# Survivability Analysis of A Cluster System with 4th Generation Security Mechanism: Regeneration

Khin Mi Mi Aung[1], Kiejin Park[2], and Jong Sou Park[1]
(Corresponding author: Khin Mi Mi Aung)

Network Security Lab., Computer Engineering Dept., Hankuk Aviation University[1]
412-791, Seoul, Republic of Korea (Email: {maung, jspark}@hau.ac.kr)
Division of Industrial and Information System Engineering, Ajou University[2]
443-749, Suwon, Republic of Korea (Email: kiejin@ajou.ac.kr)

(Received Aug. 18, 2005; revised and accepted Sept 29, 2005)

## Abstract

Cluster systems have been gradually more popular and are being broadly used in a variety of applications. On the other hand, many of those systems are not tolerant to system failures and moreover we cannot prevent all faults and attacks. In this paper, we analyze a cluster system's ability to maintain the critical services even in face of faulty and intrusions with 4th Generation Security Mechanism: Regeneration. The experiments have been demonstrated that the proposed mechanism can be used to analyze and proactively manage the effects of cluster network faults and attacks, and restore accordingly. The survivability level is greatly enhanced by the addition of functionality to targeted systems while maintaining the critical service and avoiding large incremental costs.

*Keywords: Cluster system, regeneration, survivability*

## 1 Introduction

The 1st generation security mechanism: protection mechanisms [7] are being degenerated, as the Internet has become an emerging technology with more complicated and interconnected all around the world. Then the 2nd generation: detection mechanisms were arising to be able to detect the successful intrusions. Although, if some attacks, especially for new attacks, will succeed without able to detect them. So the 3rd generation: tolerance provides to operate in face of such successful intrusions. But even in an intrusion tolerant system (ITS), the resources will be fatigued if the intrusion is long lasting because of compromising iteratively or incrementally. In due course, the system will not provide even the minimum critical functionality. Thus, the new paradigm of survivable information systems is the 4th generation security mechanism: regeneration mechanisms [7] and we propose a model of restore system for this.

This research mainly addresses the vulnerabilities of denial-of-service attacks (DoS) and proposes a cluster restore model with a software rejuvenation as a regeneration mechanism. Software rejuvenation is a proactive fault management technique aimed at cleaning up the internal system state to prevent the occurrence of more severe future crash failures. It involves occasionally terminating an application or a system, cleaning its internal state and restarting it. IBM Software Rejuvenation is a tool to help increase server availability by proactively addressing software and operation system aging [4]. The effect of aging is captured as crush/hang failures [2]. Survivability is the ability of a system to continue operating in the presence of accidental failures or malicious attacks [3].

## 2 Related Works

In particular, 4th generation security mechanism is not a generally applicable clear definition of it. Thus, the present study looked for a restore system to introduce on it with cold standby cluster system, which is composed of a primary server and a secondary server. Cold standby is a method of redundancy in which the secondary (i.e., backup) system is only called upon when the primary system fails. The system on cold standby receives scheduled data backups, but less frequently than a warm standby. Cold standby systems are used for non-critical applications or in cases where data is changed infrequently.

Currently, DARPA's organically assured and survivable information system (OASIS) is emphasizing and sponsoring several projects that address a number of related issues in 4th generation security mechanism involving diagnosis, learning, reconfiguration, software rejuvenation, natural immunity, reflection, self aware and reconfiguration [7, 13]. Jha et. al. have studied reliability, latency and cost benefit model [5]. They have analyzed survivability of network systems, which are service dependent; therefore a system architect should focus on the design of the system by analyzing only the service required of that system. They use a Constrained Markov Decision Process (CMDP) to form the basis of the survivability analysis, which is composed of reliability, latency, and cost-benefit.

Moitra et. al. simulated the model for managing survivability of network information systems [10]. They propose a model to assess the survivability of a network system. Different parameters affect survivability such as the frequency and impact of attacks on a network system. The authors finally conclude that there is no "absolute survivability" and sites other measures of survivability such as relative survivability, worst-case survivability, and survivability with expected compromise. Simulations to analyze survivability used the Poisson model.

Today, denial-of-service (DoS) attack cause significant disruptions to the Internet [8, 12, 14] and in this paper, we have mainly addressed DoS attacks. An attacker carries out a DoS attack by making a resource out of action. The nature of attacks is very dynamic because attackers have the specific intention to attack and well prepare their steps in advance. So far no respond technique able to cope with all types of attacks has been found. In most attacks, attackers overwhelm the target system with a continuous flood of traffic designed to consume all system resources, such as CPU cycles, memory, network bandwidth, and packet buffers.

In this paper, we present a cluster restore model using cold standby cluster with a software rejuvenation methodology. We illustrate our abstract model for a restore system from the malicious attacks. We evaluate the survivability of systems and services as well as the impact of any proposed changes on the overall survivability of systems. The paper has also presented aspects of the operational requirements for information systems such as the ability to operate through attacks and graceful degradation. We had analyzed the attack datasets and injected the attacks events into a system and then apply the restored the system to a healthy state within a set time following the predictive alerts [14]. And we introduces a method that determines survivability by applying a stricter concept of availability through setting a deadline for the mean sojourn time ratio under attack on the state transition diagram of a restore system.

Other related work includes the ITS Projects and reconfiguration such as a scalable intrusion tolerant architecture for distributed services SITAR [1], malicious and accidental fault tolerance for Internet applications (MAFTIA) [9] and secure overlay services SOS [6]. SOS addresses a proactive approach for the problems of existing IP infrastructure from DoS attacks and MAFTIA had developed an open architecture for transactional operations on the Internet and applied approaches developed in the realm of fault tolerance.

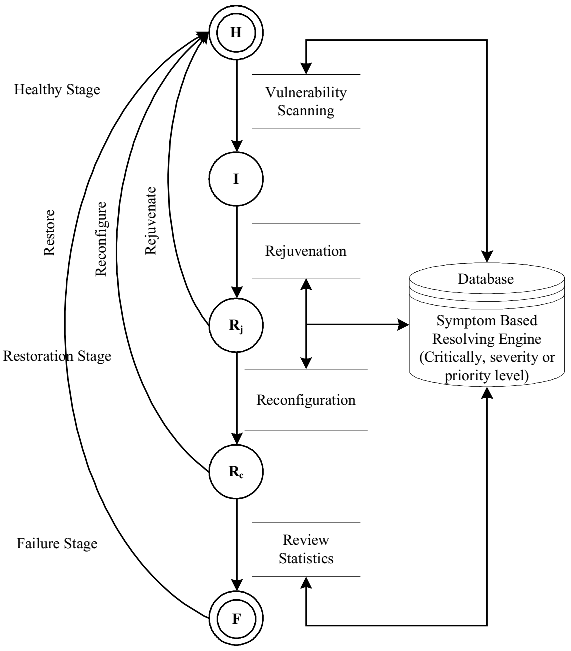

Figure 1: Proposed model

The organization of the paper is as follows. In Section 1 and 2, we define the problem and address related research. Section 3 presents a proposed model which can be used to analyze and proactively manage the effects of cluster network faults and attacks, and recover accordingly and in the following section, the model is analyzed and experimental results are given to validate the model solution. We had analyzed the attack datasets and injected the attacks events into a system, and learned the prior knowledge. Finally, we conclude that software rejuvenation and reconfiguration are viable methods and present further research issues.

# 3 Proposed Model

Significant features of various system resources may differ between specific attacks. And the response and restore methods would differ as well. In this work, the system has divided into three stages; healthy stage, restoration stage and failure stage (refer to Figure 1). The model consists of five states: healthy state ($H$), infected state ($I$), rejuvenation state ($R_j$), reconfiguration state ($R_c$) and failure state ($F$). The healthy state represents the functioning and service providing phases. In the healthy stages, the systems aware to resist by various policies and offer proactive managements which are periodic diagnostics and automatic error log analysis, scheduled tasks (checking routine) based on experiences to assess the approximate frequency of unplanned outages due to resources exhaustion, monitoring server subsystems and software processes to ascertain common trends accompanying regular failures, error logging and alerts (error logging controls).

### 3.1 Vulnerability Scanning

Vulnerability scanning service is running by timely manner that provides a low cost alternative for testing environments and to search for known security holes, flaws, and exploits on the system. Basically, vulnerability scanning has a database of known security flaws. They operate by testing for each of those flaws. That database is updated periodically as new exploits are discovered and stands as available for service stage. To scan the vulnerability and watch the symptom, we carry out the monitoring selected system events, audit event selection per process and per object, security policy events, accountability events and general system administration events. We trigger security violations as the strategies of repelling and detecting attacks, evaluation and limiting damages, restoring the compromised information or functionalities, maintaining or restoring essential services within mission time constraints, restoring full services. Based on those constraints and the knowledge gained from intrusions, the system survivability may be improved. As an experimental setup, we use Linux monitoring, logging and scheduling tools.

The immediate value of the actions has been resolving in the database engine. In that engine a set of rules, a state database, a hierarchical database of objects, a rule processor and an interpreter are involved and those factors perform the analysis, reasoning, and decision-making. And the system has been documented the types of threats or events that indicate possible signs of intrusion and how to respond if they are detected. Types of actions may include attempts (either failed or successful) to gain unauthorized access to a system or its data, unintended and unauthorized disclosure of information, unwanted disruption or DoS, the unauthorized used of a system to process, store, or transmit data etc. The system is periodically compared this information with the current state. And it is determined if anything has been altered in an unexpected way. Based on priority and sequence of actions, it will consider safer features temporarily by suspending administrator privileges or disabling write operations for a while. Focusing on symptoms, we can expect since many kinds of attacks produce similar symptoms. The more capacity we cope with a finite number of symptoms, the higher ability we can achieve less severe effects of many attacks.

### 3.2 Rejuvenation

At the rejuvenation performing state, we need to be able to weigh the risk of policy with further damage against the policy of shutting the system in an emergency stage. In this case, the tools not only detect an attacker's presence but also support to get the information containments. The events are preconditions and are related to compromised system states. Susceptible to attack is an action or series of actions that lead to a compromise. Multiple defense mechanisms are the set of actions that may be taken to correct vulnerable conditions existing on the system or to move the system from a more compromised state to a less compromised state. To this end, software rejuvenation methodologies are reviewed and synthesized by the policies. The main strategies are occasionally stopping the executing software, cleaning the internal state and restarting by means of effectiveness of proactive managements, degrading mechanism, service stop, service restart, reboot and halt.

We start this approach with simple structured software. Based on the updated collecting databases, the temporal rejuvenation is performed to solve an error forecasting and accumulation of errors for the software reliability estimation and prediction. All intelligent organisms fuse knowledge to create situational awareness. Humans continually create and redefine systems and situational knowledge. Process replication and unpredictable adaptation will be added to tolerate the faults. It will be well suited for tightly scheduled projects by which rejuvenate a system via periodical duty switching between system components, slowing down a system's aging process and enhancing mission reliability.

### 3.3 Restoration

At the restoration stage, they may be decomposed into three types according to their specific attacks such as

- Performing rejuvenation only,
- Performing reconfiguration only and
- Performing both rejuvenation and reconfiguration

When a node is attacked, the impacts within the node can be categorized into the three groups in general, attacks exhaust resources such as CPU power, memory space, and I/O bandwidth, attacks create unauthorized files or modify existing files, and attacks executed the systems and commands in the node [11]. For example, if an attacker carries out attack by overloading processes, causing resources to become unavailable, we will perform a rejuvenation process by gracefully terminating processes causing the resource overload and immediately restarting them in a clean state. But for the other kinds of attacks, we have to reconfigure the system according per their impact. In this case we have considered the reconfiguration state with various reconfiguration mechanisms, such as

- Patching (operating system patch, application patch),
- Version control (operating system version, application version),
- Anti virus (vaccine),
- Access control (IP blocking, port blocking, session drop, contents filtering), and
- Traffic control (bandwidth limit)

As an example, performing rejuvenation only could deter the attacks, which cause the process degradation such as spawn multiple processes, fork bombs, CPU overload etc. For the cases of process shutdown and system shutdown attacks, the attackers intend to halt a process or all processing on a system. Normally it happens by exploiting a software bug that causes the system to halt could cause system shutdown. In this case, just as with software bugs that are used to penetrate, so until the software bug is reconfigured, all systems of a certain type would be vulnerable. An example of attacks called mail bombardment or mail spam, the attacker accomplishes this attack by flooding the user with huge message or with very big attachments. Depending on how the system is configured, this could be counteracted by performing both reconfiguration and rejuvenation processes.

To perform the various reconfiguration mechanisms, we have implemented the event manager, which contains the various strategies with respect to the various impact levels of the specific infected cases. Each type of event has its own routine, to be run when the attack takes place [14].

# 4 Survivability Analysis

## 4.1 Steady-state Analysis of Non-cluster System Through Conventional Approach

According to the state transition diagram of Figure 2, we denote as,

$\lambda_{h,i}$ = Infected rate from the healthy state
$\lambda_{i,j}$ = Rejuvenation rate from the infected state
$\mu_{j,h}$ = Rejuvenation service rate to the healthy state
$\lambda_{i,c}$ = Reconfiguration rate from the infected state
$\lambda_{c,j}$ = Reconfiguration service rate to the rejuvenation state
$\lambda_{c,f}$ = Failure rate from the reconfiguration state
$\mu_{c,h}$ = Reconfiguration service rate to the healthy state
$\lambda_{i,f}$ = Failure rate from the infected state
$\mu_{f,h}$ = Service rate from the failure state

According to the assumption for Markov process, the sojourn time of all states in Figure 2 follows exponential distribution. And let the steady-state probabilities of the state of the system be

$\pi_h$ = The probability that the system is in Healthy State
$\pi_i$ = The probability that the system is in Infected State
$\pi_r$ = The probability that the system is in Rejuvenation State
$\pi_c$ = The probability that the system is in Reconfiguratio State
$\pi_f$ = The probability that the system is in Failure State.

Using principle of the rate at which the process enters each state with the rate at which the process leaves can derive the balance equations for the system (refer to Figure 2).

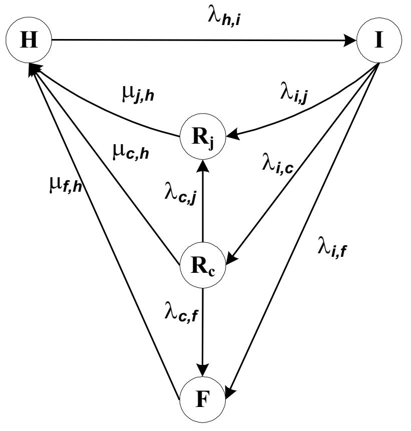

Figure 2: State transition diagram of non-cluster system

$$\begin{aligned}
\lambda_{h,i}\pi_h &= \mu_{j,h}\pi_j + \mu_{c,h}\pi_c + \mu_{f,h}\pi_f \\
\pi_i &= E\pi_h \\
\pi_c &= \frac{\lambda_{i,c}}{F}E\pi_h \\
\pi_j &= (\lambda_{i,j} + \frac{\lambda_{c,j}\lambda_{i,c}}{F})E\frac{1}{\mu_{j,h}}\pi_h \\
\pi_f &= (\lambda_{I,f} + \frac{\lambda_{c,f}\lambda_{i,c}}{F})E\frac{1}{\mu_{j,h}}\pi_h.
\end{aligned}$$

By solving above equations in terms of $\pi_h$ and the condition $\pi_h + \pi_i + \pi_r + \pi_c + \pi_f = 1$, we get

$$\pi_h = \begin{pmatrix} 1/1 + E + \frac{\lambda_{i,c}}{F}E + 1/(\lambda_{i,j} + \frac{\lambda_{c,j}\lambda_{i,c}}{F})E\frac{1}{\mu_{j,h}} \\ +1/(\lambda_{i,f} + \frac{\lambda_{c,f}\lambda_{i,c}}{F})E\frac{1}{\mu_{j,h}} \end{pmatrix}$$

where $E = \frac{\lambda_{h,i}}{\lambda_{i,j}+\lambda_{i,c}+\lambda i,f}$ and $F = \lambda_{c,f} + \lambda c, j + \mu_{c,h}$.

The availability for the steady-state analysis on a single node through Markov Process can be expressed as:

$$A = 1 - (\pi_f + \pi_j + \pi_c).$$

## 4.2 Steady-state Analysis with Two Nodes Through Semi-Markov Process

Semi-Markov models contain a Markov chain, which describes the stochastic transitions from state to state, and transition or 'sojourn' times, which describe the duration that the process takes to transition from state to state. We address the survivability model with semi-Markov process. We consider a cold standby cluster with two nodes through Semi-Markov process. One node is as an active (primary) and other as a standby (secondary) unit. The failure rate of the primary node and secondary node are different, and also the effect of failure of the primary node is different from that of secondary node. Initially the system is in state (1,1). When the primary is infected by active attacks, the system enters state $(I, 1)$. In the infected state, the system has to figure out whether rejuvenate or reconfigure to recover or limit the damage that may happen by an attack. If the primary node has to reconfigure, the system enters state $(R_c, 1)$ otherwise enters state $(R_i, 1)$. If both strategies fail then the primary system enters the fail state. When the primary node fails a protection switch successfully restores service by switching in the secondary unit, and the system enters state (0,1). If the node failure occurs when the system is in one of the states : $(0, I)$ or $(0, R_c)$, the system fails and enters state $(F, F)$.

To calculate the steady-state availability of the proposed model, the stochastic process of Equation 1 was defined. Through SMP (Semi-Markov Process) analysis applying M/G/1, whose service time is general distribution; we calculated the steady-state probability in each state.

$$X(t) : t > 0 \tag{1}$$

$$XS = \left\{ \begin{array}{l} (1,1),(I,1),(R_j,1),(R_c,1),(F,1), \\ (0,1),(0,I),(0,R_j),(0,R_c),(F,F) \end{array} \right\}$$

Semi-Markov models contain a Markov chain, which describes the stochastic transitions from state to state, and transition or 'sojourn' times, which describe the duration that the process takes to transition from state to state. We address the survivability model with semi-Markov process. We consider a cold standby cluster with two nodes through Semi-Markov process. One node is as an active (primary) and other as a standby (secondary) unit. The failure rate of the primary node and secondary node are different, and also the effect of failure of the primary node is different from that of secondary node. Initially the system is in state (1,1). When the primary is infected by active attacks, the system enters state $(I, 1)$. In the infected state, the system has to figure out whether rejuvenate or reconfigure to recover or limit the damage that may happen by an attack. If the primary node has to reconfigure, the system enters state $(R_c, 1)$ As all the states shown in Figure 3 are attainable to each other, they are irreducible. Additionally, as they do not have a cycle and can return to a certain state, they satisfy the ergodicity (Aperiodic, Recurrent, and Nonnull) characteristics. Therefore, there is a probability in the steady-state of SMP for each state and each corresponding SMP can be induced by embedded DTMC (Discrete-time Markov Chain) using transition probability in each state.

If we define the mean sojourn times in each state of SMP as $h_i$'s and define DTMC steady-state probability as $d_i$'s, the steady-state probability in each state of SMP $(\pi_i)$ can be calculated by Equation 2 [16].

$$\pi_i = \frac{d_i h_i}{\sum_j d_j h_j}, i, j \in X_S \tag{2}$$

Whereas, steady-state probability of DTMC di's will have the following relationship as shown in Equation 3 and Equation 4.

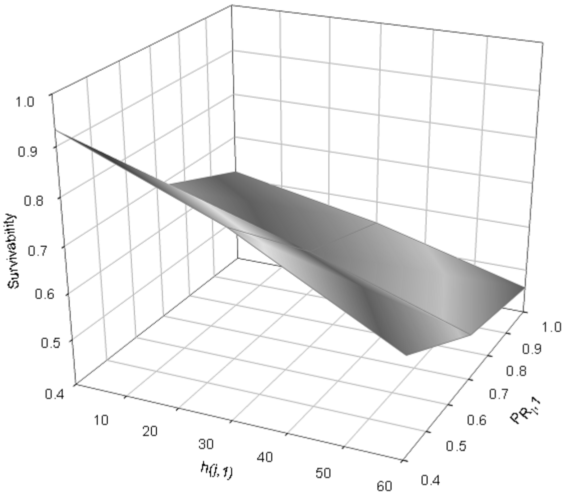

Figure 3: Analysis of survivability according to the transition probability and the sojourn time

$$\begin{aligned} \vec{d} &= \vec{d} \cdot P & (3) \\ \sum_i d_i &= 1 \quad i \in X_S & (4) \end{aligned}$$

where,
$\vec{d} = \{d_{(1,1)} d_{(I,1)}\ d_{(R_j,1)}\ d_{(R_c,1)}\ d_{(F,1)}\ d_{(0,1)}\ d_{(0,I)}\ d_{(0,R_j)}\ d_{(0,R_c)}\ d_{(F,F)}\}$
and $P$ is the transition probability matrix of DTMC expressed by the transition probability in each state of $X_S$ in Figure 3 $(P_{(i,j)})$.

The system availability in the steady-state is defined as Equation 5, which is the same as the exclusion of the probability of being in $(F, 1)$ and $(F, F)$ in each state of $X_S$ in the state transition diagram.

$$A = 1 - (\pi_{(F,1)} + \pi_{(F,F)}). \tag{5}$$

When $D^*$ indicates the deadline of the mean sojourn time ratio $(d_i h_i)$ for determining whether the system survives in case either of the primary server or the secondary server is in the attack state $((R_j, 1), (R_c, 1), (0, R_j), (0, R_c)), Y_i(i = (R_j, 1), (R_c, 1), (0, R_j), (0, R_c))$, the indicator variable for determining the survivability of the system in the corresponding state is decided as follows.

$$d_i h_i \le D^* : Y_i = 0, d_i h_i > D^* : Y_i = 1.$$

Using the indicator variable decided above, the survivability measure of a restore system is defined as Equation 6.

$$S = A - \left[ \begin{array}{l} Y_{(R_j,1)}\pi_{(R_j,1)} + Y_{(R_c,1)}\pi_{(R_c,1)} \\ +Y_{(0,R_j)}\pi_{(0,R_j)} + Y_{(0,R_c)}\pi_{(0,Rc)}] \end{array} \right] \tag{6}$$

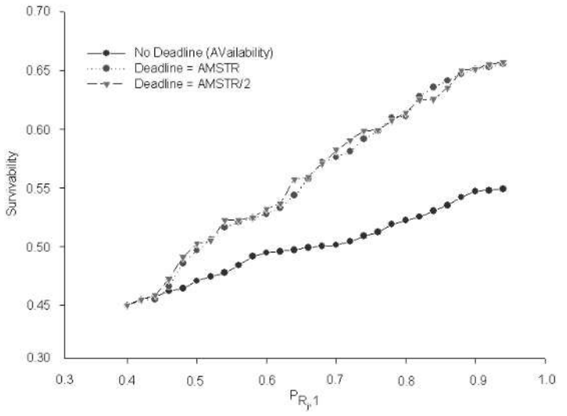

Figure 4: Analysis of survivability according to early coping ability and the change of deadline

Table 1: Simulation parameters [14]

| | |
|---|---|
| Mean Sojourn Time | $h_{(1,1)} = 50$, $h_{(I,1)} = 30$, $h_{(Rj,1)} = 25$ |
| | $h_{(Rc,1)} = 50$, $h_{(F,1)} = 20$, $h_{(0,1)} = 50$ |
| | $h_{(0,I)} = 30$, $h_{(0,Rj)} = 25$, $h_{(0,Rc)} = 20$, $h_{(F,F)} = 50$ |
| Transition Probability | $0 < P_{(Rj,1)}$, $P_{(Rc,1)}$, $P_{(0,1)}$, $P_{(0,Rc)}$, $P_{(0,Rj)} < 1$ |
| Deadline | AMSTR/3< D$*$ <AMSTR |

# 5 Numerical Results

In this section, we illustrate the evaluation of model with numerical results. To evaluate the conventional approach with non-cluster system, we have to set parameters for the service rates and the failure rates. To evaluate our SMP model, we need to set parameters for the transition probability and the mean sojourn time in each state. The accurate model parameter values are unknown and we assume the relative differences for various model parameters. The use of various model parameters makes it relevant to get the sensitivity level of different measures to variations in the model parameter values.

Simulation parameters for SMP were made based on the values shown in Table 1. Because the mean sojourn time in each state has a general distribution, values are meaningful only as relative differences. In order to analyze the effects of the transition probability at each of the five junctions appearing in the state transition diagram on the availability of a restore, the probability was set at a value between 0.0 $\sim$ 1.0. On the other hand, deadline ($D^*$) for analyzing survivability was set based on $(d_{(R_j,1)}d_{(R_c,1)} + d_{(0,R_j)}d_{(0,R_c)})/2$, which is the average of mean sojourn time ratio (AMSTR) in the attack state.

After computing steady-state probabilities of all states by Equation 2, we compute their availabilities for each state. We obtain the following graphs by using the above values of input parameters for the steady-state probabilities of the semi Markov process.

Figure 3 analyzes the survivability with the transition probability and the sojourn time. It shows that understanding the impact of sojourn time is also critical and described how to enhance survivability and regenerate the restore the system.

Figure 4 shows the changes the level of system survivability according to the rejuvenation probability when the servers are attacked. The difference between no deadline and deadline is insignificant in the section where transition probability is less than 0.4. It means that the affect of performing rejuvenation is not sensitive up to that section but after this section the coping ability of restore system is decreasing and by using this graph we can analyze the effectiveness of applying our approach. In case the sojourn time ratio does not have a deadline, the survivability has the same value as that of availability, and in case a deadline is applied strictly, the survivability measure tends to be lowered. The cold-standby restore system suggested that survivability is maximized when the primary-secondary servers detect abnormal behaviors as early as possible when each of them is exposed to attacks and vulnerable situations. On the other hand, the decrease of survivability grows slower as $P(R_c, 1)$ and $P(0, R_c)$ approach 0 and, for $P(R_j, 1)$ and $P(0, R_j)P(0, R_j)$ survivability shows a rapid decrease in the early stage but the decrease becomes slower as $P(R_j, 1)$ and $P(0, R_j)$ approach 1. This is because, even if the system is easily exposed to a vulnerable situation in the early stage, survivability can be guaranteed through recovery to the initial state by transition to the rejuvenation if the diagnosis function of a restore system can detect significantly lowered performance in the state of $P(R_c, 1)$ and $P(0, R_c)$ in which the primary-secondary servers are exposed to attacks.

# 6 Conclusion

In this paper, we analyze the cluster system's survivability using the 4th Generation Security Mechanism: Regeneration. We also analyzed the availability and survivability considering a deadline of mean sojourn time ratio. Based on these results, we have learnt that considering a deadline of mean sojourn time ratio is an obvious prerequisite to define the availability of general systems. Understanding the impact of sojourn time is also critical and described how to enhance survivability and regenerate the restore the system.

We have demonstrated the model can be used to analyze and proactively manage the effects of cluster network faults and attacks, and restore accordingly. The result shows that the system operates through intrusions and provides continued the critical functions, and gracefully degrades non-critical system functionality in the face of intrusions. According to the system operating parameters, we have modelled and analyzed steady-state probability and survivability level of cluster systems under DoS attacks by adopting a software rejuvenation technique. We have validated the closed-form solutions of the mathematical model with experiments based on the above parameters. We have also found that software rejuvenation can be used as a preventive fault-tolerant technique and it improves the survivability of cluster system. The results in this paper represent the first step in the development of a model of 4$^{\text{th}}$ generation security mechanism. As an ongoing work, we are performing our model with the real sojourn times of specific attacks in order to generalize it with various attacks. We are analyzing a variety of probability distributions in the real attack data, which is, described the attackers' transitions and the sojourn time that they spend in every state. The integration of response time and throughput with downtime cost will provide a more accurate evaluation measure.

## References

[1] *A Scalable Intrusion-Tolerant Architecture for Distributed Services*, http://www.ee.duke.edu/∼kst /sitar.html.

[2] V. Castelli, R. E. Harper, S. W. Hunter, P. Heidelberger, K. Vaidyanathan, and W. Zeggert, "Proactive management of software aging," *IBM Journal of Research & Development*, vol. 45, no. 2, pp. 311-332, 2001.

[3] R. Ellison, D. Fisher, R. Linger, H. Lipson, T. Longstaff, and N. Mead, *Survivable Network Systems: An Emerging Discipline*, Technical Report CMU/SEI-97-153, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 1997.

[4] Y. Huang, C. Kintala, N. Kolettis and N. Fulton, "Software rejuvenation: analysis, module and applications," in *The 25th International Symposium on Fault-Tolerant Computing (FTCS-25)*, pp.381-390, Pasadena, CA, USA, 1995.

[5] S. Jha and J. Wing, "Survivability analysis of networked systems," in *The 23rd International Conference on Software Engineering (ICSE' 01)*, pp. 872-874, Washington, DC, USA, 2001.

[6] D. Keromytis, V. Misra, and D. Rubenstein. "SOS: secure overlay services," in *ACM SIGCOMM 2002*, pp. 61-72, Pittsburgh, PA, USA, 2002.

[7] J. Lala, *Introduction to the Proceedings of Foundations of Intrusions Tolerant Systems (OASIS 03)*, pp. x–xix, Los Alamitos, CA, USA, 2003.

[8] M. Long, C. Wu and J. Hung, "Denial of service attacks on Network-Based control systems: impact and mitigation," *IEEE Transactions on Industrial Informatics*, vol. 1, no. 1, pp. 85-96, 2005.

[9] *Malicious and Accidental-Fault Tolerance for Internet Applications*, http://www.maftia.org.

[10] S. D. Moitra and S. D. Konda, *A Simulation Model for Managing Survivability of Networked Information Systems*, Technical Report, CMU/SEI-2000-TR-020, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 2002.

[11] D. Moore, R. Ellison and R. Linger, *Attack Modelling for Information Security and Survivability*, Technical Note CMU/SEI-2001-TN-001, 2001.

[12] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, "Inside the slammer worm," *IEEE Security & Privacy Magazine*, vol. 1, no. 4, pp. 33-39, 2003.

[13] *Organically Assured and Survivable Information System*, http://www.tolerantsystems.org.

[14] J. S. Park and K. M. M. Aung, "Transient time analysis of network security survivability using DEVS," LNCS 3397, pp. 607-616, Springer, 2005.

[15] S. Staniford, V. Paxson, and N. Weaver, "How to own the Internet in your spare time," in *The 11th USENIX Security Symposium (SEC02)*, pp. 149-167, San Fransico, CA, USA, 2002.

[16] K. Trivedi, *Probability and Statistics with Reliability Queueing and Computer Science Applications*, John Wiley & Sons, Interscience, 2003.

**Khin Mi Mi Aung** is a PhD Candidate and a Research Assistant in Network Security Lab., Computer Engineering Department, at Hankuk Aviation University. In 1999, she received an MSc from University of Computer Studies, Yangon, Myanmar in Information Science. Her research interests include Ubiquitous Sensor Network Security, Security Engineering and Network Security.

**Kiejin Park** was born in Seoul, Korea. He received the B.S. and M.S. degrees in industrial engineering from Hanyang University and POSTECH, Korea, in 1989 and 1991, respectively, and Ph.D. degree in Department of Computer Engineering, Graduate School of Ajou University in Korea in 2001. He is currently an assistant professor in Division of Industrial & Information Systems Engineering, Ajou University. From 1991 to 1996, He worked in the Computer and Communication Research Center of Samsung Advanced Institute of Technology, Korea, as an assistant researcher. From 1996 to 1997, he was with the Software Research and Development Center of Samsung Electronics Co., Korea, as a senior researcher. From 2001 to 2002, He worked in Network Equipment Test Center of Electronics and Telecommunications Research Institute (ETRI), Korea as a senior researcher. From 2002 to 2004, He worked in the Department of Computer Engineering, Anyang University, Korea as a professor. His research interests include dependable embedded computing, cluster/grid computing, and intrusion-tolerant systems.

**Dr. Jong Sou Park** is an associate professor at Hankuk Aviation University in Computer Eng. Dept., Hankuk, Korea. In 1986 he received an MSc from North Carolina State University in Electrical and Computer Engineering and in 1994 he received his PhD in Computer Engineering from The Pennsylvania State University. From 1994 - 1996 Dr. Park worked as an assistant Professor at The Pennsylvania State University in the Computer Engineering Department and was the president of the KSEA Central PA. Chapter. He was held his current position since 1996. Dr. Park's main areas of interest include Network Security, Computer Architectures, VHDL Hardware Design, Embedded Systems and Mobile Computing. Dr. Park is a member of IEEE, KICS and the Korea Information Science Society. HE is an executive board member of the Korea Institute of Information Security and Cryptology as well as a member of the editorial board of KICS.