A PROBABILISTIC APPROACH TO RISK MANAGEMENT IN MISSION-CRITICAL INFORMATION TECHNOLOGY INFRASTRUCTURE

By

Gadi Oren

Submitted to the System Design and Management Program in

Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering and Management

At the

Massachusetts Institute of Technology

February 2008

Signature of Author ______________________________
Gadi Oren
System Design and Management Program
February 2008

Certified by ______________________________
Professor George E. Apostolakis
Thesis Supervisor
Engineering Systems Division and
Department of Nuclear Science and Engineering

Certified by ______________________________
Patrick Hale
System Design and Management Program
Director

THIS PAGE WAS LEFT EMPTY

A Probabilistic Approach to Risk Management in Mission-Critical Information Technology Infrastructure

By
Gadi Oren

Submitted to the System Design and Management Program

February 2008

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Engineering and Management

ABSTRACT

In the nuclear, aerospace and chemical industries, the need for risk management is straightforward. When a system failure mode may cause a very high cost in lives or economic value, risk management becomes a necessity. In its short history, Information Technology (IT) came to be a crucial part and sometimes the platform of business activities for many large companies such as telecommunication or financial services organizations. However, due to scale and complexity, risk management methods used by other industries are not widely applied in IT.

In this thesis, we investigate how probabilistic risk assessments methods used in other industries can be applied to IT network environments. A comparison is done using a number of possible approaches, improvements to these approaches are suggested, and different tradeoffs are discussed. The thesis examines ways to apply probabilistic risk assessment to a Service Oriented Architecture environment (where each service is an application or a business process that depends on other services, local and networked resources) to estimate the service reliability, availability, expected costs over time and the importance measures of elements and configurations. Finally, a method of performing cost benefit analysis is presented to estimate the implication of changing the services-supporting infrastructure, while taking into consideration the varying impact of different services to the business.

A case study is used to demonstrate the methods suggested in the thesis. The case study compares four different configurations, showing how equipment failure and human error can be placed into a single framework and addressed as a single system. The implications and application of the results are discussed and recommendations for further research are provided.

Thesis Supervisor:
Professor George E. Apostolakis,
Engineering Systems Division and Department of Nuclear Science and Engineering

ACKNOWLEDGMENTS

I had many figures affecting my life and my work in recent years. I feel grateful and fortunate to receive their advice and to listen to their wisdom.

I would like to thank Professor George E. Apostolakis for providing guidance and knowledge and for opening up a new and fascinating field. I was fortunate to find an advisor that comes from a very different engineering field and yet shares my interests in this topic.

I feel so lucky to have so many teachers who were kind enough to share their valuable knowledge with me. I would like to thank Professors David Simchi-Levi and Eric von Hippel of MIT, as well as Professors Joseph Lassiter and Frank Cespedes from Harvard Business School and Professor J. Bradley Morrison from Brandeis University. Special thanks to Co-director John M. Grace for listening, advising and sharing insights from years of experience.

I owe a debt of gratitude to a unique group of friends and peers, my MIT cohort. I learned from their perceptive comments in and out of class and I would like to thank them for being there for me, for challenging me and expanding my horizons.

Finally, I would like to thank my wife, Shirly and my parents Yael and Moshe. Their love, unconditional support and advice are what enabled me to follow my dreams and overcome any challenges. Without that support my academic work and thesis would not have been possible. Special thanks to Susana and Misha Kramer for listening, providing advice and support.

TABLE OF CONTENTS

TABLE OF FIGURES

INTRODUCTION

Background

For most companies today, Information Technology (IT) is one of the most important infrastructures of the organization. As technology progresses and becomes ubiquitous, an increasing number of operations that used to take place in the physical world are performed in the virtual world. There are different opinions about whether or not IT by itself is strategic to the organization, but few argue with the claim that innovative usage of IT can lead to a strategic advantage even if temporary (Carr, 2003). There are a number of examples for strategic advantages that were created using IT infrastructure like Wal-Mart and Dell with vertical integration, airlines with on-demand price adjusting ticketing systems, and others.

For some types of organizations, IT is an enabler for a substantial part of their business. For example, the banking sector is composed of organizations that are information intensive and a large part of their business uses IT extensively. Although the banking industry operated successfully prior to the existence of computer networks, in today's technological landscape it is unlikely that a bank can use its assets effectively without it. For other industries, IT is an inseparable part of the business. An example for such an industry is the telecommunication industry. With large carriers, the majority of the business is carried over IT infrastructures. Without it, carriers would not have been able to operate altogether.

As an organization grows and becomes a more sophisticated user of IT, it may start using applications that are more and more business-critical[1]. Consequently, it becomes less and less tolerant to disruption at the IT service level because of possible financial impact as well as other less tangible ramifications, such as negative impact on reputation and customer satisfaction.

Problem Statement

Information Technology infrastructure is extensively shared between many services (for example, see page 33, Figure 2 - Two services sharing network infrastructure). The same

[1] The definition of 'business critical' here is something that enables an important part of business operations and without it, the business would be disrupted in a way that has financial ramifications.

equipment is being used for many types of business processes. As a result, in many cases the infrastructures become complex and contains many interdependencies.

Common best practices lead organizations to use risk management where business critical services are implemented. Usually a part of the risk management is done by introducing multiple layers of redundancy into the infrastructure and by doing so, ensuring that services will keep on running if a part of the infrastructure fails. However, a byproduct of the redundancy is a substantial addition to the infrastructure complexity.

It is very common to see such infrastructures going through very frequent changes (Wallin and Leijon, 2006) driven by business needs. At the same time, these infrastructures are dealing with high scale of information transfer, in terms of amount of data and in terms of how many end users or services are connected to the infrastructure.

As a result of the infrastructure scale, complexity and redundancy, issues originating from equipment failure or human errors around logical and physical configuration changes, will remain undetected until an actual interruption of service occurs (Reason, 1990, p 179). The latent quality issues lead to unknown financial exposure (stochastic expected loss) to the organization and misalignment between IT management and the organization business goals. Today there is no standard analytic way of measuring and managing that exposure.

Problem Importance

In the nuclear industry, the concept and need for risk management is almost straightforward. Under proper management, a nuclear reactor provides tremendous benefits but if miss-managed, the results can be disastrous. The same logic applied to many aerospace initiatives and large hazardous chemical installations. When a system's failure mode(s) may cause a very high cost in lives or economic value, risk management becomes a necessity. In its short history, Information Technology (IT) came to be a crucial part and sometimes the platform of business activities for many large companies. Addressing this problem and providing better tools to measure risk and benefit as a function of the individual components of the system, the system's topology and the human behavior that is a part of the system, will provide a way to measure service-oriented system robustness and quality of service and open opportunities for more proactive management practices such as Six Sigma. It may also allow measuring the expected risk in terms of financial value and improve communication between management, business units and IT management. Furthermore, an epistemic approach to underlying stochastic assumptions may assist organizations with measuring their improvement over time. Finally, in a Service Oriented Architecture (SOA) environment, where business processes are composed of small distributed services, using a certain service as a building block does not only imply using its functionality, but also assuming the reliability and availability characteristics of that service. These characteristics are often ignored, or may change over time and the newly formed business process may have unknown level of quality. Knowing the service risk profile will allow building bigger systems while making the right architectural decisions.

Conceptual Framework

This thesis aims to achieve the following:

1. Analyze probabilistic risk analysis methods used in industries such as nuclear, aerospace and chemicals to see whether these methods can be adapted to IT infrastructures and how this may be accomplished.

2. Examine how to address IT infrastructure as a single system that is affected both by technical issues as well as human behavior dynamics in a business environment.

3. Suggest enhancements to existing methods and create a new method to estimate risk in service oriented, complex and frequently-changing IT infrastructures.

In the process of achieving the above goals, the following questions were explored:

1. What is the implication of a change in the infrastructure, with regards to the following aspects:

    a. The implication of a change on the profile of different services, in terms of reliability and availability.

b. How does a change to the infrastructure effect the financial exposure (expected loss), or benefit to the business, considering that some services (or business processes) may have a different business impact than others.

2. How can the traditional concept of a problem root-cause be broadened to include physical events, human errors and logical configuration events?

3. What is the importance of different infrastructure components and logical configurations and how can it be measured?

4. How can risk be successfully managed to reduce exposure (expected losses), and what are the tradeoffs?

5. How can the calculated exposure (expected losses) be used for cost benefit assessment.

The following methodology was used:

1. Related work was reviewed to examine previous research.

2. A number of possible methodologies were considered and the most appropriate was selected.

3. Modifications to existing methodologies were considered to formalize a new method that can address the thesis goals and research questions.

4. A framework was developed using the suggested method to allow finding of the required measurements, incorporating both the network physical model, logical model and the effects of human behavior on the managed infrastructure.

5. The suggested method was compared with traditional PRA, using a very simple comparison case.

6. A case study was chosen to demonstrate the result measurements.

7. Conclusions were drawn and recommendations presented based on the results.

RELATED WORK - LITERATURE REVIEW

This chapter describes the research that was done in the subject, what were the data gathering and research methodologies and conclusions. Key findings that are relevant to the subject of this thesis are reviewed in the context of the thesis. As this thesis brings a wide array of subjects together, it is not the intention of the review to provide full detailed review of each one of the individual subjects.

Reliability of network infrastructure using PRA

A number of papers were found related to application of PRA methods to network infrastructures such as computer communication, power transmission, transportation, oil, gas or water production and delivery, and logistical networks. In their paper (Jain and Gopal, 1988), Jain and Gopal describe an algorithm that allows finding the minimal cut-set for Binary State Networks (BSN). Then, using the nodes probability model, global reliability can be calculated. While the algorithm is more efficient than previous algorithms (Aggarwal and Rai, 1981) its complexity and the amount of mathematical operations required to find reliability is related to the number of spanning trees in the network. Depending on the network topology, the number of spanning trees in a network bears an exponential relationship to the number of arcs and nodes, and thus the method is only applicable to small problems. Additional algorithms for accurate probability calculation were suggested by Ball (Ball, 1979), Buzacott (Buzacott, 1980), Rosenthal (Rosenthal, 1977), Satyanarayana (Satyanarayana, 1982) and Hasanuddin (Hasanuddin, 1988). All these methods have a similar issue of an exponential computational complexity. Generally, the problem of accurate calculation of reliability for a network is NP-hard (Ball, 1979). Furthermore, the space required for the calculations is often very large, and even when this issues are resolved, the excessive amount of calculations result in accumulation of round off errors and inaccurate results (Shanthikumar, 1988).

In order to address these issues Shanthikumar (Shanthikumar, 1988) proposes a different approach that calculates only the lower bound (LBR) and upper bound (UBR) of the network reliability. Additional work in this field was done in order to further improve the efficiency of the algorithms and the tightness of the bounds found (Elmallah and AboElFotoh, 2006).

Another area of research is the Multi State Networks (MSN). With MSN, the assumption is that each component has a certain capacity and that capacity may change with a certain probability (hence – a multi state element). With some systems (for example water delivery network) the change in capacity may be continuous. In order to simplify the problem, the continuous range of capacity is usually broken down into discrete steps, represented as a state. Nevertheless, under that assumption the notion of 'failure' becomes more complex and is usually defined with relation to with how much capacity is required between two terminals for the network to serve its goal: capacity related reliability (Soh and Rai, 2005). Furthermore, the complexity issues with Binary State Networks, mentioned above, dramatically increases as a result of the amount of possible spanned combinatorial states (Colbourn, 1987). As before, in order to avoid the complexity, the approach taken by many researchers is to focus on the LBR – lower bound reliability (Satitsatian and Kapur, 2006) for a Multi State Network (MSN).

Other methods for network infrastructure reliability assessment

An alternative method to PRA was examined by a number of researchers, in the context of network related systems. In order to deal with the inherent complexity of the problem, a simulation is performed to produce the network functionality results in different conditions. Monte Carlo Method (MC) is used iteratively to create component level failures (based on individual component failure probability), and simulation to find network functionality and enable estimation of network reliability. In their paper (Zio et al, 2006), Zio et al assert that while this subject can be addressed using PRA and information about the system's cut-sets, path-sets or depth first methods, these approaches are NP-hard (Non deterministic polynomial-time hard) problems, and require mathematically intensive methods for any real life systems. The paper is dealing with a new approach to overcome these issues – using cellular automata (CA) – from general class of mathematical models that are used in modeling various types of complex systems. The paper limits the problem space to a single source - single target network and verifies a binary existence of connection between source and target. Another assumption is that the system is a renewable network and that broken nodes or arcs are fixed after a time that is a simple independent random variable. Using CA and Monte Carlo

(MC) simulation, it is possible to estimate reliability and availability of the system, as well as estimating importance measures.

Next, the authors describe how CA is used to find single source to single target connection. The stochastic failure model of arches and nodes is used to perform MC analysis and estimate reliability. Reliability is calculated as the number of scenarios with a resulting connection, over the total number of all scenarios. In order to find reliability, the paper uses the assumption of an independent random variable (that is identical to all components) that determines break to fix time. Then, a set of random walks are produced according to that stochastic model. Each random walk is limited in time but may have unknown number of steps. Along every system stochastic transition, the system connection state is found using CA simulation. The results are placed in a lattice of fixed time intervals, and the availability can be found per each time interval, and for the entire timeline. The paper explains how CA is very efficient in finding the new state after a small change to the network, as change propagation is sufficient to find the system state instead of a full blown calculation.

The authors also present a number of importance measures and explain how they can be calculated within the same framework. A case study is used to show availability, reliability, and the importance measures (Risk Achievement Worth, Risk Reduction Worth, Fussell-Vesely and Birnbaum), along with a discussion of their implications. In the conclusions, two approaches to risk management are suggested: the first is to invest in areas that proved to influence availability and measures, and the second is to further 'sacrifice' the areas that proved to be non-influential.

The authors offer an alternative to the traditional PRA for network systems. The rationale and benefits of using this methodology are explained and demonstrated well. The context of this thesis is related to the application of IT infrastructure networks and their usage as a platform for business activity. Within that context, the proposed model may not be general enough because of a number of points:

1. The way that reliability is found hides an implicit assumption that the behavior of the system is immutable over time (for example – the stochastic model of break to fix). This assumption limits the ability to investigate system where there is a dependency on

time or external events that depend on time. Specifically, when the event of fixing an element depends on people's awareness to the failure, or human behavior patterns, this model may not be sufficient.

2. The proposed model is a very hardware driven model. In today's networks, many issues may be caused by a logical configuration change (see: Redundancy Types, Logical). A changed logical configuration will not be a "broken" network element, but still be a cause for loss of redundancy or even a lost connection. As such, most of the value metrics do not apply to logical configuration change the way they apply to element break, because there is no notion of 'reliability' to a logical configuration change. It is functioning as designed even when it is causing a system level failure. This model does not lend itself well to systems that include a complex set of logical rules overlaying the physical structure.

3. In order to find availability, a part of the suggested method is to use discrete time periods, but no insight is offered into how to choose them or information on the relation between the selected time period and the amount of MC cycles required.

4. The model is trying to enable usage in real world systems but the failure mode of each node and arc are too simplistic. In many network applications the nodes and arcs may have internal structure that affects their activity. Furthermore, the internal structure will likely impose different failure characteristics for different nodes and arcs and will add dependencies.

5. In an environment where a logical configuration change event may be a cause of an issue, the provided importance measures may not be a good measure for both hardware failure and logical configuration change event.

6. Finding the measurements for extremely rare events may be difficult as a result of very low amount of data points.

7. When using this methodology for IT infrastructures, the network will be used by a large number of 'services'. Each network element (physical or logical configuration) will be shared by these services, and these services may carry very different economic

value to the organization. In order to be used by upper level management, the importance measurements should be enhanced to reflect true economic value as they affect a cluster of services.

The methodology presented in this paper is promising and may be used with IT infrastructure networks. However, in the context of this thesis, enhancements to the proposed model and its assumptions must be incorporated to allow real world applications.

IT infrastructures complexity and architecture

For a number of industries, the IT infrastructure and network are constantly changing, growing and become more complex. At the same time, business pressures require the network services to be as reliable as a utility, while cost savings dictate faster turn around time and fewer expenses on operations. In their paper "Rethinking Network Management Solutions", (Wallin and Leijon, 2006), Wallin and Leijon, discuss these subjects in the Telecom industry. The research methodology used was gathering data by interviewing Telecom Operators, drawing qualitative conclusions from these interviews (an example to a Telecom Operator would be Verizon, although no specific Telecom Operator is mentioned in the paper). The authors define three types of users to networks in the Telecom industry: The *network operators*, which must earn money on their services, the *network service users* (business and customer), who pay for services and the *network administrators*, who staff the network operations center. The authors present a current problem in this industry around the following issues: A. the network generates 100,000 to 1,000,000 alarms a day (see explanation in the next paragraph) (!), B. the network changes constantly, C. there is a very complex service structure, D. Customer interaction and SLA (Service Level Agreement) management is very difficult, E. Cost reduction – smaller teams manage larger networks, F. Multiple interfaces and lack of sufficient standards.

An alarm is a diagnostic notification that can be sent from any piece of equipment in the IT network infrastructure. The alarm is usually a message that the equipment manufacturer estimated may be of value in certain conditions to the network administrators or operators. For example, the following events will generate an alarm (a very small subset of the full list): a loss of signal on a network connection, re-establish connection, a change in logical configuration, failure to negotiate a protocol with another equipment element, internal fan or power supply failure, hi or low communication buffer levels. Many network failure or change situations may lead to 'event-storms' – detailed alarms sent from multiple equipment elements. Alarms, as they are generated today, are raw, uncorrelated and flood the operators with too much information and no context. This is creating more load and little benefit in performing RCA (root cause analysis). The alarms are not prioritized according to their impact on the business (or SLAs) and no topological information related to the alarms is kept. As a result, the escalation process (the process of event triage to classify and address the most urgent issues first) is heavily dependent of human experts and distributed across multiple departments. Furthermore, because the alarms are generated from an equipment element that does not understand the context in which it is working, alarms are cryptic and provide no hint with regards to the root cause. This leads to steep training and productivity cost incurred by operators.

Telecom IT infrastructure networks are in a state of constant change. Different business and technical requirements drive daily changes that are very challenging to operators – constantly changing the network while keeping all existing services running. The expected time-to-change moved from months to hours. Very few operators have change management practices fully implemented, and new elements that are introduced to the network drive new unfamiliar types of alarms. Although SLA is signed and paid for in contracts, the network management and the IT infrastructure today provides no assistance in the actual management of the SLA. Some operators are starting to implement methodologies such as Information Technology Infrastructure Library (ITIL) to deal with change management.

Complex service structure is mentioned as an underlying problem in network management solutions according to interviewed operators. The operators do not have real visibility of services across processes and systems. Operators are looking for a service management or even an SLA management oriented solution. Several issues should be resolved to achieve this: A. Network topology management – mapping network elements relations, B. service management – mapping services relations and SLAs and C. a services centric management to all activities such as customer care, fault management and others. Customer interaction requires transparent high level billing and status reporting in a customer consumable way.

The authors suggest a number of ways to improve but states that the field has to go through a paradigm shift and start using a different set of principals. Service model and real time status should provide mapping of resources, network topology and customers. For example, a service that is an aggregated set of services should present the proper severity as a result of the service topology. A high severity event on a low importance sub service may ultimately mean a medium severity event from the top level service point of view. The topology and changes to the topology is mentioned as something that should be a core component, but according to the authors, existing solutions do not handle that important aspect today. The paper also enumerate a number of other factors that should improve such as better standards, allowing to manage heterogeneous networks with dynamic services, and high frequency change management. Another issue today is the fact that the system relay on human experts that take years to acquire their level of knowledge, but this knowledge is hard to reuse because of the manual nature of the operations today. Operators need more automated solutions in the form of expert systems that are self learning and self adapting.

In summary, the authors state that the Telecom industry changed from network management to service management, and operators need solutions that are working from a service point of view allowing automated RCA and correlation to service impact.

Short-term goals vs. mid- and long-term goals

A number of papers discuss the constant predicament that is faced by management at different levels: with a limited amount of resources what is the right balance of resource allocations towards short-term, mid-term and long-term goals? Investing in short term goals is very compelling since the results can usually be measured within a timeframe of a quarter. In their article (Lordish and Mela, 2007), Lordish and Mela discuss the fact that some business activities such as building a brand name are inherently long, while at the same time, the management is being measured on a quarterly financial performance metrics. This places the management in a difficult situation whenever an investment for a long term is required, especially if it hurts the short term performance. A solution that is commonly used is creating balanced score cards that measure not only the immediate impact of resources that were invested for the short term, but also other measurements that are by nature more long term. Although this paper considers operations that are non engineering related - the 'marketing infrastructure', the principle applies to any type of infrastructures. Reason also mentions the same issue in his book (Reason, 1990) when discussing the fact that safety issues are always mid-term issues with a feedback loop that is both very long as well as hard to associate in terms of cause and effect. As a result, in a production environment, the short term result created by increasing production capabilities over investing in maintenance or safety, is a path that is chosen often by management. Reason posits that these choices are, in many cases contributors to later accidents.

In a paper by Repenning and Sterman (Repenning and Sterman, 2001), the authors presents a behavioral pattern called the 'Capability Trap'. The pattern may happen when an organization starts leaning excessively towards short term thinking, investing more in short term results over long term infrastructures and capabilities. While this provides the expected short term result, the organization is slowly losing capabilities, and in order to perpetuate the short term results, more and more short term efforts are required. This leads to a "better-before-worse" situation, where the emphasis is around 'work harder'. This cycle is very hard to reverse because it requires a recognition and acceptance from the organization that a long "worse-before-better" cycle is required where the emphasis will be 'work smarter'. The authors discuss how companies build long term capabilities through process improvements, and hypothesize that a set of powerful thinking patterns drive companies into the 'Capability Trap', where the capability can be related to different types of organizational infrastructures such as engineering, marketing, sales and other capabilities. Repenning and Sterman present a case study on Du Pont, where in 1991, Du Pont was spending 10-30% more than industry average on maintenance while the uptime of plants was 10-15% lower than industry average. Du Pont was deep into a capability trap that stemmed from the preceding economic and industry conditions. Through a slow process of recognition, training and implementation of reinvestments in pump maintenance the company went through a "worse-before-better" cycle and after two years enjoyed impressive turnaround in plant uptime and reduced costs.

As with Du Pont's 'network' of pumps, the issue of short term investments vs. mid term or long term investments has a substantial effect on IT infrastructures and network environments. As with other industries, one can choose to work harder or smarter. The IT environment and team is constantly driven to generate results faster and that sometimes hurts the ability to build long term capabilities and remove latent issues. In the context of the thesis, this pattern is considered as an important contributor to the creation of latent conditions.

Latent conditions in complex systems

The subject of latent conditions is discussed in a number of papers. Reason devotes a chapter in his book (Reason, 1990) on that issue. Reason speculates that the more complex and automated a system becomes the more likely latent conditions to occur in the system. The reason is that as a system has more layers of protection and redundancy built in to it, people that make mistakes are left without indication for the mistake, as the system keeps on working, and the problem is more likely to remain undetected. Reason provides a number of data points indicating that latent conditions can remain in the system for a very long time – months and even years. In many cases the problem is discovered as a result of the system getting into a failure mode, after a number of redundancy layers failed. Examples are presented from the Three Mile Island (1977, nuclear, USA), Bhopal (chemical, India), Chernobyl (nuclear, Russia) and the capsizing of "Herald of Free Enterprise" (1987, shipping, Belgium).

Human behavior as a contributor to latent conditions

In 1990, James Reason published a book "Human Error" (Reason, 1990), focusing on accidents and safety issues that manifest themselves in modem technology environments, resulting from human errors. The book spans a number of themes, psychological research of human error to accidents investigations.

The author presents a short history of the research done on human error and the motivation behind it. The main thesis of the book is that human error manifests itself in a limited number of ways, and these ways are closely related to the cognitive mechanisms that are in charge of storing knowledge structures and retrieving them in response to situational demand. The author discusses variable errors and constant errors and presents the concept of predictable errors. One goal of the field is to be able to predict under which conditions errors are more probable. The prediction of errors takes a probabilistic form (during January, some percentage of people will make mistakes writing checks). Next, the relationship between error and intention is discussed to show that the concept of error must be related to the intention. The author presents a framework of three types of errors based on questions about what the intention was, what executed steps were taken and did they achieve the goal. Using these question, errors are categorized to "mistakes" that are due to an incorrect intention, and slips or lapses that relate to correct intention but incorrect execution. A slip is an unintended action while a lapse is related to incorrect store or retrieve of planned actions. The author states that mistakes by nature are much more dangerous and may stay undetected for a very long time. Along these lines the author distinguished three types or errors: planning (mistake), storage memory (lapse), and execution (slip). The author then distinguishes two forms of errors for an underspecified problem: "similarity matching" and "frequency gambling". The author covers different methods of research including collection; questionnaires, laboratory, simulator, and case studies. No single method is considered the best practice for human error research.

Next, the book covers the theoretical background and presents a framework for human error. A distinction of two control modes is presented – conscious and unconscious, and two memory areas are mapped to these modes – working memory and knowledge base. The attentional control mode is closely identified with the consciousness and working memory, and is sequential, slow and difficult to sustain for more than a short period of time. This mode uses 'attentional resources' and is in charge of setting future goals, means to achieve them monitoring and error correction. The schematic mode manages on the background a number of processes of regularities detection with different aspects of the world. This mode is very fast, parallel and triggered by certain conditions (activators). It is assumed that the schemata processes may work based on activation that is not only from the conscious area and that is how coherent actions and information is available in an unintended way. Specific activators can be used in an intentional way but use for an extended period of time, moments of preoccupation or distraction would be a very common cause of slips. General activators happen in the background and would be activated by context like a familiar environment.

Some error forms related to the system defaulting to the highly frequent schemata that is deemed the most suitable for a familiar situation (even if it is not the most suitable).

The book presents a generic error modeling system and error types are further categorized into skill based slips and lapses, rule based mistakes and knowledge based mistakes. The skill based mistakes are related to lack of attention or over-attention that leads to missing an input or misjudging the location in a sequence of actions and then taking actions based on "frequency gambling". Rule based mistakes are related to misapplication of good rules or application of bad rules to a situation.

Reason attempts to take the conceptual models presented so far and focus on the question "what type of machine would operate correctly most of the time but can produce occasional wrong responses, similar to human behavior". The book presents a computer model that uses the principles presented so far for a very limited experiment testing knowledge about presidents of the US. The computer model allows finding information in an unintentional way that is heavily affected by the frequency of exposure to the data with a level of randomness when coming to decisions that are underspecified. The results of the model presented good correlation with human subjects' performance participating in the same experiment along a number of measurements.

Following the presentation of human error theory and modeling, the book discusses latent errors and system disasters. The author states that in many cases, the actual operator accident trigger is just the final touch in a series of mistakes that were done before, by high level decision makers, construction workers, managers and maintenance people. The author also present the irony embedded in very complex nearly automated systems, where the operator is trained and becomes experienced with monitoring, but one of his implicit (and very important) functions is to deal with whatever the designers could not predict and thus inherently are complex situations and requires extremely high level of training, experience and in depth familiarity of the system, that the operator does not have. This opinion is reinforced with examples from the timeline details of accidents in the nuclear industry. The nature of latent error is described to be potentially more dangerous than other errors, and the can stay latent in the system for years without detection. Next, latent errors that originate from insufficient maintenance are discussed. As any organization is in a constant conflict between short term goals, like increasing production, capabilities and mid to long term goals, like improving safety, it is often seen that management chooses the short term goals over the long term ones. The nature of the maintenance related feedback is very long and it is hard to connect between cause and effect. The chapter is summarized by stating that **“most root causes of serious accidents in complex technologies are present in the system long before the event”.**

RISK MODEL

Mapping the Risks

In order to measure the risk, a careful definition of what the undesirable end states are is needed. Within the IT infrastructure world there may be a very large set of undesirable conditions. In the context of this thesis, the undesirable conditions are defined as a set of conditions that prevent a company from engaging in a certain activity (whether internal or external) and complete this activity within a predefined amount of time. Furthermore, it is assumed for this thesis that the immediate loss of revenue incurred because of the incomplete operation is tangible and can be quantified (if not, it may not be an important part of the business, and maybe risk analysis is not required). This type of undesirable condition will be defined here as Interruption of Service.

For example, if the web site of an on-line retailer can not process credit card transactions, the loss of revenue can be estimated. If the web site generates on average revenue from sales of $240,000 per day, a simple estimate may be to determine that one hour of service interruption translates into $10,000 loss of revenue to the organization (many financial models can be used here. Financial estimation models are outside of the scope of this thesis). However, if the web site is working but it is slow, there may be a loss of revenue but it may be hard to estimate what that loss is, especially since it may be more indirect than direct, such as lower customer satisfaction that cause customers to buy less over a long period of time. Another example – consider a Telecom Operator that provides 100,000 minutes of conversation from Boston to Moscow per hour with a margin of 3 cent per minute. If for some reason the Boston to Moscow channel should fail, the direct implications would be a loss of $3000 per every hour of service interruption (for Telecom related information see also Wallin and Leijon, 2006).

Building Blocks

Assumptions:

a. Software: The majority of elements in the IT domain involve software. Even elements that would be considered traditionally to be in the hardware category depend heavily on software. An example would be a network switch that is purchased and installed as a hardware device but is driven internally by software. It is outside the scope of this thesis to analyze software reliability. For the purpose of this work, it is assumed that software provides some functional service that works well if configured correctly and has all the required resources at its disposal.

b. Granularity: Each of the elements that are discussed in this risk model can be further decomposed. With each additional level of decomposition, the risk model may be more complete (although not always). In order to decide what level of decomposition is sufficient, the following rule of thumb is used: the decomposition is done to the level of modules that are replaceable or relevant to risk management by an IT user. This includes any equipment enclosure or sub assembly that can be replaced on site by the local staff such as power supplies, fans, internal cards or other sub assemblies. If a manufacturer does not expect local staff to disassemble further a certain component, it is assumed to fail as a whole and no additional decomposition is needed for modeling it (for example a hard drive may be decomposed but IT users would replace it as a unit). If there is a high level architecture difference between systems or devices that are presented as an advantage, it is a good idea to include that difference in the decomposition. A simple example would be a device architecture that has less power supplies than another device. Manufacturers may sell them in two categories: 'Normal' and 'High Availability', and the latter would include two power supplies. This device would provide a different risk profile, and be priced differently. Another example is a service that can provide credit card clearance and relay on more than one digital clearing house as a way to increase availability and robustness. Modeling these differences would allow the methodology suggested in this thesis to quantify these architectural advantages, to show the cost vs. the economic benefit of such differences.

c. No circular dependency: The failure mode of an element Ex may depend on other elements Ea-Eb, and they may depend on other elements, etc. The dependency tree that is spanned from Ex does not contain Ex at any depth of the tree, for any Ex that belongs to the infrastructure.

The following are elements that are used as a part of the model:

1. Service: A service is the ability to perform something that creates value within a given time frame. A service may depend on one or more other services and a number of servers. A service may also depend on a set of logical and physical resources accessible over communication networks. For example, a payroll is a business process that may involve for company ACME, checking the amount of hours done by a contractor during the last week, checking his contract, finding the contractor's hourly rate and home address and sending the check. In that example, the payroll business process can be defined as a service that depends on three other services: a. the hourly registration system, the contract management system and personnel files. If any of the three services is not working, it is possible that the payroll business process can not be performed, and that there is an interruption of service.

2. Server: A server is an enclosure that contains a set of resources and communication capabilities. A server may be used as a part of a service. A server depends on external power supply(s) in order to function. A server may have subcomponents that are redundant with the intention of risk reduction.

3. Network switch: A network switch is an enclosure that allows communication (usually broken into pieces of a limited size known as "data packets") to traverse the network. A switch depends on external power supply(s) in order to function. A switch may support logical entities that are required for certain communication routes to exist.

4. Network connections: A cable that allows a communication to exist.

5. Network Attached Resource: A resource that is available over the network. An example for such a resource would be a file that is hosted on a remote file system. Network Attached Resource depends on Resource Physical Components.

6. Resource Physical Components: Physical elements that comprise a Network Attached Resource.

7. Logical configurations. There are many types of logical configurations (depending on the type on network) that may enable or disable the usage of a Network Attached Resource by a Server or by a Service. For the scope of this thesis a simple subset of the logical elements was chosen:

   a. Resource and Server allow group: This is a logical group that allows a certain Server to access a certain resource. For example, a file on a remote file-system that is marked to be accessible by a certain Server.

   b. Port exposure group: This is a logical group that lists a set of resource enclosure ports that allow access to network attached resources. This group is set specifically per every network attached resource.

Redundancy Types

There are several types of redundancy used in IT systems:

1. Physical: This type of redundancy is the most straightforward redundancy. For example, a service may depend on one Server or two Servers. Assuming a dependency model of one out of N, it is clear that having two servers is preferred in terms of redundancy.

2. Logical configuration: This is a situation where there are different logical configurations that allow a certain Service or a part of a Service to function. Example – let's assume that in the US air traffic system, there are flights that land in Providence coming only from New York or from Boston. In that imaginary system, there is a redundancy level of two with regards to possible ways to get to Providence. If one day, the Boston authorities would decide that it is no longer possible for planes taking off from (or going through) Boston to land in Providence, the system redundancy would be reduced to one, even though there is no physical issue, and nothing in the system is

'broken'. The question "Is there any component of the system that is broken?" is likely to be answered with a "no". From a system level view, flights keep on landing in Providence and the system is working. In this example, the logical configuration of the system is the arbitrary decision to rule out a certain flight route. This is very similar to possible logical configurations that can be used in networking infrastructures.

3. Temporal: A situation where there is redundancy that is an exact replication of a resource that is done at a certain interval T. At t=0 the resource is replicated, while at 0<t<T the resource is not exactly identical. An example for this type of redundancy is a backup. If a piece of data (the resource) is replicated into another location every hour, than every hour we have full information redundancy. However as the time progresses between 1min until 59min the quality of the replication is (potentially) declining. An example for 'lower quality': If a problem occurs in the original copy after 50 min, all the changes that took place over that period of time are lost. If that data reflects for example, information on ships that left Boston harbor, that loss of quality may be acceptable, but on the other hand if the lost data represents information on planes that took off from Logan airport, this loss or quality may not be acceptable.

Understanding the Risk

Most large companies would have an IT strategy. This strategy is comprised of a set of capabilities that the company would like to develop and use to better compete in the marketplace. In order to execute the strategy, a set of services is being developed and implemented in the IT system. Each one of these services may be of very different value to different people in the organization. For example, an HR system may be important to the HR personnel and an interruption of service in this system for a single day may cause some damage but this damage is hard to measure monetarily. However, other services may lead to higher and more tangible value. These tend to be the outbound services. For example, a web site failing with a large on-line retailer can be quantified very easily: If today is similar to yesterday (both are mid week, and no holiday days), and yesterday the web site generated a revenue of $10,000 between 12:00 and 1:00PM, a simple estimation model would be to assume that an hour of downtime today (around the same time) will lead to a loss of $10,000. Obviously, much more accurate and complex estimation models can developed based on the available data and statistics. Another example would be an airline ticketing service system. The failure of such a service for a single hour may lead to a very high loss of revenue, and again this cost can be estimated by the airline company.

Another aspect of the risk is probability. Based on the failure profile of all the physical elements that comprise a service (a service depends indirectly on a large number of physical elements), there is a probability P(t) that there is going to be an interruption of service for a certain service between time 0 and time t. If the IT system does not change, P(t) will be increasing as t increases. P(t=infinity) = 1 because after a long time it is certain to have an interruption of service. However, the function P(t) may itself change over time. For example, if there is a redundant resource that has failed, an interruption of service is likely and P(t) would reflect that. However, if a maintenance person fixed the broken resource, P'(t) will replace P(t) and will show lower probability for a failure. The opposite may happen if a resource got broken at some point. P"(t) may reflect a higher probability for a failure or may actually be 1, if the broken resource actually caused an interruption in service.

For a given set of services:

$$\{S_1, S_2, S_3, \ldots, S_n\}$$

There is a matching set of failure probabilities:

$$\{P_1(t), P_2(t), P_3(t), \ldots, P_n(t)\}$$

And a set of monetary estimation of price of interruption of service:

$$\{V_1, V_2, V_3, \ldots, V_n\}$$

The exposure (the expected loss) of the organization from time t=0 until t=T is the expected "price" to pay for all possible interruptions of service during that time:

$$e(t) = \sum_{i=1}^{n} V_i P_i(t)$$

Assuming that e(t) can be found, an organization can choose a number of methodologies to reduce or bound the value of e(t).

In order to try and find e(t), two methods were considered.

1. PRA – Probabilistic Risk Assessment

2. A combination of Cellular Automata and Monte Carlo method

It should be noted that the first method was found difficult to implement for this type of problem. The next chapter documents the work that was done, and can be skipped if the reader is more interested in the chosen methodology only.

Risk Analysis Using PRA

In Figure 1, we can see a network diagram that represents a simple service:

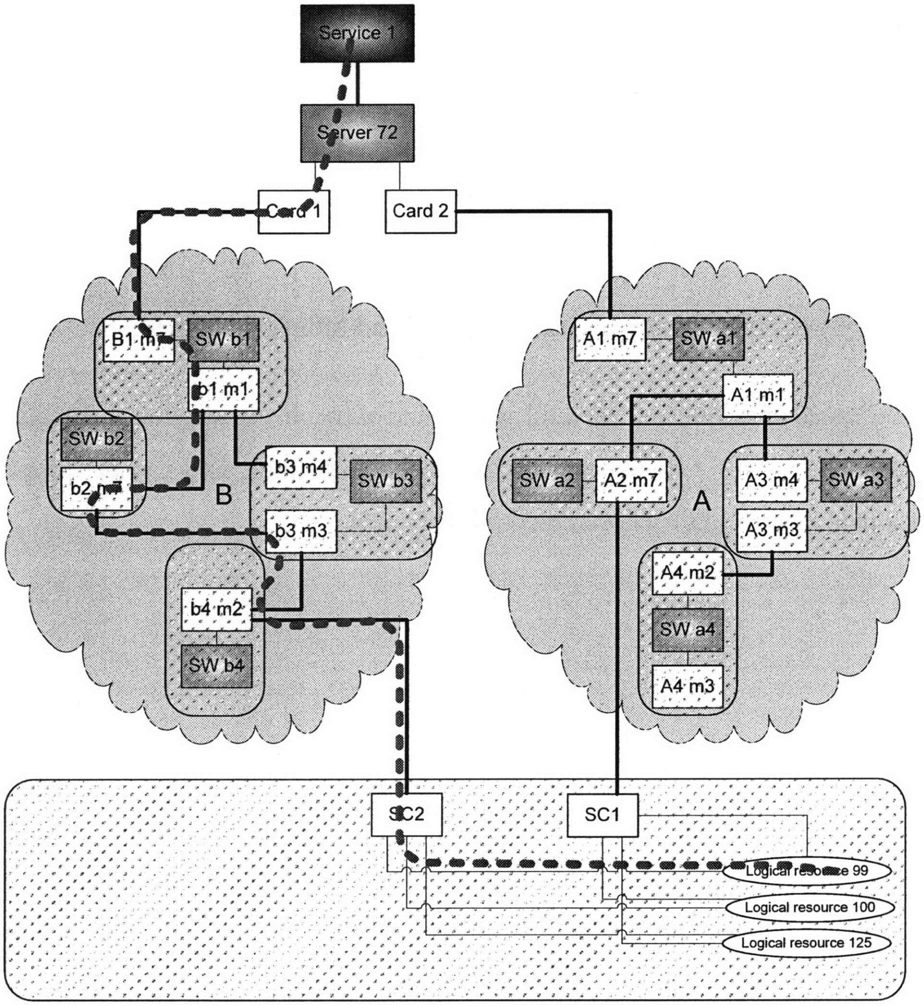


Figure 1 - A Simple service accessing a network resource

The service depends on a single server - Server 72. For redundancy purposes, Server 72 can access the network through two network cards. Each one of the cards can access through a separate set of switches, a number of logical resources. It is important to note that each card gets access to the information through a physically different array of network switches.

Eventually through both cards, Server 72 can access three network attached resources (99, 100 and 125). The dependency model here is that all resources are needed for the service to operate. In reality, the same network resources may be used by other services:

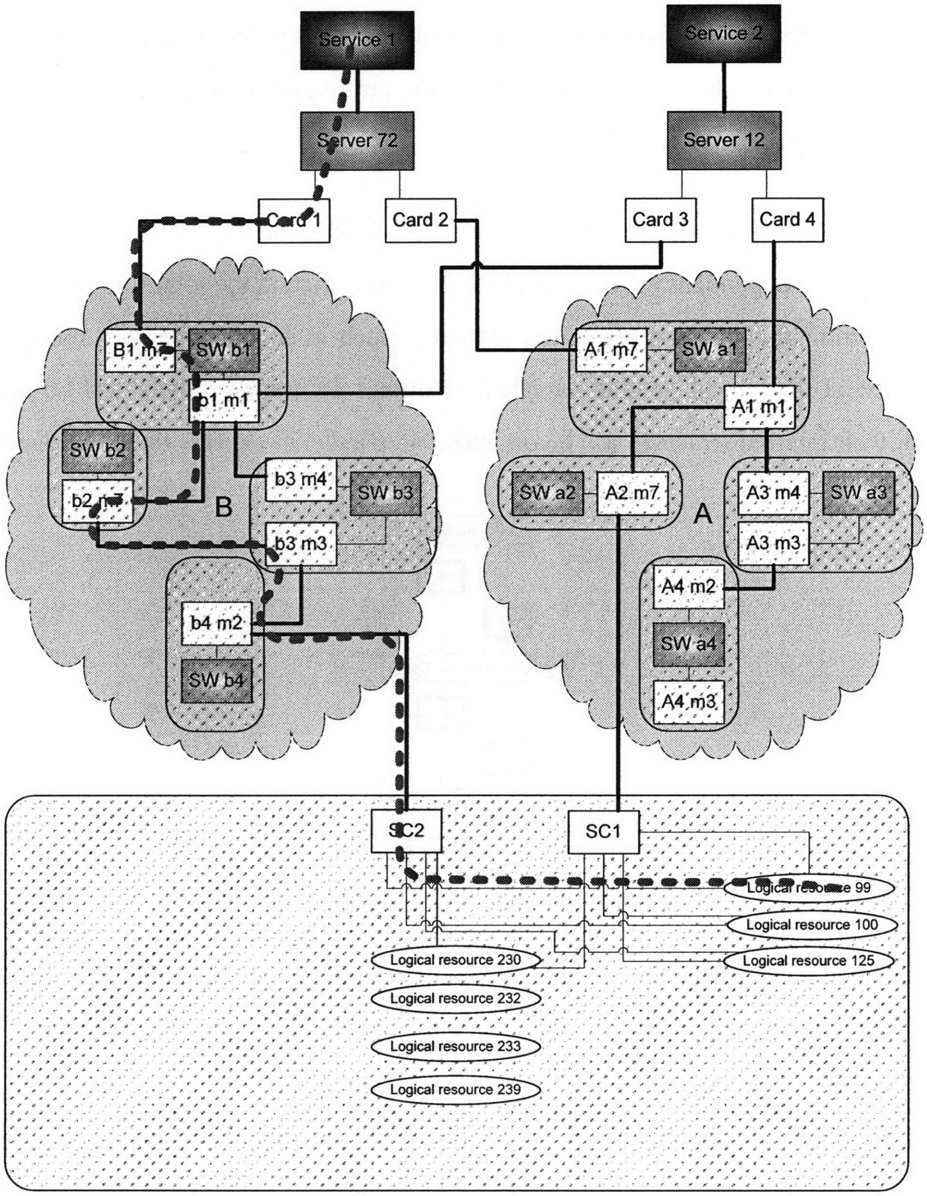


Figure 2 - Two services sharing network infrastructure

As can be seen in Figure 2, service 1 and service 2 are both using the same switches to access different network attached resources.

Dealing with logical configurations: Logical configurations are an overlay on the physical configuration. In the most complex case, the logical configuration may allow or deny the flow of information based on the specific path that the information passed through. As this information is not captured in graph flow algorithms, a modification of the suggested method would have to be developed. Two options are possible. One is to 'color' information based on its path. Another approach would be to replicate the entire graph per each one of the logical configurations. Both approaches are challenging.

Dealing with physical dependency: Whenever a physical dependency exists we will change the graph to reflect this dependency. For example, lets assume that all the logical network-attached resources depend on a physical replicated resource (such as in the case of a redundant file system). Since all the logical network-attached resources are required, they will be organized serially, but their internal structure will be organized in parallel to express the redundancy:

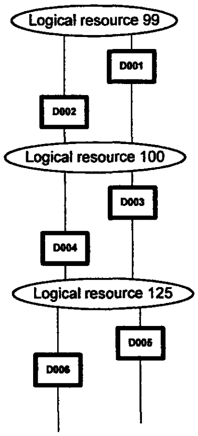


Figure 3 - Logical resources and physical resources presented as a graph

A failure of the service depends on a failure of (D1 and D2) or (D3 and D4) or (D5 and D6). Another example would be to express the dependency of network elements in power supplies and power grids (or generators):

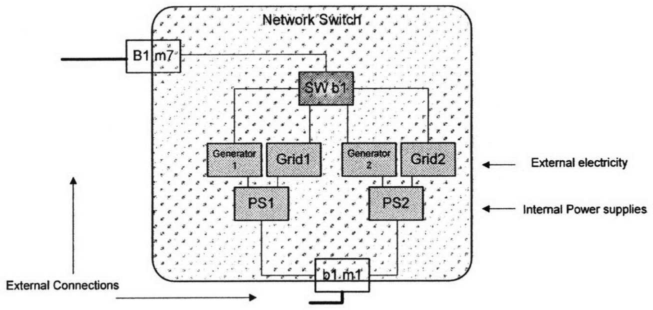


Figure 4 - Internal structure expressed as a graph

Although the actual data does not flow through the grid and power supply, their proper operation (or a subset of them) enables that flow. In a similar way, other elements can be introduced, such as geographical location (see Common-Cause Failures). This may be very important to judge the robustness of a system as was proven in many physical disasters and terrorist attacks where redundant network elements should be place in separate geographical location.

Minimal Cut Set

After constructing a graph that represents both the flow and the dependencies, the following algorithm is suggested in order to find the minimal cut-sets.

Procedure: MinimalCutset(S,D)

```
1. R=0;

2. Let S be the source and D be the drain.

3. $C = \{P_1, P_2, P_3, ...P_n\} \mid x \in C, connected(S, x)$

4. What is size(C) ?

    o size(C) = 1: R=R+S, S=first(C)
```

```
        o  size(C) >1: B=FirstMutualPoint(S,D); $R = R + \prod_{x \in C} MinimalCutset(x,B)$; S=B

        o  size(C) = 0: Stop → result is R

    5.  Go to 3
```

```
Procedure: FirstMutualPoint(S,D)

    1.  $C = \{P_1, P_2, P_3, ...P_n\} \mid x \in C, connected(x, S)$

    2.  $G_x = \{Q_1, Q_2, Q_3, ..., Q_n\} \mid Q_1 = P_x, P_x \in C, Q_n = D, 0 < y \le n \rightarrow connected(Q_{y-1}, Q_y)$

    3.  All mutual points between branches: $G_{\mathrm{m}} = \bigcap_x G_x$

    4.  Stop → result is $first(G_{\mathrm{m}})$
```

For example, observe the following network connection from a source a to the drain p:

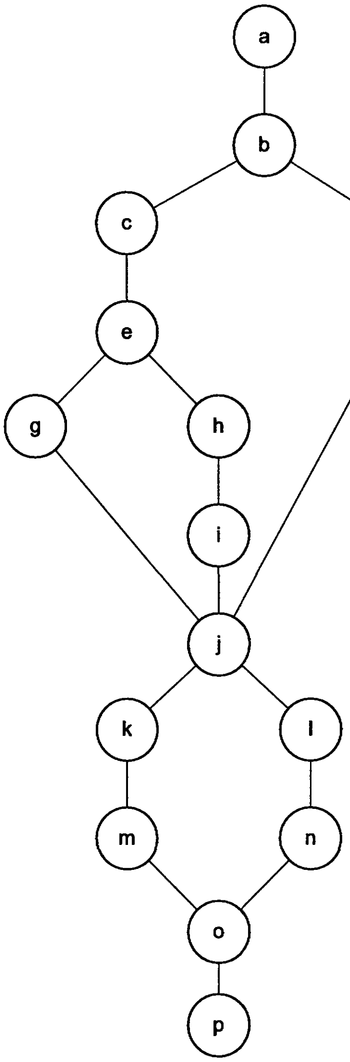


Figure 5 - An example network for minimal cut-set algorithm

1. R=0

2. C={b}

3. size(C) = 1

    a. R=a; S=b

4. C={c,d}

5. size(C)>1: B=FirstMutualPoint(b,p)=j;R=a+b+(d+f)*(c+e+(g)*(h+i))+j; S=j

6. $C=\{k,l\}$

7. $size(C)>1:B=FirstMutualPoint(j,p)=o;R=a+b+(d+f)*(c+e+(g)*(h+i))+j$ $+(k+m)*(l+n)+o; S=o$

8. $C=\{p\}$

9. $size(C)=1 : R= a+b+(d+f)*(c+e+(g)*(h+i))+j +(k+m)*(l+n)+o+p$

$$R= a+b+(d+f)*(c+e+(g)*(h+i))+j +(k+m)*(l+n)+o+p$$
$$R=a+b+(d+f)*(c+e+gh+gi)+j+(kl+kn+ml+mn)+o+p$$
$$R=a+b+dc+de+dgh+dgi+fc+fe+fgh+fgi+j+kl+kn+ml+mn+o+p$$

Other algorithms were suggested for this issue (Jain and Gopal, 1988). The suggested algorithm is more suitable to the specific problem because it incorporates a hybrid model that allows both the pure network aspect as well as the internal and external structure to be taken into account.

Model Limitations

This proposed model, using PRA and network graph theory contains a number of limitations:

1. Some of the elements in the model are based on software. Currently there is no simple way of finding the reliability of software.

2. The model assumes that there is no circular dependency. This assumption may not be always true as IT systems may end up with circular dependencies.

3. The general problem of network reliability using cut-sets suffers from the following issues:

    a. The issue may be so computationally intense that the benefit may not justify the cost. See (Ball, 1979).

   b. After a calculation of a very complex cut-set with many parts, the accumulated numeric error may render the result unusable.

4. The above model is not generic enough to deal with certain types of network logical configurations.

Given the limitations, other directions were explored in order to compare with this approach. The next chapter presents an alternative approach to PRA.

Risk Analysis using CA and MC

Using PRA to assess reliability and availability leads to an NP-hard problem (Ball, 1979). In the article "A combination of Monte Carlo simulation and cellular automata for computing the availability of complex network systems" (Zio et al, 2006), the authors suggest a different method to assess reliability and availability of networks (see also literature review). The suggested method is a good base to address the problem but requires some enhancements for a number of reasons:

1. The suggested method is good for one source to one destination, whereas network services use multiple sources to multiple destinations connection model.

2. The authors used a limited set of possible failures (arcs and nodes). In a more realistic scenario, nodes and arcs may be influenced by internal structure and exogenous conditions (e.g. power, cooling, weather disasters, common cause failures and others). In order to allow assessing the risk as it is effected by the technical management of the infrastructure, it is necessary to represent other structures that the technical management can make a decision on (will I buy a device with one internal power supply or redundant?). This will create a hybrid model of CA and a set of dependencies that may change its activity state.

3. The paper assumes that the system is not changing over time. This assumption is not sufficient for the problem that this thesis is trying to address, which include a high amount of changes to the network (see also Wallin and Leijon, 2006 on changes in Telecom networks).

4. The suggested method is very hardware driven. However, a large number of issues can be caused by network logical configuration changes and human errors related to these configurations.

5. The model assumes a stochastic constant fix profile for failed elements. This assumption is not sufficient. One of the claims of this thesis is that elements will be fixed based on their impact to the business.

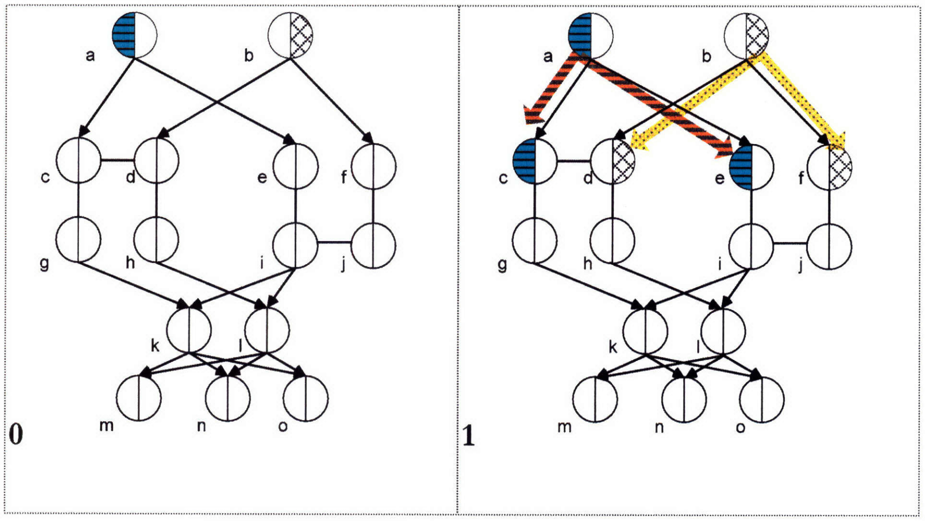
6. The suggested importance measurements may apply well to a hardware driven model but are limited when the cause to failure may be also logical configuration changes.

7. The suggested importance measurements do not present the importance based on the impact to the business. The importance should be presented from a service point of view.

A simulation package was developed to enhance the method suggested by Zio et al, using MC (Robert and Casella, 2004) and specific PRNG (Matsumoto and Nishimura, 1998). The following section describes the enhancements done to the suggested method in order to address the above points.

Multiple Sources to Multiple Destinations

In order to allow the same CA infrastructure to address multiple sources to multiple destinations, the individual Automaton had to be enhanced. Each of the new Automaton can be viewed as having internal multiple automatons. Here is an example of multiple flows:



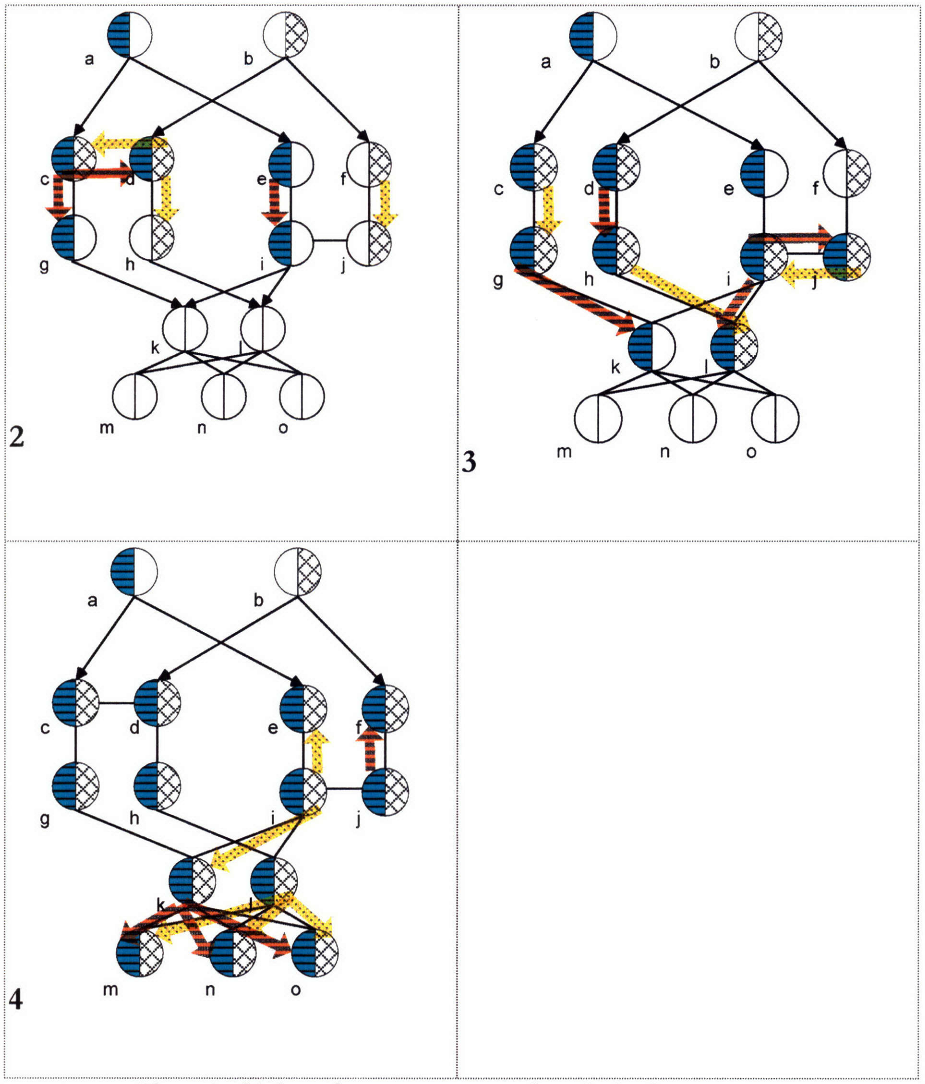


Figure 6 - CA Steps from two sources to three destinations

In Figure 6, we can see a multiple source to multiple target CA flow example. Following the connection from b, the first step (1) (right hand side) d and f are activated. The next step (2) c and h are activated because d is active, and j is activated because f is active. In the next step (3) g is activated because of c, l is activated because of h and i because of j. In the final step (4), e and k are activated because of i, and m, n, o are activated because of l. In general, there can be any number of sources and any number of destinations.

A hybrid model

In order to allow assessing decisions of infrastructure management and architecture, one level of decomposition into some of the elements is required, as well as some exogenous inputs.

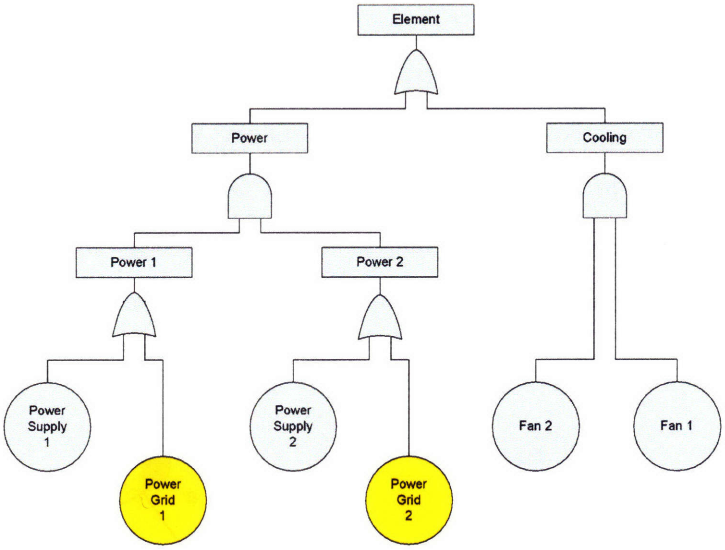


Figure 7 - A typical fault tree for a network element

Figure 7 presents a typical fault tree of a high-availability network element. The element may be very simple, for example a single CA, or a more complex like in the case of a server that has two network cards. In that case, as each network card has its individual stochastic failure characteristics, a number of CAs would represent the server network element.

A service is an abstraction of a business process performed by the organization. As such, it can depend on network elements (for example, a number of servers running an application), other services, or both. Figure 8 describes a service that depends on at least two network elements to perform as well as an additional service (that may also depend on other elements)

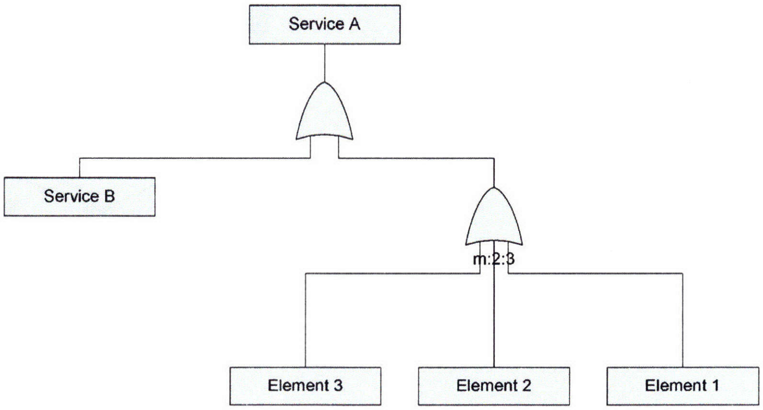


Figure 8 - Service fault tree

Changes over time

Changes can originate from a number of sources such as a failure event, a physical change event (network operator is changing the network topology) or a logical configuration change, but all changes are treated in a uniform way. First the fault trees are realized to determine, next changes to the topology or logical configuration are applied locally and affect a subset of the CA nodes. Finally the change propagates through the network and may change the system's state.

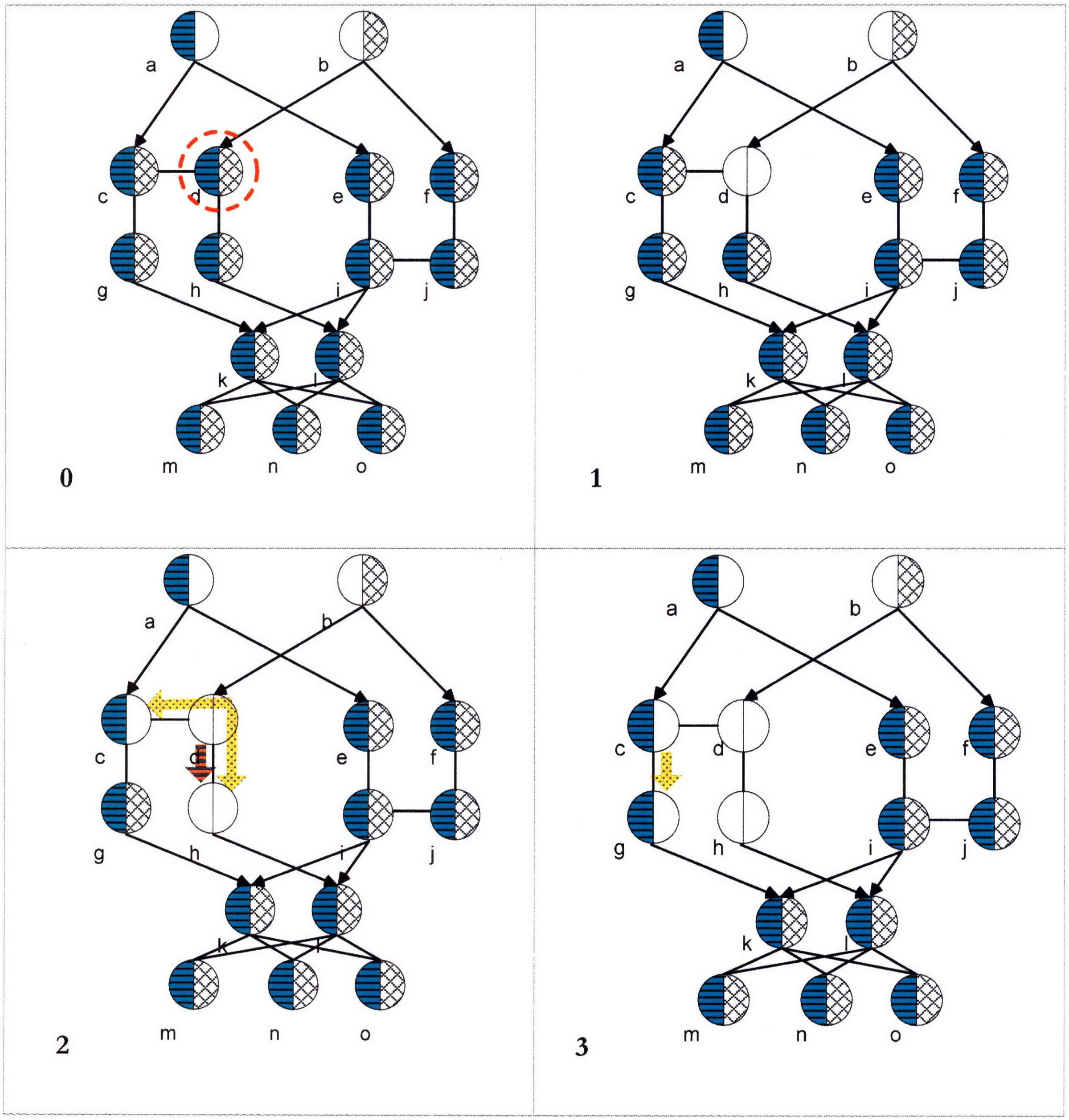


Figure 9 - Change propagation

Figure 9 presents a change causing node d to stop to working (0). Next, the node is no longer passing information (1). Later the effects are propagating to nodes c and h (2). Note that node c is only partially effected because of the connection to node a. Finally, node g is affected by node c (3), and the propagation stops. In that case, as a result of redundancy in the system, failure of node d did not change the ability of information to flow from sources a and b to destinations m, n and o. Calculating changes by propagation is computationally more efficient than calculating the entire state of the system (Zio et al, 2006).

Network Logical Configuration

As network is by its nature a shared resource, almost any practical implementation of computer networks contains logical configuration that aim to determine who can access what resources. The reasons for such mechanisms include security, scalability and efficiency. When any type of network resource is being accessed (by sending information from a source to a destination) it is important to know how is the source in order to accept or deny communication. An example to such a logical configuration was given earlier in the text, around a decision of airport authorities to accept plans coming from a certain airport or from a certain source. This is a policy than affect the behavior of the system although no physical element in the system changed (or broke).

There are many types of logical configurations, depending on the type of network and the type of communication protocol used. For the purpose of the thesis two relatively generic mechanisms where chosen. It is relatively simple to change or add new mechanisms to match a specific type of network. One of the mechanisms is "allow list": Each destination has a list of sources that it is allowed to get information from. This mechanism is equivalent of an airport authorities stating that flights originating from place X can not land in the airport. The second mechanism is "exposure list": Each resource can be exposed to the network through a number of entry points. While all entry point may be cabled into the network, the resource will selectively accept communication coming through certain entry points. This is equivalent of stating that only flight going through London can land in Boston airport and only flights going through Paris or Frankfurt can land in New York. This kind of configuration allows funneling all international flights into two groups – the one going to Boston (and thus must go through

London) and the ones going to New York (and must go through Paris or Frankfurt). By setting this configuration (and assuming that there are flights possible both to New York and Boston), the 'designer' of the system ensured redundancy with regards to flying into the US, and balances the loads through the incoming airports (London, Paris and Frankfurt). Without that definition, if all airlines chose to fly through London, apart from being highly overloaded, the London airport might have become a single point of failure.

Human Error and Equipment Failure

The study of human mistakes and errors is of high importance. With certain industries the stakes are very high and in many cases measured in lives. For example, in chemical and nuclear industries many accidents that caused loss of lives and injuries were investigated showing a wide range of human errors (Reason, 1990). As a result of very tangible human losses these industries developed high degree of awareness to human errors and a large amount of data on errors was collected. In the IT infrastructure and network management the stakes are different (more economic than safety oriented) and very little information can be found. In his book, *Human Error*, Reason makes the distinction between operators, designers, different levels of management and maintenance personnel (Reason, 1990, p 173) as different contributors of errors and eventually to an accident. It is interesting to note that in the networking management domain, a network engineer may play more than one of these roles. For example, the operator maintenance person and the designer of the network are almost the same person. Changing the network logical configuration is in many ways changing its design. Reason also claims that creating more technologically complex systems with internal protections and redundancy levels prevents people from revealing their mistakes immediately (Reason, 1990, p 179). Case studies also shows that latent errors in complex systems may remain undiscovered for long periods of time, sometimes years (Reason, 1990, p 189) and that issues left behind for a long period of time increase drastically the probability of a coincidence of multiple faults (p 180). This type of pattern is also true for IT infrastructures. The teams that manage the networks are getting smaller and they manage larger infrastructures (Wallin and Leijon, 2006). Under these conditions, any available resource is directed at "production capabilities" that is generating tangible revenue in the short term and not investing time on maintenance issues that are of mid-long term nature. This reinforces the information available from other industries, that latent issues would remain undiscovered.

Using this information, human error was defined as follows: The daily changes that are applied to IT infrastructure aim to add or remove a connection between a source and a destination. Assuming that the change was implemented successfully, all previous connections should work the as before, with an additional connection operational. Errors of the type lapse or slip would manifest themselves as a missing or incorrect action. As mistakes are more diverse, more potent and can take a very wide range of forms, the thesis does not try to model them (this will make an interesting direction for further research). Also, in order to allow simulation, an amount of tasks per week is assumed, and instead of actually changing the network, the change is assumed to be one that ends up with the infrastructure in the same state as before (eventually changing nothing) with one difference – in the process, the operator can have a lapse or a slip, that would appear as an incorrect configuration. It is assumed that a person would make a mistake with a constant of λ_H. If such an error has no effect on any of the services, it would remain latent. However, if a service fails as a result of the error, every possible reason for that failure (both the immediate cause and latent causes) is fixed within a given amount of time. The time to discover, and time to fix (both stochastic) are attributes of a service and per instance, the numbers that are used depend on the effected service.

Hardware failures are determined per component type to have different MTBF coefficients. The case study presented later in the text includes a set of coefficients that are typical for the hardware class. In general, it is very typical for equipment MTBF to be very long (tens of years). The combination of rare hardware fail events, the large amount of changes to the infrastructure and human errors, creates causal chains that lead to failures. Both hardware failures and human errors would eventually be found and fixed after a long time. For hardware this time is **MTTFix**$_C$ and for human errors **MTTFix**$_H$.

This type of modeling also give rise to examining interesting questions like: What is the effect of replacing a network element with certain **MTBF**$_{C1}$ with another component with **MTBF**$_{C2}$? What is the effect of replacing an element with a certain internal structure with a newer model with a better design? What is the implication of changing the mistake rate of the team (by training or getting people with a higher skill set)?

Importance Measures

While the traditional importance measures: $F_c^-(t)=F\left(t\,|\,\omega_c\left(\tau\right)=1,\tau\in\left[0,t\right]\right)$, and $F_c^+(t)=F\left(t\,|\,\omega_c\left(\tau\right)=0,\tau\in\left[0,t\right]\right)$ are common and are used as a base for other measures such as Risk achievement worth, Risk reduction worth, Fussell-Vesely measure and Birnbaum measure, they have two issues: a. they do not related well to logical configuration change (there is no meaning to attempt and make a logical configuration change 'more robust' to avoid a failure), and b. they do not reflect the importance of the component of configuration in terms of business impact. Furthermore it can be argued that there is no single measure, as different stakeholders may have a different subset of business services that are interesting for them. This thesis suggests another importance measure. Every time a component fail-event or a logical configuration change-event takes place, and it is a cause for a service failure (either immediate or latent), it is assigned with a weight to reflect its contribution to the failure (for example, if two redundant power supplies failed inside an element, each one would have 50% contribution) and with the amount of actual loss created by the failure of this services. Later, a weighted average of expected loss related to the specific event (and a subset of available services) can be generated to reflect the expected business impact.

There are a finite number of possible events. For each failure the contribution of each event is defined as follows:

Let E be a set of all possible events,

$\mathit{timeline}(t)=\{e_1,e_2,\cdots,e_q\},e_u\in E$ - oredered set with possible repeats
$\forall a,e_a\in \mathit{timeline}(t)$, let $\mathit{time}(a,t)$ be defined as the time of e_a in timeline t
$\forall a,b\,|\,e_a\in \mathit{timeline}(t),e_b\in \mathit{timeline}(t),b>a\Rightarrow \mathit{time}(b,t)>\mathit{time}(a,t)$

let i: $i\in\{1..N\}$, event number, $e_i\in \mathit{timeline}(t)$
let S be set of all services
let j: $j\in\{1..M\}$, service number, $s_j\in S$

$$C_{i,j}=\begin{bmatrix} c_{11} & \cdots & c_{1m} \\ \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_{nm} \end{bmatrix},\forall j\in\{1..M\}\Rightarrow\sum_{i=1,N}c_{i.j}=1$$

For each service, the fixed cost of repair and the variable cost of repair are defined as:

$j \in \{1..M\}$ - service number

$$RF_j = \begin{bmatrix} rf_1 \\ rf_2 \\ \vdots \\ rf_m \end{bmatrix}, RV_j = \begin{bmatrix} rv_1 \\ rv_2 \\ \vdots \\ rv_m \end{bmatrix}$$

Units of: rf_j - Currency

Units of: rv_j - $\dfrac{\text{Currency}}{\text{Hour}}$

The fixed cost and the variable cost are defined in order to allow different types of cases. Some service interruptions may be very simple to fix (for example replacing a cable) and thus not expensive in terms of fixed cost. At the same time, these interruptions may be extremely expensive for each hour passing. Other types of interruptions may be very expensive to fix (replacing an expensive part) while their variable cost is low.

Given an interruption of service of *h* hours, the impact can be found as follows:

let $t \in \{1..T\}$ - timeline number

let $A(t)$ - number of service interruptions in timeline t

$a \in \{1..A(t)\}$ - service interruption number

$C(a,t)$ - contribution matrix for a specific failure a in timeline t

$$IM_j(a,t) = C(a,t) \cdot RF + C(a,t) \cdot RV \cdot h =$$

$$IM_j(a,t) = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdot & c_{1,m} \\ c_{2,1} & c_{2,2} & & \\ \cdot & & & \\ c_{n,1} & & & c_{n,m} \end{bmatrix} \cdot \begin{bmatrix} rf_1 \\ rf_2 \\ \\ rf_m \end{bmatrix} + \begin{bmatrix} c_{1,1} & c_{1,2} & \cdot & c_{1,m} \\ c_{2,1} & c_{2,2} & & \\ \cdot & & & \\ c_{n,1} & & & c_{n,m} \end{bmatrix} \cdot \begin{bmatrix} rv_1 \\ rv_2 \\ \\ rv_m \end{bmatrix} \cdot h$$

$$IM_j(a,t) = \begin{bmatrix} c_{1,1} \cdot rf_1 + c_{1,2} \cdot rf_1 + \ldots + c_{1,m} \cdot rf_m + \left(c_{1,1} \cdot rv_1 + c_{1,2} \cdot rv_1 + \ldots + c_{1,m} \cdot rv_m\right) \cdot h \\ \cdot \\ \cdot \\ c_{n,1} \cdot rf_1 + c_{n,2} \cdot rf_1 + \ldots + c_{n,m} \cdot rf_m + \left(c_{n,1} \cdot rv_1 + c_{n,2} \cdot rv_1 + \ldots + c_{n,m} \cdot rv_m\right) \cdot h \end{bmatrix}$$

Finding the importance measures is defined as follows:

$A(t)$ - The number of failures in timeline t

$$IM_j = \frac{1}{T}\sum_{t=1}^{T}\sum_{a=1}^{A(t)} IM_j(a,t) = \begin{bmatrix} im_1 \\ \vdots \\ im_m \end{bmatrix}$$

If only a subset of the services is of interest, we can define a new contribution matrix:

$i \in \{1..N\}$ - event number

$j \in \{1..M\}$ - service number

$$Si_k = \{s_1 \quad \cdots \quad s_m\}, \forall k \in \{1..M\}, s_k \in S \Rightarrow \begin{cases} s_k = 1, & \text{service k is of interest} \\ s_k = 0, & \text{service k is of no interest} \end{cases}$$

$$Cs = Si \cdot C$$

$$IMS_j = Cs \cdot RF + Cs \cdot RV \cdot h$$

$$IMS_j = \frac{1}{T}\sum_{t=1}^{T}\sum_{a=1}^{A(t)} IMS_j(a,t) = \begin{bmatrix} ims_1 \\ \vdots \\ ims_m \end{bmatrix}$$

The importance measures:

$F_c^-(t) = F\left(t \mid \omega_c(\tau) = 1, \tau \in [0,t]\right)$, and $F_c^+(t) = F\left(t \mid \omega_c(\tau) = 0, \tau \in [0,t]\right)$ are not found in the process described above. In their paper (Zio et al, 2006), Zio et al discusses that for some cases certain measurements can be difficult to estimate because of a low amount of data points. This is true in the context of this thesis with regards to $F_c^-(t)$ and $F_c^+(t)$. However, if required, these values can be found specifically by creating a dedicated set of simulations for each event. However, considering that this can be done for each of the possible events, finding these measures for all events may require substantial time and computing resources.

Checking CA and MC vs. PRA

In their paper (Zio et al, 2006), the authors perform a simple test to validate that their new model works as expected and produces similar results to PRA for a small test case. After implementing and enhancing the model as described above, a similar test was performed. As the extension of the model included fault trees, the example that was chosen is a case of two fans supporting a network element. Each fan has mean time between failure (MTBF) of 13 years. The PRA expression of the Cumulative Distribution Function (CDF) is:

$$P\left(F_1,F_2\right)=\left(1-e^{\frac{-t}{\lambda_F}}\right)^2$$

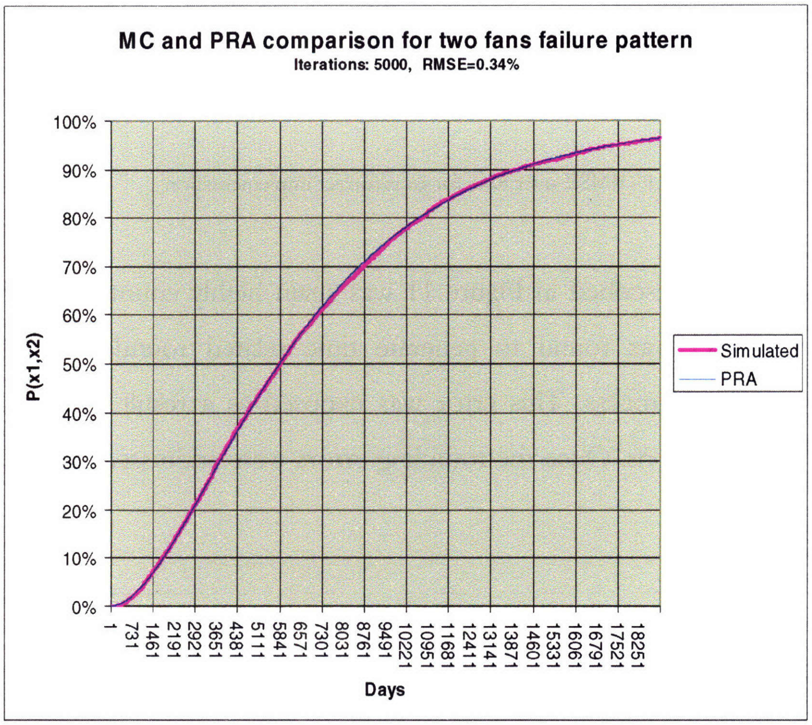


Figure 10 - MC and PRA simple comparison test case

Although the match between PRA results and the simulated results looked promising initially, the Root Mean Square Error (RMSE) found in Figure 10 presented a lower bounded behavior when increasing the amount of iterations. As this result was unexpected, more tests were needed to further verify why the average RMSE is bounded. Two more tests were performed: a. the RMSE over time was expected to be found similar to a random walk; b. the relationship between RMSE and the number of iterations was checked. When testing the RMSE over time the initial results were as follows:

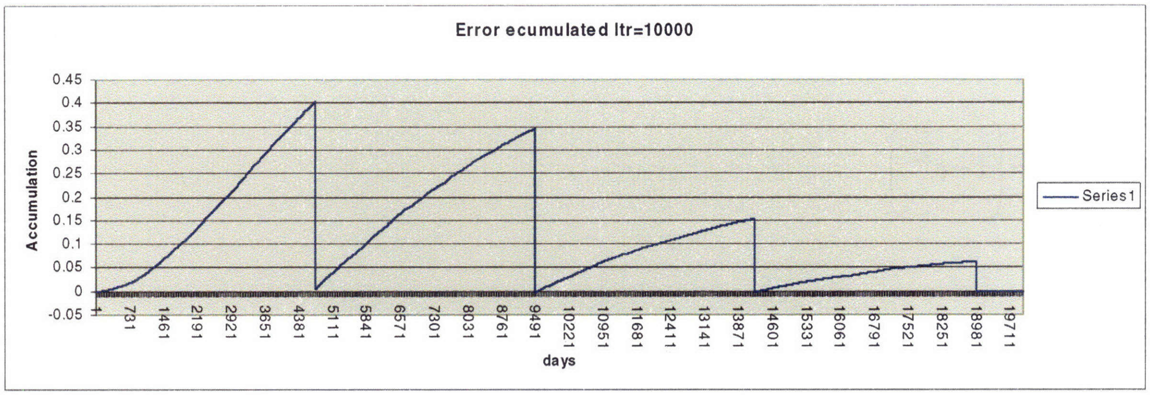
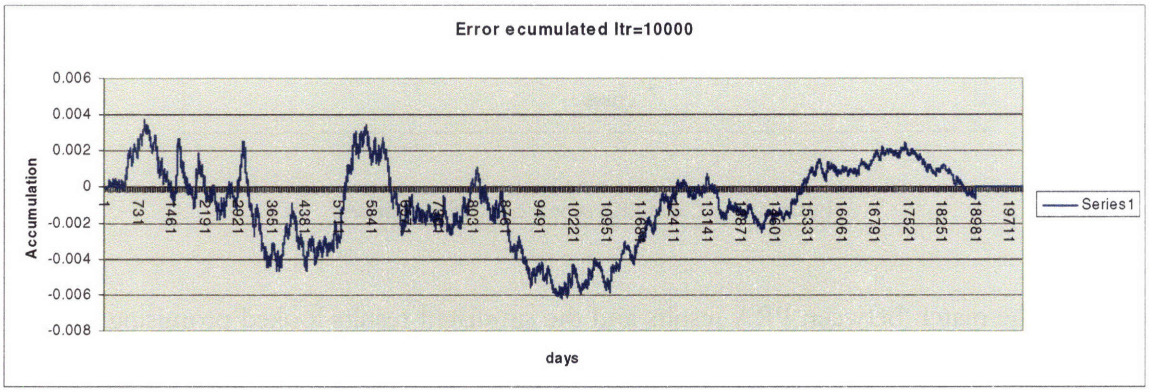


Figure 11 - RMSE over time for an incorrect implementation

The RMSE distribution described in Figure 11 was again highly counter intuitive. In a close observation, the model was found to generate time related rounding errors during the mathematical simulation process. This error was exposed as a result of a large number of iterations error accumulations. Once the rounding errors were addressed, the following results were found:



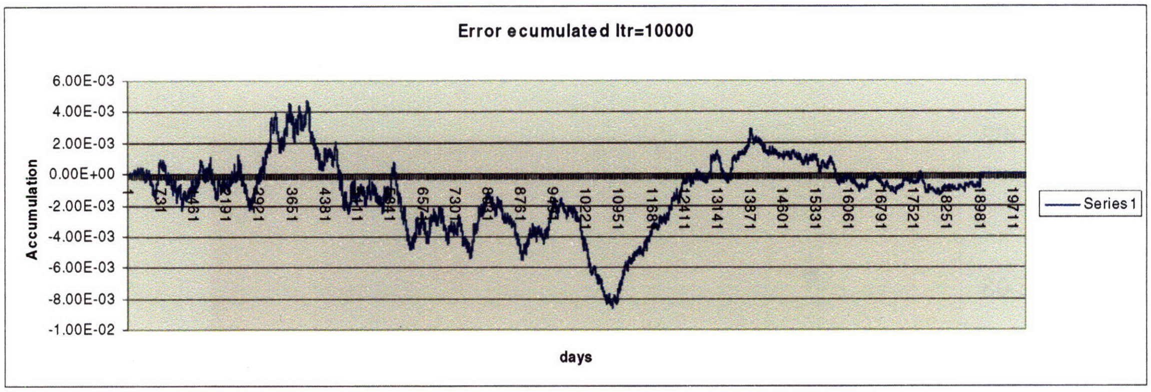


Figure 12 - RMSE over time for a correct implementation

As originally expected, the results in Figure 12 presented the qualities of a random walk. This test was performed many times and inspected visually to ensure that there are no noticeable patterns or artifacts. The two results in Figure 12 are a subset of these tests.

Next, the relationship between iterations and error was tested:

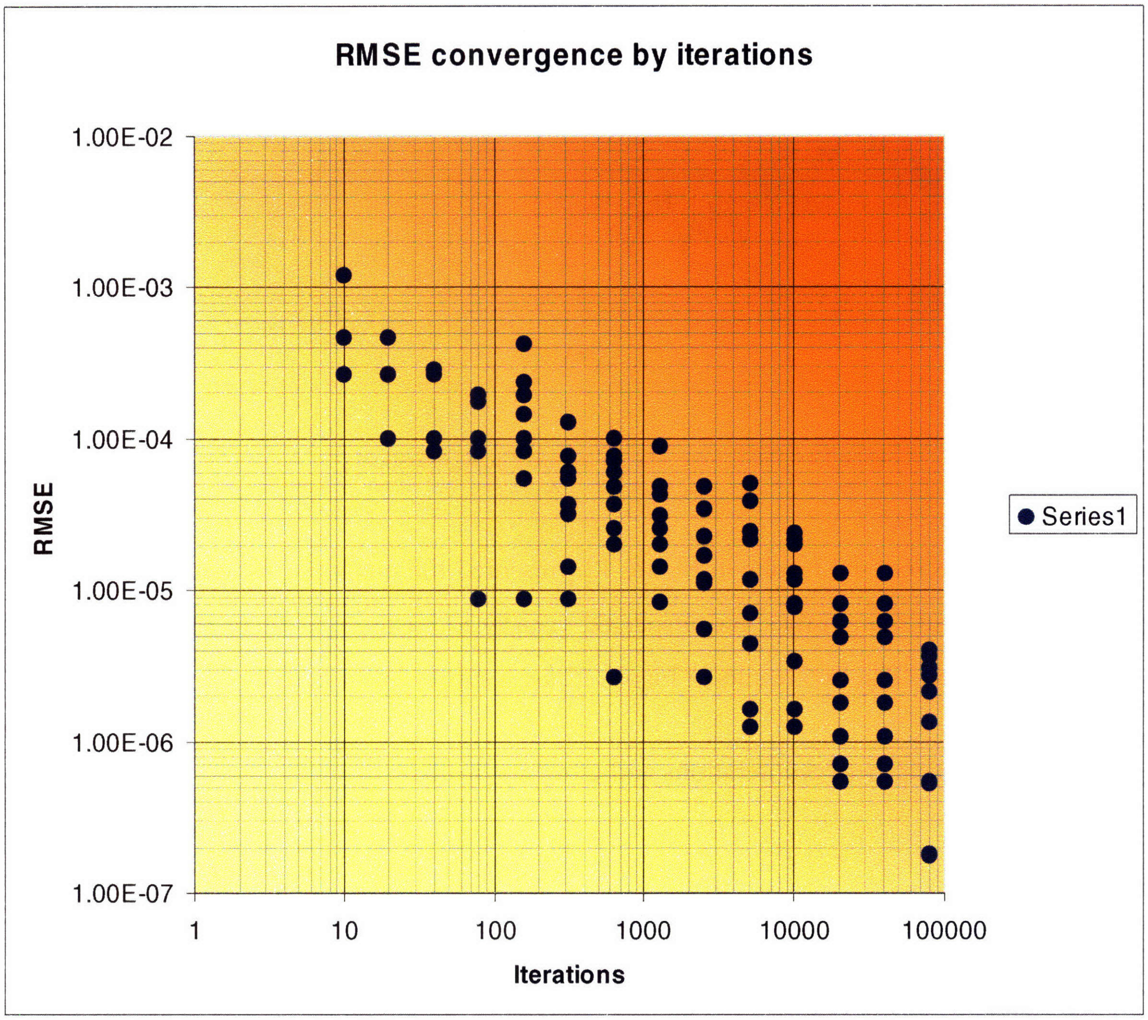


Figure 13 - RMSE as a function of MC iterations

Figure 13 shows a relationship between the number of iterations and the error. Each vertical batch of points represents 10 runs with the specific number of iterations. The iterations numbers started at 10 and moved up at x2 increments to 80192.

Cost Benefit Analysis

As described above in the importance measures section, the having the service(s) participate in the risk model allows estimating what is the monetary value of each component and configuration. As a part of the simulation, it also allows simulation of losses per each timeline. The accumulated losses have the behavior of a random walk that is bounded by the X axis, by definition. Averaging the accumulated losses at a point in time would provide the expected loss at that point in time for a given configuration. The set of simulated losses at a point in time also allow understanding what the probability density function of losses is at that time. Both these finding are presented later as a part of the case study results.

The ability to simulate future losses at a given future time, for different architectures, enables performing a cost benefit analysis. For example, if an equipment vendor has a new type of network element with superior attributes such as better internal redundancy or better MBTF profile, and the price of replacing is X, is it a good idea to replace a subset of network elements to the new model? Which subset would be optimal? A different example: if a network has a failure that is not effecting any of the services (a loss of redundancy layer), and the cost of fixing is Y, should it be fixed?

In order to answer these types of questions, the benefit should be estimated. Assuming a comparison between two architectures, *A1* and *A2*:

Let $L_1(t)$ be the losses of *A1* at future time t, and $L_2(t)$ losses of *A2* at future time t. Both are random variables with unknown distributions.

$$L_1(t) \sim D_{A1}(t)$$
$$L_2(t) \sim D_{A2}(t)$$

The ROI period should be determined. While is some industries that may be years, in IT it is not unheard of to look for an ROI of less than a year. In these conditions NPV may not even be required. However, if an NPV is required, the discount rate should be provided. Let *c* be the cost of moving from *A1* to *A2*. The benefit *b* can be expressed as follows:

$$b = NPV(L_2(t) - L_1(t)) - c$$

Correlation: As *b* is a function of random variables, *b* is also a random variable. A very important point to determine is the correlation of *L1, L2,* the two loss random variables. Correlation coefficient is defined as:

$$\rho_{L_1,L_2} = \frac{E\left(\left(L_1 - \mu_{L_1}\right)\left(L_2 - \mu_{L_2}\right)\right)}{\sigma_{L_1}\sigma_{L_2}}$$

The importance of determining the correlation is that if the two have some correlation, there will be an effect on the distribution of *b*, and as a result an effect on the confidence factors of profit and loss levels. Furthermore, since the distributions of *L1, L2* are unknown, it is reasonable to use empirical result data, construct the distributions and then find *b* using MC. However, using a simulation without understanding the relationship between the random variables may lead to incorrect results.

The two loss random variables *L1* and *L2* may be related because of the following reasons. If the architectures *A1* and *A2* are very close, one could argue that all the equipment is the same (or almost the same) in both architectures. When projecting into the future, arguably, the same equipment failures would happen in a timeline of the first architecture as well as the second. In other words – the failures are 'pre-destined' to happen in both timelines.

However, some failures are a result of operating conditions. While the concept that failures are 'pre-destined' to happen in both timeline in the same way may be true for projections to the near future, after longer periods of time the failure causes will start diverging. Also, the main contributor to failures is people changing the infrastructure (intentionally or not), not equipment failure (see – literature review). It is hard to say how human error (mistakes, lapses and slips) will correlate between two timelines. However, with even a very small difference between architectures *A1* and *A2*, and human errors depending on the architectures, there will be slightly different procedures for people to implement. As opposing to equipment that is 'destined' to fail at a certain time, people are likely to make the 'pre-destined' mistakes in different procedures, leading to very different loss results and therefore likely to diverge quickly. Finally, let's consider what is the impact of no correlation? The impact would be that the distribution of *b* has a larger variance meaning that the benefit bears more uncertainty. Given the above reasoning, this thesis uses the conservative assumption that *L1, L2* are not correlated.

$$\rho_{L_1,L_2} = 0$$

In order to estimate the benefit *b*, two factors are important: the expected benefit and the variance. As *L1* and *L2* are by assumption not correlated, there is a probability of improving an architecture and ending up with a loss. It is possible to estimate the level of confidence as follows:

$E(b)$ - expected benefit
n - number of periods
d - discount rate

$$P(b<0)=P\left(NPV(n,d,L_2(t)-L_1(t))<c\right)=$$

$$P\left((1+d)^{-n}\left(L_2(t)-L_1(t)\right)<c\right)=$$

$$P\left(\left(L_2(t)-L_1(t)\right)<c(1+d)^n\right)=$$

$$=\int_{x=-\infty}^{\infty}P\left(l_2<x\right)\cdot P\left(l_1>c(1+d)^n+x\right)\cdot dx=$$

$$P(b<0)=\int_{x=-\infty}^{\infty}\left(\int_{u=-\infty}^{x}P\left(l_2=u\right)\cdot du\cdot\int_{v=c(1+d)^n+x}^{\infty}P\left(l_1=v\right)\cdot dv\right)\cdot dx$$

Or expressed in words: the probability of a loss is the sum probability of all combinations where the discounted profit (L2 less L1) is smaller than the cost of making the change from architecture *A1* to architecture *A2*.

Epistemic System Model

The model suggested in this thesis is affected by two types of stochastic models. The first is component failure, and the second is human error. In order to use the model, information has to be found from component equipment vendors and assumptions should be made about the nature and profile of human error. However, an organization with a large infrastructure in place can use these as base assumptions only and adjust them over time. Given enough equipment, the information on component failure can be collected and fed back into the component profiles and change them. The advantage of that approach is that information can be fed back even if there were no failures (Apostolakis, 2006) and (Helton and Burmaster, 1996)

The risk profile can be adjusted by taking every period of time the results for all components that are identical, in terms of number of failures.

$\pi(\lambda)$ is the priory profile (PDF).

$L(E \mid \lambda)$ is the likelihood function of evidence E to occur given the assumed $\pi(\lambda)$.

$\pi'(\lambda \mid E)$ is the posterior profile (PDF) given evidence E.

Then the posterior distribution is given by: $\pi'(\lambda \mid E) = \dfrac{L(E \mid \lambda) \cdot \pi(\lambda)}{\int L(E \mid \lambda) \cdot \pi(\lambda) d\lambda}$

The same model can be used to assess human error profile. However, care should be taken to categorize the events well. For components that would mean choosing exactly identical components model, and for human error, defining similar environmental situations that would lead to a mistake, slip or lapse. For example, if a certain job requires two operations sequence vs. another task that requires fifteen actions in an accurate sequence are clearly different in terms of the potential for a lapse. It is outside the scope of this thesis to examine further how an epistemic model can be successfully implemented. This can be an interesting subject for further research.

Common-Cause Failures

Within an IT environment there are many factors that can lead to common cause. One of the assumptions taken here is that the network elements have a binary failure mode. However, the majority of devices today used in a networked environment are driven internally be a CPU, a Microcontroller or a PLA (Programmable Logic Array, such as FPGA from Xilinx). The software that drives these devices is known as 'firmware'. The firmware is the operating-system software of the network element and can be updated from time to time as new software revisions become available. An important common cause is the firmware version. The tendency of network managers is to upgrade all devices to the latest firmware version, once the vendor recommends doing so. However, in the rare event of a faulty firmware, the fact that many network elements are using the same faulty software version may lead to multiple failures spanning multiple geographic locations. That situation may render the redundant infrastructure design to a meaningless effort.

Another source of common causes is physical proximity. As observed in recent natural disasters and terrorist attacks, an installation such as a data center can become inactive as a whole. On a smaller scale, an event such as fire or a local flood may lead to failure of a number of devices (or cables) depending on their physical proximity to the event source.

The exact weight of common cause failure is not known yet, and best practices claim that about 10% of failures will originate from common cause. However, the common causes that are known can be added into an enhanced model. For example, if the physical location coordinates is added for each CA (in that case cables may have to be broken down to a number of CAs), best estimates for localized or large scale disasters can be modeled. Since the system failure mode can be very different in a common case situation, as a result of multiple failures, this type of modeling may be of interest to assess readiness of infrastructure for such events. It is outside the scope of this thesis to examine further how common causes can be modeled. This can be an interesting subject for further research.

CASE STUDY

Description

Figure 14 shows a simple network configuration chosen as a case study:

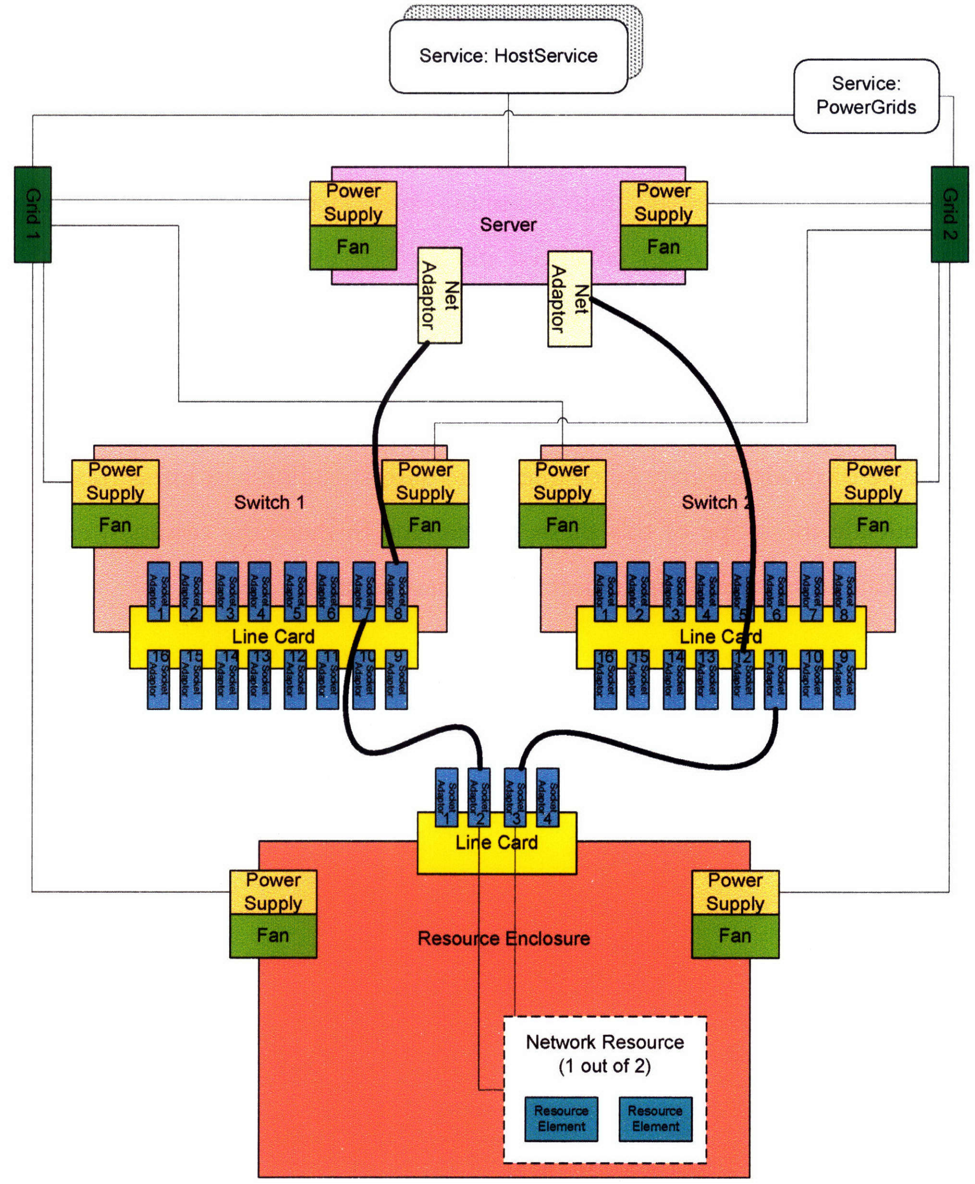


Figure 14 - Case study base configuration

The main service HostService represents a business process that uses one physical server. Every one of the blocks marked with a continuous square line represents a physical entity that has stochastic failure characteristics (see Case Parameters for the actual chosen numbers). The wide lines represent cables that allow transmission of network information. The dashed line square (network resource) represents an abstract entity that is provided to users. The network resource is composed of Resource Elements that are a physical entity.

The case contains two services marked with a rounded square line: HostService, which represents a business process that is using the server (for example, an application) and PowerGrids service. The HostService has a price associated with it, while PowerGrids does not. In contrast to all the endogenous elements that by assumption remain broken until an interruption of service, the exogenous item such as all types of utilities (specifically grid) are promptly fixed even if they did not lead to an interruption of service. The sole purpose of PowerGrids service is to ensure that the model will limit the length of external grid failure to a limited amount of time.

The intention of the case is to show the impact of a change. There are three changes chosen, presented in Figure 15, Figure 16 and Figure 17.

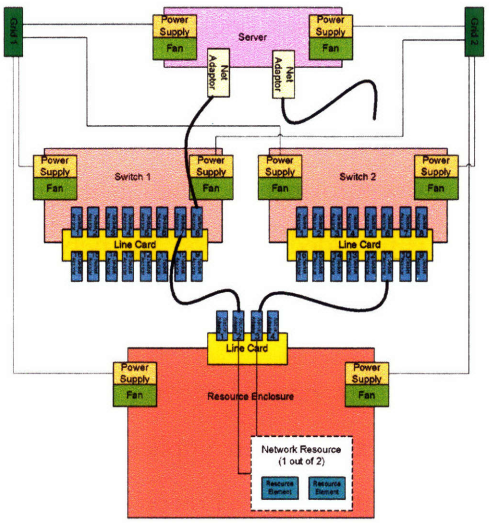


Figure 15 - Case study first change – cable disconnect

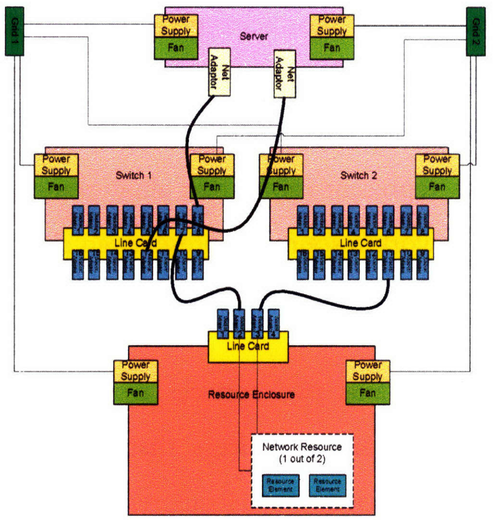


Figure 16 - Case study second change - incorrect server connection

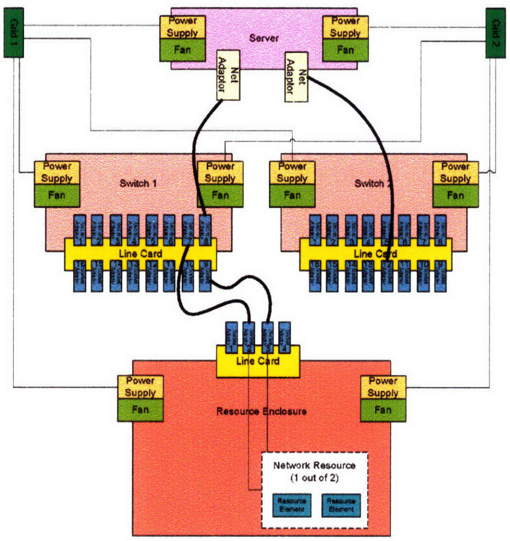


Figure 17 - Case study third change - resource enclosure incorrect connection

Case Parameters

Note: some of the following parameters were chosen after as a result of information from vendors on certain type of equipment, while others were assumed.

Table 1 - Case study, component assumptions

Component	MTBF (years)	Mean Time to Fix – MTTFix$_C$ (days)
SERVER	20	180
NET_ADAPTOR	650	365
CABLE	1000	730
SWITCH	15	180
LINE_CARD	14	1095
SOCKET_ADAPTOR	100	1095
RESOURCE_ENCLOSURE	20	180
RESOURCE_ELEMENT	45	14
FAN	13	1095
GRID	20	1
POWER_SUPPLY	10	1095

The **Mean Time to Fix, MTTFix$_C$** is the time that an element would be fixed even if it caused no interruption of service. The reason that this number varies across the different components is because some hardware components are tooled and monitored. For example with some network configurations, the 'resource element' may be a disk that can flag incorrect working conditions. In such situations where a component has failed, it would be discovered quickly. In other situations such as a failure of a 'line card', that is not tooled to report its condition, would be discovered after a long time.

The following assumptions were used for services:

Table 2 - Case study, service assumptions

Service Name	Time to discover issue	Time to fix issue	Fixed cost of failure	Variable cost of failure
HostService	Gamma distribution $K = 5$ $\phi = 1$	Gamma distribution $K = 1$ $\phi = 1$	1000($)	5000 ($/H)
PowerGrids	Gamma distribution $K = 5$ $\phi = 1$	Gamma distribution $K = 1$ $\phi = 1$	0	0

As mentioned earlier, the HostService represents an important business process that has tangible cost implications to a loss of service. The PowerGrids service was used to represent the power grid utilities. There is no direct cost associated with a failure of one of the power grids, although the indirect implications can lead to a failure of other services and indirect cost. However, it is assumed that utilities are a 'monitored' service and an interruption of a utility service would be fixed promptly (as opposing to internal issues that would be fixed only if they lead to a noticeable service interruption). Gamma distribution was used because of a better control possible in the distribution shape over lognormal distribution, and the fact that it is bounded on the low side and unbounded on the high side, a shape that lends itself well to reactive events (time to find an issue - after the issue started or time to fix a problem – after fix procedures started).

It was assumed that in the case study network there are ten weekly changes to the network. Lapses and Slips (Reason, 1990) are assumed to have Mean number of Changes Between Mistakes (MCBM) of: $MCBM_L$=200 for lapses and $MCBM_S$=200 for slips, with relation to the amount of changes applied to the infrastructure.

Logical Configuration

The resource enclosure in the case contains three network resources marked V1, V2, and V3. Each one of the resources contains two types of configurations: Port Exposure list and Allow List. The port exposure list is a configuration that sets through which resource enclosure ports it is possible to access the network resource. In Figure 18 it is possible to see how all three resources allow access through the same two ports (with IDs 30000002 and 40000003) marked by the dashed lines. This allows having only two cables go into the network while maintaining redundancy. Obviously, it is possible to use all four available ports to enhance the redundancy further.

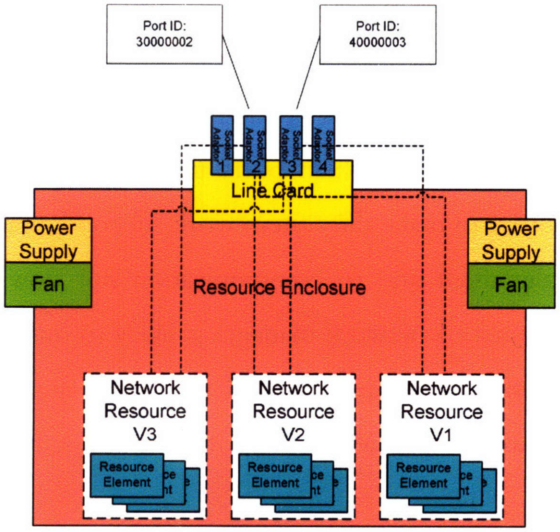


Figure 18 - Logical Configuration, Exposure List

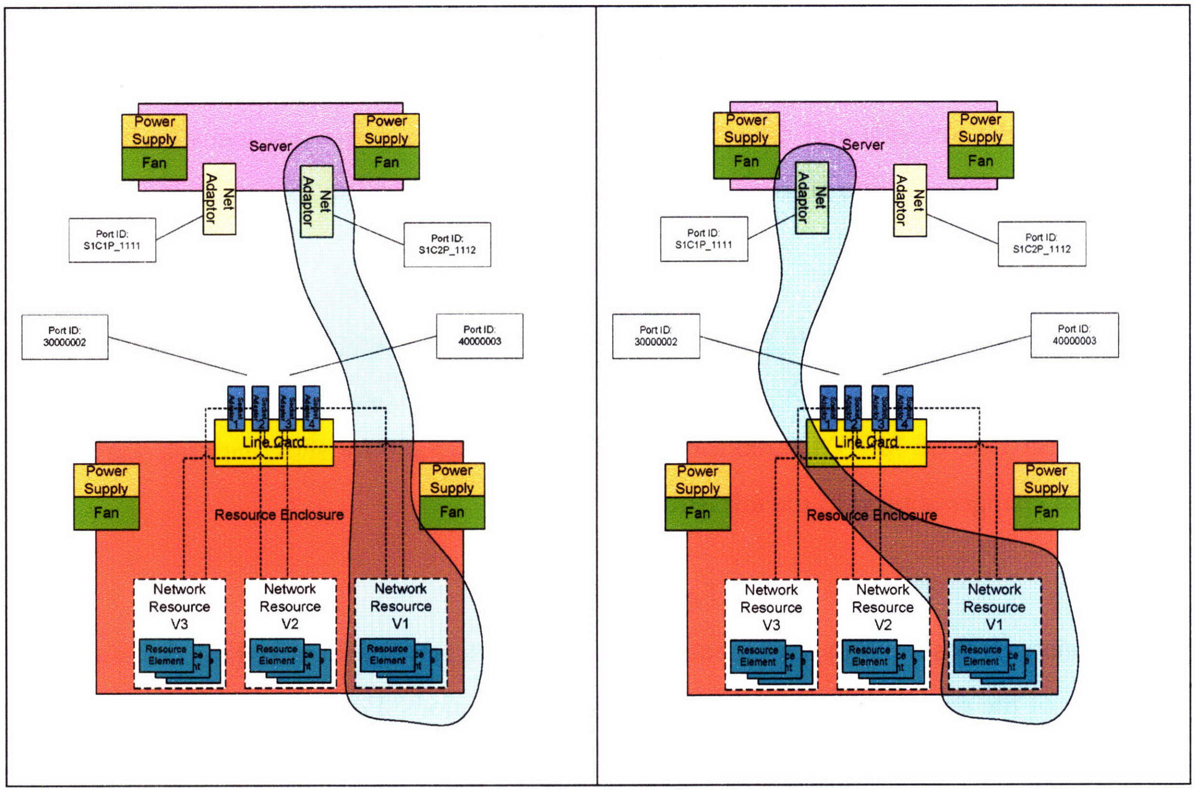


Figure 19 - Allow list configuration

Figure 19 shows the logical effect of allow list set for V1. Each one of the elements V1, V2, V3 is c0onfigured to allow communication from two source network cards, SC1P1_1111 and SC1P2_1112. Regardless of what is physically in between these two points, information coming from these sources would be accepted.

The two security mechanisms (allow list and port exposure) are relatively simple. However, even with these two mechanisms, there is apparent complexity built into the network. A network manager can decide through which ports the communication would pass and to which destination, but in order to make the system work, all configurations must be aligned. A change may lead to misalignment and to communication not flowing. However, with the redundancy built into the system, a network manager is likely to miss the error as the network will continue to work. Human errors are assumed to remain latent if the do not cause an interruption of service. However after $\mathbf{MTTFix}_H$ of 18 months, the errors would be found and fixed.

Results

Timeline

The following table represents a single future timeline. Note how some events lead to service failures (red), fix attempts (yellow) and fixes (green). Also note that events that led to a failure are fixed at the same timeframe of the service fix. Events that are marked with "from:/X/ to://" are lapses or slips of operators during logical configuration changes.

This is a single timeline that spans ten years of simulation. The case study is comprised of 5,000 simulations done for each one of the sub cases, a total of 20,000 timelines. The results were stored in a database (MySQL, 2007) and the reports were produced using a combination of the open source reporting solution from Pentaho and JReport (Pentaho, 2007) as well as spreadsheet software.

Examples in the timeline:

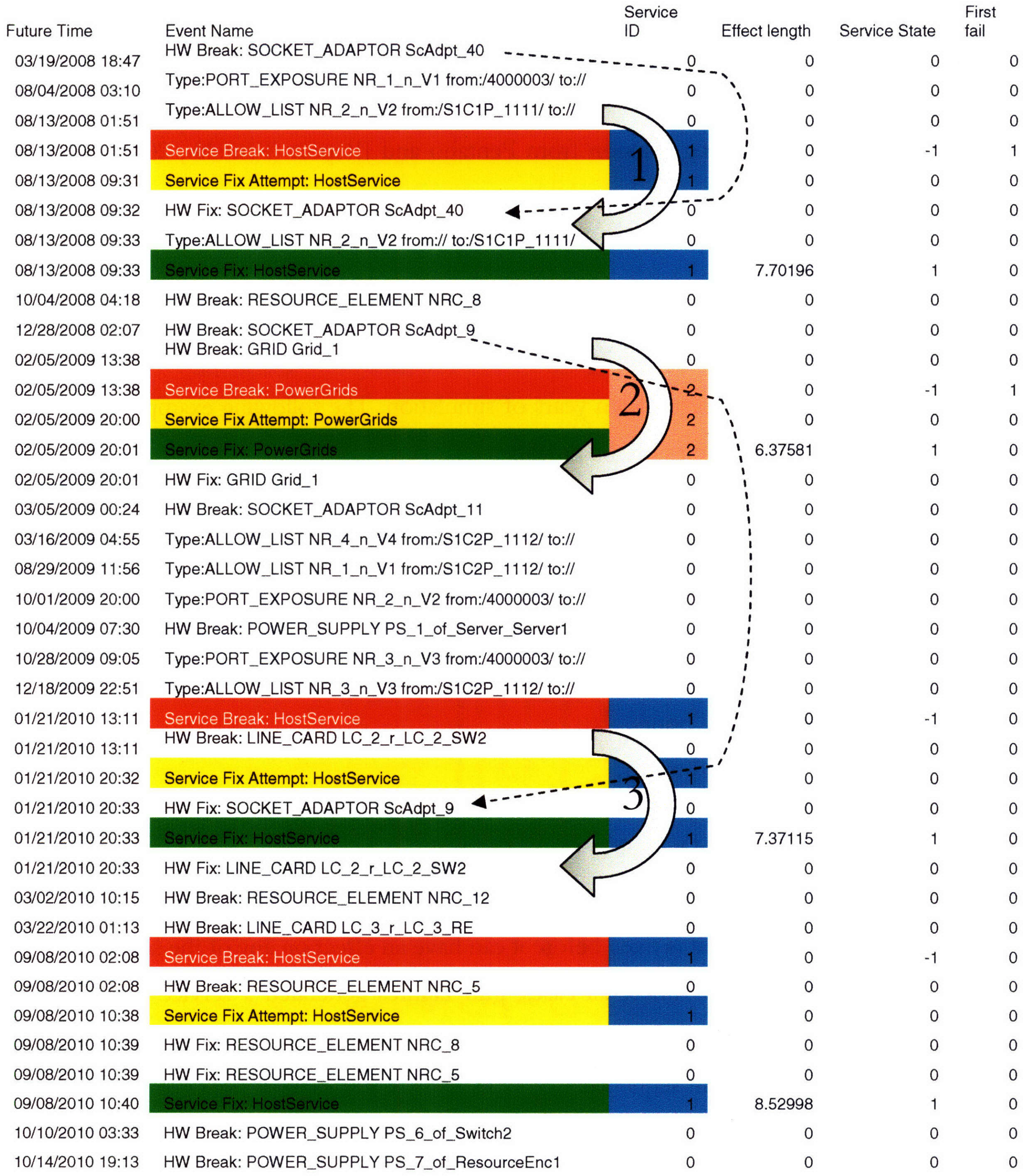
The following timeline is a typical timeline for the original configuration and contains over twenty failures over the course of ten years of simulation. The following examples illustrate a number of interesting examples on the timeline.

1. On date 03/19/2008 18:47, a socket adaptor fails. As it is not causing any immediate service failure, it remains latent until 08/04/2008 01:51 where an incorrect logical configuration change (as a result of a lapse or a slip), caused the HostService to fail. As a part of the fix procedure, both issues were fixed at 08/13/2008 09:33. Note that the effect length was 7.7 hours.

2. On date 02/05/2009 13:38, one of the electricity grid – Grid_1 failed. That leads to the failure of the grid service. Once fixed, the service is operational again after 6.37 hours. As a result of the fact that exogenous utilities are considered 'monitored' in the thesis model, the grid is fixed before it can have an effect on any of the services. Under these conditions, in almost all cases, grid failures generated a service failure only if there was a latent condition existing in the infrastructure. This timeline does not contain such an event sequence.

3. On date 01/21/2010 13:11, Line card of switch2 failed. As a result of a latent failure of socket adaptor 9 on 12/28/2008 02:07, the line card failure lead to HostService to fail. The two issues are fixed at 01/21/2010 20:33, after 7.37 hours.

Table 3 - A single future timeline



Future Time	Event Name	Service ID	Effect length	Service State	First fail
03/19/2008 18:47	HW Break: SOCKET_ADAPTOR ScAdpt_40	0	0	0	0
08/04/2008 03:10	Type:PORT_EXPOSURE NR_1_n_V1 from:/4000003/ to://	0	0	0	0
08/13/2008 01:51	Type:ALLOW_LIST NR_2_n_V2 from:/S1C1P_1111/ to://	0	0	0	0
08/13/2008 01:51	Service Break: HostService	1	0	-1	1
08/13/2008 09:31	Service Fix Attempt: HostService	1	0	0	0
08/13/2008 09:32	HW Fix: SOCKET_ADAPTOR ScAdpt_40	0	0	0	0
08/13/2008 09:33	Type:ALLOW_LIST NR_2_n_V2 from:// to:/S1C1P_1111/	0	0	0	0
08/13/2008 09:33	Service Fix: HostService	1	7.70196	1	0
10/04/2008 04:18	HW Break: RESOURCE_ELEMENT NRC_8	0	0	0	0
12/28/2008 02:07	HW Break: SOCKET_ADAPTOR ScAdpt_9	0	0	0	0
02/05/2009 13:38	HW Break: GRID Grid_1	0	0	0	0
02/05/2009 13:38	Service Break: PowerGrids	2	0	-1	1
02/05/2009 20:00	Service Fix Attempt: PowerGrids	2	0	0	0
02/05/2009 20:01	Service Fix: PowerGrids	2	6.37581	1	0
02/05/2009 20:01	HW Fix: GRID Grid_1	0	0	0	0
03/05/2009 00:24	HW Break: SOCKET_ADAPTOR ScAdpt_11	0	0	0	0
03/16/2009 04:55	Type:ALLOW_LIST NR_4_n_V4 from:/S1C2P_1112/ to://	0	0	0	0
08/29/2009 11:56	Type:ALLOW_LIST NR_1_n_V1 from:/S1C2P_1112/ to://	0	0	0	0
10/01/2009 20:00	Type:PORT_EXPOSURE NR_2_n_V2 from:/4000003/ to://	0	0	0	0
10/04/2009 07:30	HW Break: POWER_SUPPLY PS_1_of_Server_Server1	0	0	0	0
10/28/2009 09:05	Type:PORT_EXPOSURE NR_3_n_V3 from:/4000003/ to://	0	0	0	0
12/18/2009 22:51	Type:ALLOW_LIST NR_3_n_V3 from:/S1C2P_1112/ to://	0	0	0	0
01/21/2010 13:11	Service Break: HostService	1	0	-1	0
01/21/2010 13:11	HW Break: LINE_CARD LC_2_r_LC_2_SW2	0	0	0	0
01/21/2010 20:32	Service Fix Attempt: HostService	1	0	0	0
01/21/2010 20:33	HW Fix: SOCKET_ADAPTOR ScAdpt_9	0	0	0	0
01/21/2010 20:33	Service Fix: HostService	1	7.37115	1	0
01/21/2010 20:33	HW Fix: LINE_CARD LC_2_r_LC_2_SW2	0	0	0	0
03/02/2010 10:15	HW Break: RESOURCE_ELEMENT NRC_12	0	0	0	0
03/22/2010 01:13	HW Break: LINE_CARD LC_3_r_LC_3_RE	0	0	0	0
09/08/2010 02:08	Service Break: HostService	1	0	-1	0
09/08/2010 02:08	HW Break: RESOURCE_ELEMENT NRC_5	0	0	0	0
09/08/2010 10:38	Service Fix Attempt: HostService	1	0	0	0
09/08/2010 10:39	HW Fix: RESOURCE_ELEMENT NRC_8	0	0	0	0
09/08/2010 10:39	HW Fix: RESOURCE_ELEMENT NRC_5	0	0	0	0
09/08/2010 10:40	Service Fix: HostService	1	8.52998	1	0
10/10/2010 03:33	HW Break: POWER_SUPPLY PS_6_of_Switch2	0	0	0	0
10/14/2010 19:13	HW Break: POWER_SUPPLY PS_7_of_ResourceEnc1	0	0	0	0

10/15/2010 19:21	HW Break: FAN FAN_8ResourceEnc1	0	0	0	0
04/17/2011 19:17	Type:PORT_EXPOSURE NR_1_n_V1 from:/3000002/ to://	0	0	0	0
04/21/2011 06:48	Type:ALLOW_LIST NR_2_n_V2 from:/S1C2P_1112/ to://	0	0	0	0
07/10/2011 23:08	HW Break: SWITCH Switch1	0	0	0	0
01/14/2012 00:26	Type:PORT_EXPOSURE NR_2_n_V2 from:/3000002/ to://	0	0	0	0
04/10/2012 08:14	HW Break: POWER_SUPPLY PS_5_of_Switch2	0	0	0	0
04/24/2012 16:35	Type:ALLOW_LIST NR_4_n_V4 from:/S1C1P_1111/ to://	0	0	0	0
08/20/2012 10:01	Type:PORT_EXPOSURE NR_3_n_V3 from:/3000002/ to://	0	0	0	0
09/02/2012 23:57	HW Break: RESOURCE_ELEMENT NRC_14	0	0	0	0
10/25/2012 22:51	Type:ALLOW_LIST NR_1_n_V1 from:/S1C1P_1111/ to://	0	0	0	0
10/28/2012 11:27	Service Break: HostService	1	0	-1	0
10/28/2012 11:27	HW Break: LINE_CARD LC_2_r_LC_2_SW2	0	0	0	0
10/28/2012 13:56	Service Fix Attempt: HostService	1	0	0	0
10/28/2012 13:57	HW Fix: LINE_CARD LC_2_r_LC_2_SW2	0	0	0	0
10/28/2012 13:57	Service Fix: HostService	1	2.494	1	0
11/11/2012 11:01	HW Break: SOCKET_ADAPTOR ScAdpt_5	0	0	0	0
12/26/2012 05:53	HW Break: POWER_SUPPLY PS_4_of_Switch1	0	0	0	0
03/12/2013 14:25	HW Break: FAN FAN_1Server_Server1	0	0	0	0
04/18/2013 10:32	Service Break: HostService	1	0	-1	0
04/18/2013 10:32	HW Break: FAN FAN_2Server_Server1	0	0	0	0
04/18/2013 13:41	Service Fix Attempt: HostService	1	0	0	0
04/18/2013 13:42	HW Fix: FAN FAN_1Server_Server1	0	0	0	0
04/18/2013 13:42	Service Fix: HostService	1	3.16971	1	0
04/18/2013 13:42	HW Fix: FAN FAN_2Server_Server1	0	0	0	0
09/14/2013 15:27	Service Break: HostService	1	0	-1	0
09/14/2013 15:27	HW Break: POWER_SUPPLY PS_2_of_Server_Server1	0	0	0	0
09/14/2013 20:49	Service Fix Attempt: HostService	1	0	0	0
09/14/2013 20:49	HW Fix: POWER_SUPPLY PS_1_of_Server_Server1	0	0	0	0
09/14/2013 20:50	HW Fix: POWER_SUPPLY PS_2_of_Server_Server1	0	0	0	0
09/14/2013 20:50	Service Fix: HostService	1	5.39041	1	0
02/21/2014 09:51	Type:PORT_EXPOSURE NR_4_n_V4 from:/3000002/ to://	0	0	0	0
04/16/2014 01:17	Type:PORT_EXPOSURE NR_4_n_V4 from:/4000003/ to://	0	0	0	0
07/07/2014 19:52	HW Break: POWER_SUPPLY PS_8_of_ResourceEnc1	0	0	0	0
07/07/2014 19:52	Service Break: HostService	1	0	-1	0
07/07/2014 23:12	Service Fix Attempt: HostService	1	0	0	0
07/07/2014 23:13	HW Fix: POWER_SUPPLY PS_8_of_ResourceEnc1	0	0	0	0
07/07/2014 23:14	HW Fix: POWER_SUPPLY PS_7_of_ResourceEnc1	0	0	0	0
07/07/2014 23:14	Service Fix: HostService	1	3.3601	1	0
12/15/2014 03:52	Type:ALLOW_LIST NR_3_n_V3 from:/S1C1P_1111/ to://	0	0	0	0
01/22/2015 03:21	Service Break: HostService	1	0	-1	0
01/22/2015 03:21	Type:PORT_EXPOSURE NR_3_n_V3 from:/4000003/ to://	0	0	0	0
01/22/2015 07:28	Service Fix Attempt: HostService	1	0	0	0
01/22/2015 07:28	Service Fix: HostService	1	4.12194	1	0
01/22/2015 07:28	Type:PORT_EXPOSURE NR_3_n_V3 from:// to:/4000003/	0	0	0	0
01/25/2015 16:12	Type:PORT_EXPOSURE NR_2_n_V2 from:/4000003/ to://	0	0	0	0
01/25/2015 16:13	Service Break: HostService	1	0	-1	0
01/25/2015 19:35	Type:PORT_EXPOSURE NR_2_n_V2 from:// to:/4000003/	0	0	0	0
01/25/2015 19:35	Service Fix Attempt: HostService	1	0	0	0
01/25/2015 19:35	Service Fix: HostService	1	3.38099	1	0
03/26/2015 21:34	Type:ALLOW_LIST NR_2_n_V2 from:/S1C2P_1112/ to://	0	0	0	0

03/26/2015 21:34	Service Break: HostService	1	0	-1	0
03/27/2015 00:54	Service Fix Attempt: HostService	1	0	0	0
03/27/2015 00:54	Service Fix: HostService	1	3.32927	1	0
03/27/2015 00:54	Type:ALLOW_LIST NR_2_n_V2 from:// to:/S1C2P_1112/	0	0	0	0
04/04/2015 13:22	HW Break: FAN FAN_5Switch2	0	0	0	0
04/16/2015 05:17	Type:ALLOW_LIST NR_2_n_V2 from:/S1C1P_1111/ to://	0	0	0	0
05/26/2015 12:58	Type:ALLOW_LIST NR_4_n_V4 from:/S1C2P_1112/ to://	0	0	0	0
05/29/2015 21:52	Service Break: HostService	1	0	-1	0
05/29/2015 21:52	HW Break: LINE_CARD LC_2_r_LC_2_SW2	0	0	0	0
05/30/2015 02:01	HW Fix: LINE_CARD LC_2_r_LC_2_SW2	0	0	0	0
05/30/2015 02:01	Service Fix Attempt: HostService	1	0	0	0
05/30/2015 02:01	Service Fix: HostService	1	4.15949	1	0
08/11/2015 00:00	Type:ALLOW_LIST NR_1_n_V1 from:/S1C1P_1111/ to://	0	0	0	0
09/22/2015 15:42	Service Break: HostService	1	0	-1	0
09/22/2015 15:42	Type:PORT_EXPOSURE NR_1_n_V1 from:/4000003/ to://	0	0	0	0
09/22/2015 22:07	Service Fix Attempt: HostService	1	0	0	0
09/22/2015 22:08	Type:PORT_EXPOSURE NR_1_n_V1 from:// to:/4000003/	0	0	0	0
09/22/2015 22:08	Service Fix: HostService	1	6.42384	1	0
10/10/2015 17:34	HW Break: POWER_SUPPLY PS_7_of_ResourceEnc1	0	0	0	0
01/08/2016 04:07	Type:PORT_EXPOSURE NR_1_n_V1 from:/3000002/ to://	0	0	0	0
03/29/2016 04:04	Type:PORT_EXPOSURE NR_4_n_V4 from:/4000003/ to://	0	0	0	0
06/18/2016 16:58	Type:PORT_EXPOSURE NR_4_n_V4 from:/3000002/ to://	0	0	0	0
06/25/2016 16:40	Service Break: HostService	1	0	-1	0
06/25/2016 16:40	Type:ALLOW_LIST NR_3_n_V3 from:/S1C2P_1112/ to://	0	0	0	0
06/25/2016 20:31	Type:ALLOW_LIST NR_3_n_V3 from:// to:/S1C2P_1112/	0	0	0	0
06/25/2016 20:31	Service Fix Attempt: HostService	1	0	0	0
06/25/2016 20:32	Service Fix: HostService	1	3.86203	1	0
07/12/2016 06:59	HW Break: POWER_SUPPLY PS_8_of_ResourceEnc1	0	0	0	0
07/12/2016 06:59	Service Break: HostService	1	0	-1	0
07/12/2016 11:04	Service Fix Attempt: HostService	1	0	0	0
07/12/2016 11:05	HW Fix: POWER_SUPPLY PS_8_of_ResourceEnc1	0	0	0	0
07/12/2016 11:05	HW Fix: POWER_SUPPLY PS_7_of_ResourceEnc1	0	0	0	0
07/12/2016 11:05	Service Fix: HostService	1	4.10853	1	0
08/19/2016 08:03	Type:ALLOW_LIST NR_1_n_V1 from:/S1C2P_1112/ to://	0	0	0	0
08/19/2016 08:03	Service Break: HostService	1	0	-1	0
08/19/2016 13:36	Service Fix Attempt: HostService	1	0	0	0
08/19/2016 13:36	Service Fix: HostService	1	5.55716	1	0
08/19/2016 13:36	Type:ALLOW_LIST NR_1_n_V1 from:// to:/S1C2P_1112/	0	0	0	0
10/15/2016 15:51	HW Break: POWER_SUPPLY PS_7_of_ResourceEnc1	0	0	0	0
10/24/2016 12:49	Type:PORT_EXPOSURE NR_3_n_V3 from:/3000002/ to://	0	0	0	0
12/11/2016 07:51	Type:PORT_EXPOSURE NR_2_n_V2 from:/3000002/ to://	0	0	0	0
02/14/2017 16:25	HW Break: POWER_SUPPLY PS_1_of_Server_Server1	0	0	0	0
02/21/2017 04:40	HW Break: FAN FAN_2Server_Server1	0	0	0	0
03/13/2017 04:42	Service Break: HostService	1	0	-1	0
03/13/2017 04:42	HW Break: LINE_CARD LC_2_r_LC_2_SW2	0	0	0	0
03/13/2017 13:25	Service Fix Attempt: HostService	1	0	0	0
03/13/2017 13:26	HW Fix: LINE_CARD LC_2_r_LC_2_SW2	0	0	0	0
03/13/2017 13:26	Service Fix: HostService	1	8.72105	1	0
04/23/2017 07:25	Type:ALLOW_LIST NR_3_n_V3 from:/S1C1P_1111/ to://	0	0	0	0
08/31/2017 23:22	Service Break: HostService	1	0	-1	0

08/31/2017 23:22	HW Break: POWER_SUPPLY PS_2_of_Server_Server1	0	0	0	0
09/01/2017 04:33	Service Fix Attempt: HostService	1	0	0	0
09/01/2017 04:34	HW Fix: POWER_SUPPLY PS_1_of_Server_Server1	0	0	0	0
09/01/2017 04:34	HW Fix: POWER_SUPPLY PS_2_of_Server_Server1	0	0	0	0
09/01/2017 04:35	Service Fix: HostService	1	5.20991	1	0
10/09/2017 10:51	Type:ALLOW_LIST NR_4_n_V4 from:/S1C1P_1111/ to://	0	0	0	0
11/26/2017 13:29	Service Break: HostService	1	0	-1	0
11/26/2017 13:29	Type:PORT_EXPOSURE NR_1_n_V1 from:/4000003/ to://	0	0	0	0
11/26/2017 19:27	Service Fix Attempt: HostService	1	0	0	0
11/26/2017 19:28	Type:PORT_EXPOSURE NR_1_n_V1 from:// to:/4000003/	0	0	0	0
11/26/2017 19:28	Service Fix: HostService	1	5.97964	1	0
03/25/2018 01:53	Type:ALLOW_LIST NR_2_n_V2 from:/S1C2P_1112/ to://	0	0	0	0
03/25/2018 01:53	Service Break: HostService	1	0	-1	0
03/25/2018 05:46	Service Fix Attempt: HostService	1	0	0	0
03/25/2018 05:46	Type:ALLOW_LIST NR_2_n_V2 from:// to:/S1C2P_1112/	0	0	0	0
03/25/2018 05:47	Service Fix: HostService	1	3.89229	1	0
03/25/2018 10:12	Service Break: HostService	1	0	-1	0
03/25/2018 10:12	HW Break: GRID Grid_2	0	0	0	0
03/25/2018 15:15	Service Fix Attempt: HostService	1	0	0	0
03/25/2018 15:16	HW Fix: POWER_SUPPLY PS_7_of_ResourceEnc1	0	0	0	0
03/25/2018 15:16	Service Fix: HostService	1	5.07545	1	0
03/25/2018 15:16	HW Fix: GRID Grid_2	0	0	0	0

Reliability

The following graphs show the reliability of the configurations.

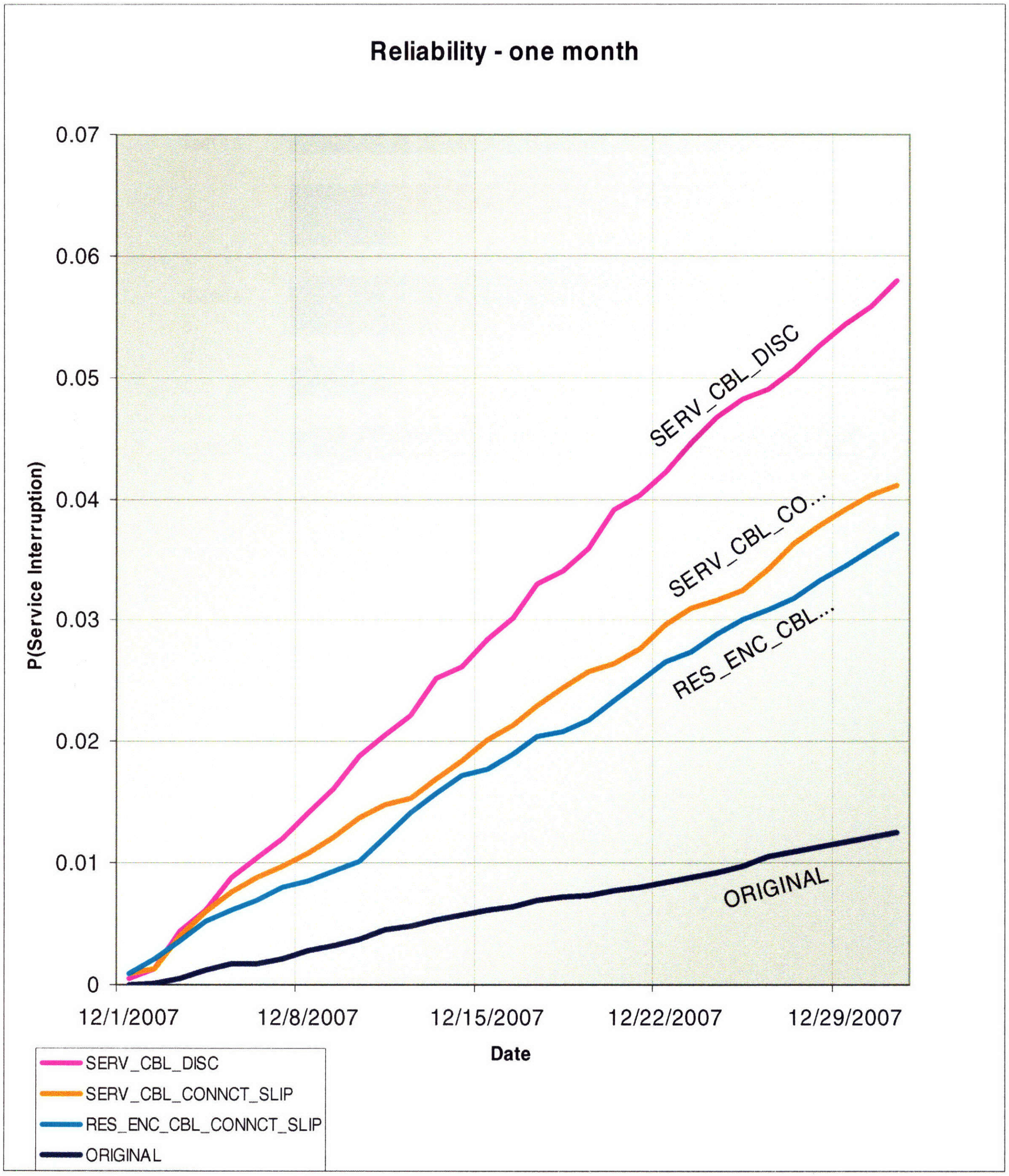


Figure 20 - Case study, reliability of first month

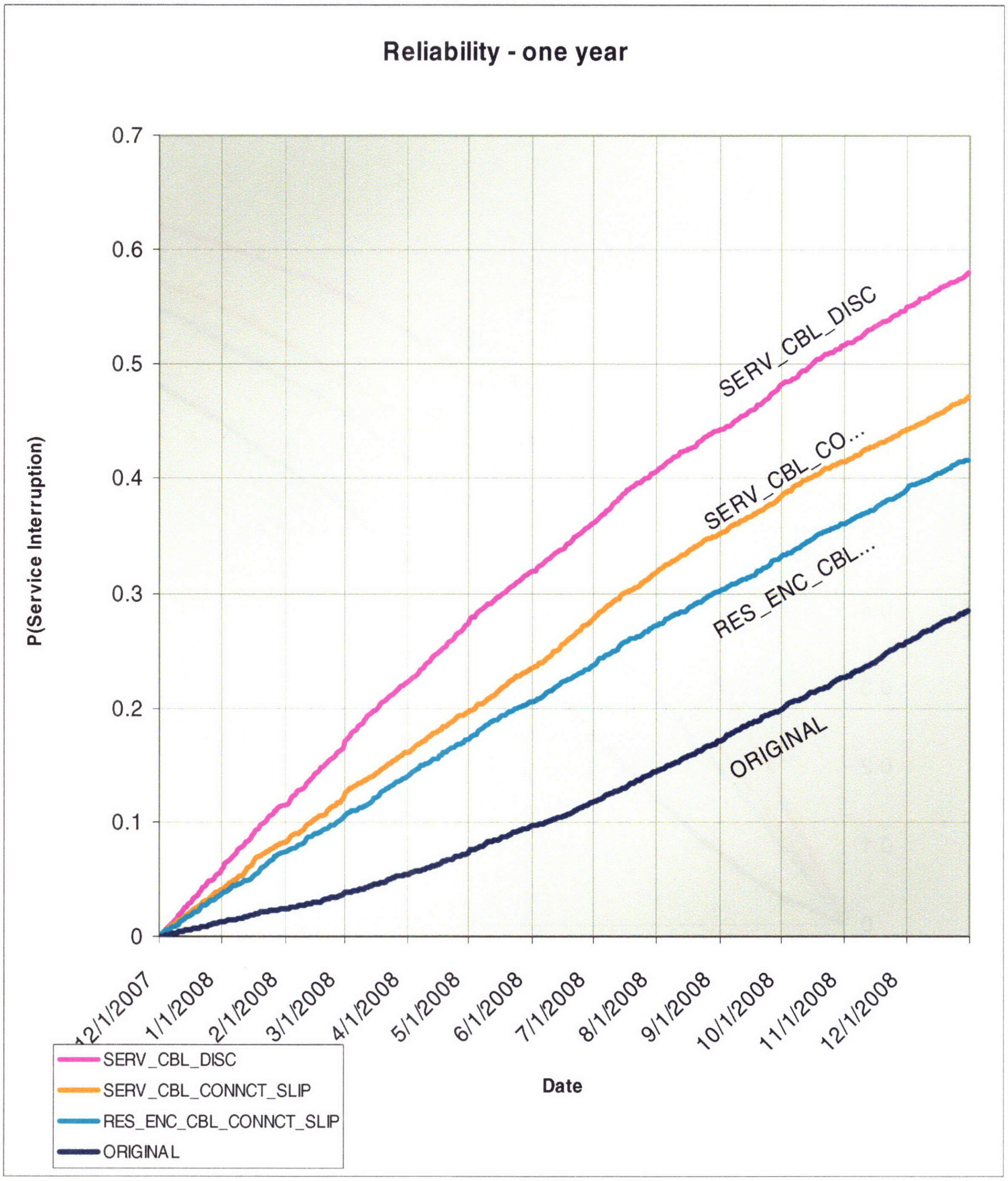


Figure 21 - Case study, reliability of first year

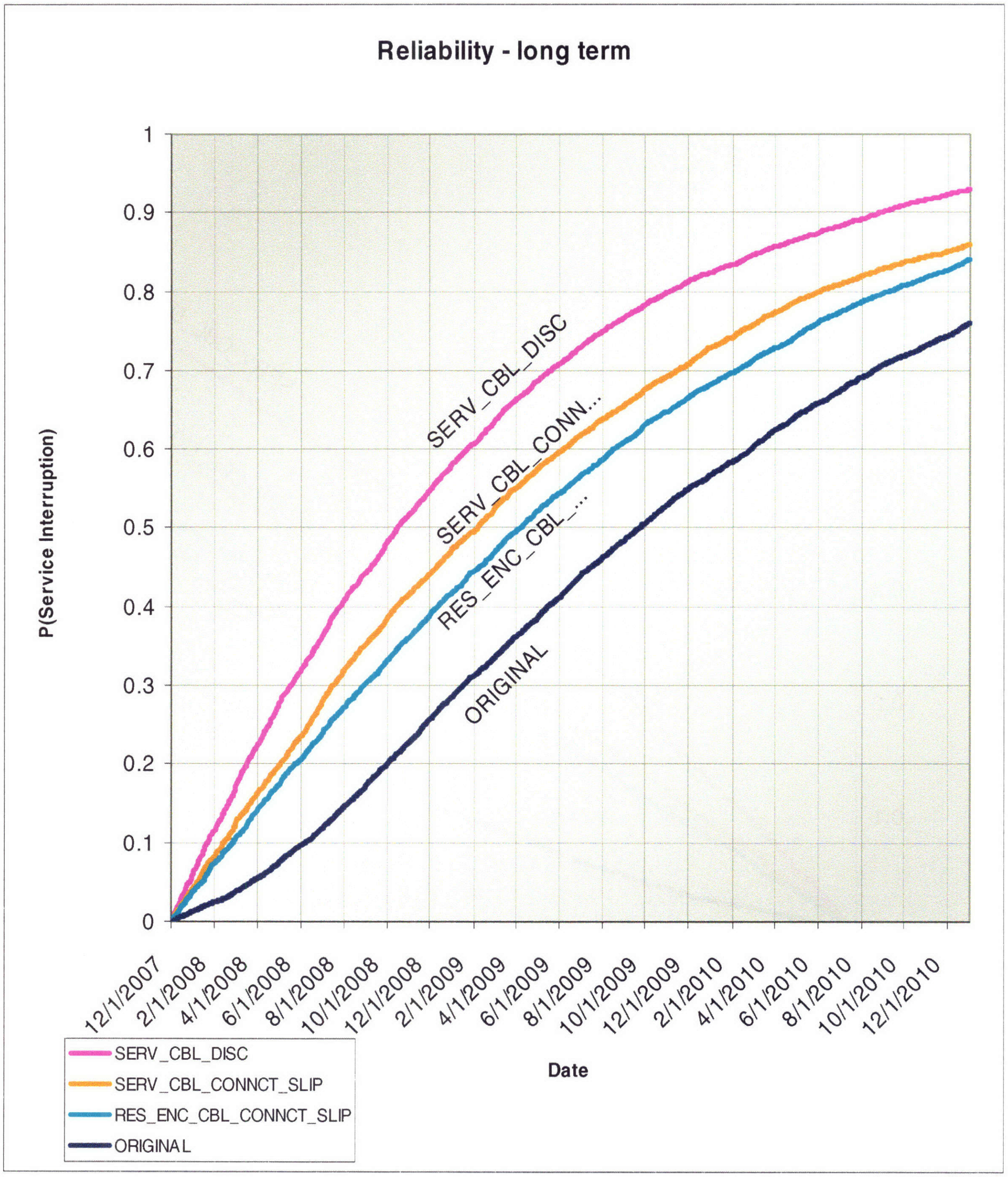


Figure 22 - Case study, reliability long term horizon

Figure 20, Figure 21 and Figure 22 shows the reliability graphs of all four configurations, in different time horizons. Looking at the probability of a service failure (HostService) after two weeks, it is clear that the ORIGINAL configuration that is comprised of multiple layers of redundancy (See Figure 14 - Case study base configuration, page 9), is superior to the other architectures that reflect three different changes. It is clear that the SERV_CBL_DISC (See Figure 15 - Case study first change – cable disconnect, Page 9) is the worst one in terms of reliability, and reflects a risk of ~10% or service interruption. That risk is about ten times more than the original configuration. However, the SERV_CBL_CONNCT_SLIP, and RES_ENC_CBL_CONNCT_SLIP configurations (See Figure 16, and Figure 17, Page 9), represents a better alternative than SERV_CBL_DISC. The reason for that is that although the connections were incorrect (a slip) and redundancy was lost to an extent, the two cables going out of the server in the SERV_CBL_CONNCT_SLIP configuration and the two cables going out of the resource enclosure in the RES_ENC_CBL_CONNCT_SLIP configuration, represents a small advantage in the form of network adaptor redundancy on the server or socket adaptor on the resource enclosure side, as well as switch ports redundancy. In the short term horizon (Figure 21) it is a significant difference. The two cable connection slip configurations are not identical in terms of reliability, and the RES_ENC_CBL_CONNCT_SLIP looks like it is slightly better than SERV_CBL_CONNCT_SLIP. This can be attributed to the fact that they reflect a slightly different risk. On the server side there are two network adaptors and on the resource enclosure – only one line card.

After a year, the chances of having a service interruption with the original configuration are around 25% while the other configurations are 38%-55%.

These results show how even changes that may seem very subtle in first observation, can be differentiated and quantified. This may be a valuable tool for comparing and evaluating different network configurations.

Un-Availability and Expected Loss

Figure 23 shows the un-availability for all configurations.

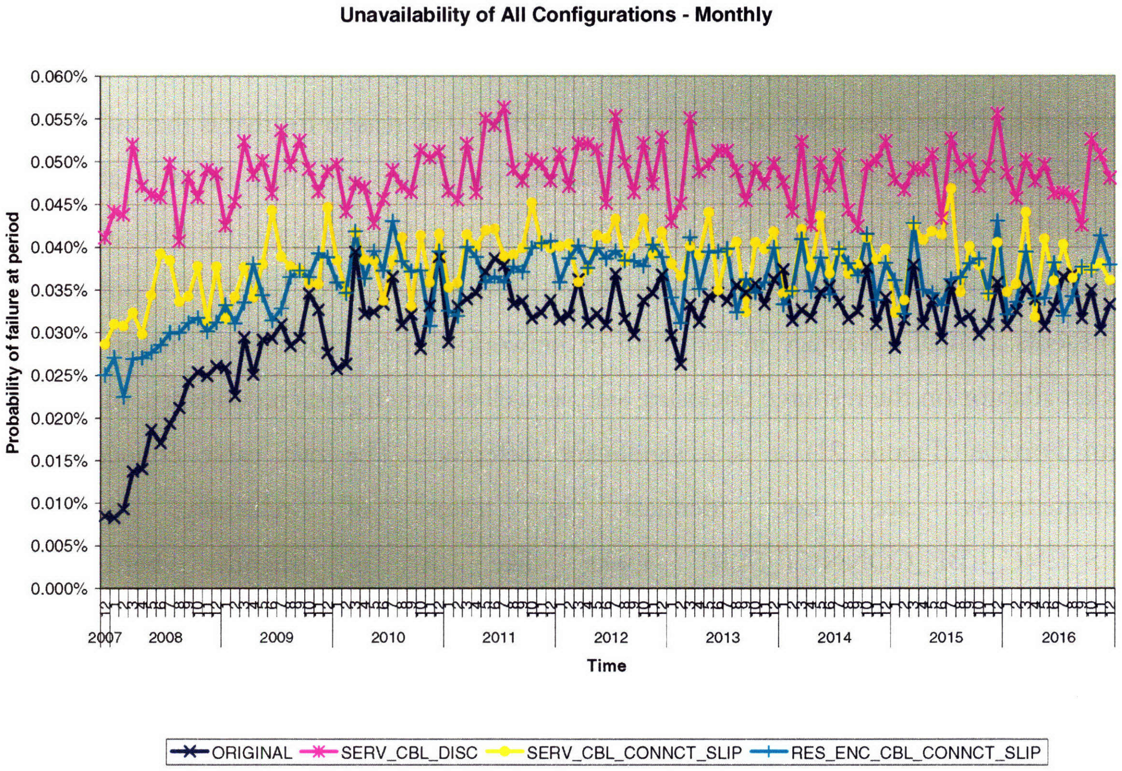


Figure 23 - Case study, un-availability of all configurations

The original configuration has more layers of redundancy and as a result, the probability of a service being unavailable at any given day is lower than other configurations. Again, depending on certain time frames, the original configuration is distinctly better than the other configurations. Note that the unavailability information is much noisier than the reliability is, because a substantially less data points are used for each point in the graph.

It is interesting to note that in some places along the graph the unavailability seems to be declining (for example serv_cbl_disc during parts of 2010). In multiple runs performed with the same conditions, this type of behavior was sometimes observed at other locations along the timeline. Again, this behavior is assumed to be related to the substantially low amount of data points comprising this graph. Since each of the scenarios was run with 5,000 future timelines, a month with unavailability of 0.04% represents an average unavailability time of about 17 min per month, or about 285 failure events at a certain month, out of the 5,000 future timelines. Since the number of failures is so low, there is a high degree of sensitivity in the results with regards to the number of failures. A much larger run (with a higher number of timelines) is needed to verify that point (for example by one or two orders of magnitude). Such run was not performed due to time and resources required.

As a result of the differences in availability, the expected losses are different. These cumulative losses are described in Figure 24 and Figure 25 comparing all four configurations. The expected loss after 6 months varies from ~$5,600 to ~$11,200 a difference of about 2x.

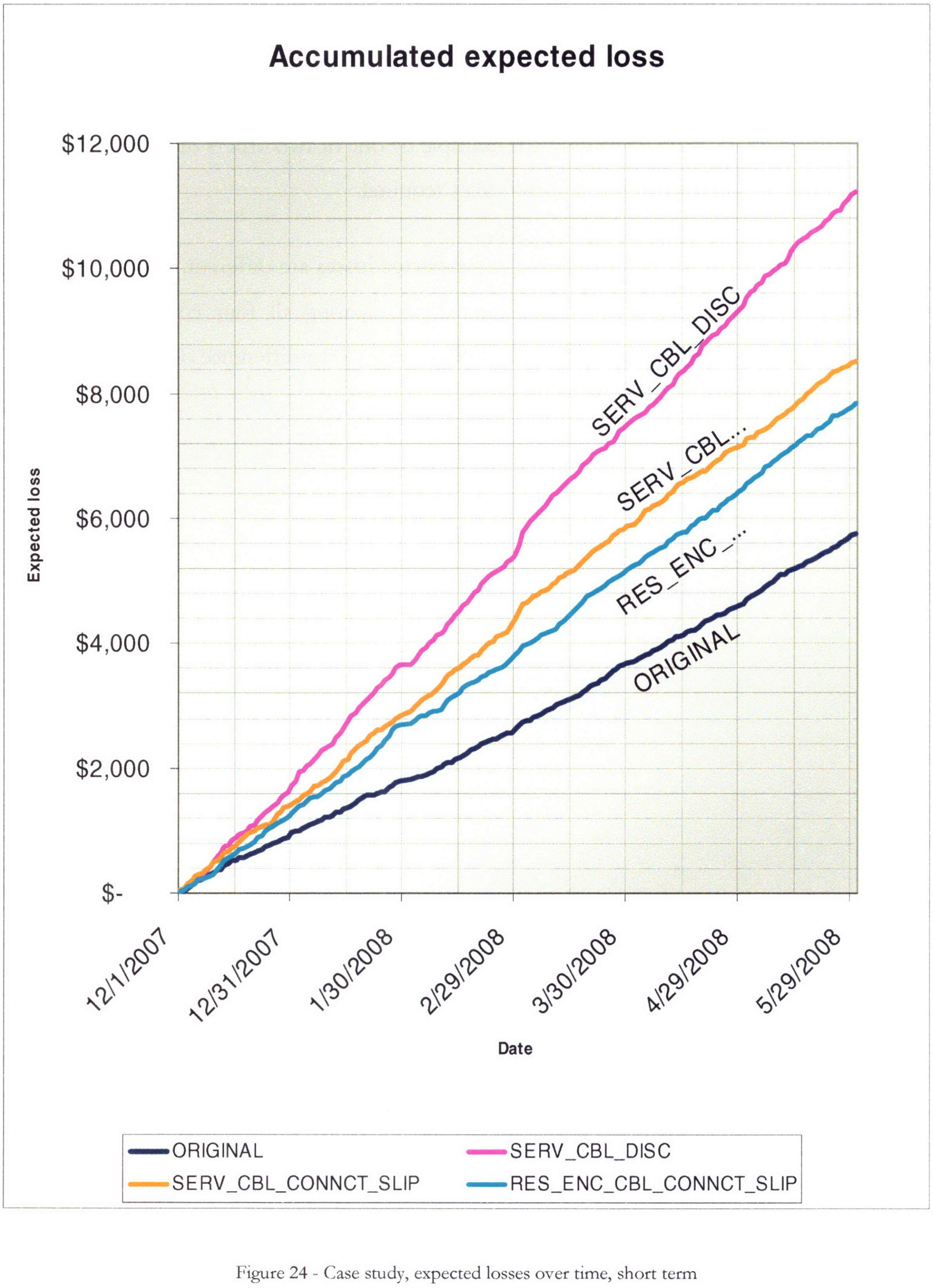


Figure 24 - Case study, expected losses over time, short term

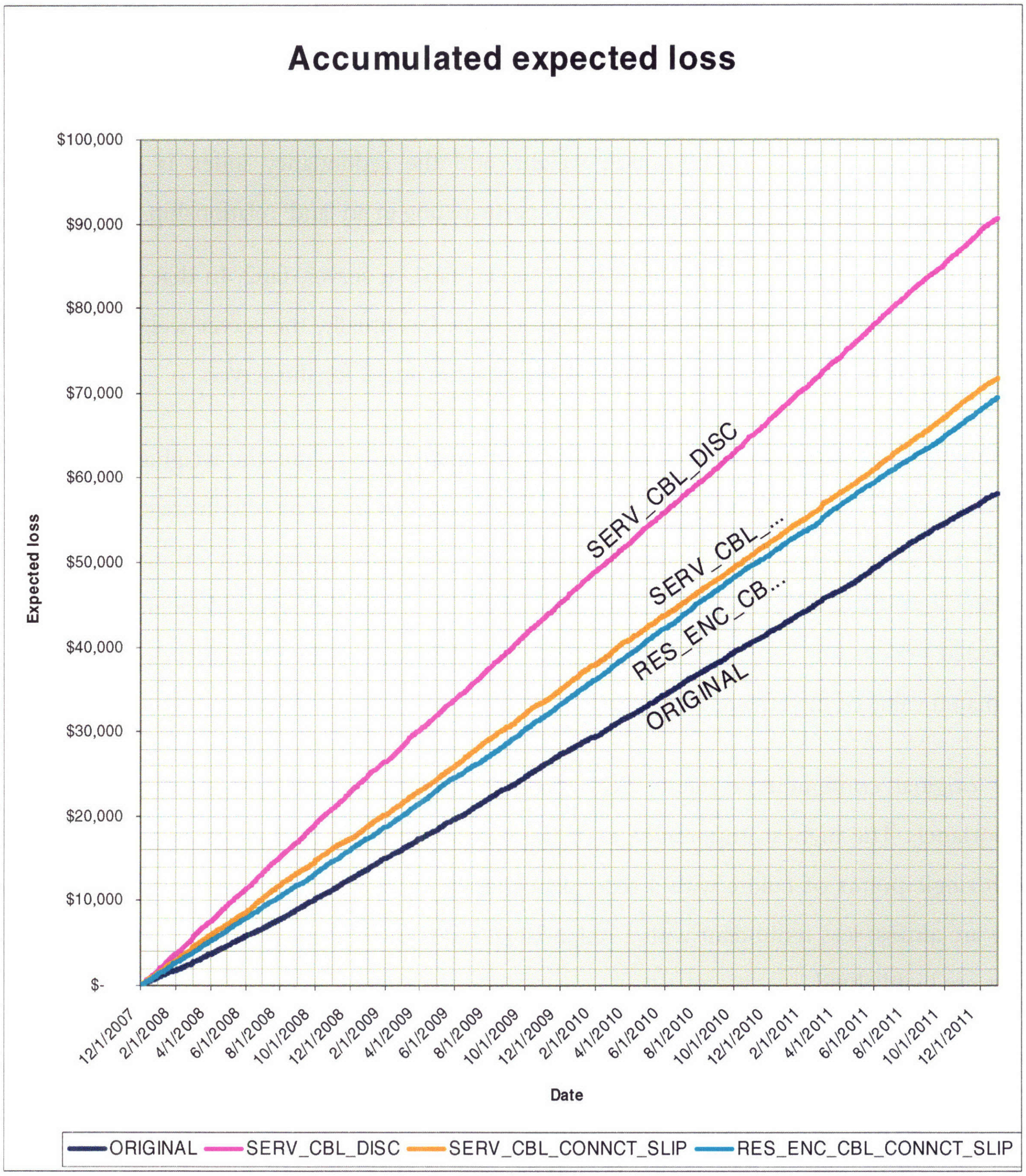


Figure 25 - Case study - expected losses over time, long term

Importance Measure

Importance measures were calculated as follows: whenever an event was found to be a part of a root cause (either immediate or latent) its contribution to the failure was measured. For example, if two fans fail and lead to the resource enclosure failure and a service failure, each fan failure even has 0.5 contribution weight, and the redundancy index is calculated as $1/_{contribution}$ and is 2. If an additional event is participating in an 'and' relationship, for example, the server failed, there are two causal chains server, fan1 or server, fan2. In that case, the first chain weighs 0.5 and each piece of the chain is 0.25. Eventually, the server weight is 0.5 and each fan is 0.25. The redundancy index is 4 for each fan and 2 for the server.

The importance report show each event that participated in a failure, the weighted average of expected cost (contribution * failure price). The report also presents the average redundancy level. A redundancy index of 1 indicates clearly on a non-redundant event (component failure or logical configuration). A redundancy index of 2 or more is a well redundant event. The expected price is not necessarily correlated to the redundancy index because it depends on the estimated value of the effected service.

The original configuration Figure 26 - page 9, presents a relatively low distribution of importance. The highest importance is of the server and the resource enclosure, as they are both single points of failure with a redundancy index of 1.0. The line cards follow closely, as they are almost less redundant with a redundancy index or ~1.8. Note that the 90th percentile rows are highlighted.

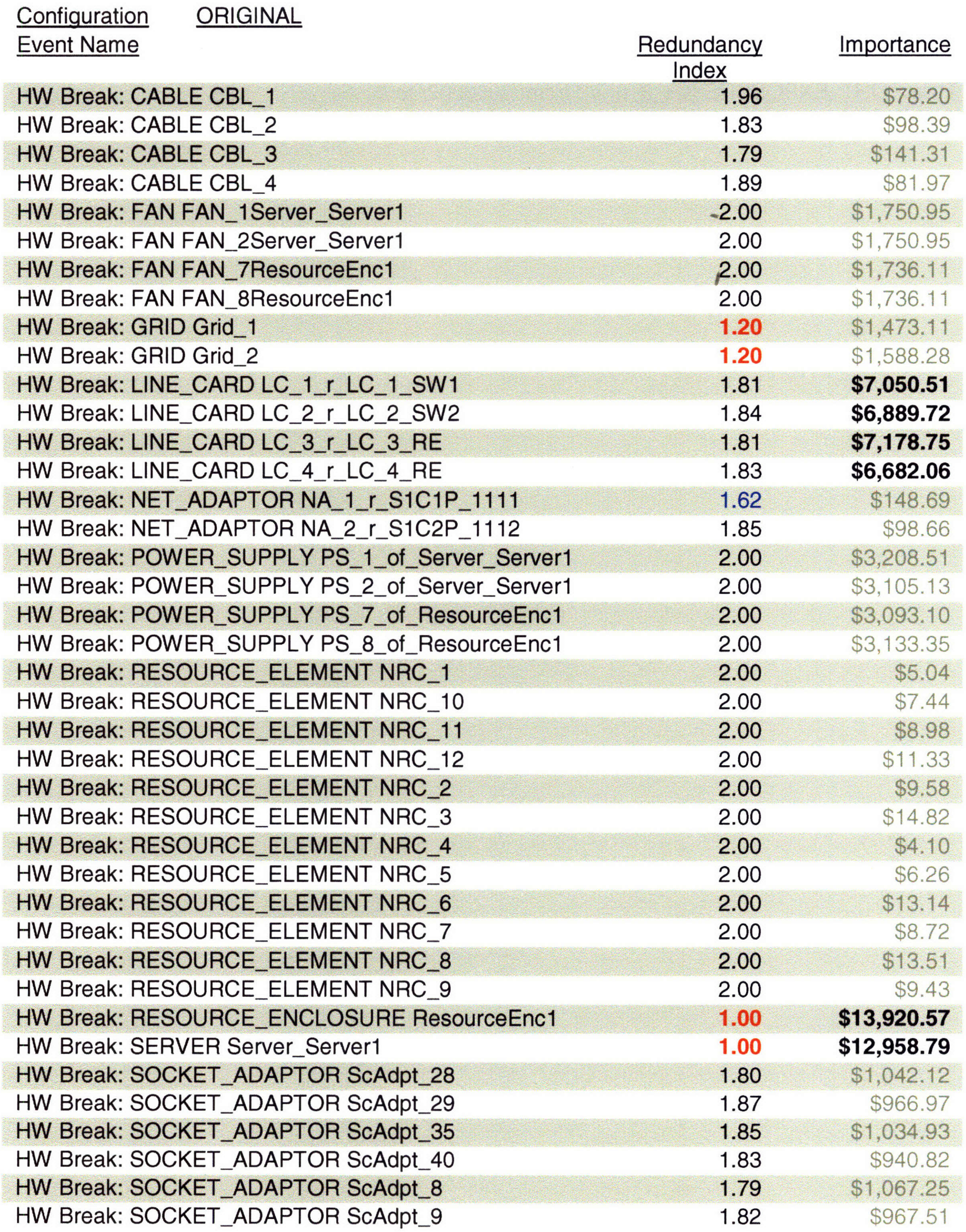
The serv_cable_disc configuration presents a substantial shift of importance to configurations on one side of the network (S1C1P_1111 represents the ID of one resource enclosure port, and 3000002 is one of the servers port IDs). The two other configurations have a cable connected incorrectly but the other side ports are still in use, and the importance is shifted accordingly. On Figure 27 - page 9, the differences between the original configuration and the first change - server cable disconnect are presented. It is interesting to note that with the original configuration, the importance of logical configurations of the network resources V1, V2, V3 are low and well distributed in the figure. However, after the change, a steep increase in the importance is associated with one 'side' of the configuration while the other 'side' loses all importance. The same happens to some of the hardware components, for example line card 1 of switch 1, and line card 3 of the resource enclosure. This clearly shows how the components or configurations that are 'out of the game' as a result of a change, loses their importance while their previously redundant counterparts become single points of failure and get a disproportionate share as a cause of expected future losses. Figure 28 - page 89, presents the measures when the two server ports are connected to one 'side'. As an entire 'side' is disabled, but the storage ports are still redundant and as a result the importance of the three configurations related to S1C1P_1111 and S1C2P_1112 reduces (as both server ports are still usable) while configurations related to the resource enclosure port 30000002 becomes very important (as only one resource enclosure port is usable). Note that even though the cable is incorrectly connected, it is allowing information to flow. The exposure configuration changes dramatically, when all the configurations related to 30000002 become the most important items with redundancy index 1.0 and configurations related to 40000003 become irrelevant because they are related to the inactive side. Figure 29 - page 91, related to incorrect connection of cable from the resource enclosure, presents a situation where the resource enclosure side connections are all somewhat more important and remain relevant (redundancy index 1.94) and here the server side configurations S1C1P_1111 become the most important items with redundancy 1.0 and their counterparts S1C2P_1112 become irrelevant. In this case, information can flow through the two cables coming from the resource enclosure to the same switch, but only one server cable is active.

By reviewing the importance measures details it is possible to understand how three different changes has such different risk profiles, while they all cause a (conceptually) similar situation: disable the information flow on the right hand side of the network. Figure 30 - page 92, shows a comparison between the two cable-slip configurations. There is symmetry between the importance shift on the resource enclosure side and the server side depending on where the slip took place. The slight difference between the two configurations in terms of reliability can be attributed to the fact the hardware related importance is different between the two and break the symmetry.

These importance-measures present information from the point of view of impact on the business. Failures or changes that may look similar in terms of engineering importance and probability may have different impacts on different services. Allowing a triage process that is driven by the business impact will assist with aligning the IT operations with business goals.

Element importance report

Table 4 - Importance report for original configuration



Configuration ORIGINAL

Event Name	Redundancy Index	Importance
HW Break: CABLE CBL_1	1.96	$78.20
HW Break: CABLE CBL_2	1.83	$98.39
HW Break: CABLE CBL_3	1.79	$141.31
HW Break: CABLE CBL_4	1.89	$81.97
HW Break: FAN FAN_1Server_Server1	2.00	$1,750.95
HW Break: FAN FAN_2Server_Server1	2.00	$1,750.95
HW Break: FAN FAN_7ResourceEnc1	2.00	$1,736.11
HW Break: FAN FAN_8ResourceEnc1	2.00	$1,736.11
HW Break: GRID Grid_1	**1.20**	$1,473.11
HW Break: GRID Grid_2	**1.20**	$1,588.28
HW Break: LINE_CARD LC_1_r_LC_1_SW1	1.81	**$7,050.51**
HW Break: LINE_CARD LC_2_r_LC_2_SW2	1.84	**$6,889.72**
HW Break: LINE_CARD LC_3_r_LC_3_RE	1.81	**$7,178.75**
HW Break: LINE_CARD LC_4_r_LC_4_RE	1.83	**$6,682.06**
HW Break: NET_ADAPTOR NA_1_r_S1C1P_1111	1.62	$148.69
HW Break: NET_ADAPTOR NA_2_r_S1C2P_1112	1.85	$98.66
HW Break: POWER_SUPPLY PS_1_of_Server_Server1	2.00	$3,208.51
HW Break: POWER_SUPPLY PS_2_of_Server_Server1	2.00	$3,105.13
HW Break: POWER_SUPPLY PS_7_of_ResourceEnc1	2.00	$3,093.10
HW Break: POWER_SUPPLY PS_8_of_ResourceEnc1	2.00	$3,133.35
HW Break: RESOURCE_ELEMENT NRC_1	2.00	$5.04
HW Break: RESOURCE_ELEMENT NRC_10	2.00	$7.44
HW Break: RESOURCE_ELEMENT NRC_11	2.00	$8.98
HW Break: RESOURCE_ELEMENT NRC_12	2.00	$11.33
HW Break: RESOURCE_ELEMENT NRC_2	2.00	$9.58
HW Break: RESOURCE_ELEMENT NRC_3	2.00	$14.82
HW Break: RESOURCE_ELEMENT NRC_4	2.00	$4.10
HW Break: RESOURCE_ELEMENT NRC_5	2.00	$6.26
HW Break: RESOURCE_ELEMENT NRC_6	2.00	$13.14
HW Break: RESOURCE_ELEMENT NRC_7	2.00	$8.72
HW Break: RESOURCE_ELEMENT NRC_8	2.00	$13.51
HW Break: RESOURCE_ELEMENT NRC_9	2.00	$9.43
HW Break: RESOURCE_ENCLOSURE ResourceEnc1	**1.00**	**$13,920.57**
HW Break: SERVER Server_Server1	**1.00**	**$12,958.79**
HW Break: SOCKET_ADAPTOR ScAdpt_28	1.80	$1,042.12
HW Break: SOCKET_ADAPTOR ScAdpt_29	1.87	$966.97
HW Break: SOCKET_ADAPTOR ScAdpt_35	1.85	$1,034.93
HW Break: SOCKET_ADAPTOR ScAdpt_40	1.83	$940.82
HW Break: SOCKET_ADAPTOR ScAdpt_8	1.79	$1,067.25
HW Break: SOCKET_ADAPTOR ScAdpt_9	1.82	$967.51

Type:ALLOW_LIST NR_1_n_V1 from:/S1C1P_1111/ to://	1.86	$3,009.32
Type:ALLOW_LIST NR_1_n_V1 from:/S1C2P_1112/ to://	1.90	$4,298.74
Type:ALLOW_LIST NR_2_n_V2 from:/S1C1P_1111/ to://	1.89	$5,425.15
Type:ALLOW_LIST NR_2_n_V2 from:/S1C2P_1112/ to://	1.90	$5,095.23
Type:ALLOW_LIST NR_3_n_V3 from:/S1C1P_1111/ to://	1.87	$5,418.37
Type:ALLOW_LIST NR_3_n_V3 from:/S1C2P_1112/ to://	1.90	$4,941.14
Type:PORT_EXPOSURE NR_1_n_V1 from:/3000002/ to://	1.88	$5,565.93
Type:PORT_EXPOSURE NR_1_n_V1 from:/4000003/ to://	1.90	$4,536.98
Type:PORT_EXPOSURE NR_2_n_V2 from:/3000002/ to://	1.87	$5,597.93
Type:PORT_EXPOSURE NR_2_n_V2 from:/4000003/ to://	1.89	$4,999.68
Type:PORT_EXPOSURE NR_3_n_V3 from:/3000002/ to://	1.88	$5,155.19
Type:PORT_EXPOSURE NR_3_n_V3 from:/4000003/ to://	1.88	$4,979.11

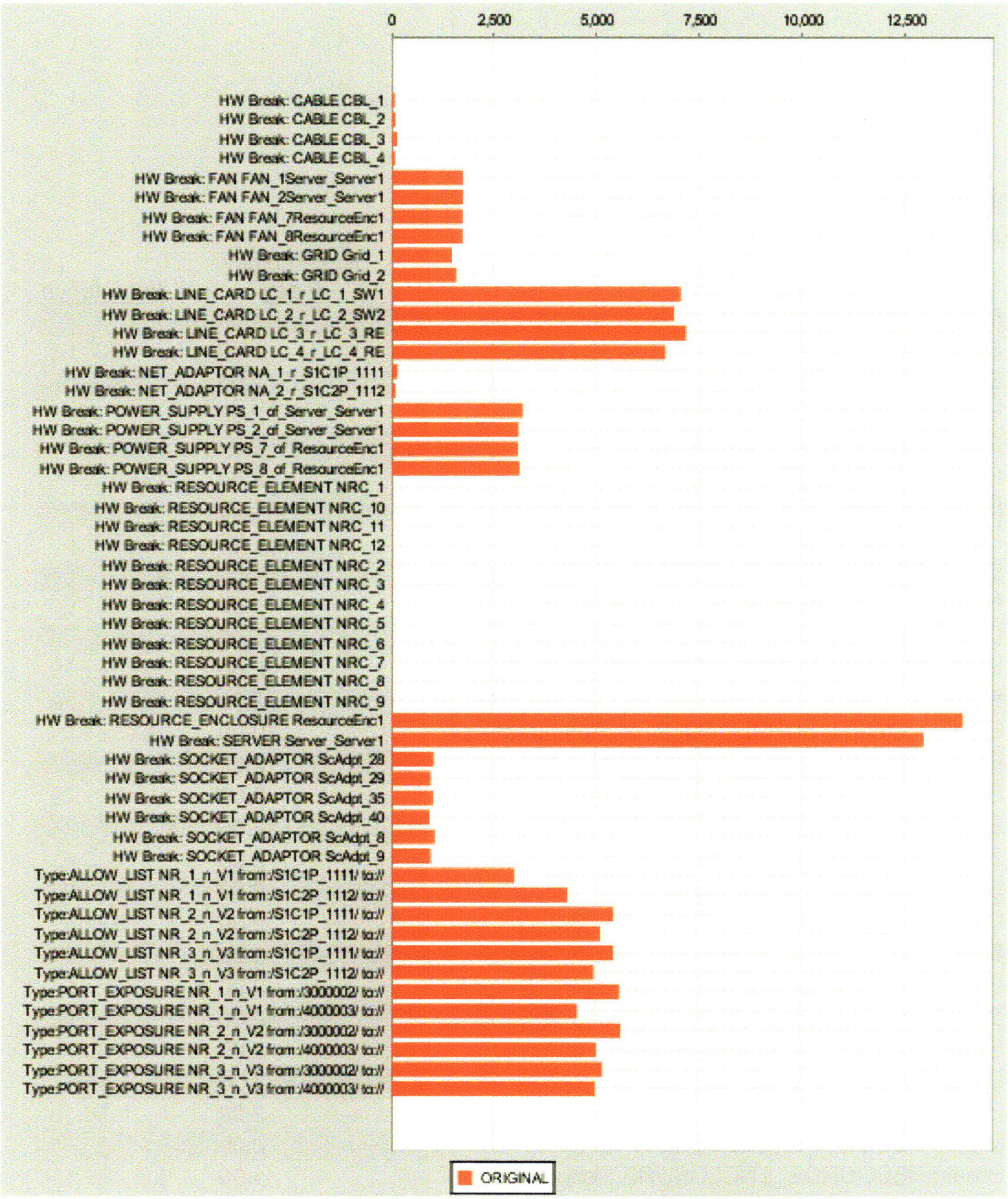
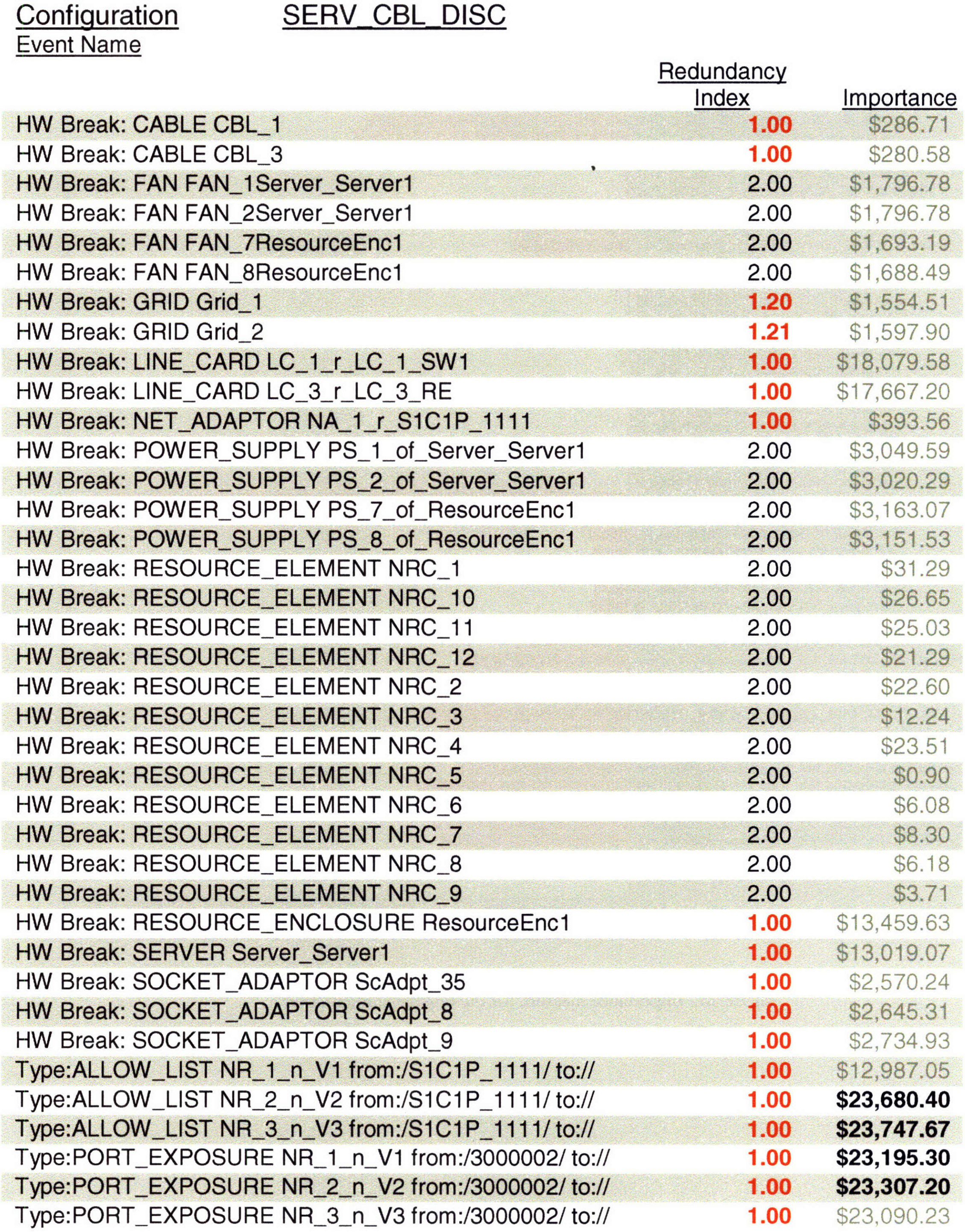


Figure 26 - Importance measure for the original configuration

Table 5 - Importance report for server disconnect slip configuration



Configuration SERV_CBL_DISC

Event Name	Redundancy Index	Importance
HW Break: CABLE CBL_1	1.00	$286.71
HW Break: CABLE CBL_3	1.00	$280.58
HW Break: FAN FAN_1Server_Server1	2.00	$1,796.78
HW Break: FAN FAN_2Server_Server1	2.00	$1,796.78
HW Break: FAN FAN_7ResourceEnc1	2.00	$1,693.19
HW Break: FAN FAN_8ResourceEnc1	2.00	$1,688.49
HW Break: GRID Grid_1	1.20	$1,554.51
HW Break: GRID Grid_2	1.21	$1,597.90
HW Break: LINE_CARD LC_1_r_LC_1_SW1	1.00	$18,079.58
HW Break: LINE_CARD LC_3_r_LC_3_RE	1.00	$17,667.20
HW Break: NET_ADAPTOR NA_1_r_S1C1P_1111	1.00	$393.56
HW Break: POWER_SUPPLY PS_1_of_Server_Server1	2.00	$3,049.59
HW Break: POWER_SUPPLY PS_2_of_Server_Server1	2.00	$3,020.29
HW Break: POWER_SUPPLY PS_7_of_ResourceEnc1	2.00	$3,163.07
HW Break: POWER_SUPPLY PS_8_of_ResourceEnc1	2.00	$3,151.53
HW Break: RESOURCE_ELEMENT NRC_1	2.00	$31.29
HW Break: RESOURCE_ELEMENT NRC_10	2.00	$26.65
HW Break: RESOURCE_ELEMENT NRC_11	2.00	$25.03
HW Break: RESOURCE_ELEMENT NRC_12	2.00	$21.29
HW Break: RESOURCE_ELEMENT NRC_2	2.00	$22.60
HW Break: RESOURCE_ELEMENT NRC_3	2.00	$12.24
HW Break: RESOURCE_ELEMENT NRC_4	2.00	$23.51
HW Break: RESOURCE_ELEMENT NRC_5	2.00	$0.90
HW Break: RESOURCE_ELEMENT NRC_6	2.00	$6.08
HW Break: RESOURCE_ELEMENT NRC_7	2.00	$8.30
HW Break: RESOURCE_ELEMENT NRC_8	2.00	$6.18
HW Break: RESOURCE_ELEMENT NRC_9	2.00	$3.71
HW Break: RESOURCE_ENCLOSURE ResourceEnc1	1.00	$13,459.63
HW Break: SERVER Server_Server1	1.00	$13,019.07
HW Break: SOCKET_ADAPTOR ScAdpt_35	1.00	$2,570.24
HW Break: SOCKET_ADAPTOR ScAdpt_8	1.00	$2,645.31
HW Break: SOCKET_ADAPTOR ScAdpt_9	1.00	$2,734.93
Type:ALLOW_LIST NR_1_n_V1 from:/S1C1P_1111/ to://	1.00	$12,987.05
Type:ALLOW_LIST NR_2_n_V2 from:/S1C1P_1111/ to://	1.00	**$23,680.40**
Type:ALLOW_LIST NR_3_n_V3 from:/S1C1P_1111/ to://	1.00	**$23,747.67**
Type:PORT_EXPOSURE NR_1_n_V1 from:/3000002/ to://	1.00	**$23,195.30**
Type:PORT_EXPOSURE NR_2_n_V2 from:/3000002/ to://	1.00	**$23,307.20**
Type:PORT_EXPOSURE NR_3_n_V3 from:/3000002/ to://	1.00	$23,090.23

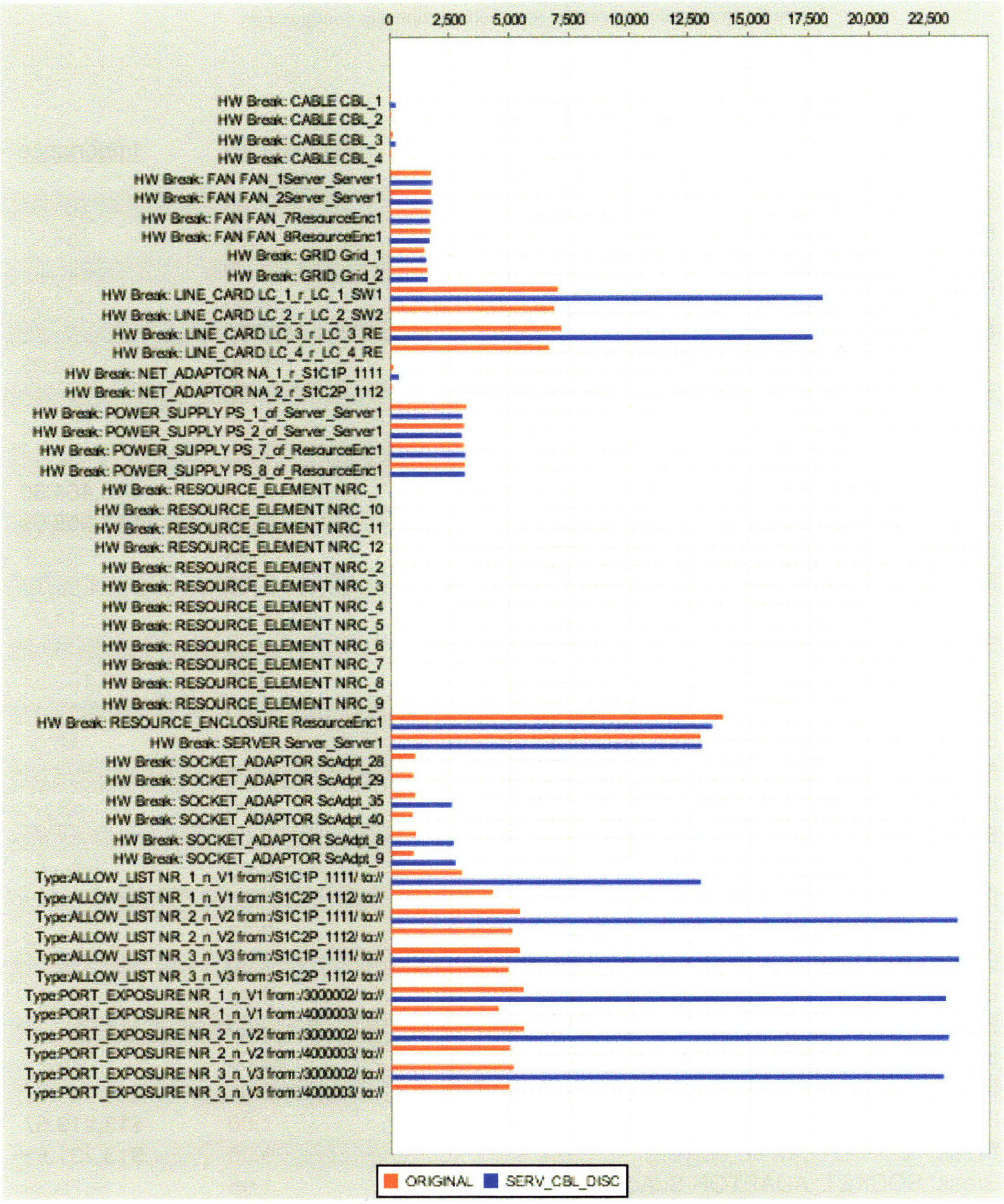
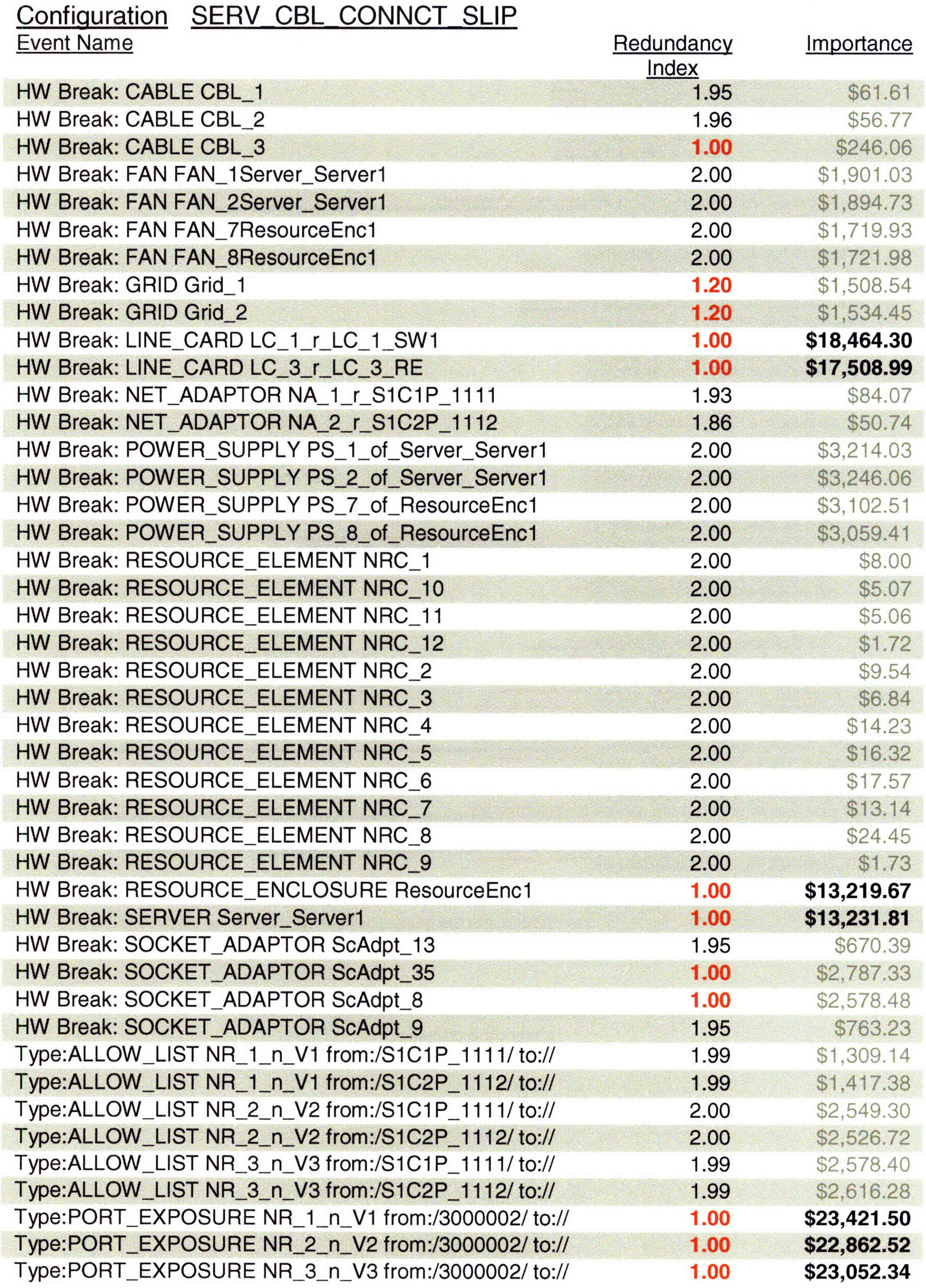


Figure 27 - Importance comparison of original and first change,
server cable disconnect

Table 6 - Importance report for server connection slip configuration



Configuration SERV_CBL_CONNCT_SLIP

Event Name	Redundancy Index	Importance
HW Break: CABLE CBL_1	1.95	$61.61
HW Break: CABLE CBL_2	1.96	$56.77
HW Break: CABLE CBL_3	1.00	$246.06
HW Break: FAN FAN_1Server_Server1	2.00	$1,901.03
HW Break: FAN FAN_2Server_Server1	2.00	$1,894.73
HW Break: FAN FAN_7ResourceEnc1	2.00	$1,719.93
HW Break: FAN FAN_8ResourceEnc1	2.00	$1,721.98
HW Break: GRID Grid_1	1.20	$1,508.54
HW Break: GRID Grid_2	1.20	$1,534.45
HW Break: LINE_CARD LC_1_r_LC_1_SW1	1.00	**$18,464.30**
HW Break: LINE_CARD LC_3_r_LC_3_RE	1.00	**$17,508.99**
HW Break: NET_ADAPTOR NA_1_r_S1C1P_1111	1.93	$84.07
HW Break: NET_ADAPTOR NA_2_r_S1C2P_1112	1.86	$50.74
HW Break: POWER_SUPPLY PS_1_of_Server_Server1	2.00	$3,214.03
HW Break: POWER_SUPPLY PS_2_of_Server_Server1	2.00	$3,246.06
HW Break: POWER_SUPPLY PS_7_of_ResourceEnc1	2.00	$3,102.51
HW Break: POWER_SUPPLY PS_8_of_ResourceEnc1	2.00	$3,059.41
HW Break: RESOURCE_ELEMENT NRC_1	2.00	$8.00
HW Break: RESOURCE_ELEMENT NRC_10	2.00	$5.07
HW Break: RESOURCE_ELEMENT NRC_11	2.00	$5.06
HW Break: RESOURCE_ELEMENT NRC_12	2.00	$1.72
HW Break: RESOURCE_ELEMENT NRC_2	2.00	$9.54
HW Break: RESOURCE_ELEMENT NRC_3	2.00	$6.84
HW Break: RESOURCE_ELEMENT NRC_4	2.00	$14.23
HW Break: RESOURCE_ELEMENT NRC_5	2.00	$16.32
HW Break: RESOURCE_ELEMENT NRC_6	2.00	$17.57
HW Break: RESOURCE_ELEMENT NRC_7	2.00	$13.14
HW Break: RESOURCE_ELEMENT NRC_8	2.00	$24.45
HW Break: RESOURCE_ELEMENT NRC_9	2.00	$1.73
HW Break: RESOURCE_ENCLOSURE ResourceEnc1	1.00	**$13,219.67**
HW Break: SERVER Server_Server1	1.00	**$13,231.81**
HW Break: SOCKET_ADAPTOR ScAdpt_13	1.95	$670.39
HW Break: SOCKET_ADAPTOR ScAdpt_35	1.00	$2,787.33
HW Break: SOCKET_ADAPTOR ScAdpt_8	1.00	$2,578.48
HW Break: SOCKET_ADAPTOR ScAdpt_9	1.95	$763.23
Type:ALLOW_LIST NR_1_n_V1 from:/S1C1P_1111/ to://	1.99	$1,309.14
Type:ALLOW_LIST NR_1_n_V1 from:/S1C2P_1112/ to://	1.99	$1,417.38
Type:ALLOW_LIST NR_2_n_V2 from:/S1C1P_1111/ to://	2.00	$2,549.30
Type:ALLOW_LIST NR_2_n_V2 from:/S1C2P_1112/ to://	2.00	$2,526.72
Type:ALLOW_LIST NR_3_n_V3 from:/S1C1P_1111/ to://	1.99	$2,578.40
Type:ALLOW_LIST NR_3_n_V3 from:/S1C2P_1112/ to://	1.99	$2,616.28
Type:PORT_EXPOSURE NR_1_n_V1 from:/3000002/ to://	1.00	**$23,421.50**
Type:PORT_EXPOSURE NR_2_n_V2 from:/3000002/ to://	1.00	**$22,862.52**
Type:PORT_EXPOSURE NR_3_n_V3 from:/3000002/ to://	1.00	**$23,052.34**

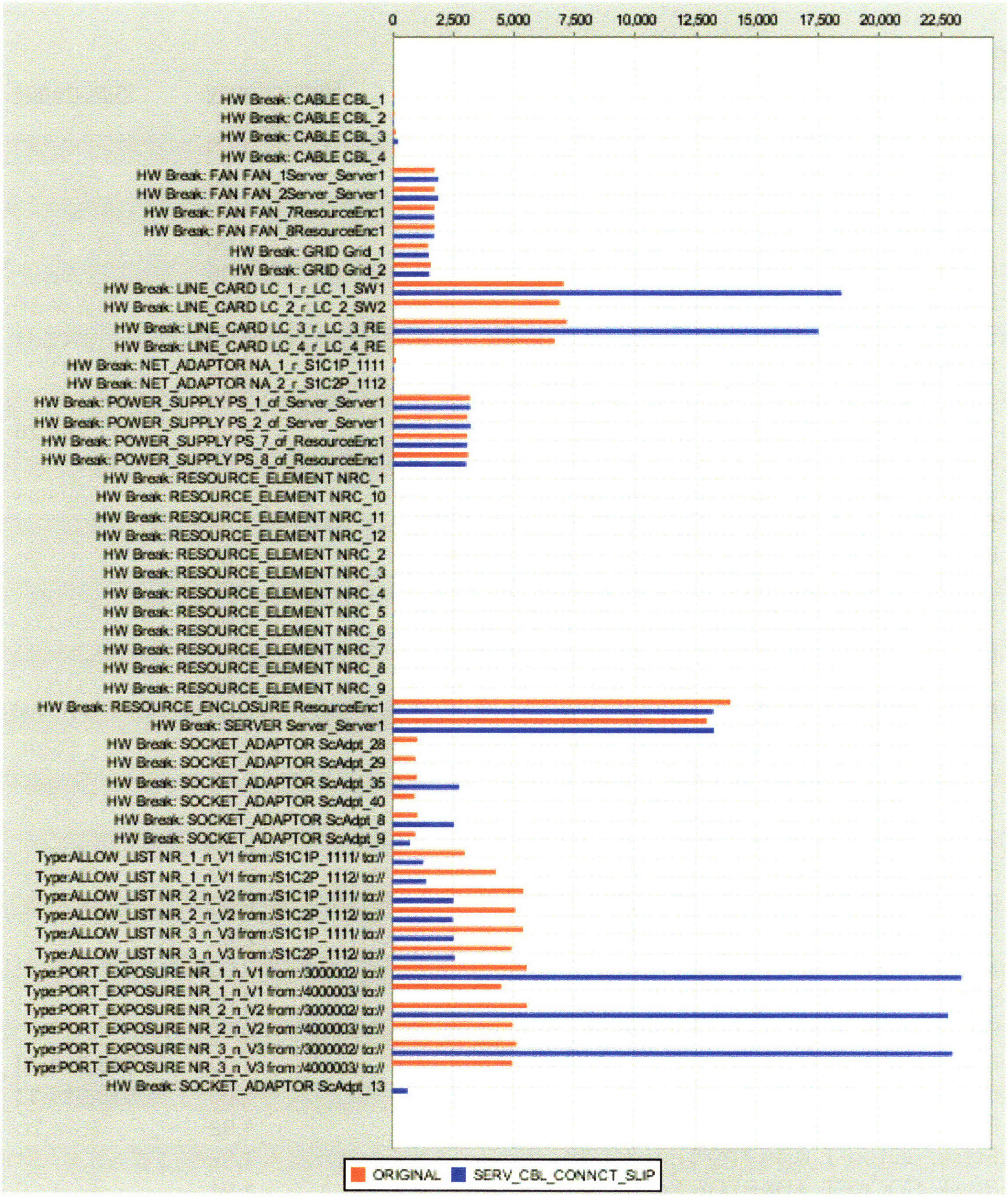
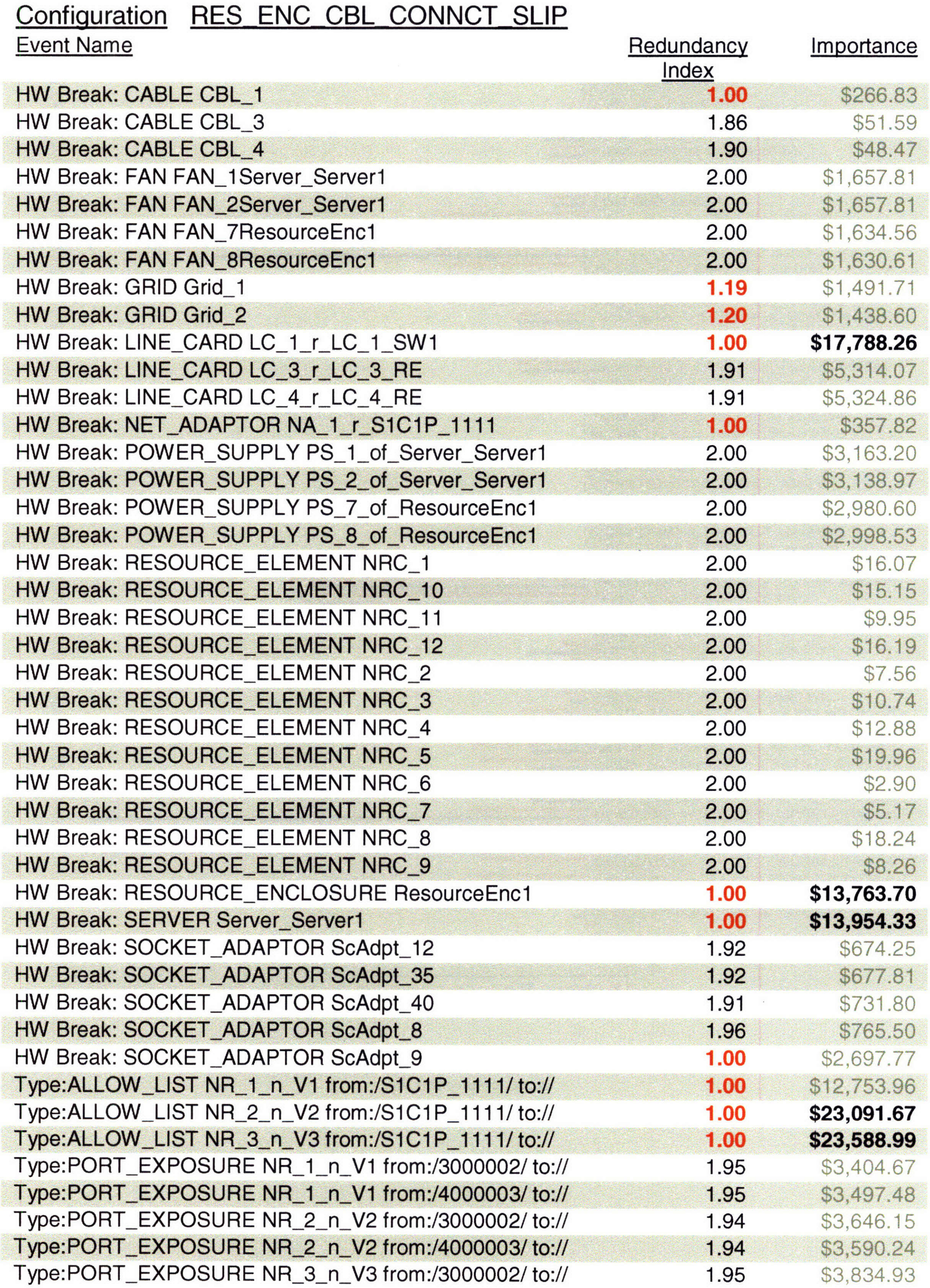


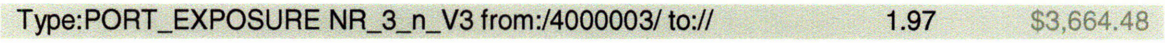
Figure 28 - Importance comparison of original and second change, server connection slip

Table 7 - Importance report for resource enclosure connection slip configuration



Configuration RES_ENC_CBL_CONNCT_SLIP

Event Name	Redundancy Index	Importance
HW Break: CABLE CBL_1	1.00	$266.83
HW Break: CABLE CBL_3	1.86	$51.59
HW Break: CABLE CBL_4	1.90	$48.47
HW Break: FAN FAN_1Server_Server1	2.00	$1,657.81
HW Break: FAN FAN_2Server_Server1	2.00	$1,657.81
HW Break: FAN FAN_7ResourceEnc1	2.00	$1,634.56
HW Break: FAN FAN_8ResourceEnc1	2.00	$1,630.61
HW Break: GRID Grid_1	1.19	$1,491.71
HW Break: GRID Grid_2	1.20	$1,438.60
HW Break: LINE_CARD LC_1_r_LC_1_SW1	1.00	**$17,788.26**
HW Break: LINE_CARD LC_3_r_LC_3_RE	1.91	$5,314.07
HW Break: LINE_CARD LC_4_r_LC_4_RE	1.91	$5,324.86
HW Break: NET_ADAPTOR NA_1_r_S1C1P_1111	1.00	$357.82
HW Break: POWER_SUPPLY PS_1_of_Server_Server1	2.00	$3,163.20
HW Break: POWER_SUPPLY PS_2_of_Server_Server1	2.00	$3,138.97
HW Break: POWER_SUPPLY PS_7_of_ResourceEnc1	2.00	$2,980.60
HW Break: POWER_SUPPLY PS_8_of_ResourceEnc1	2.00	$2,998.53
HW Break: RESOURCE_ELEMENT NRC_1	2.00	$16.07
HW Break: RESOURCE_ELEMENT NRC_10	2.00	$15.15
HW Break: RESOURCE_ELEMENT NRC_11	2.00	$9.95
HW Break: RESOURCE_ELEMENT NRC_12	2.00	$16.19
HW Break: RESOURCE_ELEMENT NRC_2	2.00	$7.56
HW Break: RESOURCE_ELEMENT NRC_3	2.00	$10.74
HW Break: RESOURCE_ELEMENT NRC_4	2.00	$12.88
HW Break: RESOURCE_ELEMENT NRC_5	2.00	$19.96
HW Break: RESOURCE_ELEMENT NRC_6	2.00	$2.90
HW Break: RESOURCE_ELEMENT NRC_7	2.00	$5.17
HW Break: RESOURCE_ELEMENT NRC_8	2.00	$18.24
HW Break: RESOURCE_ELEMENT NRC_9	2.00	$8.26
HW Break: RESOURCE_ENCLOSURE ResourceEnc1	1.00	**$13,763.70**
HW Break: SERVER Server_Server1	1.00	**$13,954.33**
HW Break: SOCKET_ADAPTOR ScAdpt_12	1.92	$674.25
HW Break: SOCKET_ADAPTOR ScAdpt_35	1.92	$677.81
HW Break: SOCKET_ADAPTOR ScAdpt_40	1.91	$731.80
HW Break: SOCKET_ADAPTOR ScAdpt_8	1.96	$765.50
HW Break: SOCKET_ADAPTOR ScAdpt_9	1.00	$2,697.77
Type:ALLOW_LIST NR_1_n_V1 from:/S1C1P_1111/ to://	1.00	$12,753.96
Type:ALLOW_LIST NR_2_n_V2 from:/S1C1P_1111/ to://	1.00	**$23,091.67**
Type:ALLOW_LIST NR_3_n_V3 from:/S1C1P_1111/ to://	1.00	**$23,588.99**
Type:PORT_EXPOSURE NR_1_n_V1 from:/3000002/ to://	1.95	$3,404.67
Type:PORT_EXPOSURE NR_1_n_V1 from:/4000003/ to://	1.95	$3,497.48
Type:PORT_EXPOSURE NR_2_n_V2 from:/3000002/ to://	1.94	$3,646.15
Type:PORT_EXPOSURE NR_2_n_V2 from:/4000003/ to://	1.94	$3,590.24
Type:PORT_EXPOSURE NR_3_n_V3 from:/3000002/ to://	1.95	$3,834.93



Type:PORT_EXPOSURE NR_3_n_V3 from:/4000003/ to://	1.97	$3,664.48

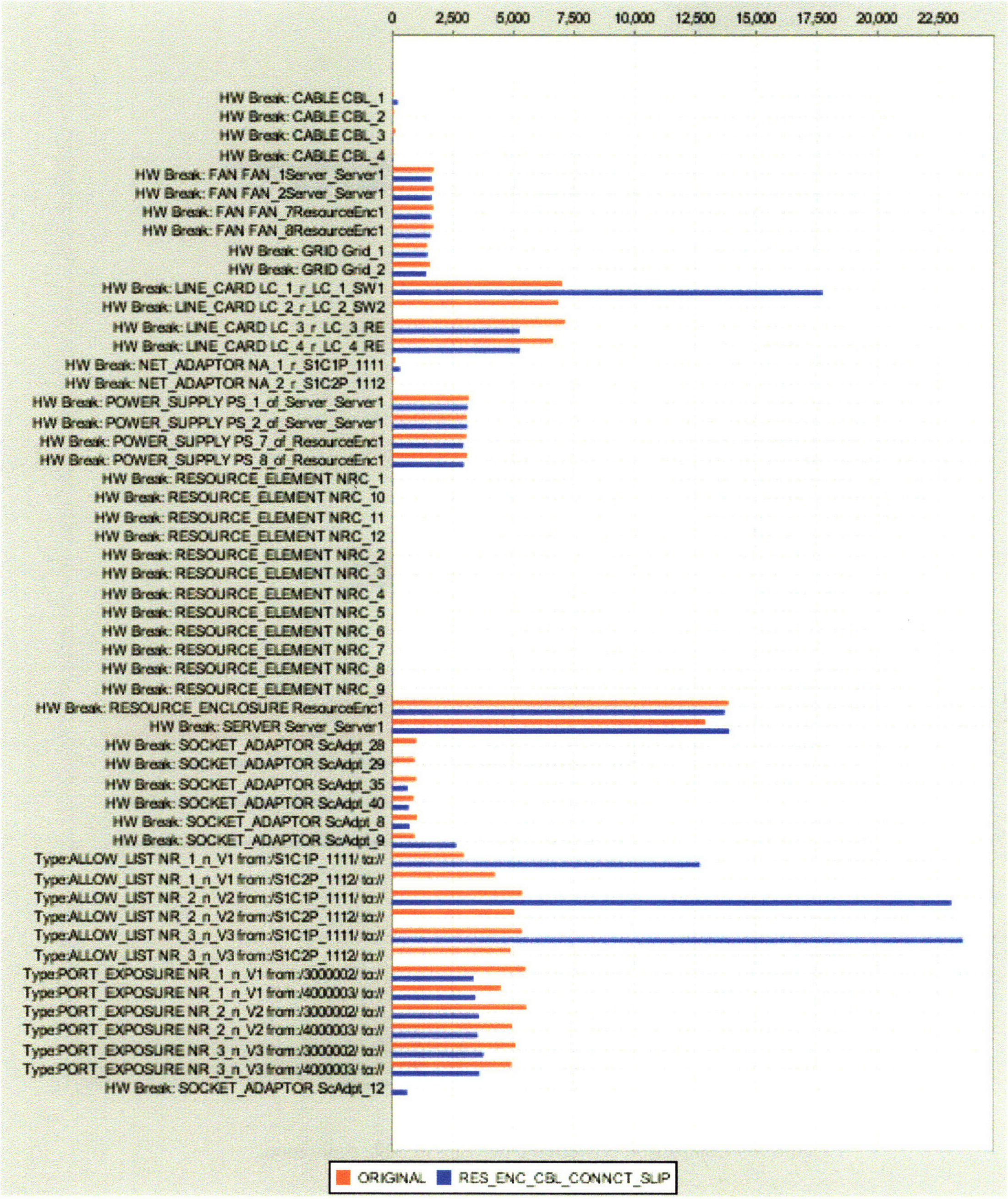


Figure 29 - Importance comparison of original and third change,
resource enc. Connection

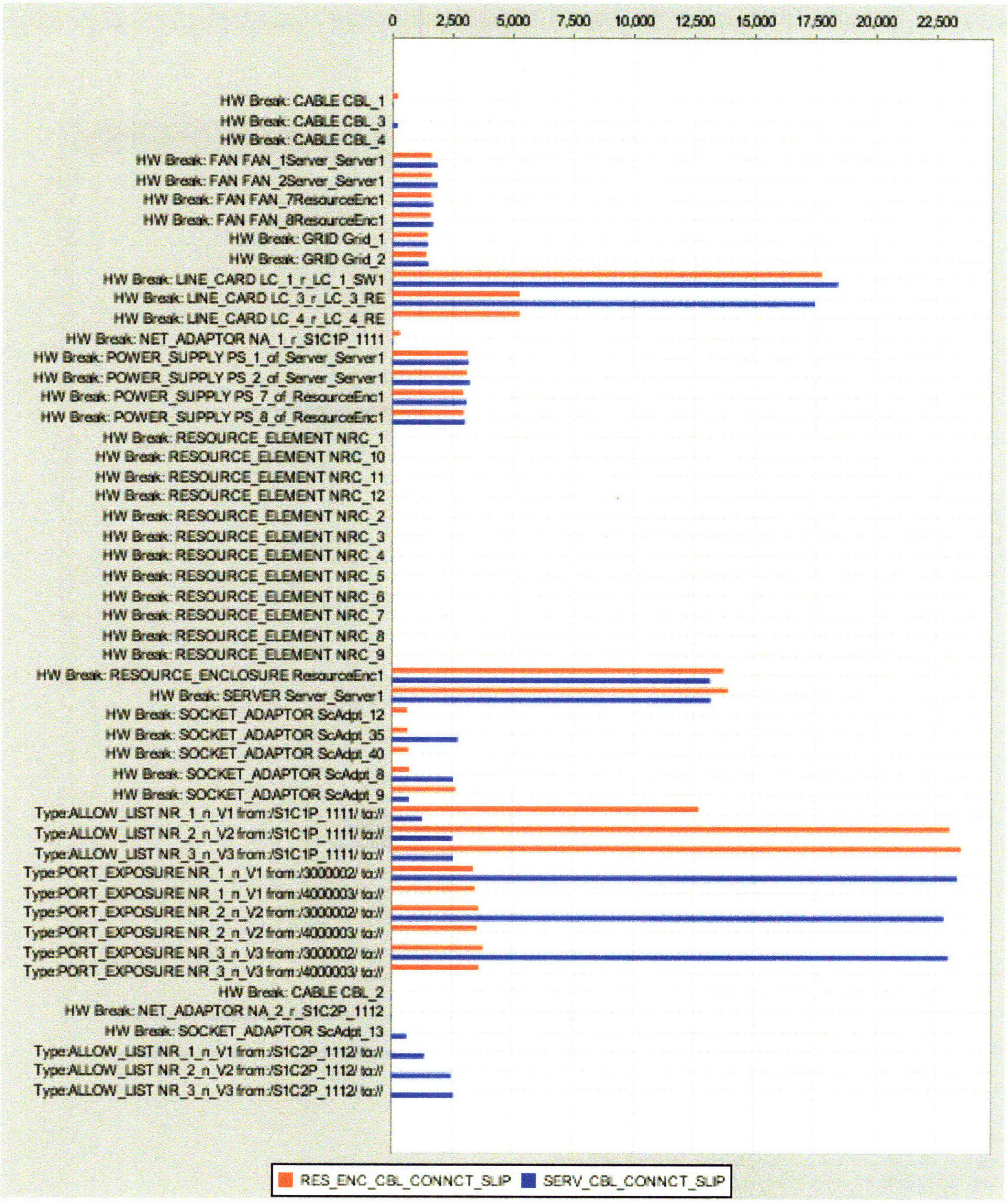


Figure 30 - Comparison between the two slips in cable connection

Cost Distribution over Time

Each one of the configuration discussed creates a different expected accumulated cost over time as can be seen in Figure 25, in page 79. However, these lines represent the expected cost

/ loss or the average of a distribution. In order to better understand what the distributions are, a detailed description of how they change over time is presented here. Original:

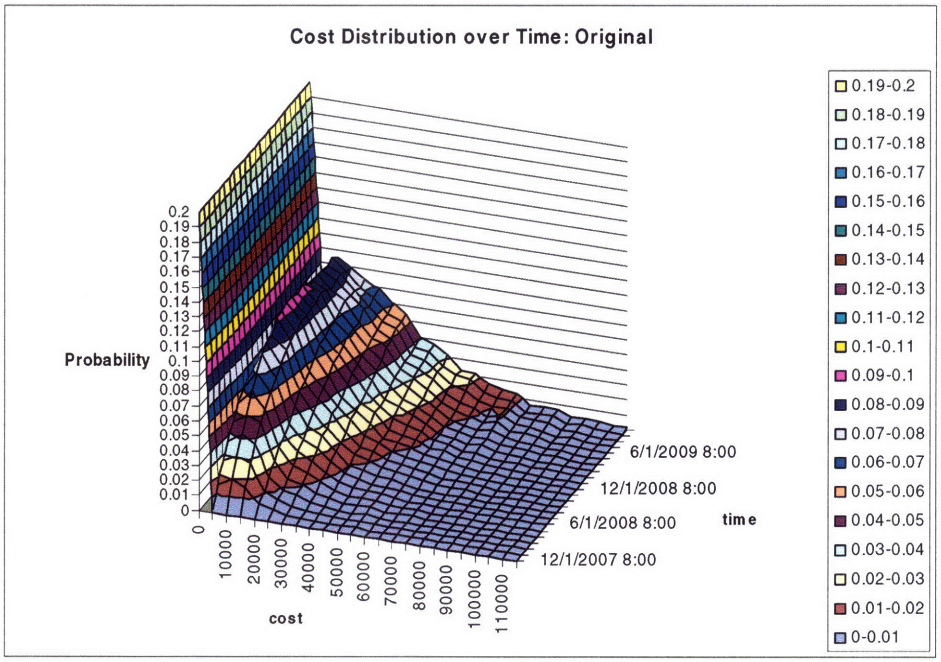


Figure 31 - Cost Distribution over Time: Original (3D)

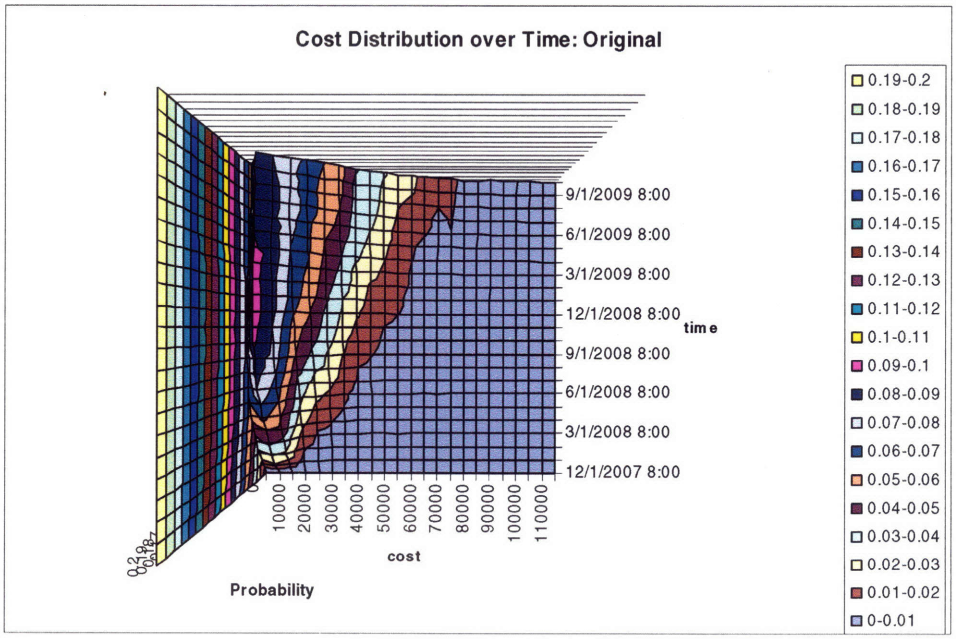


Figure 32 - Cost Distribution over Time: Original (Top)

Cable disconnect:

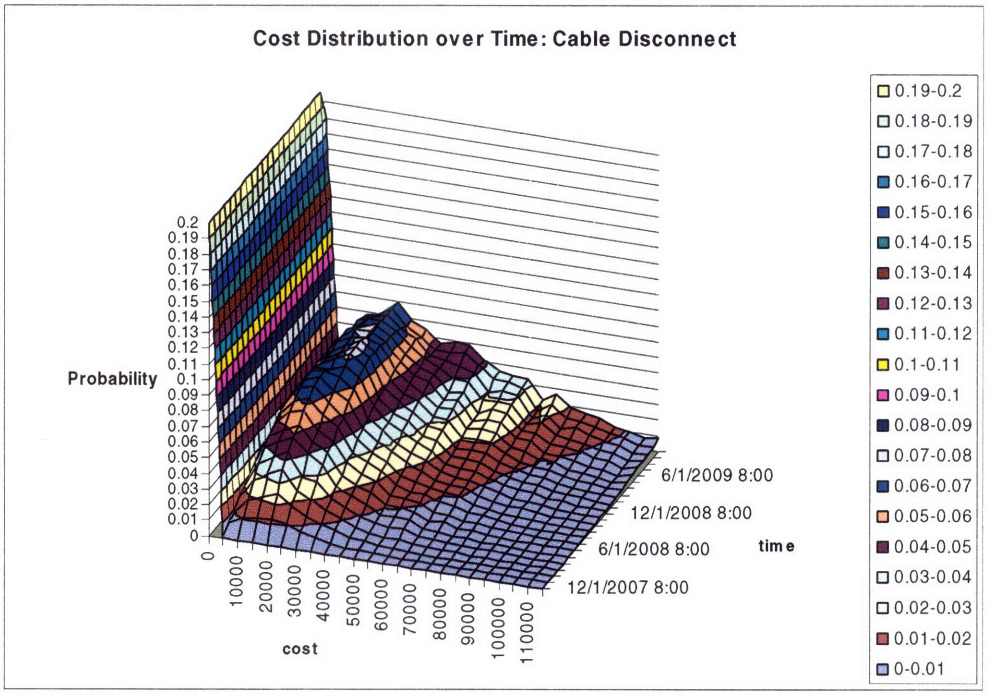


Figure 33 - Cost Distribution over Time: Cable Disconnect (3D)

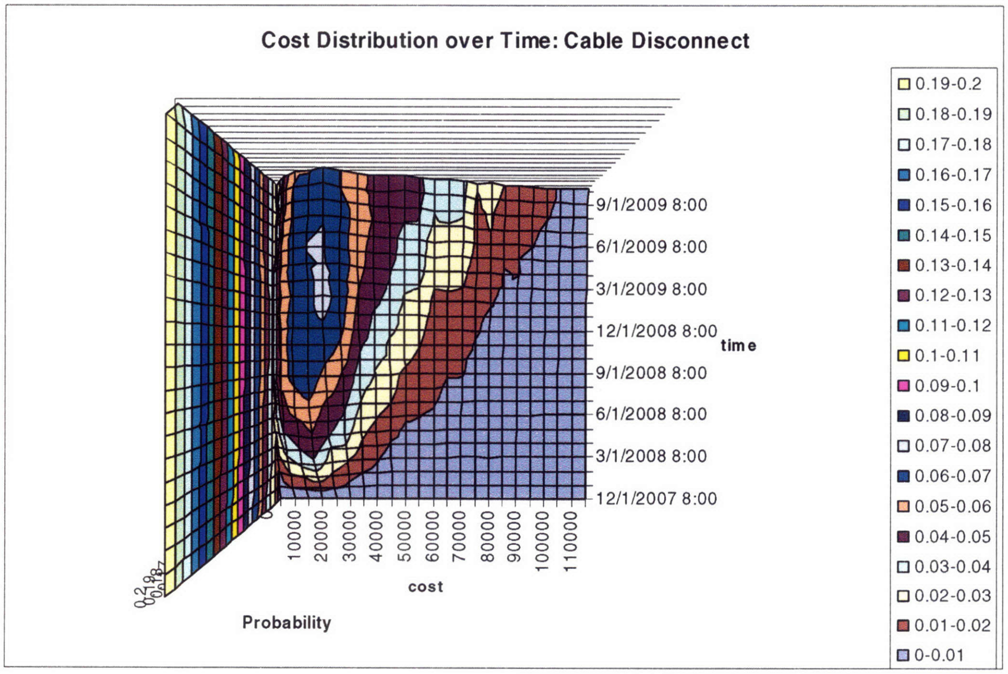


Figure 34 - Cost Distribution over Time: Cable Disconnect (Top)

Server connection slip:

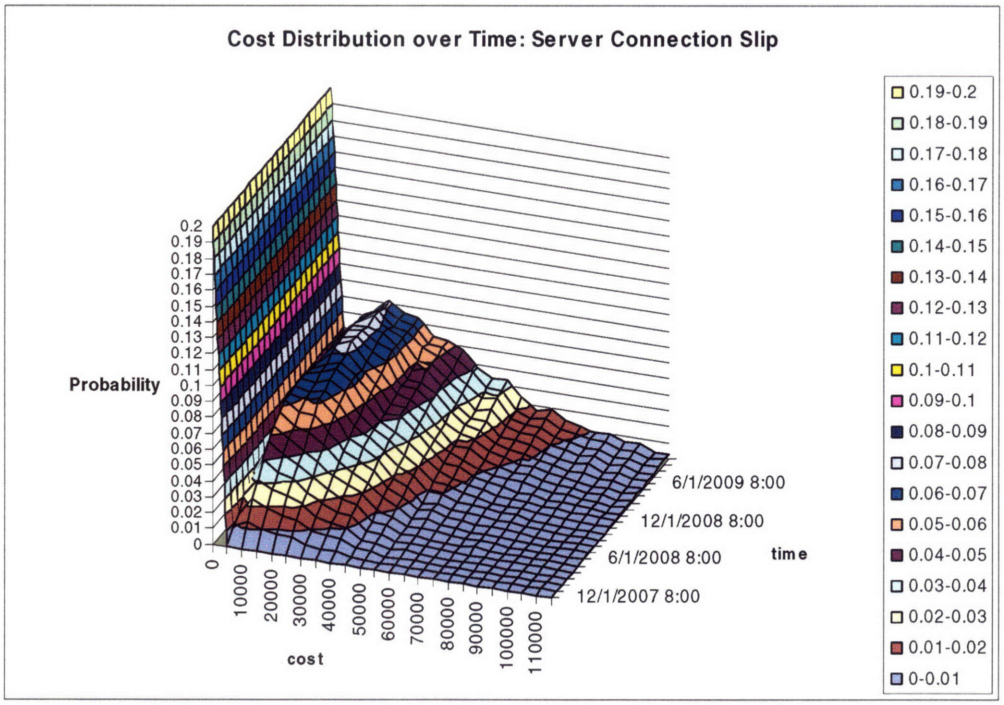


Figure 35 - Cost Distribution over Time: Server Con Slip (3D)

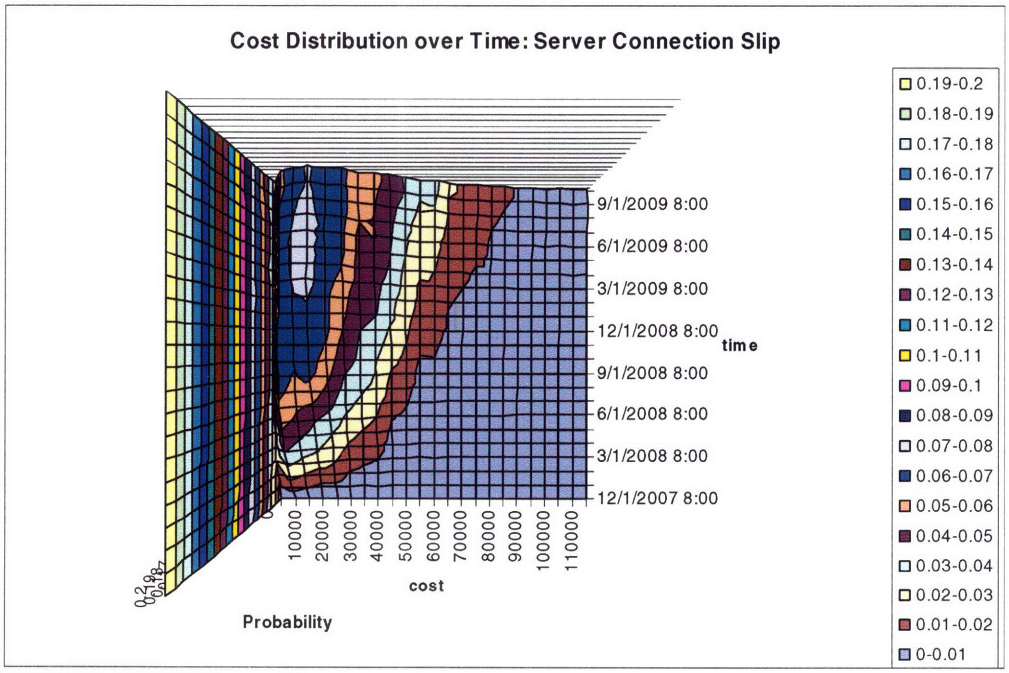


Figure 36 - Cost Distribution over Time: Server Con Slip (Top)

Resource enclosure connection slip:

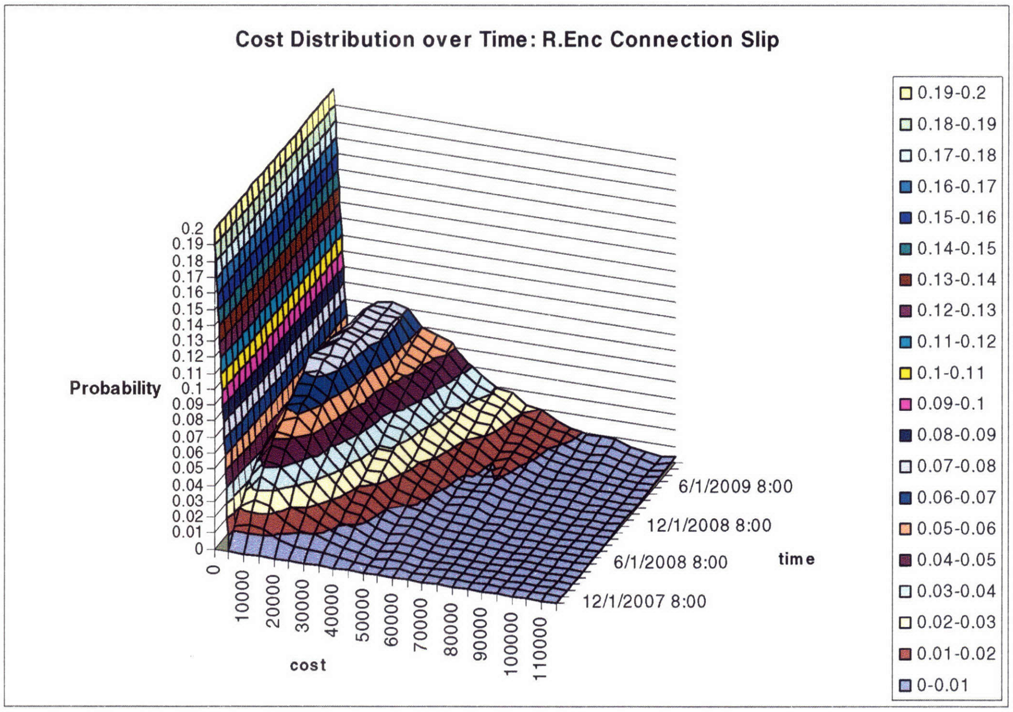


Figure 37 - Cost Distribution over Time: Res Enc Con Slip (3D)

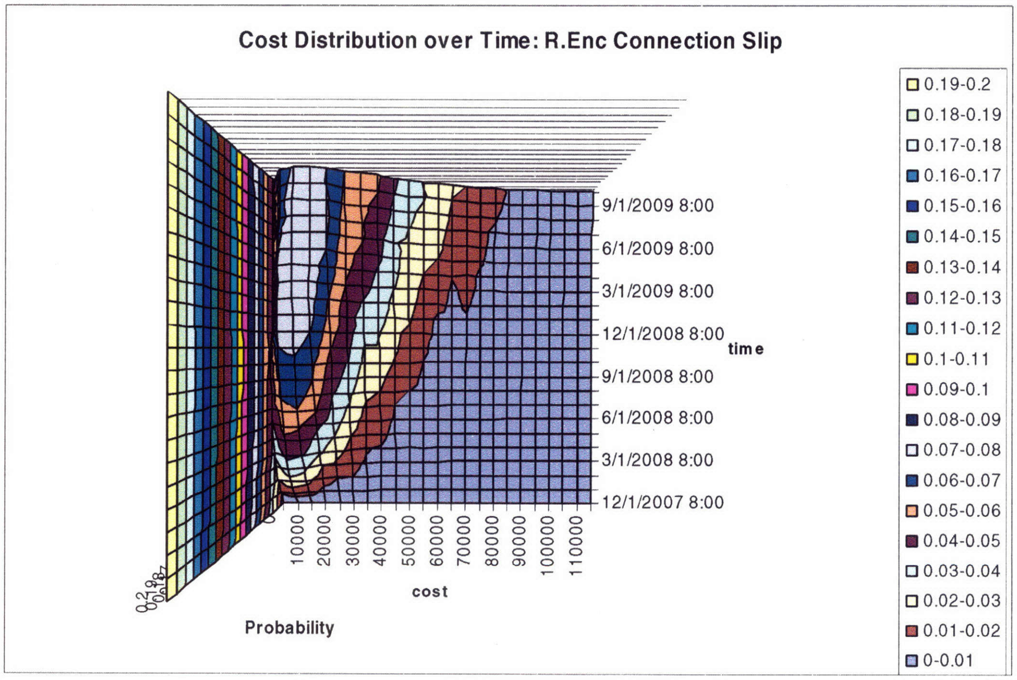


Figure 38 - Cost Distribution over Time: Res Enc Con Slip (Top)

Interesting to note that close to time 0 (the month of dec-2007), most of the volume of the distribution is concentrated on the lower left. The reason is that early in the timeline, there is a very high probability of having cost of $0. As time progresses, some of the volume is transferred from the left to the rest of the distribution, as can be seen in Figure 39. Note that the original configuration has a relatively concentrated distribution on the left comparing with the other configurations during the first year. Over time, all configurations become lower and wider spread to the right.

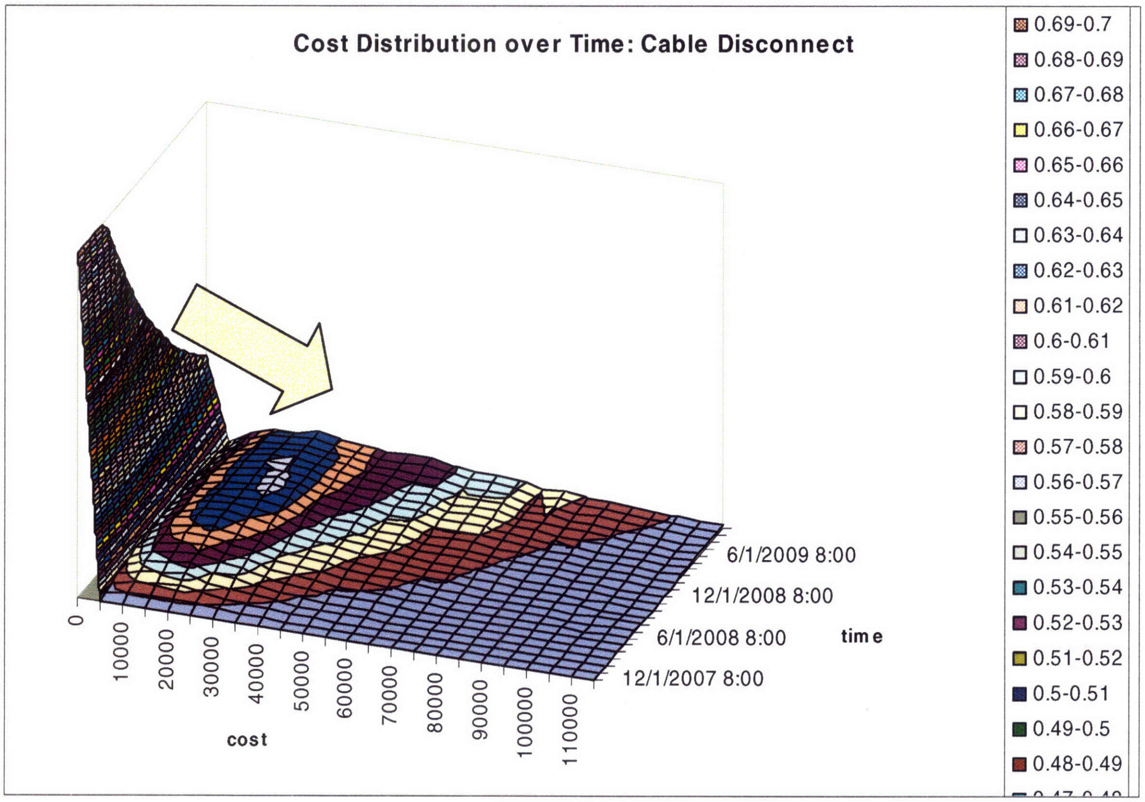


Figure 39 - Cable disconnect distribution over time - larger scale

Cost Benefit Analysis: IT Architecture Decisions and Business Alignment

One the results of this case study is the ability to compare architectures using a number of measurements. The measures of reliability and availability are always important, but are opaque in most cases because of system complexity. Furthermore, the ability to measure the expected loss of each architecture design decision, allows choosing the best design to fit the goals of the organization, as well as justifying that decision. In the case study, if the two cable-misconnect lapses were two architectures that an organization should have chosen from, this decision would have been a very difficult one to make and most likely would have been done in a qualitative manner. With the suggested methodology, this decision and even more complex ones can be made. Also, in most cases, an architecture decision would have a cost associated with it (the cost for doing the work and the cost of replacing or adding components). Figure 24 on page 78 shows the expected loss for six months. The difference between the disconnected cable configuration and the original configuration can be found by abstracting two points from the same time on the graph. However, as presented in the cost benefit section, comparing two expected values may be misleading. After understanding better how the distribution of cost / losses change over time, we can isolate one point in time, a year after the beginning of the timeline:

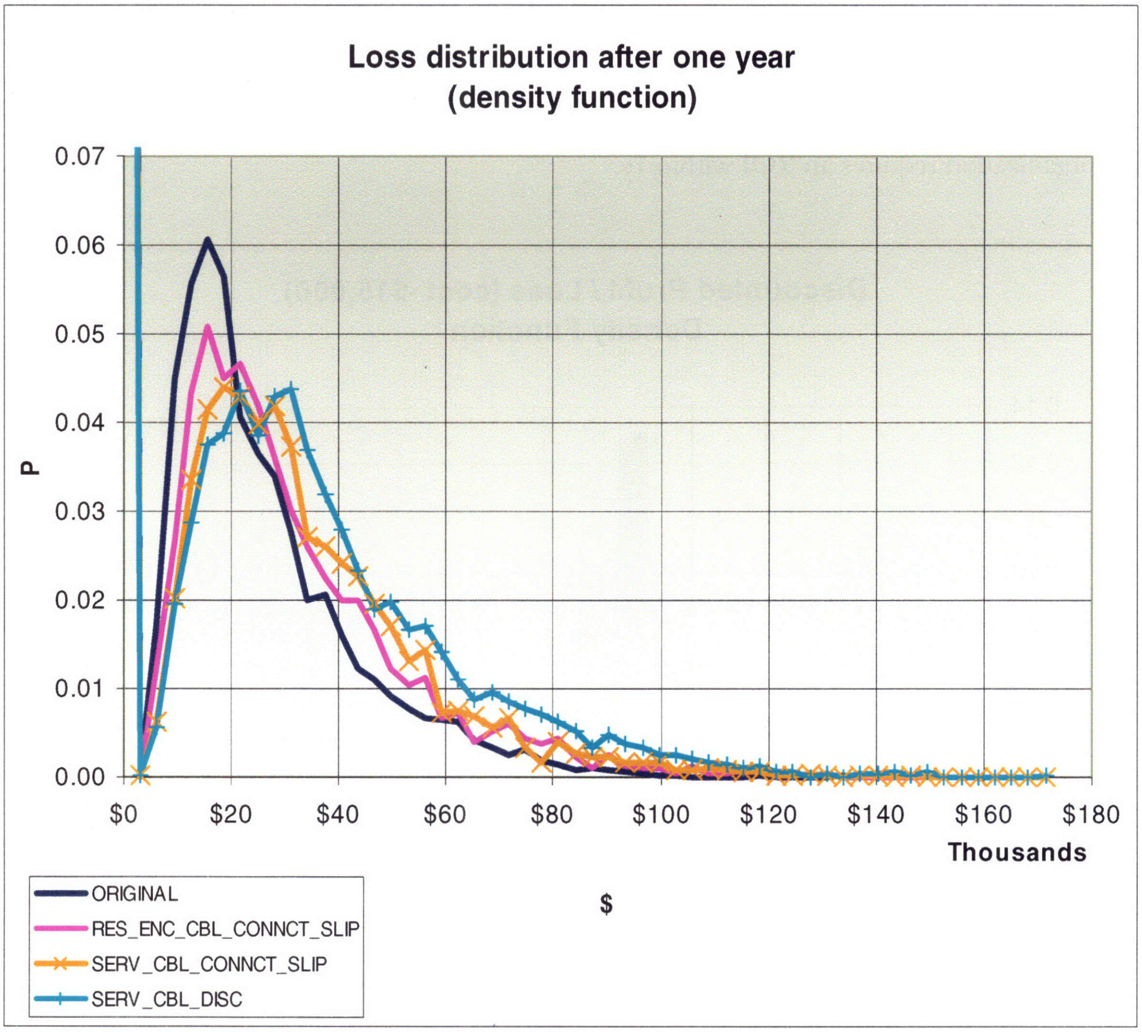


Figure 40 - Loss distribution after one year for all configurations

Each one is a random variable and for the following example, we will use the original configuration distribution and the cable-disconnect configuration distribution.

Let's consider the following example: Assuming that a cable disconnected and in three different locations:

a. The cable was disconnected in the data center. The equipment belongs to a customer that will pay for fixing it is $15,000.

b. The cable was disconnected in the data center in an accessible place. The cost of fixing is $0.

c. The cable was disconnected in a remote site in a desert. The cost of sending a technician is $8,000

The organization requires an ROI within 1y.

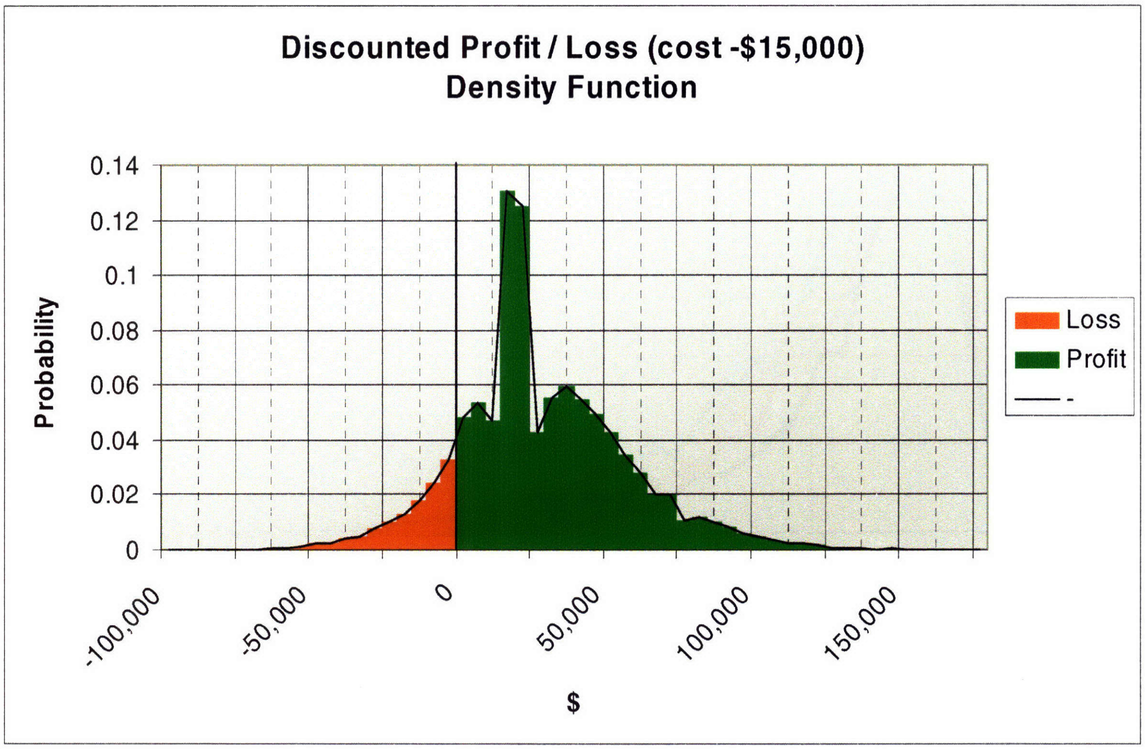


Figure 41 - Discounted profit / loss, for payment of $15,000

The discount rate is 12%. In the first case (a), the distribution is shown in Figure 41. The cost is -$15,000, the expected discounted profit in that case is $26,564, and the probability of loss is 17.16%.

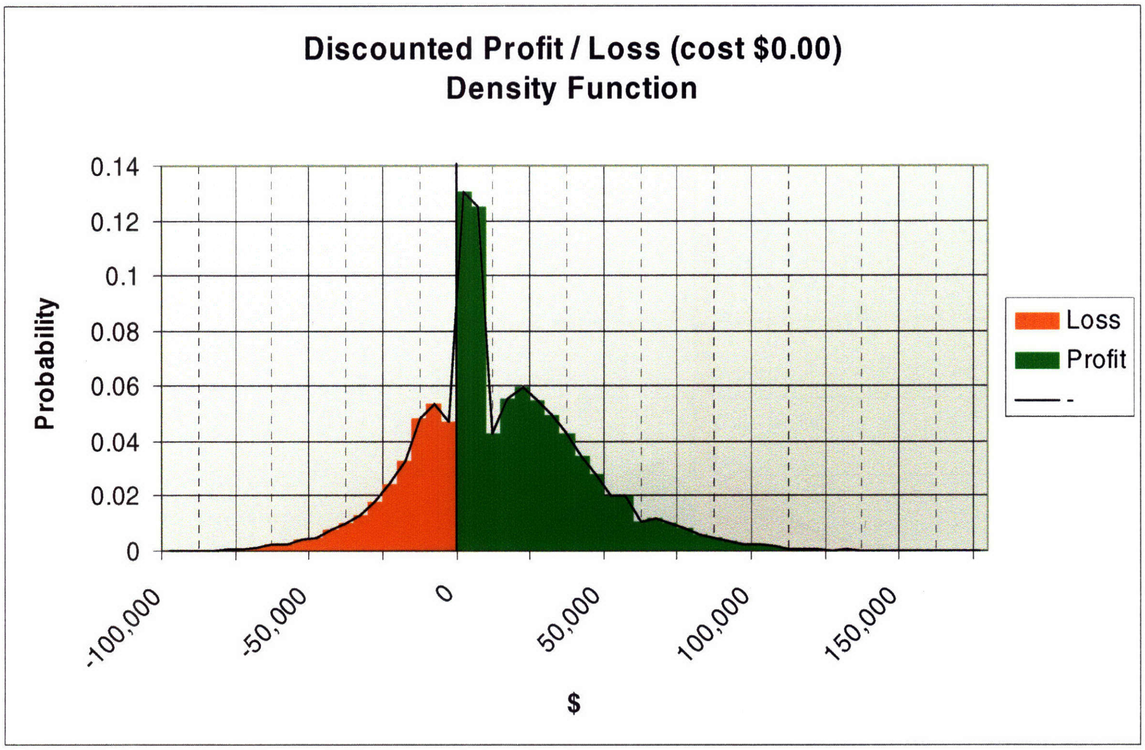


Figure 42 - Discounted profit / loss, for no cost

In the second case (b), the distribution is shown in Figure 42. The cost is $0.00, the expected discounted profit is $11,717 and the probability of loss is 40.03%

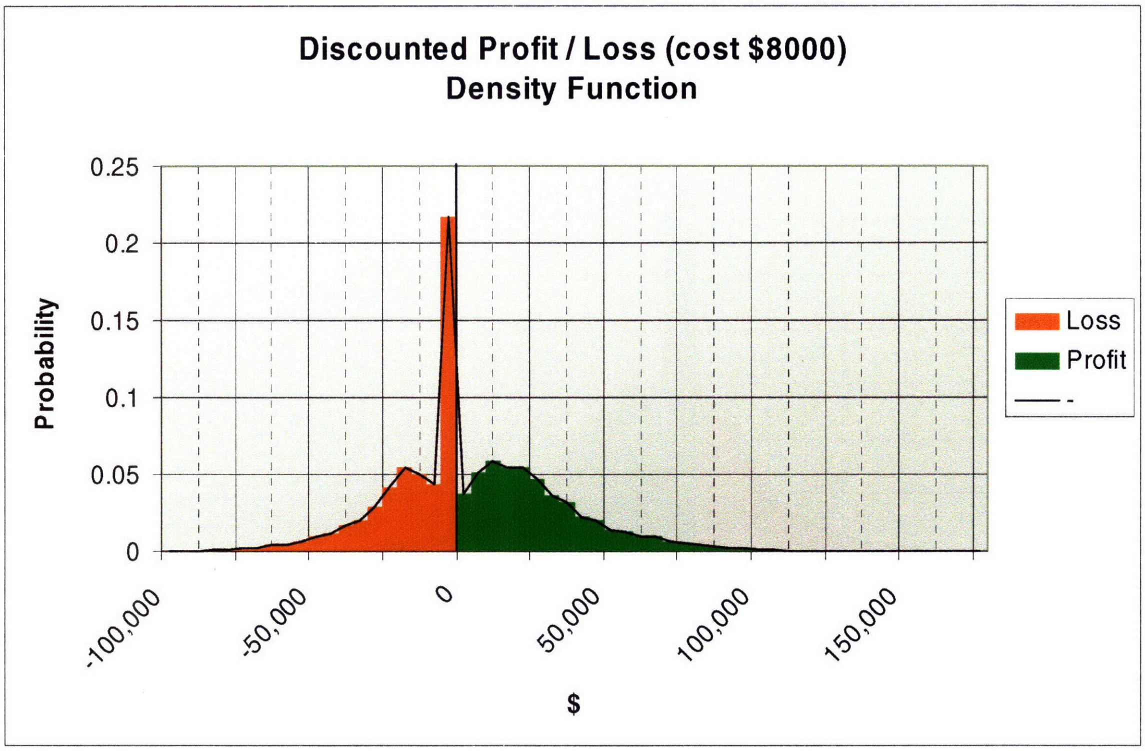


Figure 43 - Discounted profit / loss, for cost of $8,000

In the third case (c), the distribution is shown in Figure 43. The cost is $8,000, the expected discounted profit is $3,890, and the probability of loss is 54.80%. It is interesting to note that while the expected profit may be considered worthwhile, as a result of the variance, a loss probability of more than 40% indicates that options b and c may not be good options.

Addressing Exposure

The case presented two types of information that can be used to address exposure.

First, the importance measure point to the elements in the network that would generate the highest impact. The high impact can be a result of the fact that the element will affect a very costly service(s) the setting of the element in the system. It is also possible that the service(s) impacted are not very costly but the probability of failure is high. In any case, the importance measure suggests a very simple path of action: add the level of redundancy for that element.

CONCLUSIONS AND FURTHER WORK

The goal of this research was to examine how probabilistic methods of risk assessment that are applied successfully in certain industries can be enhanced and applied to service oriented, complex and frequently changing IT infrastructures. As IT infrastructure becomes an important enabler and sometimes even the platform of conducting business for different industries, the stakes involved with the failure of a service become very high. The ability to better asses the risk provides an opportunity to better manage that risk.

As noted by previous research efforts, there are a number of challenges in using methods such as PRA with IT infrastructures. The inherent physical and logical complexity in IT systems renders traditional methods such as PRA to be non-scalable. Furthermore, IT infrastructures are systems that contain both equipment and people (designers and operators), and both have a crucial impact on the risk of service failure. Although well recognized and documented, the human factor is usually not a part of most of the traditional PRA methods.

The Implication of a Change

Benjamin Disraeli said, "Change is inevitable. Change is constant". This thesis posed a number of research questions, but throughout there is a common thread: how can the implications of a change in the infrastructure be found and expressed, in an actionable way, in terms of how they impact the business. Providing a way to measure the impact of a change leads to additional questions, such as how to model and include human error as a substantial source of unintended change, how to account for infrastructure logical configuration, which physical elements and logical elements are more important than others, and what analytic method would best lend itself to deal with the estimation of the change implications. Finally, having answered the previous questions, how can these answers be applied to perform cost benefit analysis.

Two methods were considered for estimating the implication of a change from a stochastic point of view. While traditional PRA methods are successfully applied in nuclear and aerospace installations, the logical and combinatorial complexity inherent to network infrastructures limits the ability to get an accurate result. Furthermore even a relatively simple communication network that contains logical configurations leads to extremely complex PRA calculations and does not scale well. An alternative method comprised of a combination of Cellular Automata and Monte Carlo proved to be applicable within reasonable assumptions and enhancements (with regards to previous research). The thesis suggests a way of incorporating the wider context of a corporate IT infrastructure – including the human behavior and errors affecting the system. Using that methodology applied to a case study, this thesis showed how reliability and availability over time can be calculated.

Application of the Results

The thesis further showed how the profit (or the lack of profit), from a service perspective, can be used as a common metric to measure the expected loss over time and the importance measures for each of the physical components, as well as the importance of different logical configurations. Under different network topologies, logical configurations may become single points of failure (even though they are non-physical). The importance measurement for logical configurations and the ability to measure it is especially interesting because during literature review, no other related work with similar results was found. The case study showed how network changes that may seem only slightly different from each other lead to clearly distinguishable differences in the importance measure results. As a result, the suggested method may be a very good tool for comparing different architectural alternatives or to understand why two configurations are different and the implications of that difference.

The change of probabilistic distribution of loss over time was studied and based on a future distribution a financial risk benefit analysis model was presented showing both expected value as well as the distribution of the result. While certain engineering actions showed an expected profit, the distribution indicated a substantial probability for a loss. The expected profit or loss as well as the probability for a loss or a profit are all parameters that should be taken into consideration in a decision making process.

The ability to quantify the impact of a change, the results of the process described in this thesis can be used in a number of ways. First, given some operational data (or reasonable assumptions) an organization can estimate the reliability, availability and importance measures of the IT networking infrastructure. Second, this information can be used later to asses where are the weak points in the network, from a company level point of view. The weakest points can be reinforced by adding layers of redundancy, be that a physical component or a logical configuration. Third, multiple proposed engineering changes can be assessed in terms of cost benefit analysis in order to choose the option to maximize the expected return while controlling the probability of a loss.

Limitations of This Research

While this thesis found a reasonable working approach to estimate probabilistic measures for IT network infrastructure, a considerable effort was needed to achieve these results both in terms of code and in terms of actual computing resources. Although these results would have been much harder to achieve using traditional PRA, it remains to be seen how well the suggested method scales. As discussed in the description of CA, this limitation is not inherent, but any attempt to commercialize this method would require a major effort around scaling up to larger IT infrastructures.

An inherent issue in this thesis is the fact that the proposed method aims to model human error and incorporate it into a larger MC and CA model. In a real life environment, people that operate the network may find such an attempt to be threatening and potentially exposing, a challenge which may ultimately reduce the usefulness of using such a method. Furthermore, human error was modeled in a very simplistic way. As human error has high impact on the results, a different modeling for human error is likely to prove very sensitive to the results.

Finally it is interesting to note that if this method is a good indicator for future impact or exposure (expected losses) and its results would be implemented on an on-going basis, its results may become less accurate. On going correction of latent issues indicated by this method will over time render them to be inaccurate, because the basic assumption that an issue remains latent as long as there is no interruption of service will become inaccurate.

Further Work

The conclusions and the limitations above lead to a number of promising areas that were identified during the research, as possible interesting further research.

Human error is a complex phenomenon. This thesis used a relatively simple model for human error. Further work can be done around utilization of more advanced models such as the models suggested by Reason (Reason, 1990).

The thesis touched upon epistemic modeling of the world but did not go very deep into that direction. Further work can be done in order to establish how an epistemic model can be used. Using an epistemic model would address the last limitation mentioned above by constantly updating the basic probabilities of errors and failures. That would also require a detailed categorization of all physical failures and human errors. Each category would have it own failure characteristics that would change over time based on data accumulated about failures over time.

A common cause failure is yet another area that was mentioned but not examined in detail. This area would make an important possible research subject, since within the IT infrastructure there are many possible common cause failures. One example is a single firmware shared by a large number of IT elements may lead to a uniform collapse of a large part of an IT infrastructure. Another example is sharing the same physical location such as a data center that suffers a natural disaster.

Finally, modeling weather and natural disasters can be another relevant further direction of research. Dealing with weather and disasters is very important for an organization and should be taken into consideration when planning a multi-site infrastructure. Modeling disasters (fires, earth quakes, floods, hurricanes, etc).is challenging, but since some may have cyclical patterns, their probability distribution functions may lose the attribute of "memorylessness", an attribute that the framework developed for the thesis relied on. Can a distribution without memorylessness be incorporated into the model? How would conditional probability distribution be used? These would make interesting questions for further research.

APPENDIX A – DEFINITIONS

1. **BSN**: Binary State Network – a network where element failure is binary (working or not working). See also MSN (Multi State Network)

2. **CA:** Cellular Automata - A cellular automaton (plural: cellular automata) is a discrete model studied in computability theory, mathematics, and theoretical biology. It consists of a regular grid of cells, each in one of a finite number of states. The grid can be in any finite number of dimensions. Time is also discrete, and the state of a cell at time t is a function of the states of a finite number of cells (called its neighborhood) at time t - 1. These neighbors are a selection of cells relative to the specified cell, and do not change (though the cell itself may be in its neighborhood, it is not usually considered a neighbor). Every cell has the same rule for updating, based on the values in this neighborhood. Each time the rules are applied to the whole grid a new generation is created (Wikipedia, CA, 2007b)

3. **IT**: Information Technology - as defined by the Information Technology Association of America (ITAA), is "the study, design, development, implementation, support or management of computer-based information systems, particularly software applications and computer hardware." IT deals with the use of electronic computers and computer software to convert, store, protect, process, transmit and retrieve information, securely (Wikipedia, IT, 2007a).

4. **MC**: Monte Carlo Method - Monte Carlo methods are a widely used class of computational algorithms for simulating the behavior of various physical and mathematical systems, and for other computations. They are distinguished from other simulation methods (such as molecular dynamics) by being stochastic, that is nondeterministic in some manner – usually by using random numbers (in practice, pseudo-random numbers) – as opposed to deterministic algorithms. Because of the repetition of algorithms and the large number of calculations involved, Monte Carlo is a method suited to calculation using a computer, using many computer simulation techniques (Wikipedia, MC, 2007c).

5. **MSN**: Multi State Network – a network where element failure is multi state, representing different levels of capacity. See also BSN (Binary State Network)

6. **NP**: In computational complexity theory, NP ("Non-deterministic Polynomial time") is the set of decision problems solvable in polynomial time on a non-deterministic Turing machine (Wikipedia, NP, 2007d).

7. **NP-hard**: NP-hard (nondeterministic polynomial-time hard), in computational complexity theory, is a class of problems informally "at least as hard as the hardest problems in NP." NP-hard problems may be of any type: decision problems, search problems, optimization problems (Wikipedia, NP-hard, 2007e).

8. **PRA**: Probabilistic Risk Assessment – a process that is comprised of A. defining the end states of a system, B. Identification of initiating events, C. Development of event and fault trees, D. Quantification. The objectives of PRA are: 1. Identification of accident scenarios, 2. ranking these scenarios according to their probabilities of occurrence, 3. Rank systems, structures, and components according to their contribution to various risk metrics (Apostolakis, 2006).

9. **PRNG**: Pseudo Random Number Generator. An algorithm that generates a set of numbers that imitate the properties of random numbers. The numbers are not truly random and are completely determined by a small predetermined set of values. PRNGs are used in the context of cryptography and Monte Carlo Method simulations.

10. **SOA** - Service Oriented Architecture is an architectural style that guides all aspects of creating and using business processes, packaged as services, throughout their lifecycle, as well as defining and provisioning the IT infrastructure that allows different applications to exchange data and participate in business processes regardless of the operating systems or programming languages underlying those applications. SOA represents a model in which functionality is decomposed into small, distinct units (services), which can be distributed over a network and can be combined together and reused to create business applications. These services communicate with each other by passing data from one service to another, or by coordinating an activity between one

or more services. It is often seen as an evolution of distributed computing and modular programming (Wikipedia, SOA, 2007f).

REFERENCES

INDEX

BIBLIOGRAPHY

Aggarwal K. K., Rai Suresh "An Efficient Algorithm for Computing Global Reliability of a Network ", *IEEE Transactions on Reliability*, vol. R-30, no. 5, pp 32-35, 1981 13

Apostolakis, G., E., (2006), "Engineering Risk Benefit Analysis", Course Notes, RPRA 6, Spring Semester, Engineering Systems Division, MIT 108

Apostolakis, G., E., (2006), "Engineering Risk Benefit Analysis", Course Notes, RPRA 6, Spring Semester, Engineering Systems Division, MIT 57

Ball M. O., "Computing Network Reliability", *Opns. Res.*, vol. 27, Jul-Aug, pp 823-838, 1979 .... 13, 38, 40

Buzacott J. A., "A recursive algorithm for finding reliability measures related to the connection of nodes in a graph", *Networks*, vol 10, 1980 Winter, pp 311-327 13

Carr, N. G. (2003), "IT Doesn't Matter", *Harvard Business Review*, R0305B 8

Colbourn C. J. *The Combinatorics of Network Reliability*, Oxford University Press, New York, 1987 14

Elmallah E. S., AboElFotoh H. " Circular Layout Cutsets: An Approach for Improving Consecutive Cutset Bounds for Network Reliability ", *IEEE Trans. Reliability*, vol 55, no 4, 2006 December, pp 602-612 14

Hasanuddin A., "Simple Enumeration of Minimal Cutsets of Acyclic Directed Graph ", *IEEE Trans. Reliability*, vol 37, no. 5 1988 December, pp 484-487 13

Helton J.C., Burmaster D.E., " Special Issue on the Treatment of Aleatory and Epistemic Uncertainty" *Reliability Engineering and System Safety* vol. 54, no. 2-3 1996 57

Jain S., Gopal K., "An Efficient Algorithm for Computing Global Reliability of a Network", *IEEE Transactions on Reliability*, vol. 37, no. 5, pp 488-492, 1988 .... 13, 38

Lordish M., Mela C., (2007) "If Brands Are Built Over Years, Why Are They Managed over Quarters?", *Harvard Business Review*, R0707H, July-August 2007, pp 104-112 19

Matsumoto M. and Nishimura T., "Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator", *ACM Trans. on Modeling and Computer Simulation*, Vol. 8, No. 1, January pp.3-30, 1998 41

MySQL open source database, http://mysql.com/, 2007 67

Pentaho Reporting, http://www.pentaho.com/products/reporting/ 2007 67

Reason, J. (1990), *Human Error*, Cambridge University Press, New York, NY 9, 20, 21, 46, 65, 106

Repenning N., Sterman J., "Nobody Ever Gets Credit for Fixing Problems that Never Happened: Creating and Sustaining Process Improvement", *California Management Review*, vol.43, no.4, Summer 2001 20

Robert C.P. and Casella. G. "Monte Carlo Statistical Methods" (second edition). New York: Springer-Verlag, 2004 41

Rosenthal A., “Computing the reliability of complex networks”, *SIAM J. Applied Math.*, vol 32, 1977, pp 384-393 13

Satitsatian S., Kapur K., ”An Algorithm for Lower Reliability Bounds of Multistate Two-Terminal Networks”, *IEEE Transactions on Reliability*, vol. 55, no. 2, pp 199-206, 2006 14

Satyanarayana A., “A unified formula for analysis of some network reliability problems”, *IEEE Trans. Reliability*, vol 31, 1982 April, pp 23-32 13

Shanthikumar G., “ Bounding Network-Reliability Using Consecutive Minimal Cutsets ”, *IEEE Trans. Reliability*, vol 37, no 1, 1988 April, pp 45-49 13

Soh S., Rai S., ”An Efficient Cutset Approach for Evaluating Communication-Network Reliability With Heterogeneous Link-Capacities”, *IEEE Transactions on Reliability*, vol. 54, no. 1, pp 133-144, 2005 14

Wallin, S., Leijon V., “Rethinking Network Management Solutions” *IT Professional – IEEE Computer Society*, vol. 8, no. 6, pp. 19-23, 2006 .... 9, 17, 25, 40, 46

Wikipedia, “Information Technology”, http://en.wikipedia.org/wiki/Information_Technology (11-8-2007) 2007a 107

Wikipedia, ”Cellular automaton”, http://en.wikipedia.org/wiki/Cellular_automata (11-8-2007) 2007b......... 107

Wikipedia, ”Monte Carlo method”, http://en.wikipedia.org/wiki/Monte_Carlo_method (11-8-2007) 2007c 107

Wikipedia, ”NP (complexity)”, http://en.wikipedia.org/wiki/NP_%28complexity%29 (11-10-2007) 2007d 108

Wikipedia, ”NP-hard”, http://en.wikipedia.org/wiki/Np-hard (11-10-2007) 2007e 108

Wikipedia, ”Service-oriented architecture”, http://en.wikipedia.org/wiki/Service-oriented_architecture (11-18-2007) 2007f 109

Zio, E., Podofillini, L., and Zille, V., "A combination of Monte Carlo simulation and cellular automata for computing the availability of complex network systems", *Reliability Engineering and System Safety* 91, pp.181-190, 2006 .... 14, 40, 45, 50