $\overline{\delta}$. So, we must try to compute survivability of CMP_i based on C_1 to C_n survivability along with its own independent survivability. This process is called propagation in our notation. This is represented in Fig. 7.

Coefficients δ and $\overline{\delta}$ in Fig. 7 are obtained by the Eq. (14). $\gamma_{x,i}$ used in this formulae shows the dependency coefficient of CMP_i to C_x and has been taken from Fig. 3. In Eq. (14), $\pi_H^{C_x}$ shows the healthy state probability of infrastructure component C_x that CMP_i is dependent to.

$$\delta_x=\left(\pi_H^{C_x}+\pi_S^{C_x}\right)\gamma_{x,i}$$
$$\overline{\delta_x}=\pi_F^{C_x}\cdot\gamma_{x,i} \tag{14}$$

Now we define following sets for computing dependent component CMP_i survivability.

$$S=\{l\in N\mid CMP_i\ is\ depended\ to\ CMP_l\}$$
$$IC=\{l\in S\mid CMP_l\ is\ critical\ \ for\ CMP_i\}$$
$$INC=\{l\in S\mid CMP_l\ is\ not\ critical\ \ for\ CMP_i\}$$

Based on coefficients δ and $\overline{\delta}$ we compute final survivability of CMP_i through Eq. (15). In Eq. (15), $\pi_H^{abstract}$ means the probability of healthy state before including dependencies. The probability after including dependencies represented by π_H^{final}.

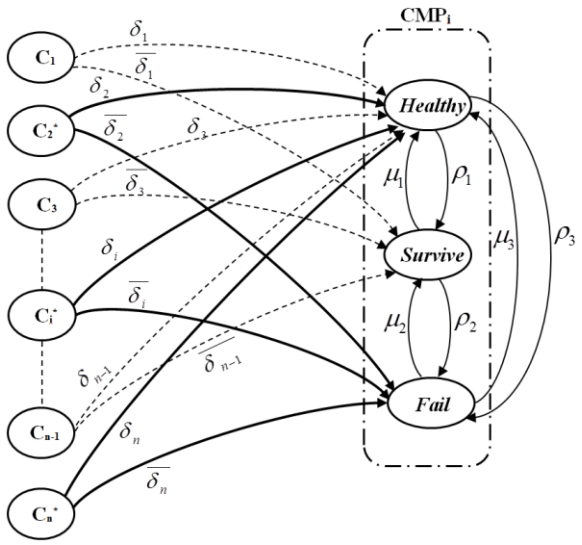

Fig. 7. Survivability Propagation in Infrastructure Layer of Model Among Dependent Components.

$$\pi_H^{final}(CMP_i)=\prod_{x\in S}\delta_x+\left(1-\sum_{x\in S}\gamma_{x,i}\right)\pi_H^{abstract}$$

$$\pi_S^{final}(CMP_i)=\prod_{x\in IC}\delta_x\left(\sum_{\substack{A\subset INC\\ A\neq INC}}\left(\prod_{j\in A}\delta_j\prod_{k\in INC-A}\overline{\delta_k}\right)\right)+\left(1-\sum_{x\in S}\gamma_{x,i}\right)\pi_S^{abstract} \quad (15)$$

$$\pi_F^{final}(CMP_i)=\sum_{\substack{A\subset IC\\ A\neq IC}}\left(\prod_{j\in A}\delta_j\prod_{k\in IC-A}\overline{\delta_k}\right)\sum_{B\subseteq INC}\left(\prod_{j\in B}\delta_j\prod_{k\in INC-B}\overline{\delta_k}\right)+\left(1-\sum_{x\in S}\gamma_{x,i}\right)\pi_F^{abstract}$$

$$\pi_H^{final}(CMP_i)+\pi_S^{final}(CMP_i)+\pi_F^{final}(CMP_i)=1$$

C. Comprehensive Model for Multilayer Survivability Computation

Now we are completely ready for developing our model toward the multilayer computing of survivability. To do so, as we mentioned at the beginning of section 5, we shoud compute the survivability of applications of software layer based on finalized survivability of components. Then we compute the survivability of operation layer functions based on applications survivability of each function. In other words, we must propagate the survivability of infrastructure layer components to software layer applications. Then propagate the survivability of applications to operation layer functions. This process exactly follows the propagation method provided in section 5.2. Fig. 8 shows the process.

In Fig. 8, we compute the survivability of applications with respect to survivability of its underlying components that depends on. This process is similar to previous one for calculating survivability of dependant component CMP_i. When computation of survivability of all applications is done, then we take them into account for computing survivability of operation layer functions in a similar way. One can say, we propagate from software layer to operation layer.

One important point in Fig. 8 is that for healthy operating of any application, it is enough that each underlying component performing its essential functions only. So we can merge the healthy and survive state of components and name it as operational state as illustrated in right portion of Fig. 8.

VI. Survivability of a Crisis Management System

For investigating about the proposed model, we imagined a crisis management system and tried to model it as well. Then we applied the model to the crisis management system for verification of our approach. Based on our studies, we extracted the general model of Fig. 9 for a common crisis management system.

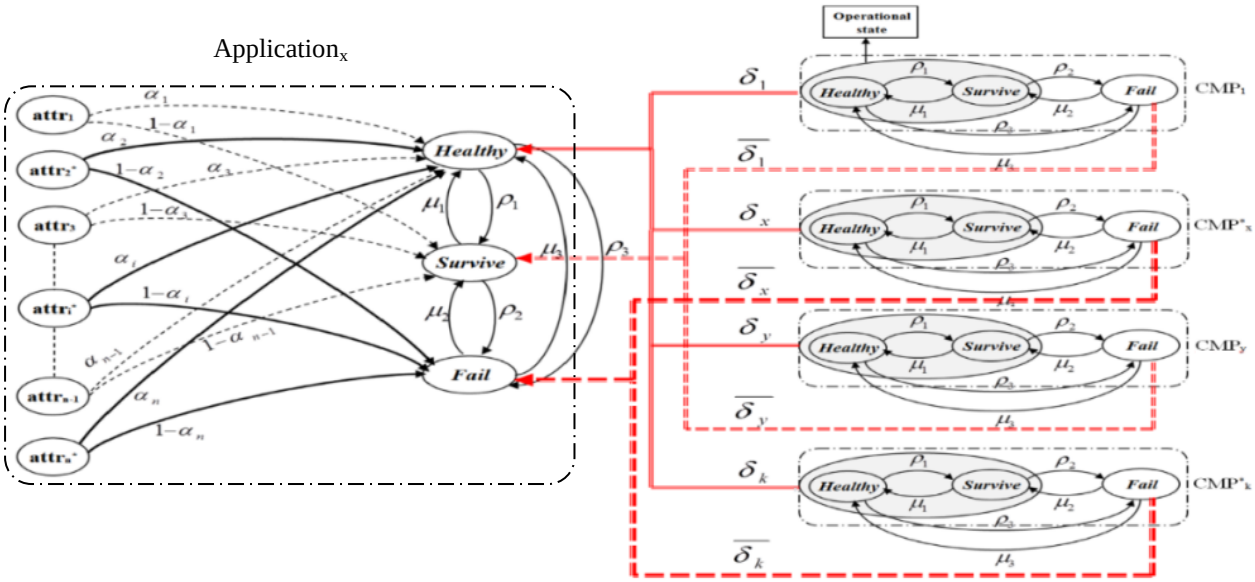

Fig. 8. Comprehensive Model of Survivability Computation.

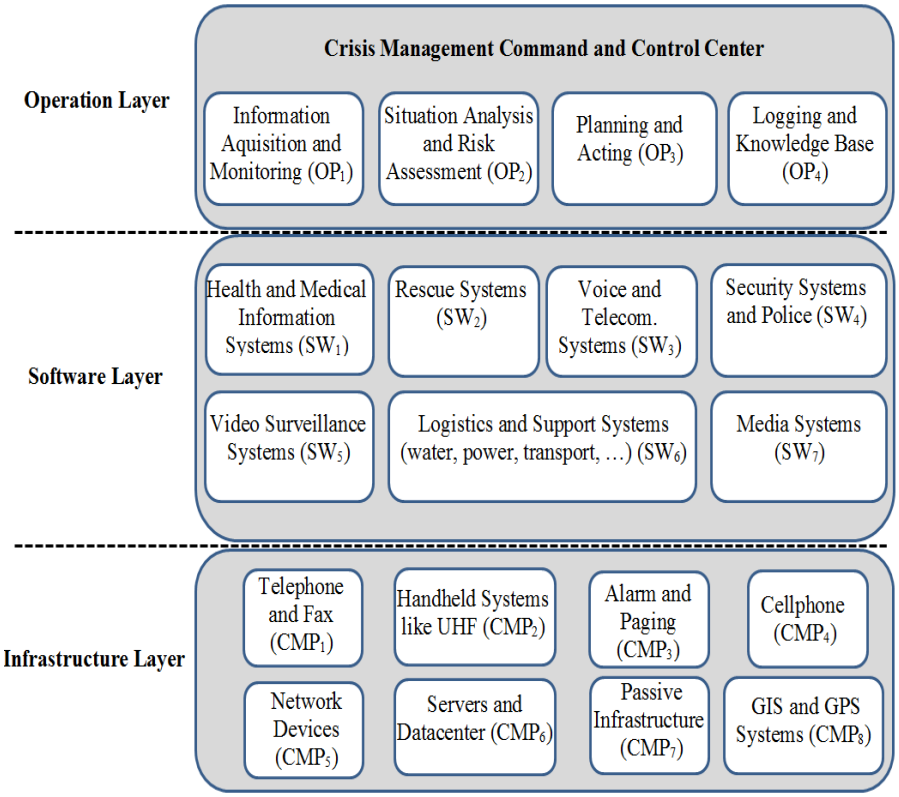

Fig. 9. Model of the Given Crisis Management System.

TABLE II. COMPUTATION OF CRITICAL OPERATION LAYER SURVIVABILITY

Process	πH	πS	πF
OP1	9.631358E-01	3.874875E-02	9.092752E-03
OP2	9.659744E-01	6.218598E-02	2.288903E-02

We have done a noticeable amount of calculations about all layers and components of the system for calculating survivability, but due to page number limitations we are not able to present all of them. Each interested reader can achieve them by email. Only for representing the achieved results at final stage, we present the calculated survivability of two critical processes OP1 and OP2 in Table II.

As seen from Table II, the essential disorder probabilities of OP1 and OP2 processes that are critical for the crisis management are 9.092752E-03 and 2.288903E-02 respectively, that are called $\pi_F^{OP_1}$ and $\pi_F^{OP_2}$.

Now, we are ready for computing the survivability probabilities of the total crisis management system overally. For this purpose, the total health probability of the crisis management structure is symbolized $\pi_H^{Overall}$ and probability of operation continuation in the failure conditions of non-critical process as $\pi_S^{Overall}$ and the probability of failure of total crisis management system as $\pi_F^{Overall}$. It is supposed that the computation processes performed for OP1 and OP2 are similarly performed for OP3 and OP4.

Since we want to find acceptable states operationally for processes, $\pi_S^{Overall}$ is also considered among those states. Thus, instead of direct computation of $\pi_S^{Overall}$ and $\pi_H^{Overall}$ values, the following quantity that is derived from the basic relation $\pi_H^{Overall}+\pi_S^{Overall}+\pi_F^{Overall}=1$ are introduced as the total system acceptable function probability.

$$\pi_F^{Overall} = 1-\left(1-\pi_F^{OP_1}\right)\left(1-\pi_F^{OP_2}\right) = 3.177366\text{E-}02$$
$$\pi_S^{Overall} = \left(1-\pi_F^{OP_1}\right)\left(1-\pi_F^{OP_2}\right)\left(1-\left(\pi_H^{OP_3}+\pi_S^{OP_3}\right)\left(\pi_H^{OP_4}+\pi_S^{OP_4}\right)\right)$$
$$\pi_H^{Overall} = \left(1-\pi_F^{OP_1}\right)\left(1-\pi_F^{OP_2}\right)\left(\pi_H^{OP_3}+\pi_S^{OP_3}\right)\left(\pi_H^{OP_4}+\pi_S^{OP_4}\right)$$

system acceptable operation probability is equal to:

$$\pi_H^{Overall}+\pi_S^{Overall}=1-\pi_F^{Overall}=96.822634E-02$$

VII. CONCLUSION

This paper provides a general multilayer structure for systems survivability computation that is extendable to all common organization systems and operations. We designed a three layer model that connects the operational processes to application systems and application systems to the infrastructure layer. Then, the dependencies among these layers have been studied vertically (interlayer) and horizontally (intralayer). On the other hand, a new conceptual model was provided based on the Markov model characteristics for

survivability. Then, this model was used in a three stage structure for achieving our goal. In the first stage, the survivability of an infrastructure layer component was computed regardless of any dependencies and independently. Then, the horizontal dependency between the infrastructure layer components was entered in the computations and the survivability was computed applying those dependencies. In the final stage, the survivability computation model was provided taking into account the vertical dependencies for upper layers. The survivability of application systems and finally system operational processes have been computed including these dependencies. Finally, applying the total model in an important and frequently used problem such as the crisis management system, we could compute the real value of survivability for such system in the level of the crisis management critical and major processes and presented the abilities of our model. Utilizing this model will result in enabling the managers and planners to detect system weak points that make the highest loss in the survivability and efficiently protecting and retaining the system critical functions in crisis condition.

REFERENCES

[1] Knight JC, Sullivan KJ. On the definition of survivability. University of Virginia, Department of Computer Science, Technical Report CS-TR-33-00. 2000.

[2] Deutsch MS, Willis RR. Software quality engineering: a total technical and management approach. Prentice-Hall Inc., 1988.

[3] Ellison RJ, Fisher DA, Linger RC, Lipson HF, Longstaff T. Survivable network systems: An emerging discipline. Carnegie-mellon Univ Pittsburgh PA Software Engineering Institute, 1997.

[4] Ellison RJ, Fisher DA, Linger RC, Lipson HF, Longstaff TA, Mead NR. An approach to survivable systems. In: NATO IST Symposium on Protecting Information Systems in the 21st Century, 1999, 754-759.

[5] Byon, Imju. Survivability of the US electric power industry. PhD Dissertation. Carnegie Mellon University, 2000.

[6] Caldera, Jose. Survivability requirements for the US health care industry. PhD Dissertation. Carnegie Mellon University, 2000.

[7] Ellison RJ, Fisher DA, Linger RC, Lipson HF, Longstaff TA, Mead NR. Survivability: Protecting your critical systems. IEEE Internet Computing, 1999, 3(6):55-63.

[8] Knight J, Heimbigner D, Wolf AL, Carzaniga A, Hill J, Devanbu P, Gertz M. The Willow architecture: comprehensive survivability for large-scale distributed applications. COLORADO UNIV AT BOULDER DEPT OF COMPUTER SCIENCE, 2001.

[9] Neumann PG. Practical Architectures for Survivable Systems and Networks: Phase-One Final Report. SRI INTERNATIONAL MENLO PARK CA COMPUTER SCIENCE LAB, 1999.

[10] Mead NR, Ellison RJ, Linger RC, Longstaff T, McHugh J. Survivable network analysis method. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, 2000.

[11] Ravindran B. Engineering dynamic real-time distributed systems: Architecture, system description language, and middleware. IEEE Transactions on Software Engineering, 2002, 8(1):30-57.

[12] Avizienis A, Laprie JC, Randell B. Fundamental concepts of dependability. University of Newcastle upon Tyne, Computing Science, 2001.

[13] Keromytis AD, Parekh J, Gross PN, Kaiser G, Misra V, Nieh J, Rubenstein D, Stolfo S. A holistic approach to service survivability. In: Proceedings of the 2003 ACM workshop on Survivable and self-regenerative systems in association with 10th ACM Conference on Computer and Communications Security, 2003, 11-22.

[14] Sames D, et al. Developing a heterogeneous intrusion tolerant CORBA system. In: Proceedings of IEEE International Conference on Dependable Systems and Networks, 2002, 239-248.

[15] Federal Standard l037C, U.S. Department of Commerce, National Telecommunications and Information Administration, Institute for Telecommunications Services, 1996.

[16] Heegaard PE, Trivedi KS. Network survivability modeling. Computer Networks, 2009, 53(8):1215-1234.

[17] Chong J, et al. Survivability architecture of a mission critical system: The DPASA example. In: 21st IEEE Annual Computer Security Applications Conference, 2005.

[18] Shrobe H, et al. AWDRAT: a cognitive middleware system for information survivability. AI Magazine. 2007, 28(3):73-80.

[19] M. Keshtgari, A. H. Jahangir,A General Framework For Network Survivability Performance Evaluation, PhD Dissertation, Sharif University of Technology, 2005.

[20] Richards MG, et al. Design for survivability: concept generation and evaluation in dynamic tradespace exploration. In: 2nd International Symposium on Engineering Systems, 2009.

[21] Zuo Y. A framework of survivability requirement specification for critical information systems. In: 43rd Hawaii International Conference on System Sciences (HICSS), 2010, 1-10.

[22] Zuo Y, Lande S. A logical framework of proof-carrying survivability. In: IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 472-481.

[23] Zuo Y. Incorporating Constraints to Software System Survivability Specification and Proof. In: Sixth International Symposium on Theoretical Aspects of Software Engineering (TASE), 2012, 67-74.

[24] Knight JC, Strunk EA. Achieving critical system survivability through software architectures. In: Architecting Dependable Systems II, 2004, 51-78.

[25] Avizienis A, Laprie JC, Randell B, Landwehr C. Basic concepts and taxonomy of dependable and secure computing. IEEE transactions on dependable and secure computing, 2004, 1(1):11-33.

[26] Trivedi KS, Xia R. Quantification of system survivability. Telecommunication Systems, 2015, 60(4):451-470.

[27] T1A1.2 Working Group on Network Survivability Performance, Technical Report No. 68, ATIS-T1.TR.68, 2001.

[28] Westmark VR. A definition for information system survivability. In: Proceedings of the 37th IEEE Annual Hawaii International Conference on System Sciences, 2004.

[29] Wang XG. Network System Survivability: A Survey. In: IEEE International Conference on Industrial and Information Systems, 2009, 232-235.

[30] Sterbenz JP, et al. Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. Computer Networks, 2010, 54(8):1245-1265.

[31] Rodríguez RJ, Chang X, Li X, Trivedi KS. Survivability Analysis of a Computer System Under an Advanced Persistent Threat Attack. In: International Workshop on Graphical Models for Security, 2016, 134-149.

[32] Necula GC. Proof-carrying code. In: Proceedings of the 24th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, 1997,106-119.