University of Piraeus
School of Information and Communication Technologies
Department of Digital Systems

Survivable Mobile Telecommunication Systems.

PHD Thesis
MYKONIATI MARIA

Piraeus, July 2022

University of Piraeus
School of Information and Communication Technologies
Department of Digital Systems

Survivable Mobile Telecommunication Systems.

Supervisors: Prof. C. Lambrinoudakis

This thesis is submitted for the degree of
Doctor of Philosophy

Piraeus, July 2022

APPROVAL SHEET

UNIVERSITY OF PIRAEUS

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGIES

DEPARTMENT OF DIGITAL SYSTEMS

This is to certify that the Thesis presented by Mykoniati Maria, entitled **"Survivable Mobile Telecommunication Systems",** submitted in fulfillment of the requirement for the degree of Doctor of Philosophy, complies with the regulation of the University of Piraeus and meets the accepted standards with respect to originality

Costas Lambrinoudakis Professor University of Piraeus (Supervisor)
Sokratis Katsikas Professor NTNU
Stefanos Gritzalis Professor University of Piraeus
Christos Xenakis Professor University of Piraeus
Athanasios G. Kanatas Professor University of Piraeus
Kalloniatis Christos Associate Professor at University of the Aegean
Demosthenes Vouyioukas Professor University of the Aegean

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this thesis are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This thesis is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. I additionally declare that the opinions expressed in this document are my sole responsibility and do not necessarily represent the official position of the University of Piraeus.

Μυκονιάτη Μαρία
Piraeus, March 2022

Acknowledgements

I would like to thank my supervisor Dr. Konstantinos Lambrinoudakis for his guidance, feedback, and precious support, throughout the current research.

Abstract

Survivable systems are systems that have the ability to maintain their critical services and to make them available in a timely manner even in cases of accidents, failures or attacks. The main objectives of a survivable system are robustness, fault tolerance, security and recovery of its critical services. An open issue in the field of survivable systems is the determination of standard processes which ensure that a service is indeed survivable. During the current research, general approaches on providing and evaluating system's survivability are firstly examined. Secondly, the applicability of these approaches on telecommunication systems is investigated, with final target the provision of a framework in form of Software Development Lifecycle (SDLC), for design and implementation of a survivable mobile telecommunication system. The research has been based on investigation of survivability approaches already defined by 3GPP organization's standardization, which should be followed by any organization, which constructs nodes of a mobile telecommunication system. After presenting the already defined survivability controls, the current thesis extends 3GPP standardization survivability mechanisms to provide a SDLC for a survivable mobile telecommunication system. Then, application of framework proposed has been applied to 4G Voice Bearer Connection establishment as a case study. Finally, quantitative and qualitative results of the proposed process capability are presented, and conclusions are extracted.

Περίληψη

Επιβιώσιμα συστήματα είναι αυτά που έχουν την ικανότητα να διατηρούν τις υπηρεσίες τους λειτουργικές κατά την διάρκεια καταστροφών, αστοχιών, επιθέσεων κλπ. Ο βασικός στόχος αυτών των συστημάτων είναι η αντίσταση στην αποτυχία του συστήματος να παρέχει τις κρίσιμες υπηρεσίες του. Η παρούσα διατριβή περιγράφει αρχικά την έρευνα γύρω από τον ορισμό της επιβιωσιμότητας και των βασικών χαρακτηριστικών ενός επιβιώσιμου συστήματος. Στη συνέχεια, η έρευνα συνεχίζεται με τον καθορισμό των ιδιοτήτων που κάνουν ένα δίκτυο κινητής τηλεφωνίας επιβιώσιμο, έτσι όπως αυτές ορίζονται από το 3GPP Standard, και τη βιβλιογραφία. Δύο σημαντικές ελλείψεις σχετικά με τα χαρακτηριστικά ενός επιβιώσιμου συστήματος εντοπίζονται, και αναπτύσσονται δύο μεθοδολογίες που καλύπτουν αυτές τις ελλείψεις και ορίζονται ως απαραίτητες προϋποθέσεις που θα πρέπει να ληφθούν υπόψιν κατά την ανάπτυξη τέτοιων συστημάτων. Στη συνέχεια, η έρευνα ολοκληρώνεται με την ανάπτυξη της ζητούμενης μεθοδολογίας (SDLC) για την κατασκευή επιβιώσιμων δικτύων κινητής τηλεφωνίας. Η εφαρμογή της μεθοδολογίας γίνεται στη μελέτη περίπτωσης δικτύων 4^{ης} γενιάς (4G LTE Networks) ώστε να διερευνηθεί η αποτελεσματικότητα της σε σχέση με την προϋπάρχουσα μεθοδολογία που ακολουθείται από τις εταιρίες ανάπτυξης συσκευών δικτύων κινητής τηλεφωνίας. Τέλος μία ποσοτική και ποιοτική ανάλυση των αποτελεσμάτων παρουσιάζεται ώστε να φανούν τα αποτελέσματα της απόδοσης της προτεινόμενης μεθοδολογίας.

Contents

Glossary

MME: Mobility Management Entity

SGW: Serving Gateway

QoS: Quality of Service

PGW: PDN gateway

PDN: Packet Data Network

PDP: Packet Data Protocol

eNB: E-UTRAN Node B

HSS: Home Subscriber Server

VLR: Visitor Location Register

HLR: Home Location Register

SGSN: Serving GPRS Support Node

PLMN: Public Land Mobile Network

TAU: Tracking Area Update

RAU: Routing Area Update

GSM: Global System for Mobile communications

PSTN: Public Switched Telephone Network

LTE: Long-Term Evolution

KPI: Key Performance Indicator

GTP: GPRS Tunneling Protocol

UE: User Equipment

TMSI: Temporary Mobile Subscriber Identity

GUMMEI: Globally Unique MME Identity

RRC: Radio Resource Control

APN: Access Point Name

DDN: Downlink Data Notification

ARP: Address Resolution Protocol

ISR. Idle mode Signalling Reduction

EPS: Evolved Packet System

NAS: Non-Access Stratum

AKA: Authentication and Key Agreement

E-UTRAN: Evolved UMTS Terrestrial Radio Access Network

HO: Handover

1. Introduction

Availability and continuity of the critical information technology infrastructures is a matter of concern in a lot of scientific fields like security, robustness, fault tolerance etc. Actually, the unavailability and failure of critical information technology infrastructures is a synonymous of great economical loss.

A main objective is the survival of the information technology infrastructures, like information systems or network systems. This means that these systems should serve their critical services even during attacks, failures or accidents. The scientific field that involves these disciplines is called survivability with **security, robustness, fault-tolerance and recovery** of systems to be among survivability's main disciplines.

A very accurate definition of the survivability may be described by the following definition: *"survivability is the capability of a system to fulfil its mission, in a timely manner, in the presence of threats such as attacks or large-scale natural disasters". [1].*

What may be firstly observed is that survivability is focused on the survival of the **mission** of the system and not on the survival of the system itself. This is the core principle of survivability.

There is much research on survivability measures and approaches that should be performed and adapted by a system in order to be characterised as survivable. As part of Research and Development team of an organization that develops such systems, I discovered that what should be improved is the perception of managing and handling system failures. All possible failures should be considered through an organized way in order to decrease the possibility of service failure. Additionally, whenever it is possible, the system shall provide a "recovery from failure" mechanism. The current research focusses on examination of current methodologies for providing survivability to systems' mission, and focusses on providing solutions for missing survivability methodologies, so that a general framework for developing a survivable mobile telecommunication system to be implemented.

During the literature review, survivability as a term is examined so that the main principles and requirements of developing a system of which the critical services may survive from any kind of failure, to be defined. Additionally, mobile systems and their principles regarding survivability are presented. Finally, any survivability control deficiency is pinpointed.

What follows is the presentation of survivability controls proposed by the current research in order to cover any survivability control deficiency. The controls proposed are focussed on recognition of failure and resistance to failure. Furthermore, a general framework in form of software development lifecycle for mobile telecommunication

systems is presented. The main characteristic of these systems is that they should be considered as multi-layered since most of the times network nodes (ex. MME, SGW etc) are connected to form a system (2G, 3G, 4G etc) which is connected to other systems to form an intersystem. Considering this intersystem as a whole when a new functionality is under development, interoperability threats and properties from interconnection of network nodes, systems and environment will be considered instead of just focussing on a single node to provide a new functionality. In other words, by this way, when a new functionality arrives "legacy code", which is referring to features that are already implemented will be considered.

Continuing with survivability, the current research focuses on 4G mobile system, in the fourth chapter, as a case study to the framework presented. Results from using the proposed methodology will be gathered and analysed.
Finally, conclusions and future work will be described.

2. Scope

Scope of the current thesis is to provide a framework for design and implementation of survivable mobile telecommunication systems. To accomplish this, detailed research on literature related to survivability has been conducted. Additionally, research has been extended to cover all already defined **survivability requirements** for telecommunication systems. This research includes literature review, and anything defined by 3GPP which is the standardization followed by all organizations that build nodes for telecommunication networks. After gathering all survivability requirements proposed through literature, the current thesis focuses on survivability control deficiencies, and tries to cover two deficiencies that have been depicted. The first one is the incomplete approaches for **self-diagnosis** of failure, and the second one is the lack of appropriate **self-configuration** approaches based on failure. To continue, the next step of the current research is to define two methodologies that could cover these two deficiencies that are considered the main deficiencies found through literature review. After having also defined these survivability controls as system requirements, all survivability requirements for building a survivable telecommunication system have been defined. The next part of the scope of the current thesis is the presentation of the **Software Development Lifecycle** for building a survivable mobile telecommunication system. More emphasis has been paid on requirements and testing phases of the SDLC proposed since design and implementation are organization centric. Proposed SDLC has been applied to build a critical service. As a case study a voice bearer establishment from 4G mobile system has been used. Results on the case study have been presented to discover if project objectives have been met. Finally, conclusions and future work will be presented.

2.1. Research Objectives:

1. Theoretical research on definition of survivability and on methods of providing survivability of system's mission.
2. Extending already defined survivability measurements by 3GPP or literature, to cover any missing survivability controls.
3. Constructing a framework for developing survivable mobile telecommunication systems.
4. Apply methodologies proposed to 4G Voice Bearer establishment procedure, that will be used as a case study of a critical service.
5. Gather results and analyse them in order to ascertain that proposed process's capability is improved against service and system failure.
6. Present conclusions and future work.

2.2 Research Contribution

Survivability of systems' critical services is of vital importance for today's systems. Survivability's main objective is the continuity of these system's critical services in a timely manner, even if the system is under any kind of attack, failure or disaster. For this to be succeeded, the system itself should not of course be out of scope, but continuity of services shall be the main objective, even if the system is not operating at all. The first part of the contribution of the current thesis is the provision of extensive **literature review on theory of survivability** and the definition of survivability's' main objectives that should be followed during design of any system with critical services.

Being a software developer of NOKIA 4G mobile networks for 6 years, I was always trying to find possible ways to ensure system's critical services continuity, mostly based on Software Development and Quality Assurance or Testing approaches. So, the current thesis focuses on mobile network services survivability from Software Development Lifecycle perspective and on validation of services survivability through Software Testing perspective. In other words, the research focused on how to develop a network node that will provide survivable services and how to verify that the services running over the implemented system are indeed survivable. For this to be succeeded, extensive research on **already implemented survivability controls** by 3GPP, which is the standardization for mobile networks, and by approaches described through literature review, took place and results of this research are presented to the current thesis. This also constitutes part of the current research contribution.

After defining the currently implemented survivability controls, the current research describes the deficiencies discovered. Firstly, the approaches described seem to be system centric and not service centric. By this what is meant is that for example, the solution indicated by 3GPP in case of node failure, if the failure has been recognised, is all other nodes to delete all connections with this node and wait for the node to be restored. Though, these services, that are dropped, might be critical and there is no temporary solution for all services that will be lost until the node is restored. Additionally, there is not always solution provided in case the failure has not been recognised by the system. Finally, what if the network node is not under failure but certain services fail because of a software error?

Part of the current research contribution is the proposal of a **framework for prediction of possible failure of a service (number 1 paper of table 1 below)** when a network node is selected in order nodes with increased failure possibility to be avoided. This solution improves system's failure resistance. Additionally, part of the current research's contribution is providing a **framework for fault detection and categorization of failure (number 2 paper of table 1 below)** to improve failure **recognition** survivability controls which seem to be not taken into consideration by 3GPP.

The main contribution of the current research follows, which is the presentation of a

SDLC where, contrary to traditional SDLC approaches, **survivability is integrated in all phases of the SDLC (number 3 paper of table 1 below)** in order to provide a survivable system by design.

After defining the proposed SDLC, the current research applies the proposed methodology to 4G Voice Bearer establishment service, which constitutes a critical service of 4G Networks as defined by 3GG (please refer to literature review). Finally, comparison of the traditional SDLC used by organizations for building mobile networks and the proposed SDLC, shows the value of adopting the proposed methodology for improving survivability's main objectives which are recognition of failure, resistance to failure and recovery from failure due to literature review presented. So, the last part of contribution of the current research is presenting the **advantages of focusing on survivability** while building mobile telecommunication systems by investigation of appliance of the proposed methodology on a real service that is implemented to 4G networks.

	Research	Authors	Analysis
1	Fault Prediction Model for Node Selection Function of Mobile Networks	Mykoniati Maria, Costas Lambrinoudakis	This paper focuses on provision of a framework for system's **resistance** to failure.
2	Self-Diagnosis Framework for Mobile Network Services	Mykoniati Maria, Costas Lambrinoudakis	This paper focuses on provision of a framework for **recognition** of failure
3	Software Development Lifecycle for Survivable Mobile Telecommunication Systems	Mykoniati Maria, Costas Lambrinoudakis	The last paper describes the **SDLC** proposed for building a survivable telecommunication system.

Table 1: Papers related to the current research.

3. Literature Review

In this part of dissertation, the literature relevant to the scientific fields of survivability systems is to be examined. Additionally, the word system and its characteristics and requirements are to be presented so as the survivability systems (system of systems / critical systems etc.) to be examined thoroughly. Finally, the research will focus on survivability approaches of telecommunication networks.

3.1. Survivability

Firstly, what is to be presented is a definition of survivability and the definition of relative terms. During an extend research, there are a lot of terms that have been encountered and their definition must be presented in such a way that the similarities or differences with the term survivability to be revealed. Examples of these terms are the reliability, the dependability, the adaptability etc. Furthermore, some of these terms are countable and this may give a brief opening for measuring survivability.

Secondly, a research on the term system and its association with survivability is have been presented. General requirements that system should follow in order to be survivable have been presented.

3.1.1. Definition of Survivability.

Nowadays, there have been dramatic changes in information systems, and in network infrastructures. New highly distributed systems, with no central administration and with anonymous participants are introduced and play an important role in everyday life. These systems participate into various infrastructures which are of vital importance for the public, like the Public Power Infrastructure or Telecommunication Systems. This makes principles as the availability and the continuity problems of greater concern. Additionally, the operation of these infrastructures, their availability and continuity of service may rely on computer systems which must provide the system with these characteristics.

But what really may affect any system and what are the disasters that the systems must be protected from? The word **disaster** is any inability of the system to fulfil its purpose in a timely manner mostly because of attacks or failures of the system. Examples of such disasters are the presence of viruses, Trojans worms, network intrusion or failure, software and hardware failures, cut off power supply, human error, physical disasters like earthquake and fire or even Cyber terrorism. Though, it is important to pinpoint that in case of a business, the difference between a natural

disaster and an attack or failure, is that in the first case the customer is expecting a degraded performance in services provided and the fame of the company is less threatened than the second case. Survivability in general focus on principles applied before, during and after an attack to a system

By the changes to the needs of networked information systems, and by realising a *"fundamental assumption that no system is totally immune to attacks accidents or failures" [2]*, there is a need for reconsidering security issues from a different perspective. The main objective is not only to manage these issues, by for example setting up firewalls or applying recovery techniques, but also ensure that the **essential services** of a system **will be delivered,** and that the system will remain functional **even in the presence of catastrophic events**. This is succeeded with the introduction of the principles of survivability which enables the designing and installation of the system according to the principles of security, fault – tolerance and recovery, etc. By this, what is meant is that security is a feature that the installed system will have by default, if principles of survivability are applied.

Survivability is not only a technical issue but also an issue of all participants of the system and of the **management of the system as a whole**. This may include executive management, security experts, inside and outside participants involving with the system with the main objective to be to protect **critical services** from attacks, failures or accidents. By critical infrastructure or systems what is described is systems to which a minor damage may lead to a complete cessation of system services. A definition of survivability that communicates this idea may be an extract from Howard F. Lipson et al [2] of CERT institute: *"Survivability is an emerging discipline that blends computer security with business risk management for the purpose of protecting highly distributed information services and assets." [2].* From business perspective, contingency plans may not always be able to confront with disasters that may disrupt the critical networked information systems and degrade their performance in a way that the survival of the business to be an issue. The sustainability of these critical infrastructures and of the delivery of essential services is ensured by survivability principles. This ensuring is mostly based on the **risk management** of the business as the survivability is mostly focused on the **mission** of the business as the main objective.

Starting with an attempt to define survivability, there are a lot of definitions found, examined and presented. The most complete source for research on survivability has been proved to be the CERT Coordination Center of Software Engineering Institute of Carnegie Mellon University [2]. What is being supported by Howard F. Lipson et al [2] of CERT institute is that survivability is a new security perspective as it covers issues intractable by traditional security techniques.

The definition provided by Howard F. Lipson et al [2] of CERT institute is the following:

"Survivability is the capability of a system to fulfil its mission in a timely manner, in the

presence of attacks, failures or accidents." [2].

This is the definition used by most of the sources focussing on survivability. Focussing on exploring all available different definitions of survivability, Vickie R. Westmark [3] has been trying to answer the questions "What survivability means, what is composed of and how it may be measured or computed?". These are the questions that should be answered so as a complete definition of survivability to be officially formulated and standards on survivable systems to be developed. A list of all definitions available, definitions according to Vickie R. [3] may be depicted by APPENDIX – A.

Additionally, according to Vickie R. Westmark, for giving a "context – specific" definition of survivability, generally approved, the definition of terms "**system", "usage", "minimum level of service", "threats", "continuity", "time",** and their association with survivability must be examined and be part of the definition of survivability. Though, defining these terms is of vital importance while defining and examining certain systems from the scope of survivability.

System:

Firstly, the term **"system",** is the subject to which the principles of survivability should be applied to. Since usually these systems are large scale distributed network systems, in an undefined environment and with anonymous participants, the system that is to be examined must be defined as clearly as possible. The term system is examined in detail below.

Usage of system:

Secondly, the **"usage of the system",** or more precisely the end-user services must be clearly stated and defined. Furthermore, the "minimum level of service" acceptable from the user, due to functional specifications and predefined quality features must be clearly stated, so as the performance of the systems not to go beyond that level. What should be mentioned here is that these attributes are most of the time subjective and focused on the nature of the system and the service it provides. An example is given from Vickie R. Westmark [3]:

*"For a network **distributed system**, a required service may be a specified **response time** to the end user. For a **time – critical system**, the description of a required service may include the **maximum time allowable between user request and system response**." [3].*

Something similar is supported by John C. Knight et al [4]. What is supported is that the main objective of survivability is to keep the "more" functionality that may be provided, which is a synonymous to the "usual" and "expected" functionality by customers. John C. Knight et al [4] exactly support that, *"by defining survivability precisely, system designers can state exactly what the user of a system can expect over time in terms of the*

provision of service". [4] Supporting to this idea is the term "**mission**", presented at Howard F. Lipson et al [2], as the most critical objective of survivability. What is presented is that *"it is the mission that must survive, not any particular component of the system or even the system itself." [2].*

Threat:

Thirdly, the term **"threat"** must be clearly specified so as the purpose for application of survivability or security or fault tolerance etc. to be revealed. According to Vickie R Westmark [3], a threat for survivability is anything that may prevent the system from providing essential services at all, or under the "minimum level of service", as described before, or for more time than the one predefined as acceptable. Though, every system has at some level a capability to adapt to the threats and continue its functionality. This is described by the term "adaptability". For avoiding any degradation of the performance of the system, "scenarios" according to these threats must be developed taking into consideration the cost/benefit analysis of businesses.

Continuity:

Fourthly, the term that describes this challenge that survivability must overcome is the term **"continuity"**. Continuity of service and of certain quality of services is the key for survivability. Actually, what is surprising is that from survivability perspective, as long as there is continuity of services under certain requirements and quality features, no action is needed even though a threat is present. Additionally, according to John C. Knight et al [4] a damage to the system affects its dependability. As a result, a system that will absorb damage is of vital importance to be built. This system is supposed to be a critical information system due to John C. Knight et al [4]. Survivability for John C. Knight et al [4] is a way for a system to continue to operate after a damage or after a sequence of damages.

Time:

As for the term **"time",** it focuses on the interval to which the services must be available to the user according to system requirements' and to users' expectations. In other words, how much time the system may not provide its services or provide quality of service below the minimum acceptable.

Another definition of survivability is provided by John C. Knight et al: *"A system is survivable if it complies with its survivability specification" [4].* By the term specification what is meant is:

- a detailed definition of the **operating environment** of the system,
- a detailed definition of the **functions** of the system,
- a detailed definition of the **order in which certain functionality must be provided,**

– a definition of the **probabilities** that these specifications will be provided.

Finally, according to M. Al-Kuwaiti et al [5], survivability has been emerged from military needs and its approaches have been applied to critical systems and system of systems or complex systems. The threats, requirements and means of survivability are depicted at figure 1. at the right.

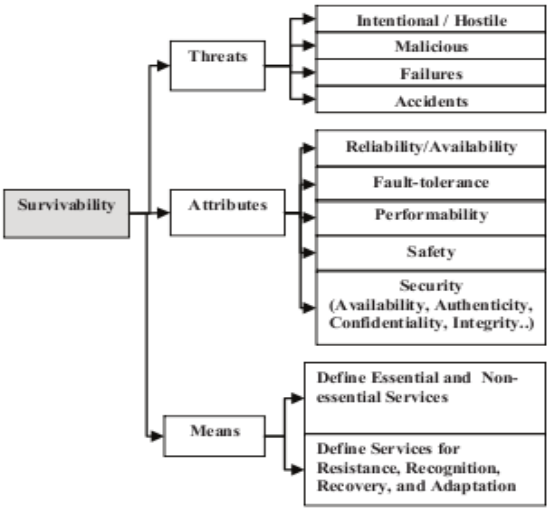


Figure 1: Survivability concept [5].

The only and single goal of survivability is the mission fulfilment in a timely manner and the full recovery of the system when possible.

By this definition, specifications of survivability seem to be a big concern. Additionally, the "usage" of the system is a combination of its well-defined functions or services and the probability that these functions may be provided. This imply a certain level of functionality and if this functionality may not be achieved, an order of functionality levels should have been defined so as the next accepted level of functionality to be applied. Finally, for John C. Knight et al [4], the environment of the system and its participants play a very important role for the succeeding of designing and implement a survivable system. As a result, it should be also defined precisely.

The whole image of survivability definition may be depicted by figure 2 bellow. What is depicted by this figure is that survivability is tightly connected to the quality (value) of service provided and it may be distinguished to two approaches. Firstly, the reduction of the likelihood of a disturbance, and secondly, the satisfaction of the predefined minimum acceptable level of service. Additionally, recovery from disturbance and the time needed for the system to recover up to that acceptable level seem to play a very important role.

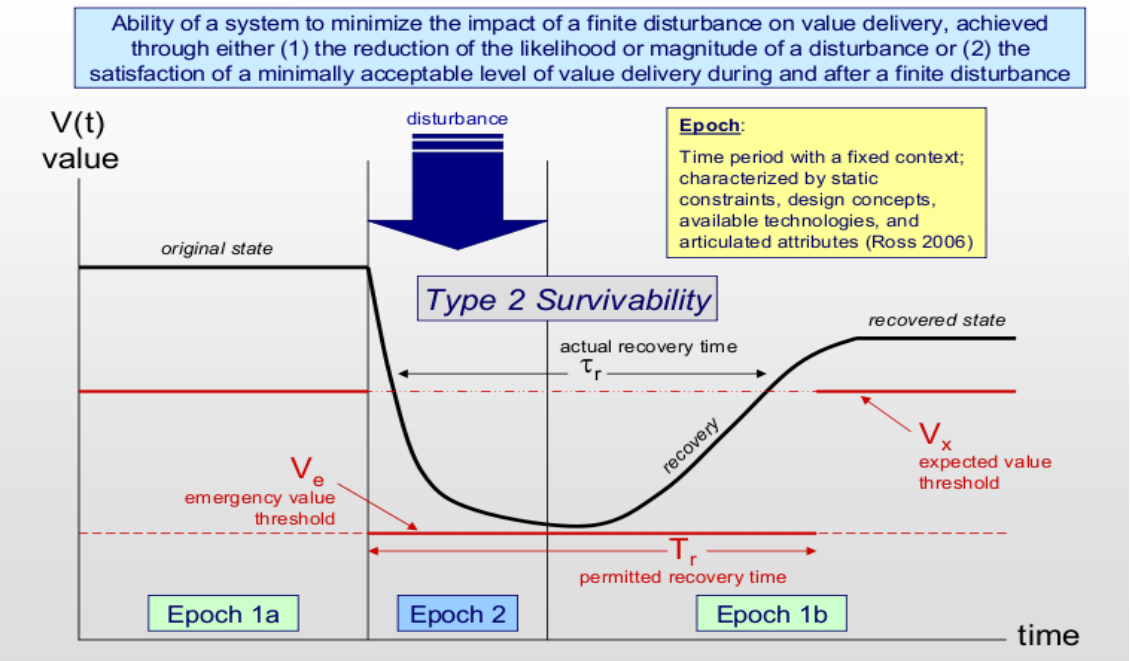


Figure 2: Survivability definition [6].

3.1.2. Related terms to Survivability.

While searching an exact definition for survivability, someone will come across various related terms and their definitions. As a result, before proceeding with survivability, an attempt of shortly defining these terms and their relationship to survivability is very essential. These terms may be **characteristics of survivability,** according to literature, they may have been characterised as **synonymous with survivability**, or they may **overlap with survivability**. Though, what should be pinpointed is that survivability itself is a property or characteristic of a system. More specifically, it is an emergent property resulting from a system designed in such a way (survivable system).

Firstly, what should be mentioned is that the main contents of survivability are **security**, **fault – tolerance** and **recovery** according to definitions of CERT [7]. As a result, all the characteristics of these terms are also characteristics of survivability. More precisely what has been presented by this Institute is that:

*"Survivability goes beyond **security** and **fault tolerance** to focus on **delivery of essential services**, even when systems are penetrated or experience failures, and **rapid recovery of full services** when conditions improve." [7].*

Though a most careful and detailed look should be taken upon some terms which are very similar or close to the term survivability.

According to references [3], survivability is a set of certain characteristics and quality features that may be found to quality standards like the ISO/IEC 9126. Though, by a closer look to these characteristics, what it may be observed is that they are characteristics that a system may have. As a result, it would be safe to imply that these characteristics may be the key for characterising a system as survivable. What may enforce this idea is a definition of survivability provided by John C. Knight et al: "*A system is survivable if it complies with its survivability specification." [4]*

A list of **characteristics of survivable systems** according to literature may be as follow:

1. Availability: "*The degree to which a system is in a specified operable and committable (available) state." [8].*

Mathematical expression: $$A = \frac{E[\text{Uptime}]}{E[\text{Uptime}] + E[\text{Downtime}]}$$

where E is the expected value, Uptime is the time the system is up and Downtime the exact opposite.

The highest value of availability is the five 9s = 0.99999 or 99,999% [8].

2. Software and Hardware Dependence: the degree to either of each depends on specific environments of the other.

3. Connectivity: "*the degree to which a system will perform when all nodes and links are available" [3].*

4. Correctness: "*to which degree a software component or an algorithm is correct according to some specification" [8].*

5. Endurability: "*the degree that a system can tolerate a threat and still provide services" [3].*

6. Fault – tolerance: Fault – tolerance is assumed to be one of the three most important components of survivability. It may be defined as the "*building of systems that are able to react in a requisite way to prescribed faults is fault tolerance." [4].* Fault – tolerance is a set of mechanisms and techniques used to avoid or attenuate the effect of threats to the output of a system. The faults may be masked, and their affects do not influence the services of the system. On the opposite case the faults are non – masked and in this case the damage is so extensive that the services of the system may not be delivered. Fault – tolerance mechanism is a characteristic by which if faults are masked properly survivability requirement may be achieved.

From the same source it may also be found that fault is an "*event that causes damage"* and **fault avoidance** is "*building a system in such a way that certain faults do not arise."*

[4]. Fault avoidance is also a very strong achievement for survivability as it may for example reduce the failures or eliminate a security attack by using the suitable measures respectively.

According to Pentti Tarvainen [9] fault-tolerance consists of four phases:

1. error detection
2. damage assessment
3. state restoration
4. continued services.

Additionally, according to Pentti Tarvainen [9] a fault tolerant system is a system designed to be so and its characteristics are reliability, availability and safety. It is a system that may improve systems' dependability accompanied with fault avoidance, fault forecasting and fault elimination.

The **faults** that the system will be tolerable to or that the system may avoid must be clearly defined. The way that this is going to be succeeded including mechanisms and processes must also be defined. For faults that have not been predicted the system will not be able to react under survivability specifications as it will not be "designed" to do so.

Finally, according to M. Al-Kuwaiti et al [5], **dependability** is *"the ability of a system or component to continue normal operation despite the presence of hardware or software **faults**."* The threats, requirements and means of dependability are depicted at figure 3. at the right.

Fault – tolerance is supposed to be a main objective of survivability, dependability and reliability and its means are of vital importance to a success to these properties.

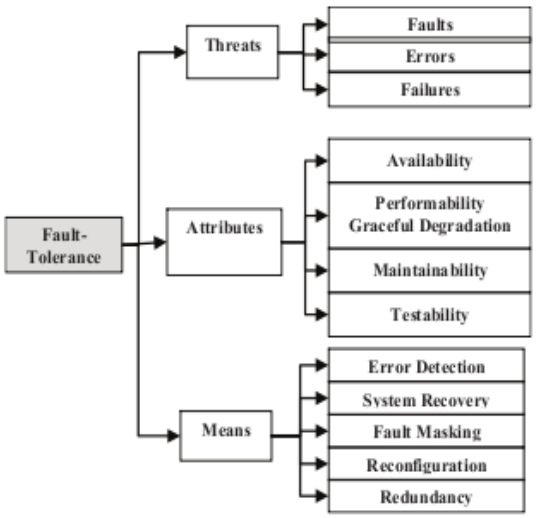


Figure 3: Fault-tolerance concept [5].

7. Interoperability: *"The degree to which a system may be connected and operate with other systems." [3].*

8. Modifiability: *"The effort required to modify the efficiency of the system." [3].*

9. Predictability: *"The degree of a systole providing countermeasures to failures in the*

event of a threat". [3]

10. Recoverability: "*The degree of a system providing countermeasures to failures in the event of a threat". [3]*

11. Restorability: "*The ability of a system to recover from **threat** and provide services in a timely manner". [3]*

12. Safety: "*The ability of a system not to cause harm to the network or personnel" [3].* Safety is describing what must never happen during the operation of a unit. Additionally, according to Matthew Richards et al [10], "*safety emerges from an aggregate of system components, subsystems, software, organizations, human behaviours, and their interactions.*

13. Secure-ability (Security): The degree to which a system is secure.

According to M. Al-Kuwaiti et al [5], security "*must encompass dependable **protection** against all relevant concerns, including **confidentiality**, **integrity**, and **availability** despite attempted compromises, **preventing denials of service**, **preventing and detecting misuse**, **providing timely responses to threats**, and **reducing the consequences of unforeseen threats**."*

Additionally, security protects the assets, predicts threats etc. In other words, it is the resilience of the system to attacks. The threats, requirements and means of security are depicted at figure at the right.

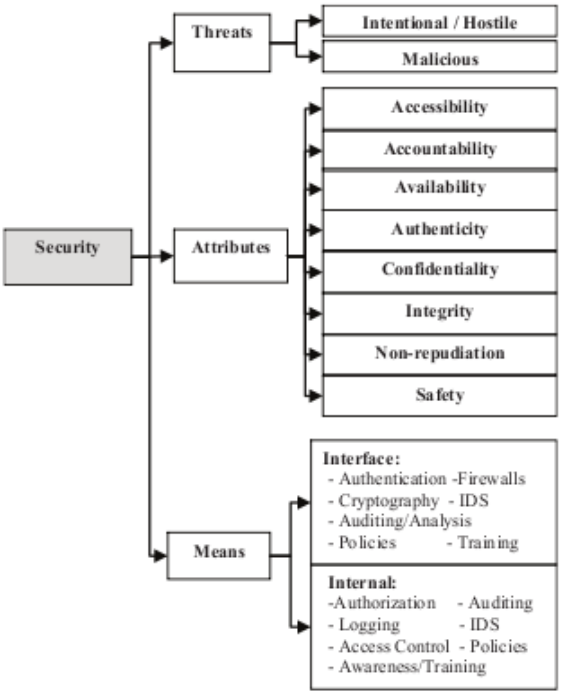


Figure 4: Security concept [5].

14. Maintainability: "*ability for a process to undergo modifications and repairs" [11].*

15. Sustainability [12]**:** it is the ability of a system to sustain. It includes tactics that the nature uses so as ecological systems to exist.

16. Resilience: Resilience is the ability of a system to return to normal state of services provisioning after disaster. According to Matthew Richards et al [10], **"***to be resilient,*

*systems must not only be **reliable** but also able to **recover** from disturbances through the design of proactive organizations and processes.***"**

Figure 5: Characteristics of survivable system.

Apart from the terms that may be characteristics of a survivable system, there are other related terms that have been implied to be synonymous or terms for which survivability is an objective. For example, in literature dependability is considered to be a synonymous of survivability and performance, for which survivability is an objective in accordance with many other factors.

1. Dependability: According to John C. Knight et al [4], survivability is a new dependability perspective. Dependability is *"the trustworthiness of a computing system which allows reliance to be justifiably placed on the service it delivers" [11].* The elements of dependability are *"to avoid mistakes, detect and remove errors and limit damage caused by failure" [13].* Its characteristics are availability, reliability, safety, integrity, and maintainability. As a result, dependability is a system's property like survivability, and it is succeeded after the compliance of the system with certain levels

of those characteristics. Though, the importance of these characteristics and their levels depends on the nature of the system that they are applied to. These features already imply a similarity between the two terms mostly on their subsistence and what John C. Knight et al [4] support, is that survivability is the dependability needed to be defined for critical networked infrastructure systems:

"The reason for making survivability a new aspect of dependability is that it is a primary form of dependability needed by critical networked infrastructure systems." [4]

Contrary to John C. Knight et al [4], Howard F. Lipson of CERT team [2] supports that dependability is not a synonymous with survivability as it is not only focused on software quality attributes, as dependability does, but it is focused on the *"functional and non-functional requirements determined by the mission" [4]* of the system. Additionally, the requirements of dependability are determined by the definition of dependability whereas the requirements of survivability are always focused on a **mission** and sometimes may be completely different from those implied by standard principles. For example, a software component may have no quality attributes according to dependability, but it may constitute to the whole systems' survivability.

Finally, according to M. Al-Kuwaiti et al [5], dependability is a more general sense of reliability and not so strictly mathematically described. The threats, requirements and means of dependability are depicted at figure at the right.

The exact definition given is:
"Dependability measures the degree to which a system is operable at any random time during a specific mission profile, given that its services are available at the start of the mission."[5]

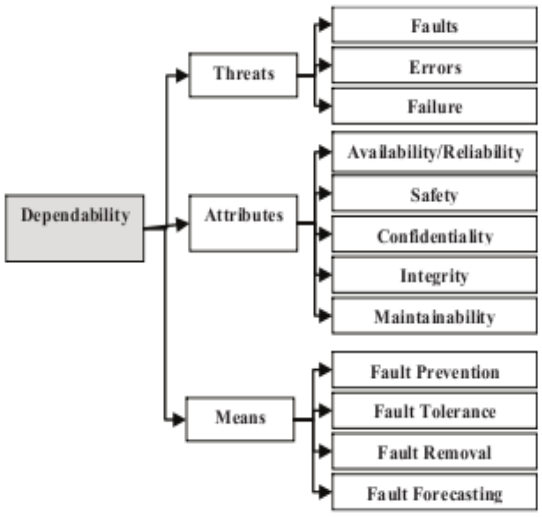


Figure 6: Dependability concept [5].

2. Viability: This term refers to the *"ability of companies to produce and deliver their products and services in a timely manner." [2]* This concept along with continuity are the extensions that should be added to security according to today's needs. Additionally, according to Stafford Beer [14], viable is a system *"capable of independent existence".*

3. Reliability: Reliability may be defined as *"a set of attributes that bear on the capability of system to maintain its level of performance under stated conditions for a stated period of time" [3].* According to references [15], a system may be reliable if it never fails

(informally) or if it meets a probabilistic goal (formally). This means that a system may fail but still be considered as reliable.

Additionally, according to M. Al-Kuwaiti et al [5], reliability may be mathematically defined.

The threats, requirements and means of reliability are depicted by figure at the right. They support that reliability and availability are very close terms with the only difference to be that *"reliability refers to failure-free operation during an interval, while availability refers to failure-free operation at a given instant of time" [5]*

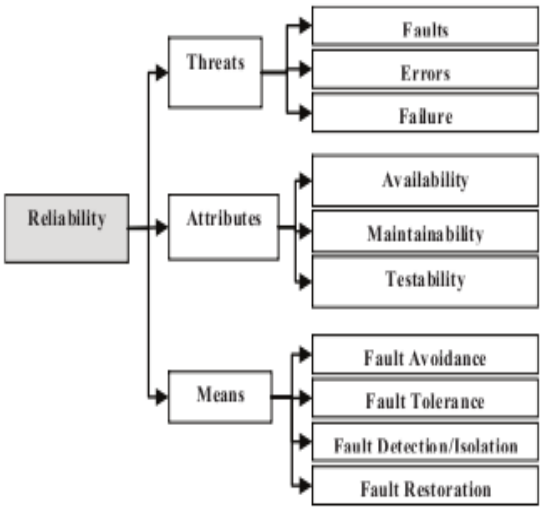


Figure 7: Reliability concept [5].

4. Changeability: According to Matthew Richards et al. [10], survivability is *"conceptualized as a meta-framework for robustness and changeability."* A changeable system, by the same source *"is able to have its design variables modified by either an internal change agent (adaptability) or by an external change agent (flexibility) to maintain or improve value delivery in the presence of shifting environments and requirements".* In other words, it is the ability of a system to maintain its services values in a changing environment.

5. Robustness: According to references [16], *robustness "is the ability of a computer system to cope with errors during execution or the ability of an algorithm to continue to operate despite abnormalities in input, calculations, etc."*

The main objective of a system, according to Matthew Richards [10], is the robustness the unchangeability and adaptation in different levels of the system. What is also supported by Matthew Richards [10] and may be depicted by figure below is that the characteristics may be following three different axes and as a result, may be gathered to three categories:

1. The physical system – focus on design parameters.
2. The stakeholder values – focus on utility function.
3. The environmental context – focus on environmental factors.

The system's design parameters may be modified so as improvement to value of services to emerge even in changing design parameters. Changeability is on physical axis in order to provide physical robustness. The environmental axis includes

robustness (maintaining value of services) in changing environments. For the value axis, the unchangeability is being maintained by versatility (adaptability). Finally, survivability is the function that maintains the services up to a certain value or recover the functionality of this service during or after environmental disturbance. As a result, survivability is a combination of changeability and robustness so as the system to be maintained in a *"plane value of robustness" [10].*

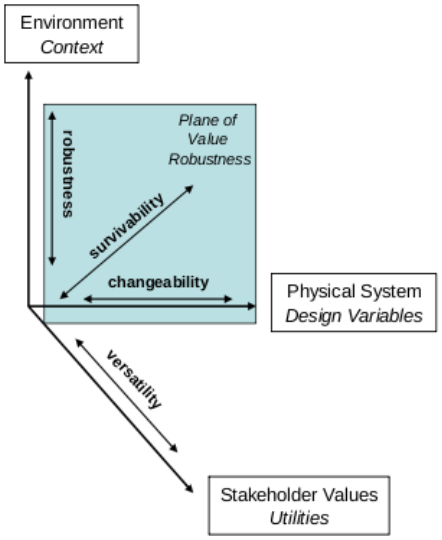


Figure 8: The - ility space. [10]

What is important and it is the key for understanding survivability principles is that by **improving the survivability of a system** it *"also improves the capacity to survive accidents and system failures that are* ***not malicious in nature****." [17].* As a result, what may be concluded is that a survivable system is also a secure, a fault – tolerant, a reliable, a recoverable, etc. system. The whole image of the aforementioned terms may be depicted by the figure bellow.

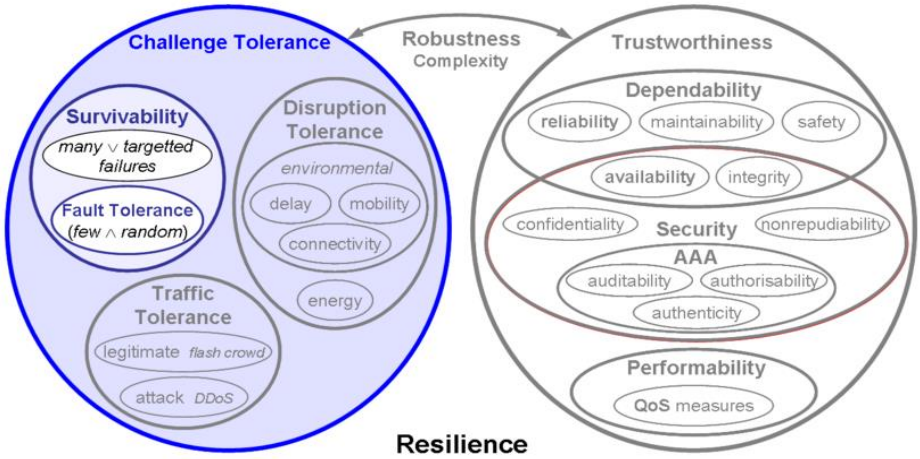


Figure 9: The whole image [17].

6. Business Continuity: Finally, another research area that is firmly connected to survivability is business continuity. Business continuity may be described by the following: "*Business continuity management is a holistic management process that identifies potential **impacts** that threaten an organization and provides a **framework** for building resilience and the capability for an effective response which safeguards the interests of its key stakeholders, reputation, brand and value creating activities*" [18]

The main difference of all these terms with survivability is that for survivability, protection and continuity of **system's mission and its critical services** is the main objective. In some cases, this makes system or some system's services out of scope of survivability. Any attributes that could provide an acceptable level of service and protect system's mission should be considered as survivability main objectives each time. So, defining survivability compared to other related terms is not so important. What is important is each time to find the key attributes that would provide survivability to system's critical services. For example, for a telecommunication system, protecting the survival and continuation of call establishment is a key objective. Additionally, another key objective is ensuring the confidentiality of communication. Finally, keeping the system robust is a key objective in the extend of keeping this critical service in an acceptable level of quality.

3.1.3. The word "system" - Survivable systems.

What may be observed from the attempt to define survivability, is that there is a need to define and "bound" if possible, the term "system" and how it is associated with the term "survivability".

Systems in general.

A definition of **systems** according to references may be*: "A system is defined as a **set of objects** together with relationships between the objects and between their attributes r**elated to each other and to their environment** so as to form a whole. Schoderbek, Kefalas (1990)." [19]*

Additionally, according to Von Bertalanffy in general systems research, a system is characterized by the **interactions of its components** and the non-linearity of those interactions. In addition, he supported that "*…there **exist models, principles, and laws** that apply to generalized systems or their subclasses, irrespective of their particular kind, the nature of their component elements, and the relationships or "forces" between them. It seems legitimate to ask for a theory, not of systems of a more or less special kind, but of u**niversal principles applying to systems in general.**" [20].*

As a result, what may be implied is that a system has:

1. **Components**: the parts of the systems which may be systems themselves.
2. **Structure**: the interactions and hierarchy of components
3. **Behaviour**: the input / output and processing of the system as well as the characteristics of its behaviour.
4. **Interconnectivity**: the interaction of the system with other systems and the environment.
5. **Functions**: the functionality of the system.

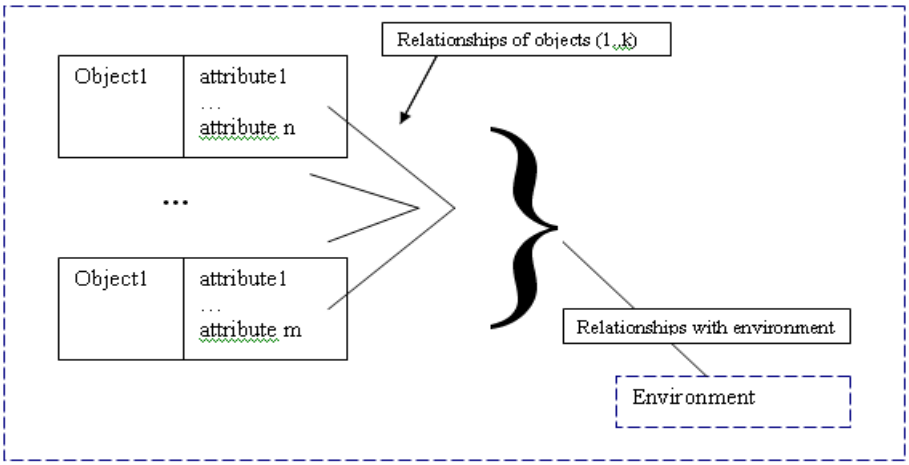


Figure 10: Schematic approach of system's definition.

What may be pinpointed from the above definitions is that a system is a set of objects with attributes related to each other and to the **environment (***Figure 10).* Though, systems may be distinguished between **bounded** and **unbounded** *(Figure 11).* In the first case, the components of the system are clearly distinguished from external components. In unbounded systems, according to references [2], components of the system are not clearly defined, there is no centralized control or control at all, and the insiders of the system may not be trusted. Examples of such systems, according to references [2] may be: *"the Internet, any systems with distributed administrative control without central authority, any system with remote access, any system with unknown users, and any system containing commercial – off – the – shelf (COTS) software." [2].*

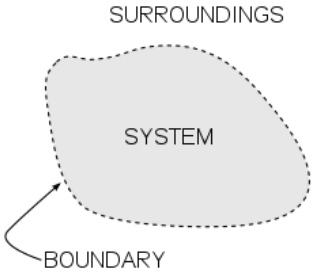


Figure 11: System Boundaries.

The **analysis of systems** involves the identification, reconstruction, optimization and controlling a system. For analysing a system, the risks, costs and benefits of actions applied on it must be examined. What may be observed from the systems theory and the known systems in general is that their main property is the **complexity** arising from **variety** of possible **states** of the system. Additionally, a characteristic of the system that has major influence on the way that will be chosen for organising a system is the **variety**.

Variety is the multiplicity of distinctions or the measurement of the possible distinct states of a system. For example, a system with 2 different states may be an On/Off system. Variety is growing as the system becomes more complex. "*Variety may be defined as the number of distinguishable elements it contains. Thus, if the order of occurrence is ignored, the set {c, b, c, a, c, c, a, b, c, b, b, a} which contains twelve elements, contains only three distinct elements —a, b and c. Such a set will be said to have a **variety** of three elements. (A qualification is added in the next section.)" [21].*

The complexity of an organization may be depicted by figure 12. The complexity of systems is needed to be **controlled.** For understanding the nature of complex systems, **reductionism** is the main approach. By reductionism, a complex system is being reduced to **parts** and **interactions** of their parts. Analysing and reducing a system to sub-subsystems is an approach for controlling the complexity of the systems. This is called **Hierarchy**. An example of hierarchy is an organization which is reduced to departments. Besides reducing systems to sub-systems, systems interact with each other making bigger dynamic and more complex systems. Moreover, the boundaries between systems are not always clearly visible.

At each level of hierarchical model of each system, new properties arise which did not exist at the components (or sub-systems) alone. These are called **Emergent** properties and are needed to be controlled as well. According to references [22], survivability is an emergent property and as a result it depends upon affection and extend of other scientific fields as security, fault – tolerance, safety, reliability, reuse, performance etc.

According to references, for designing a system there are 5 principles:

"*The 5 Cs form a useful set of systems design principles:*
• ***Complementary**. The parts / sub-assemblies of a system complement each other to create a whole, i.e., the set of parts is complete, comprises a full complement.*
• ***Co-operative**. The parts of a system act and interact cooperatively and harmoniously within the whole.*
• ***Coordinated**. The functions and processes within a system coordinate and synchronize their actions and activities to create requisite capabilities, behaviours and synergies*
• ***Contributory**. The parts/subsystems contribute, separately and together, to the objectives of the whole – this defines their value in the context of the whole.*
• ***Concinnity**: The parts/subsystems are constructed, configured and conformed to synthesize a dynamic, balanced whole.*

*… and then there is homeostasis. **Homeostasis** is the tendency toward a relatively stable dynamic equilibrium between interdependent elements… Design for homeostasis is essential for the viability of the created whole in its future operational environment." [23]*

The key problems of **a complex system** are modelling and simulation. As a result, the complex systems' theory is applied when reductionism is not enough. **System dynamics** is the scientific field that approaches the **behaviour of complex systems** over time. By examining the system dynamics, the dynamic behaviour of complex systems may be analysed. The main objective of the dynamical systems theory is the discovery and definition of **steady states** of the system for a long – term, and if the long-term behaviour of the system depends on its initial position. The behaviour of a system is affected by **feedback loops** and **time delays.** For controlling the behaviour of any dynamical system, the main scientific field is the **Control Theory**. The control theory is the theory of how to influence the behaviour of dynamical systems. Firstly, the output of a system is called reference. If one or more outputs of the system need to follow a certain reference over time, then the input is manipulated by a **controller** and the desired effect on the output is obtained.

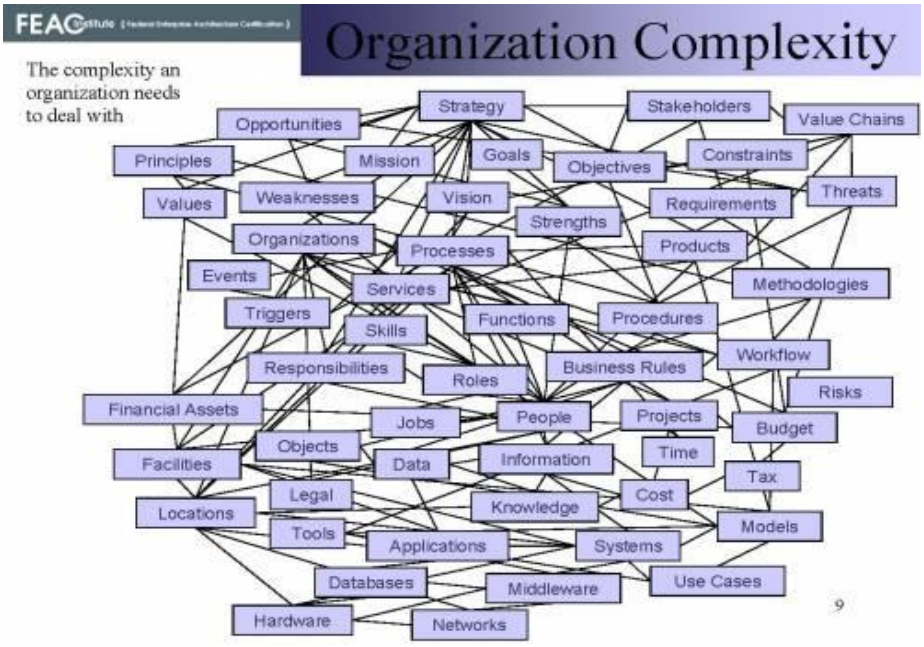


Figure 12: Organization Complexity [24].

There are two different types of controllers. The open – loop controllers and the closed – loop controllers. For the open – loop controllers there is no connection between the output of the system and the unexpected conditions mostly coming from the environment. For the closed – loop control system a sensor monitors the output and feeds the input with data related to the output and as a result, the input is adjusted so as to provide the system with the appropriate output. This is called **feedback** (figure

13.) and is a mechanism, process or signal, looped back to control a system by its own data.

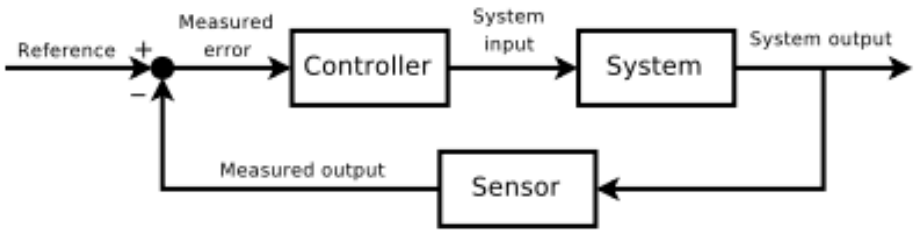


Figure 13: Feedback loop to control the dynamic behaviour of the systems.

According to references, *"when feedback acts in response to an event/phenomenon, it can influence the input signal in one of two ways:*

1. *The feedback signal can **amplify** the input signal, leading to more modification. This is known as **positive feedback**.*
2. *The feedback signal **dampens** the effect of the input signal, leading to less modification. This is known as **negative feedback"** [25]*

The **negative feedback**, as implied from the above definitions, reduces the effect of the input. This is also known as self-correcting. An example is a thermostat that compares the input temperature to desired temperature and tries to reduce the differences.

Complexity means a lot of things but for management, complexity seems to be the major problem. According to Stafford Beer [26], for handling complexity computers are to be used to handle organisation of large, complex, probabilistic systems. This is the science of cybernetics and its main objective should be the **self-regulation and self-organization** of large interactive systems. Finally, according to Norbert Wiener, Negative feedback is the **basis** for cybernetics.

Cybernetics

Cybernetics is characterized as the *"science of communication and control in the animal and the machine"* by Wiener [27] the man who first introduced the term. Cybernetics is the" theory of machines" and focus on the way they **behave**. Cybernetics is the domain of all possible machines and the framework of all machines to be understood and related. In fact, cybernetics includes machines that do not exist yet. By cybernetics the understanding and definition of the processes and functions of a system are examined. In addition, the cybernetics is focused on how these systems act, the evaluation of the output of this action and the re –action after adjustments of input to get the desired output. The main objective is to make the systems more **efficient** and **effective**.

Cybernetics includes the theoretical concept of how to **control** systems. According to Ashby the themes of cybernetics is **co-ordination**, **regulation** and **control** of living

organisms, machines and organizations**.** Finally**,** it includes the study of **feedback, and self-organization.**

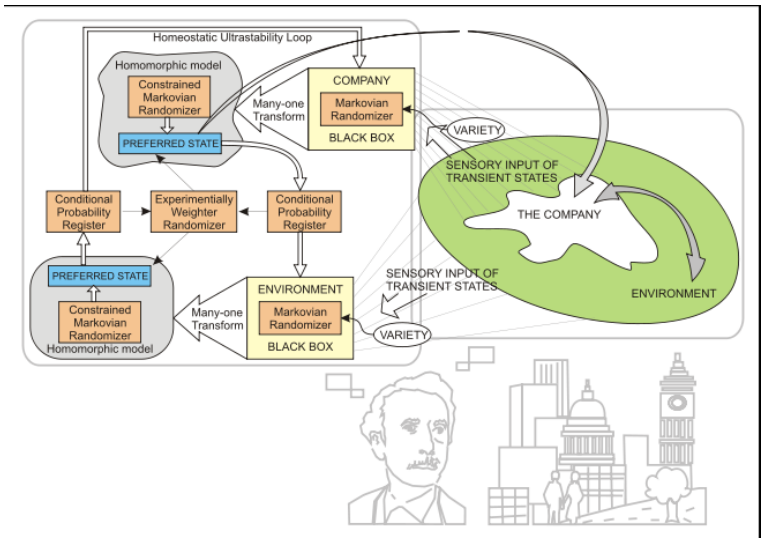


Figure 14: The first approach is of the system in the environment. The second approach is the control system. [27]

Regulation is the control of human or societal behaviour by rules or restrictions. A kind of regulation is the **self-regulation or self - policing** a process by which an organization monitors itself and enforces its standards. The control of the system as it has already been mentioned is achieved by keeping the output of the system within a target rate. For achieving the main elements need to be controlled are:

Control: The type of control that involves the use of feedback**,** the error detection and the correction processes to maintain the desired output.
Effectors: Components that carry out the desired action.
Comparator –Detector: The error-detection mechanism which compares the desired state to the real state.
Executive: Actions necessary to maintain the desired state.
Feedback: Data produced from system sensors. In cybernetics the feedback is found in the relation of affection with each other. *"When this circularity of action exists between the parts of a dynamic system, **feedback** may be said to be present." [21]*

An example of these may be a thermostat depicted at the figure *16*.

In addition, according to references [19], **control systems** which are systems that manage, command, direct or regulate the behaviour of other systems, follow three generic principles.
*"1. A control system should (a) **compare** automatically system behaviour with a pattern and (b) continuous automatic **feedback**, corrective action.*
*2. **Control is synonymous with communication***

*3. **Variables are brought back into control** in the act of and by the act of going out of control. [19]"*

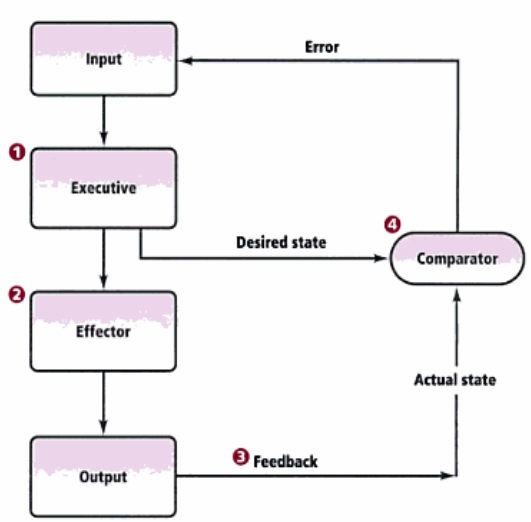


Figure 15: Control elements of systems [28].

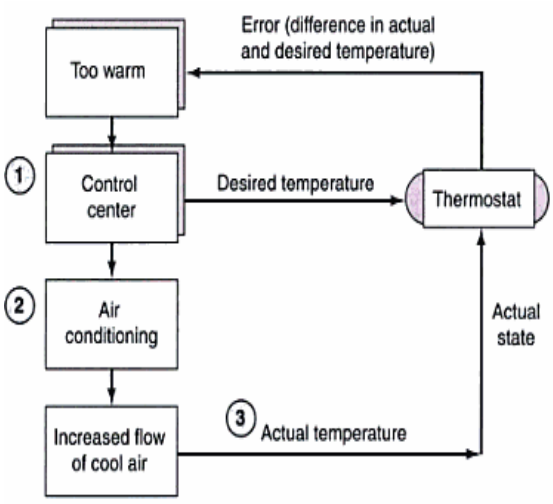


Figure 16: Example: the thermostat [28].

Another scientist who worked on fields of cybernetics and more precisely the management of cybernetics is **Stafford Beer**. Beer is assumed to be the first who applied cybernetics to management. Management cybernetics is the application of cybernetic laws on all types of organizations made of human. Stafford Beer also developed the viable system model in his book "Brain of the Firm", which is presented below.

Survivable systems.

After a short description of the term "system" and some system engineering on how to control systems, what should be examined is the kind of systems that survivability approaches will be applied to.

As it may already has been implied, survivability in general, focuses on the **purpose or scope** of the system, the **critical functions** that serve this scope and the approaches that should be adopted so as the system to **continue serve** this scope at an a**cceptable quality level** and in **a timely manner**. As a result, to define what survivability approaches must be applied to, these issues should be defined. Additionally to these issues, most of the available sources support that for identifying survivability requirements of a particular system, a case study analysis or cost/benefit analysis with specific threat scenarios and response scenarios should be conducted.

According to Vickie R. Westmark [3], the scientific focus of security has now turned to be focus on survivability as network and information systems turn to be more

distributed, decentralized, unbounded, uncontrolled and **impersonal**. Moreover, the other characteristic that makes survivability an emerging discipline is that these infrastructures support more **critical** and **essential** services than ever. Survivability focusses on the concept of **continuity and quality of these services** for which all the participants of a system are responsible. Robert J. Ellison et al [29], also support this opinion on the kinds of systems that survivability is to be focussed on. They characterise these systems as large-scale, highly distributed and operating in unbounded network environments like the Internet. What is also supported is that, despite the fact that these systems may not be invulnerable to attacks, they should "*deliver essential services and maintain essential properties such as integrity, confidentiality, and performance, despite the presence of intrusions*" [29].

Before proceeding with characteristics and requirements of survivable systems, the kinds of systems that survivability approaches are focused on is important to be presented. According to references, the systems that survivability focuses on are the critical systems often made up of system of systems.

Critical Systems

Most of the sources that examine the term survivability characterise the systems examined as Critical. According to references [30], **critical systems** are systems "*in which defects could have a dramatic impact on human life, the environment or significant assets".* What is also supported is that requirements that should be satisfied are reliability, security, safety, correctness, maintainability and availability. The examples provided for software critical systems are "*the controlling signal lights at a traffic intersection"* and "*the Anti-lock Brake System (ABS) in an auto-mobile that relies on software to determine when and how to regulate braking force", (figure 17).*

Figure 17: Examples of Critical Systems [30].

Additionally, according to Ian Sommerville *[13],* the main objective of critical systems is dependability and its four dimensions; availability, reliability, safety and security. Critical systems may focus on safety if the threat is the loss of life, on mission if the threat is the failure of a goal and on business if the threat is a high economic loss. What may be implied is that a critical system must be available when required (availability), must deliver its purpose correctly (reliability) and at no case should be threatened (safety). For example, a medical examination system that exports automatically a diagnosis, must be always available (availability), must export a correct diagnosis (reliability) and at no case must export, wrongly, such a diagnosis that may threat the patients' life (safety). A key feature of these systems is that they should be trusted by their users. In the opposite case, economic loss from their rejection may be huge.

System of Systems.

Additionally to critical systems, while examining the term "survivability" one may come across the term **system of systems**. The scientific field of system of systems remains an open issue. This term focuses on the interconnection of independent systems to provide a bigger, more complex system with emergent properties, controlling mechanisms, quantitative analysis, design methods etc, to be main objectives of system of systems' engineering processes. These characteristics of system of systems are to be managed by theory of survivability. An example of system of systems is the Internet as various communication systems (network infrastructures, communication nodes, etc) interact with each other. An exact definition of system of systems may be:
"A system of systems exists when a group of independently operating systems—comprised of people, technology, and organizations—are connected, enabling emergency responders to effectively support day-to-day operations, planned events, or major incidents." [12]

The main challenges of system of systems according to references [31], are:

- *Operational Independence of Elements*
- *Managerial Independence of Elements*
- *Evolutionary Development*
- *Emergent Behaviour*
- *Geographical Distribution of Elements*
- *Inter-disciplinary Study*
- *Heterogeneity of Systems*
- *Networks of Systems [31]*

System of systems' engineering is the scientific field of managing these challenges. The methodology provided may be used in various system architectures which may be problematic, and systems requirements are not always precisely defined. The difference against system engineering is that the system of systems' engineering focuses on building systems' interactions according to requirements, whereas the system engineering focus on building the systems right. System of systems lack standards and general accepted approaches as their design and implementation is based on the interoperation of different systems with different characteristics and emerging properties each time. As a result, there is a need for a more general approach so as rules for this interoperability to enact.

Survivable Systems characteristics and requirements.

During this sub-chapter, the objectives, characteristics and requirements of survivability according to literature are to be defined. Before examining the requirements and characteristics of a survivable system, what is important to be pinpointed is that the main objective of survivability is that the system must deliver its critical services in an acceptable level and in a timely manner no matter what. A

definition of survivable system may be, according to John C. Knight et al [15]: *"A system is survivable if it complies with its survivability specification".* It is obvious that for this to be succeeded, every part of an organization must be involved (management, operational, administrator and so on). The security and information specialists can only provide the specifications decided by an organization or a company as a whole.

Definition of system's purpose.

According to Douglas Lancaster [32], the main objective before defining survivability requirements is to define the system. Then the **main operation of the system** and the **steps to achieve this purpose (services)** must be defined. As a result, the business or organization operation will be the system of systems and the sub-systems of the business or organization must be pinpointed. Thereafter, the **critical services** of sub-systems must be defined, as effect on these sub-systems may result to a whole failure. A general **solution** to this would be the whole system to remain unaffected in case of the failure of one or more **sub-systems.** As survivability is proven to be an emergent property, if each sub-system is survivable by applying survivability approaches, then the whole system is survivable.

Definition of essential and non-essential services and of quality of services.

Furthermore, the **quality of the services** must be defined so as the acceptable level of service to be depicted and any deviation of this level to be the alarm for the survivability measures to "react". The acceptable level of service is supported by many sources to be defined by the users of the system. It is what the user expects and for how long. Though, different user requirements and different circumstances, even for the same system, determine different **value of services.** According to John C. Knight et al [15], the systems' requirements result from ordering its services due to their values. Though different sets of acceptable services may be defined by scaling from "full" to "normal" service for example. Acceptable services in general are what the users expect and for how long they are expected in case that normal functionality is not able to be delivered. Then the system may degrade the value of service, from one set of acceptable service to another. The degradation must be pre – decided to which conditions will be realised and each acceptable level of services may be associated with a probability of avoidance of such a degradation. This probability could be a requirement for the quality of services of a system. Additionally, according to John C. Knight et al [15], another important issue is to define **when** the essential services must be provided (e.g. during which attacks, failures, accidents or after which attacks, failures, accidents). Consequently, the system could have **more than one alternative service** to deal with different threats.

Operating environment.

According to John C. Knight et al [15], the value of a set of acceptable services, may

vary according to factors affected by the **operating environment** (ex. weather, time, date etc.). As a result, these environmental factors must be also defined in order to define and order different values of services of the system. Gathering these factors to environmental conditions, scenarios of varying operating environment may be developed, and different sets of acceptable levels of quality of services may be defined.

Definition of threat scenarios.

For succession of survivability appliances, according to references, after the aforementioned definitions, **threat scenarios** must be defined according to the nature of the system. Then, the measurements to survive from these threats must be applied to the system and its sub-systems so as the systems of systems to **maintain** its **essential services** even in cases that most of the system has failed. For this to be succeed, measurements to avoid, eliminate or tolerate threats, must be applied, Though, for the cost of measurement to be decreased or a balance between loss of failure and cost of measurements to be found, some classes and sets of threats must be ignored while designing the system. Finally, **threats** must be also modelled, for example by attack trees, and intrusion scenarios must be defined so as the reaction of the system (recovery, adaptation) to be also defined as a requirement for a system to be survivable. Focussing on survivability, threats that should be considered is any impact that the system may have from any accident failure or attack, since the root cause of failure is not always known. What is known though is the **impact** and this may be handled in order the system to be able to repel any failure.

The -ilities of a survivable system.

Before examining the different approaches to design and implement a survivable system, the main objective, which is the examination and representation of **characteristics of survivable systems**, or the quality attributes as they are called from Pentti Tarvainen [9], should be firstly examined. Additionally, to quality attributes, the balance of acceptable levels between these attributes must be considered. A list of characteristics of survivable systems is also presented in sub-chapter of "Definition of Survivability". Some of these characteristics are performance, **security**, **fault-tolerance**, **reliability**, **recoverability**, **modifiability** and **affordability**. Each of these characteristics has its own attributes. For example, security's attributes are integrity, availability, confidentiality. Furthermore, some of these characteristics may be emergent characteristics provided by the interconnectivity of sub-systems of the system examined. The trade-off between quality levels of these features must also be analysed according to the nature of the system examined.

The three Rs (resistance, recognition, recovery).

The examination here is based upon the research of Richard C. Linger et al. [33]. The most important characteristic of a survivable system, as it has already been presented,

is the delivery of essential services and the maintenance of levels of quality attributes of services. The essential services that will be maintained in case of accident, failure or attack must be defined in order the system to be protected. For this to be succeeded, according to references, the four key capabilities of a survivable system are the three Rs, resistance, recognition, recovery and adaptation and are depicted by the table of figure 18. From the same table, various measurements for these capabilities may be depicted.

Key Property	Description	Example Strategies
Resistance to attacks	Strategies for repelling attacks	Authentication Access controls Encryption Message filtering Survivability wrappers System diversification Functional isolation
Recognition of attacks and damage	Strategies for detecting attacks and evaluating damage	Intrusion detection Integrity checking
Recovery of essential and full services after attack	Strategies for limiting damage, restoring compromised information or functionality, maintaining or restoring essential services within mission time constraints, restoring full services	Redundant components Data replication System backup and restoration Contingency planning
Adaptation and evolution to reduce effectiveness of future attacks	Strategies for improving system survivability based on knowledge gained from intrusions	New intrusion recognition patterns

Figure 18: Key properties of survivable systems [33].

What may be pinpointed is that for system's survivability to be succeeded, firstly, the system must be able to **resist** with appropriate measurements. Secondly, the systems must be able to **recognise** the disaster, and to recognise which sub-systems are affected. After recognising the attacks, the system must be able to **recover** essential services and system's operation or to devolve the system to a backup system.

Additionally, according to Nancy R. Mead [34], a discrimination between **essential services** and **non-essential services** of a system must be conducted. According to this discrimination, non-essential services may be recovered after intrusions, attacks or failures contrary to essential services that must be maintained even during successful intrusions, attacks or failures. What is meant by the word "maintain" is providing specified and acceptable level of services. Finally, the survivability of system must be evaluated and redesigned in order to be more robust to future threats, as the feedback mechanism examined. This is called **adaptation** and is the last vital characteristic of survivable systems. According to Pentti Tarvainen [9], a kind of adaptation may be the situation at which if an essential service is lost, after all, it may be replaced by another

service for the mission of the system to be succeeded.

Characteristics of critical survivable systems.

Additionally, according to John C. Knight et al [15], the **characteristics that affect survivability** of a critical system are:

1. The **size** of the system. Critical systems are large and complex, and their control is infeasible.
2. **Externally Observable Damage**: the case when damage is visible to users of the system.
3. **Damage and repair sequences:** the sequence of events that may damage the system and the repair techniques that must be partially applied.
4. T**ime – Dependent Damage effects**: the damage tends to increase with time and may reach a point where survivability approaches may not be applied at all.
5. **Heterogeneous Criticality**: Requirements for survivability or dependability change as the nature of the system changes, the usage of same system changes, as the time passes for the same system, etc.
6. **Complex Operational Environment**: the large variation of the infrastructures, of the types of threats (accidents, failures, attacks), of usage, of damage (over time for example), of criticality of services, of level of quality of services, etc.

Usage Requirements.

According to Nancy R. Mead [34], **usage requirements** are also important for survivability as they may depict the essential and non-essential services. Usage requirements define various usage scenarios with another objective, apart from the depiction of essential services and non-essential services, to be the discrimination between intrusion and legitimate usage scenarios.

3.1.4. Theoretical Approaches on providing Survivability.

The main objective for applying survivability approaches is to protect or diminish the risk of critical infrastructures from damage and of system – of – systems from catastrophic failures. According to Matthew Richards et al [10], methodologies for survivable systems may be categorized as follow:

1. methodologies which *are often* ***"reductionist"*** *in nature*: *risk analysis, bottom-up verification.* These approaches are based on standard processes definition and appliance.

2. methodologies which arrive at **subjective definitions** of survivability: these approaches are based on operating and environmental scenarios and predefined threats in order to provide certain processes according to these scenarios.

3. methodologies which *consider survivability at* ***higher levels of system architecture****: considering the trade-off between cost, utility and survivability.* These methodologies focus on the system as a whole and the survivability as an emerging property of the system.

Through this literature review methodologies of all these categories are to be presented and examined as the most appropriate to use to the application of the dissertation to be pinpointed.

Passive and Active Survivability.

According to Matthew Richards et al. [10], who presented the categories of methodologies, survivability includes active and passive techniques. **Passive** techniques focus on resistance and robustness, which are the main objectives. With passive techniques the system has the ability to maintain value delivery despite environmental disturbances. **Active** focus on the ability of the system's regeneration, reallocation, retaliation etc.. and changeability is the main objective. With active techniques, the system reacts to the environmental disturbances. To continue, they discriminate the developing of a survivable system into two levels; the system architecture level and the component level. For the first level survivability is reliability, node hardening, operational behaviour, human factors, supporting infrastructures, etc. This level also includes a cost-benefit analysis so as utility, cost and survivability trade-off to be depicted. For the second level, survivability is an emergent property from system architecture.

	Passive Survivability	Active Survivability
Philosophy	Survivability is something that a system *has*	Survivability is something that a system *does*
Characteristics	proactive, resistant, robust	reactive, flexible, adaptive
Design Principles	hardness, stealth, redundancy, diversity	regenerate, evolve, relocate, retaliate
Forecasting	Presupposes knowledge of disturbance environment	Acknowledges uncertainty in projection of future disturbances
Architecture	Closed (static)	Open (dynamic)
Design Focus	Defensive barriers at system-level to resist disturbances	Architectural agility to avoid, deter, and recover from disturbances
Failures	Causal chain (often linear)	Tight couplings, functional resonance (nonlinear)
Relevant Disciplines	Component reliability, safety engineering, risk analysis, domain-specific technologies	Real options, organizational theory, process design, domain-specific technologies

Figure 19.: Passive vs Active Survivability approach [10].

Survivability Analysis Framework (SAF).

Starting with a subjective survivability method, Robert J. Ellison and Carol Woody proposed the **Survivability Analysis Framework (SAF)**. This approach is focussed on organizational survivability which they consider to be a matter of capabilities of people, actions and technology working together to apply operational effectiveness. Additionally, the framework proposed, has been developed for managing the complexity that arises from interaction of people, actions and technology that work together for the purpose of the system, from integration of systems to large system of systems, and from interoperability of components. This framework provides a technique to *"analyse complexity and integration issues throughout development life cycle"* [35]. Focussing on operations of the organization the framework also proposes that organization should capture interactions of the organization with people, actions and technology, depict and analyse critical operations and failure conditions, depict the operations as input to the development of the system, evaluate dependencies between the parts of the organization.

The processes needed for the analysis are:
- the depiction of the operational process.
- the parts involved with the organizations (people, actions, technology).
- the critical steps of the operational process.
- the step failure that may lead to overall failure.
- the impact of each of these steps of failure.
- plans for mitigation for unacceptable failure.

The survivability of these features will result to an end-to end operation of the organization. As a result, a list of organisational components should be constructed, containing hardware, software, people, policies and practices.

Then scenarios on how these pieces will work together are generated, and critical steps will be depicted through this process. Each step may be model based on the SAF approach (figure below). Each critical step is considered to cover a portion of the operational purpose. For each critical step, the resources provided are called preconditions and are part of the analysis. These preconditions trigger the step's actions. Though, there are other actions to the step that are working in continuous mode like a sensor. Each step outputs the post conditions. These may interact with other steps.

As a result, by this approach a number of critical steps and their analysis is provided so actions of mitigation the risk to be applied if necessary.

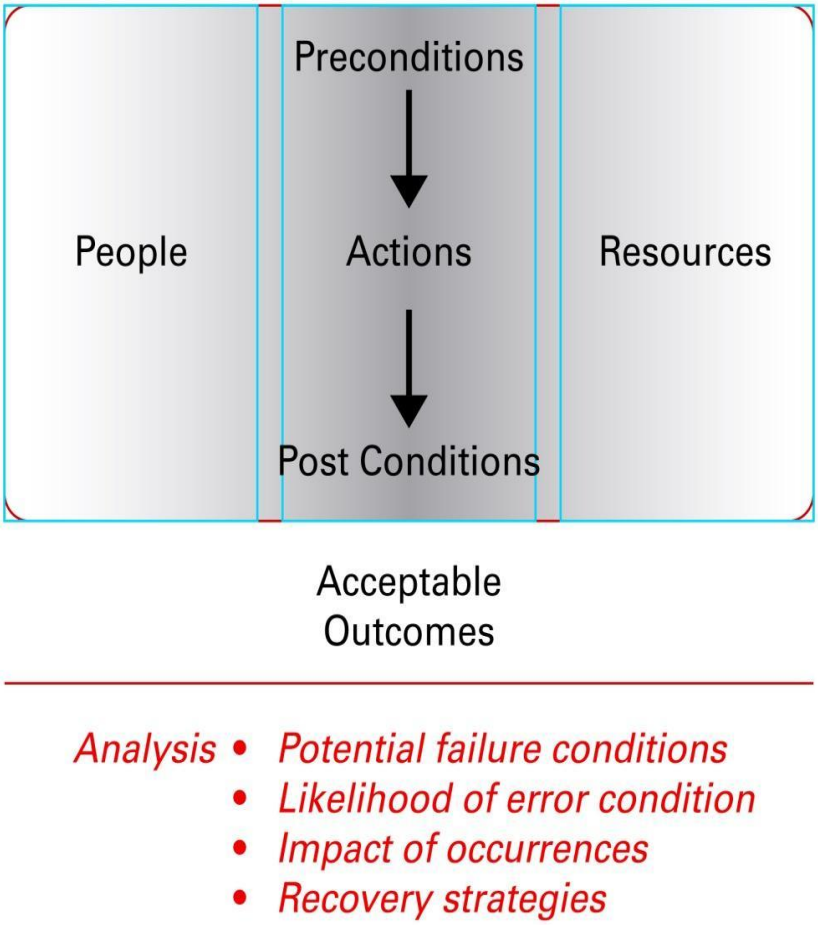


Figure 20.: Survivability Analysis Framework [35].

Viable System Model.

Another representative approach, focussing on the higher level of system architecture is the one given by Stafford Beer and his **Viable System Model.**

For Stafford Beer the cybernetics is the science of **effective organization**. The viable system model is applied for the organisational structure of any viable or autonomous system. A viable system is organised in such a way that it survives in any change of the environment or capable of independent existence. In addition, the viable systems are recursive.

For a system to be viable the key is the control of **variety.** Ashby proposed a law for controlling the variety; the **Law of Requisite Variety**. By this law Ashby proposed that only variety may destroy variety. This means that for controlling a system the mechanisms must have the same variety as the system to be controlled. In real life the

way of coping with variety is by amplifiers and attenuators (figure 21). In real life we attenuate or filter all the senses that we get from the environment and keep those relevant to us. Furthermore, we amplify our variety to increase our influence on the environment.

The ability to maintain identity is called self – organisation. **Self- organising systems** have many purposes which need to be under hierarchy so as the system to remain viable. Self – organising systems have **elements** which **do** things, **control the doers** and all that are acting in a particular **environment**. These elements are called **Operations, Management** and **Environment** respectively. These three elements **interact** with each other and this interaction needs to be controlled by attenuators and amplifiers in order to equally spread processes and provide the organization with a balance in functions need to be conducted. This is called **Homeostasis** and happens by attenuating the incoming variety and amplifying its own variety. This is because the variety of the Environment is greater than the variety of the Operation which is greater than the variety of Management. Furthermore, the Management is enclosed within the Operation and Operation within the Environment. As a result, the information travels through the elements with that direction; from management to operation and from operation to environment and via versa. The Ashby's Law is the mechanism for successfully applying the Homeostasis. This law is "*Control can be obtained only if the variety of the controller is at least a as great as the variety of the situation to be controlled" [36]*

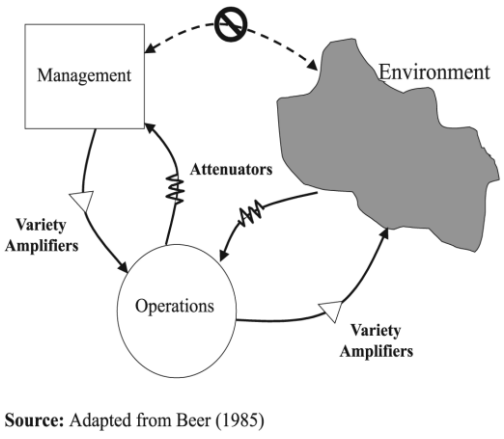


Figure 21.: Ashby's Law on Centre of the systems [old 26].

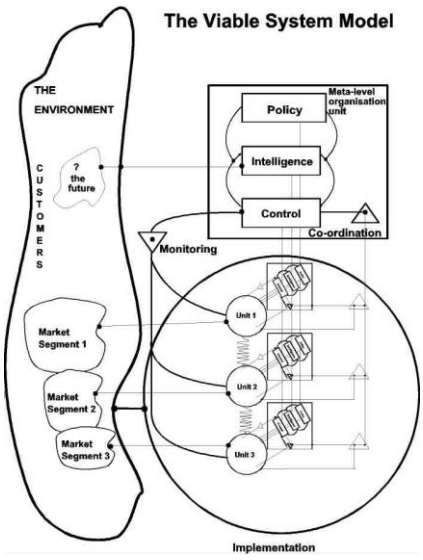


Figure 22.: Viable System Model (VSM).[37]

The viable system is more complex of figure 22. Viable systems are **recursive** which means that viable systems obtain other systems which may be modelled by the same way. In other words, the viable systems contain a number of Operations each of which has a number of Management functions and operates in its own environment. This is depicted by figure 22 in the circle. These operations must **co-operate** with each other and remain stable which means that they must have production plans or senior managers who oversee operations.

Before examining the Viable System Model according to references there are four principles of Organization. [14]

First Principle of Organization: The first principle says that managerial, operational and environmental varieties tend to equate. They should do that with the less consumption of resources as possible.

Second Principle of Organization: The directional channels between the environment, operation and management must have satisfactory capacity in order to transmit information relevant to variety in a given time.

Third Principle of Operation: Whenever the information of such a channel crosses a boundary, it undergoes transduction which means that it is translated in order to be transferred.

Fourth Principle of Operation: The operation of the three previous principles must be cyclically maintained through time. In other words, management should be a continuous process to cope with the changes of the environment.

Stafford Beer proposed five interacting sub-systems of operational structure. System 1-3 is focused on organization's operations. System 4 is focused on the strategical responses to the effects of external, environmental and future demands on the organisation. System 5 is focuses on the balancing through policing directives which maintain the organisation viable. These are the five essential functions for a system to be viable and operate effectively with the environment.

System 1 (Implementation): Contains **primary activities** and each system 1 is a viable system too. This is the **recursive** nature of systems. What should be pinpointed here is the fact that a viable system contains other viable systems. These sub-systems help the handling of complexity of their environment.

System 2 (Co-ordination): Contains the channels of **communication** between the primary activities of system 1. According to references [37], the stronger the communication links are, the less the management requirements of control and the more autonomous the sub- systems become.

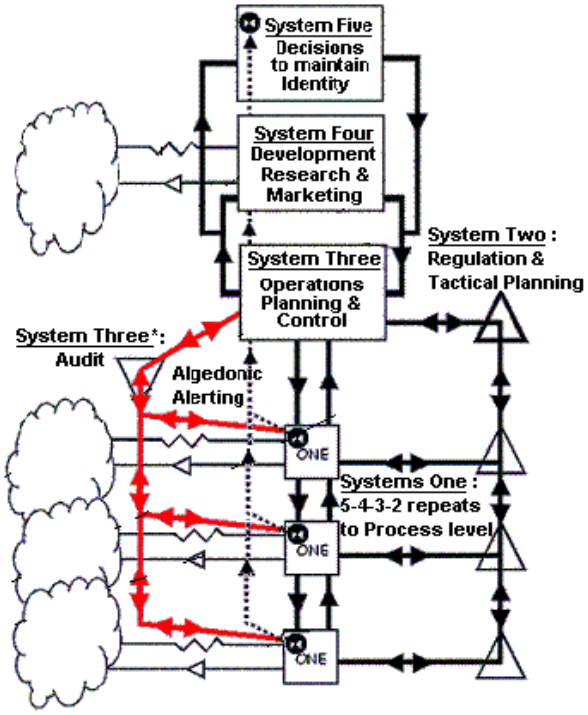


Figure 23: The Viable system Model and its sub-systems (VSM). [36]

System 3 (Control): **Monitors** and **co-ordinate** activities of system 1. Additionally, system 3 is the representation of structures and **controls** and contributes on establishing the rules, resources, rights and responsibilities of system 1. Finally, it is the interface for System 4 and 5. The communication between the sub-unit and the Meta – level is of vital importance.

System 4 (Intelligence): This system is responsible for **comparing** the desired state with the one that comes from the environment. It is a two – way link between the primary activity and its environment. Part of this subsystem is the **feedback loop** for keeping the variety balance.

System 5 (Policy): Responsible for **policy decisions** so as to keep the balance in the organisation. It provides with an overall direction, and the purpose of organisation.

Another methodology that focuses on the **viable system model** and uses a systemic approach for survivability is the one proposed by Maria Karyda, Spyros Kokolakis and Evagelos Kiountouzis [38]. The approach adopted focuses on the viability of a system to provide required services and on the goal achievement and cost of the context of the organization. The systemic model proposed, deals with predefined threat scenarios for the functionality or performance of the system, and with the selection and implementation of the appropriate countermeasures for survivability to be achieved. What should be pinpointed from those presented to this methodology is that IS security is considered to be a build-on characteristic and not an add-on one. By this

what is meant is that IS security is a problem that must be dealt with on the design of the whole processes and structures of an organization and on the functionality of these processes. As a result, IS security should be considered in a holistic and systemic way.

The **model proposed for building a viable system,** consists of a three-phase interactive processes (figure *24* bellow):

1. **Diagnosis**: During this phase, vulnerabilities and threats of the system are detected. The I.S. is also being examined from performance, risk and cost perspective against the goal of the organization. For this to be succeeded, corresponding parameters adding the estimation of operation cost are estimated. After estimation of these parameters, the VSM is used as a diagnostic tool. The main five actions of the model are developed and examined against these parameters. This may lead to re-designing processes. Then by VSM techniques to control variety, using attenuators and amplifiers, threats may be decreased.

2. **Re-design:** It follows diagnosis for the re-design of the system. The steps of re-designing as presented are:

- Design **processes** that implement the **missing, underdeveloped** or **flawed** VSM **functions**.
- Add **processes** that serve as **attenuators** or **amplifiers**.
- Add **controls** and **mechanisms** to **mitigate risk** for the processes with a high-risk factor.
- **Re-evaluate.**

3. **Transformation:** Implementation of changes from the previous step.

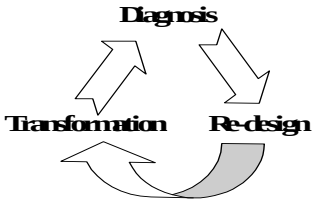


Figure 24: *Three phases for building viable information systems [38].*

Continuing with another approach, John C. Knight et al [15], considers that survivability must have three characteristics related to the system that is being serving. Firstly, it must be suitable to deal with several damages of the system. Secondly, it must include various forms of service which are able to achieve certain levels of function and costs, determined under certain operating conditions. Additionally, the information of what level of service is to be provided under what circumstances if full service may not be provided must be gathered to be input of the design process of the critical system.

Thirdly, there should be a defined probability related to if each of these services may be available for use in accordance with the conditions that the system may encounter. Additionally, survivability is considered to be related to the characteristics that must survive each time. For example, a system may have to be highly reliable, highly available or to have a high level of safety.

Survivability, is defined by John C. Knight et al, as a six-tuple {S, E, D, V, T, P} where:

S [specifications]: is the set of acceptable forms of service specifications for the system. Each specification must include the appropriate and predefined capabilities, and the operating conditions under which these specifications must be realized.
E [service value factors]: is a function *"from the set of service value factors to the set of values that each factor can take" [15]*. These are the factors that change to provide different levels of services.
D: is the *"set of all combinations of values of the service value factors that the application might encounter" [15]*. These factors vary with the different environmental conditions. These variations provide different values of E which provide different services.
V: the relative services that the system may provide to the user under certain environmental conditions. It is the service values that are perceived by users.
T: the set of transitions between acceptable levels of service. The combinations may be represented by a two dimensions' table and the reasons of the transition may be changes on environmental factors, the unavailability of operational acceptable services, or the availability of a higher acceptable service.

Table 1: V for C2 example

	s_1	s_2	s_3	s_4	s_5
d_1	5	4	3	1	2
d_2	5	4	3	1	2
d_3	5	2	4	1	3
d_4	5	2	4	1	3
d_5	5	3	2	1	4
d_6	5	3	2	1	4
d_7	5	1	3	4	2
d_8	5	1	3	4	2
d_9	5	1	3	4	2

Figure 25: Transition table [15].

P: the set of probabilistic requirements on the operation of the acceptable forms of service. It is defined for each member of set S and it is the probability that a service may meet each dependability requirements.

Example of survivability specifications of an example system may be examined from figure bellow:

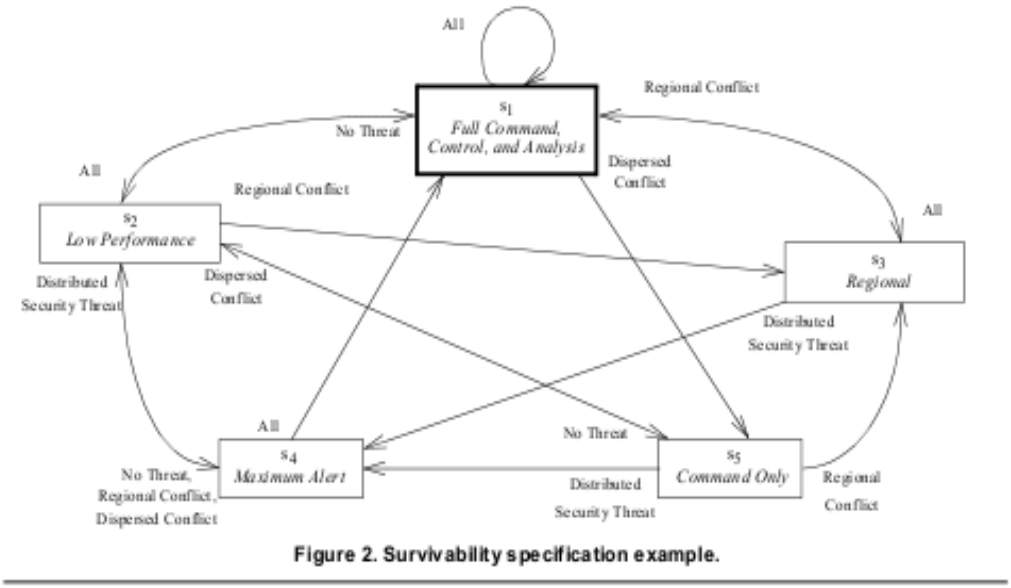


Figure 26: Survivability specifications example [15].

What may be observed to the figure above are the different acceptable states of the system. After the appearance of changes to environmental variables or the emergence of threats to the system, the system may transit to the next level of service. When the conditions change the system may go over to higher level of service. To sum up, the approach proposed is the definition of set of services and survivability for each system according to each characteristic and the environment the system functions in. Then the situations that transition from each level of service to another must be defined and when conditions change this transition is realized.

Another approach has been proposed by Richard C.Linger et al [33]. Firstly, what is firmly supported is that survivability should be integrated into the primary development phase of system and not treated as an add-on property of an already implemented system. If the system is a software element, the development phase may be represented by a life cycle. Additionally, considering the development of a system as a life-cycle model, then, according to Richard C.Linger et al [33], survivability goals and **survivability methods must be addressed for each action of the life-cycle.** Examples, of such methods and approaches may be examined to table below:

Life-Cycle Activities	Key Survivability Elements	Examples
Mission definition	Analysis of mission criticality and consequences of failure	Estimation of cost impact of denial-of-service attacks
Concept of operations	Definition of system capabilities in adverse environments	Enumeration of critical mission functions that must withstand attacks
Project planning	Integration of survivability into life-cycle activities	Identification of defensive coding techniques for implementation
Requirements definition	Definition of survivability requirements from mission perspective	Definition of access requirements for critical system assets during attacks
System specification	Specification of essential service and intrusion scenarios	Definition of steps that compose critical system transactions
System architecture	Integration of survivability strategies into architecture definition	Creation of network facilities for replication of critical data assets
System design	Development and verification of survivability strategies	Verification of data-encryption algorithms for correctness
System implementation	Application of survivability coding and implementation techniques	Definition of methods to avoid buffer overflow vulnerabilities
System testing	Treatment of intruders as users in testing and certification	Addition of intrusion usage to usage models for statistical testing; use of independent verification and validation
System evolution	Improvement of survivability to prevent degradation over time	Redefinition of architecture in response to changing threat environment

Figure 27: Life-Cycle Activities and corresponding survivability elements.[33]

Starting with survivability characteristics, as it has already been described to the relative sub-chapter, the system is supposed to deliver essential services even under attacks, failures or accidents. Additionally, essential services may be served into various predefined levels of service. As a result, the minimum level of quality attributes must be defined. During this approach, characteristics of a system like security, performance, fault-tolerance, etc., are considered in terms of quality attributes. Additionally, survivability is considered in terms of trade-off analysis and it is the balance of these quality attributes that emerges survivability at last. As a result, the essential services of a system and their essential properties must be depicted and maintained within the operating environment of the system. When the environmental conditions change, for example, if threats for the system have arisen, the essential services must be maintained to predefined quality levels and not exceed the minimum quality levels. For this to be achieved, the system must have four properties; resistance, recognition, recovery and adaptation, as defined to the relevant sub-chapter. The realization of these properties for the system, by application of relative survivability methods may emerge a survivable system.

To continue with the same concept Richard C.Linger et al, propose that system survivability is achieved if survivability is considered at each activity of system life cycle. What firstly presented are the requirements definitions of the system as the first critical step. Survivability requirements must be addressed for functionality, usage, operation, and evolution of the system and these types of functionalities may be examined at

figure bellow.

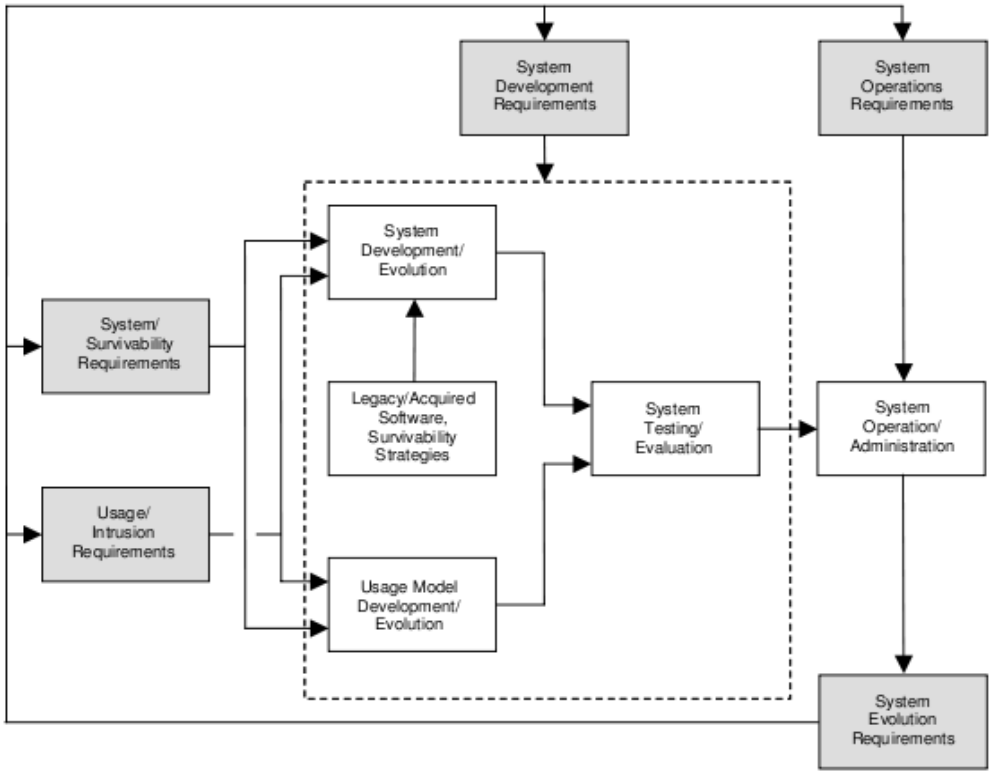


Figure 28.: Survivability requirements [33].

What may be depicted from figure above is the survivability requirements integration in system's requirements.

Starting with **systems / survivability** requirements, systems' requirements are the **users' functions** that the system must provide for monitoring operations, performance etc. Survivability requirements are focussed on the maintenance of these functions under attack, failure or accident and the recovery of full services. These are **recognition, recovery, resistance and adaptation**. Finally, there are the **emergent requirements** for the system that result from the connection and communication of different parts of the system and interconnection of the system with other systems. The desired result would be a system that would adapt systems' behaviour, functions, and resource allocation after intrusions. As a result, another requirement's category should be **adaptation and reconfiguration requirements** of the system. Finally, the nature of the system, like those that are related to its size, mission, or interconnections, or those related to other features like recovery time intervals, may provide with more requirements that a survivable system must preserve.

What is supported to the concept proposed is that system requirements may be discriminated to two categories: essential services and non-essential services. Essential services are those that must be maintained under attacks, failures or accidents. These

services must be maintained under certain levels, not going beyond a predefined minimum and go to upper quality levels gradually, as the system is repaired. Non-essential services are those which are recovered after the system has been repaired.

Next, there are **usage / intrusion requirements.** These requirements consider the quality of performance of the system at usual or under intrusion performance of the system. Survivable systems must be tested under these circumstances. For this to be achieved, usage of system and environments of usage scenarios must be developed and tested, under usage requirements. Intrusion usage scenarios must not be excluded from the usage scenarios.

Development requirements are the next category that is proposed to be integrated with system's requirements and especially during the development activities of system's life cycle.

Additionally, there are **operational requirements** for survivability like "*communicating survivability policies, monitoring system use, responding to intrusions, and evolving system functions [33]".* These requirements focus on operation of the system and administration activities and on the continuation of the system.

Finally, there are the **evolution requirements** which are the users' requirements for the new functions of the system. These must be examined in accordance with the intruder's increasing knowledge of the system.

After essential and non-essential services survivability requirements must be tested against three phases of attack, the **penetration phase,** when the intruder attempts to gain access to the system, the **exploration phase**, when the intruder is exploring through the integral system organization and capabilities, and the **exploitation phase** when the intruder performs operations due to the system facilities. According to these phases, survivability strategies for resistance, recognition, recovery, adaptation and evolution must take action.

For example, the survivability approaches for **resistance** are equivalent for penetration and exploration phases, before the actual exploitation. Resistance contains any approach that may deter attacks like firewalls, encryption, authentication etc. Additionally, the survivability approaches for recognition are essential during all phases of attacks.

Approaches for **recognition** of attacks may be used like intrusion detection techniques, investigation of reports effective techniques as logging and frequent auditing, etc. Survivability approaches for recognition include examples as self-awareness, trust maintenance and black – box reporting. Starting with trust maintenance, it is being achieved by periodically testing trust relationships among elements of a system. Black-box reporting is recognition of a pattern of information that it comes out of a crashed

system. This may be helpful to stop the expansion to other systems. Finally, a **self-awareness** system is a system that can process what has been asked to do through a high-level semantic model of computations. In that way the system may refuse to execute actions that may be dangerous to the system or may develop a legitimate behaviour.

After recognition, come recovery requirements. The application of recovery approaches may be essential during exploration and exploitation phases and may consist of a react to the intrusion and survive even if the intrusion may not be completely repelled. Recovery according to Richard C.Linger et al, is what distinguishes survivable systems from secure systems as the secure systems react only by resisting and protecting. The most known approach of recovery is the use of backup systems.

Finally, adaptation and evolution may be a part of survival during the exploitation phase as they may maintain resistance against the increasing knowledge of the intruder about the system. According to Richard C.Linger et al, dynamic adaptation permanently improves a system's ability to resist, recognize, and recover from intrusion attempts. For example, security fixes provided from newer attacks and vulnerability discoveries may be applied to the systems' elements. The same may be applied for intrusion-detection rules of known intruders' behaviours. This may be succeeded if the system is being frequently reviewed and continuously is being improved. Maybe it would be safe to say that this may be succeeded if there was a business continuity plan for the system which is though developed at the design phase of the system and not as an ad-on characteristic.

After requirements definition, Richard C.Linger et al provide an architectural level of survivability design principles, through a survivable network design method depicted at figure bellow. As with software development life cycle model, for the **network designing model,** survivability approaches are not ad-on processes to the system, but they are part of the design steps of the model. As it may be observed, the initial step of the model is the requirements definition (mission detection, system / survivability requirements, usage / intrusion requirements, operations requirements) as they have already been described. Then, the architecture of the system is designed and analysed so as system elements and connections between them are defined and essential services and intrusion scenario are analysed. What follows is the survivability analysis with application of resistance, recognition, recovery strategies and survivability map generation. According to these, architecture definition and analysis and system requirements are modified properly and periodically.

What may be pinpointed from this analysis is the architecture analysis step where survivability properties of the system are examined in terms of:

"1. essential services (services that must be maintained during attack);
2. essential assets (assets whose integrity, confidentiality, availability, and other

properties must be maintained during attack);
3. mission objectives and the consequences of failure." [33]

Firstly, **essential components** that must be maintained after and during an attack are chosen. Then intrusion scenarios are selected due to system's environmental properties, risk assessment and intruder capabilities. From these scenarios **comprisable** ("vulnerable") **components** are depicted. Then essential and comprisable components are chosen as "**soft spot" components.** These "soft spot" components are then analysed against their recognition, recovery and resistance properties to provide the survivability map. This is a mapping of intrusion scenarios, "soft spot" scenarios of each and resistance, recognition, and recovery approaches that should be adopted. This map is a feedback for improvements and may be used to a cost/benefit analysis of the system.

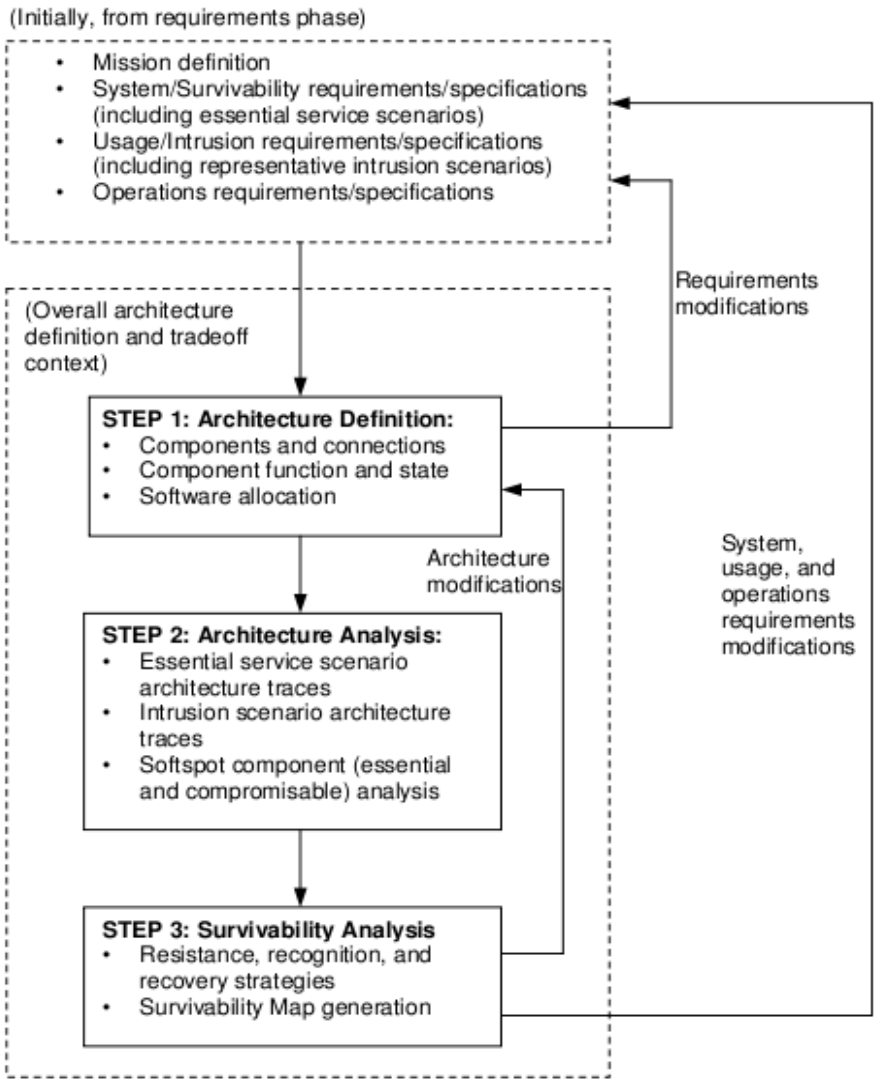


Figure 29.: Architectural level of a survivable network design method [33].

After the system has been developed to be survivable, the testing against survivability of the system may be evaluated by penetration testing, statistical, usage-based testing, etc. Though, the testing of survivability must be focussed on mission of the system, critical services, quality attributes and systems' life cycle.

Finally, there must be a continuity plan for survivability of the system as the factors that affect it may change over time, as survivability is a mix of security and business risk management according to references [33].

Another approach is that of Partha Pal et al [39], who present an approach of providing survivability based on defence – enabling. Contrary to security approaches of preventing the attacker to gain access, they start with the assumption that the attacker has gained access and they try to find ways to prevent the attacker from interfering with systems' services. What is actually proposed is to provide the system with methods that protect the system. This means that the attacker is being prevented from gaining access to the system, and there are methods that frustrate the attacker. The two approaches appear at that order, which means that if the attacker gains access may be frustrated so as critical systems to be protected. Additionally, by using this approach corruption of services of the system may be delayed as well, as the attackers' gaining access slows down, and the defence mechanisms may respond and adapt to attackers' actions. For this to be achieved, they propose the creation of multiple domains of security in order to distribute the single point of failure of security methods. Additionally, quality of service of the system must be monitored, as a decrease in quality of service may be caused by an attack. Finally, the system must be able to adapt to environmental changes or to degraded services.

Willow architecture

Another approach for providing survivability comes from John Knight et al [40] and is called the **Willow architecture**. It is a proposal focusing on proactive and reactive reconfiguration of a system to provide survivability to the system. With proactive reconfiguration what is meant is the adding, replacement and removal of components and interconnections of the system, or the changing to their mode of operation. This is called posturing and it is used to limit the vulnerabilities of system's components when threats are realized. Example of such reconfiguration may be the turn-off of non-essential services and networking links and strengthening of cryptographic keys if a worm has been depicted to the network or infections have been observed. The reactive configuration does the same actions with scope to restore a system from damage or intrusions, in specific time intervals. In fact, the most appropriate approach for reacting is fault tolerance, as proposed. An example of such reconfiguration is the activation of copies of application modules as a reaction to a coordinated attack or damage.

The architecture that has been developed is based on this reconfiguration and:

*1. "**ensures** that the correct configuration is in place and remains in place during normal operation.*
*2. facilitates the reconfiguration of such systems in **response** to anticipated threats before they occur (including security threats);*
*3. **recovers** from damage after it occurs (including security attacks)" [40].*

As it has already been designed from other architectures, the reconfiguration takes place after a not normal ("steady – state") has appeared. Additionally, the discrimination of services of essential or non-essential services is being used as in previous approaches and threats scenarios are also defined. The willow architecture defines reconfiguration of the system, during or after any scenario that may result to any deviation of the function of the system than that provided by steady-state, or normal function. Reconfiguration of the systems provides:

- *"**Initial application** system deployment.*
- *Periodic application and operating system updates including **component replacement and re-parametrization.***
- *Planned posture changes in **response** to anticipated threats.*
- *Planned fault tolerance in **response** to anticipated component failures.*
- *Systematic best efforts to deal with **unanticipated failures."** [40]*

As a result, the monitoring of the system is of vital importance for the reconfiguration of the system to react and response to attacks or failures. The components of architecture may be depicted from figure bellow.

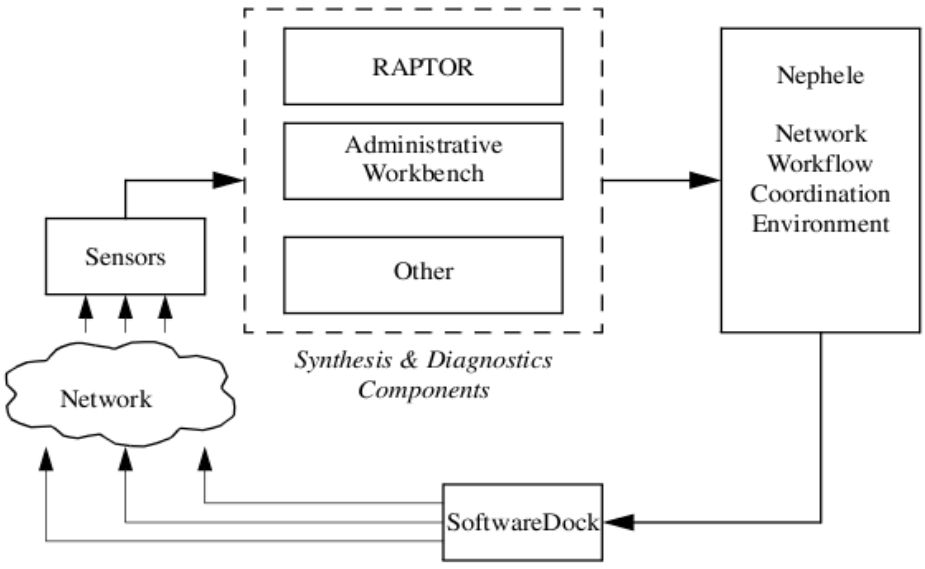


Figure 30: The WILLOW architecture [40].

The first element of the architecture is the **sensors** that are sensing the system's state and produce reports for this state. These may include reports from applications, application heartbeat monitors, intrusion detection alarms etc. Then the **synthesis and diagnosis components** take advantage of these reports to provide the system with models of the state and the changes that are required. **RAPTOR** is one component of this sub-system, analyse and provide with appropriate changes for the system, in bounded time. The **Administrative Workbench** provides an environment for administrators of the system to monitor applications and adjust their properties. **Other** components may be also added. Then the changes provided by the synthesis and diagnostics components come in form of work-flow requests to be executed by an appropriate environment, to coordinate such an execution. Except from ordering execution of changing processes, the environment is responsible to handle possible conflicts of execution of processes. Finally, the components may communicate with each other for an event-driven reaction to a deviation of normal state of operation of components.

A relevant approach of reconfiguration of the system is also provided by LI WANG et al [41]. What is supported to this paper is that QoS and survivability are firmly connected. As a result, if QoS may be measured, reconfiguration approaches may be triggered under certain measurements to provide survivability for the system. Firstly, survivable system is considered to be a system that may repair itself or degrade in such a way that will provide as much functionality as possible. This may be done as the system is able to switch between alternatives of accepted functionality. Secondly, survivable system is a system that may adapt threats in its environment and environmental changes and reallocate essential processing to most robust resources. All these may be provided by dynamic reconfiguration.

Such reconfigurations may be *"process/host restart, migration of objects to alternate hosts, replication, transparent rebinding of clients and servers, use of service alternatives, and approximate services".[41]* These reconfigurations may be based to several metrics like *"available battery power, varying communication bandwidth, available memory or faults in software components" [41]* and must be done in bounded time and must be based on QoS service levels. QoS for each subsystem of a system may be measured by different factors. For example, at network level QoS factors may be the bandwidth or the error rates, and for the service level QoS factors may be precision, accuracy, timeliness, etc. Then a survivable system must provide a minimum level of QoS under changing environments. For that purpose, the best-suited elements are to be chosen at each time, based on these QoS factors.

What is proposed is a reconfiguration framework based on QoS. Starting with **Reconfiguration User Interface**, it is a GUI for friendly interaction with users. Through this environment users may construct and manage QoS metrics, may specify constraints for the overall services, and initialize values of QoS metrics. **Reconfiguration Engine** is responsible to trigger reconfiguration actions, design a

dynamic reconfiguration algorithm and realize the reconfiguration process. The triggering of reconfiguration is realized, after the computed QoS performance of the system is found to be under a certain threshold. Additionally, the reconfiguration algorithm may be described as follow:

*"**Step1. Initialize** the QoS metrics, allocate the **different weight** to the metrics, and make the **reconfiguration trigger threshold**.*

***Step2.** Compute the n**ew performance value** of the candidate configuration in the presence of failure using the **information** provided from the resource management and monitor management, judge if it is **under the threshold**, if yes, the **reconfiguration action will be triggered**.*

***Step3**. **Execute** the reconfiguration action.*

***Step4**. **Renew** the QoS **metrics**. Return to step1**."** [41]*

Resource Management, manages and evaluates the state of resources, and predicts possible future states of resources. This is also responsible to allocate the best-suited resource to the examined service. Then the **QoS metric** management constructs the metrics by adjusting the weight of the metrics. These metrics may provide the reconfiguration decisions. **Monitor management,** finally, gathers reports from system's state to feed the reconfiguration engine.

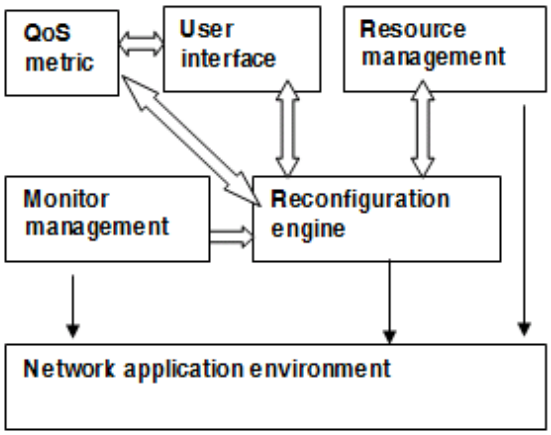


Figure 31.: Survivability reconfiguration framework base on QoS [41].

CACTUS methodology

Another approach that focuses on **find-grain customization** of services and Quality

attributes and **dynamic adaptation** of detected intrusions or failures is the **CACTUS methodology** [42]. Firstly, customization provides the system with the ability to conform to users' requirements for services and to environmental changes. Customization is being applied after taking into consideration trade-offs like the cost of detecting an intrusion versus the coverage of its appliance to the system. Then dynamic adaptation provides the ability to the system to customize its services at limited time during run-time. Again, this methodology helps the system to react to anything that may degrade its quality levels of service and to recover services after or during attacks or failures. All these make survivability a framework that may leverage existing and future approaches.

After the detailed presentation of approaches that are proposed by bibliography available, general rules based on these methodologies are to be adopted and form a methodology for the current research. To sum up the main objectives of a survivable system may be presented as below:

1. System's **mission** should be clearly defined. Services that serve system's mission should be separated to **critical and non-critical**. Critical services should be maintained under attacks or any kind of failures while non-critical services may be restored after failure.

2. Survivability is not an add-on characteristic. The system should be built to be survivable. For this to be succeeded, survivability should be part of all steps of **SDLC** of any system.

3. Survivable systems should be able to **recognise, resist and recover** from any failure or attack. For this to be succeeded, the system's services should be **monitored**, and if degradation of acceptable level of service is detected, the system should react. Reaction of the system may include **self- (re) configuration or self-healing** mechanisms in order the system to provide the best possible quality of services.

4. **Threat** to survivability is anything that may harm critical services. Since threats may not always be predicted, the impact of a threat realization may be considered each time in order the system to be able to resist to and recover from failure.

3.1.5. Theoretical Approaches on evaluation of Survivability

Evaluation of system survivability at literature review, is mostly based on defining different acceptance levels of system performance and on evaluating the impact by measuring the key properties like number of outages, time needed for system recovery etc. Though, these evaluation models are mostly based on node failures or link failures, and they are not giving a whole idea about the quality of service the system provides to end users. As a result, at the end, they seem to be based on system availability and

continuity and not on service or system's mission availability. Of course, system's availability is of vital importance for supporting system's mission and providing end to end system serving should be part of any survivability analysis and evaluation plan. As a result, many of these evaluation models could be very useful to pinpoint any possible network failures and include these to a test suite that would test if the system could recover from these kinds of failures or if it could function as expected during the system suffering from these failures. But it is very important to provide a guidance for testing survivability of systems from requirements specification up to delivery of new product release phase of system development.

Starting with Dongyan Chen et al [43], Markov model is used to map the failure possibility. They base survivability measurements on the frequency of failure events, on the duration of outages and on the impact of failure. As failure, since the research is conducted on wireless networks as a case study, what is included is node failure, power faults and link failures.

A very similar approach is proposed by Alex Hai Wang et al [44], who are using a semi-Markov survivability evaluation model for intrusion tolerant database systems. This is a very valuable research since there was no other quantification model focused on database systems until then. As key attributes for quantification of a database survivability, integrity and availability are proposed. Much focus is paid on system's functionality under failure and how system performs against these attributes.

To continue with quantification of system's survivability, Abdul Jabbar Mohammad et al [45], proposed network condition metrics which are density (based on topology and its changes), mobility (speed of node, predictability etc.), channel (bit error rate, capacity distribution etc.), node resources (memory, computing power etc.), network traffic (QoS, packet size, distribution etc.), derived properties (degree of connectivity, queueing delay, propagation delay etc.). Additional to those metrics, service requirements are also defined. Again, every adverse event, transits system's performance to another state which is quantified by these measurements (network performance based, and service performance based) so as to be marked as acceptable or no.

Another approach based again on Markov model is from Yun Liu et al [46]. It is focused on call losses of a telecommunication switching system because of various system failures like hardware/software faults, human errors, impairment damage from adverse environments etc. As key survivability metrics, system performance, availability and performability are used and the measurements proposed are measurements that can be used to describe system survivability such as the number of functioning units, the number of connected nodes, the maximum traffic capacity, blocking probability, throughput/goodput, and the service restoration time.

To continue with evaluation methods, Ming Liang et al [47], propose a testing survivability framework with focusing again on recovery part of survivability attributes. They firstly present the idea of 5-step phases of survivability of a system under failure, normal phase, resistance phase, destroyed phase, recovery phase and adaptation phase. Then they propose a scheme for representing the different stages of system performance against time during these phases. For quantification of network performance, two factors are proposed to be used, Node Connectivity Factor (NCF) and the Link Connectivity Factor (LCF). Practically though, they try to focus on the availability of end-to-end activity for the end user which is what really matters. Therefore, their research focuses on source-destination pairs "SD-pairs", to describe connectivity and service quality "SD-quality" and try these factors by applying different failures so as to calculate SD Recovery time for each pair. Finally, NRD metric is calculated to give an overall idea about the whole system survivability.

Another very important research on evaluation of survivability has been conducted by Chunlei Wang et al [48]. The framework proposed, is based on developing a general measurement model, which may be specified based on specific domain requirements, a network survivability testing model, which is based on testing network performance against survivability metrics during different steps of system performance (resistance, destroy, recovery), and the network survivability evaluation, which includes measurement of the whole system survivability based on different metrics, evaluation models or algorithms. The method concludes to a mechanism which if applied to a system under test, may provide all possible combinations of test schemes to test failures of a network and to measure them so as conclusions on overall system survivability to be extracted.

Another research by Ming Liang et al [49], propose measuring survivability by four attributes, Process-Weighted Average Availability (PWAA), Process-Weighted Average Controllability (PWAC), Process-Weighted Average Robustness (PWAR), Process - Weighted Average Adaptability (PWAD). These will depict the state of system through survivability life cycle, which is normal state, resistance state, destroy state, recovery state and adaptation phase.

Finally, another important approach for quantifying survivability is coming from Le-Jun Zhang et al [50], who propose to base quantification, on system's reaction to specific attacks and vulnerabilities modelled by attack graph. The attack graph represents the nodes that the attacker may exploit, and the way chosen to transverse these nodes to cover all possible system functionality states is forward-search, breadth-first and depth-limited algorithm.

To conclude, what may be observed is that most approaches on quantifying survivability are based on measuring availability and robustness characteristics of the system. Though, survivability is a more complex attribute that the system as a whole

should emerge and should be based on the ability of the system to continue serving critical services.

As evaluation method for the current research, software and system testing techniques will be chosen to evaluate system status. Furthermore, to measure systems performance key performance indicators will be used.

3.2 Mobile telecommunication networks

Mobile devices have become part of everyone's life. Mobile networks are wireless networks which allow data transmission to any geographical place on earth since such networks are free of the restrictions of wired network infrastructures. With the number of people using them and the number of applications that they serve increasing day by day, it would be safe to consider mobile networks as very critical systems for our lives.

Mobile systems are system of systems that nowadays have become a large part of the whole image of the internet as they serve transmission of voice, messages and data (music, video, images etc) by using the IP protocol. The image bellow presents the whole image of today's mobile networks which include 2G (second generation), 3G (third generation), 3.5 G (the release between third and fourth generation) and 4G (fourth generation).

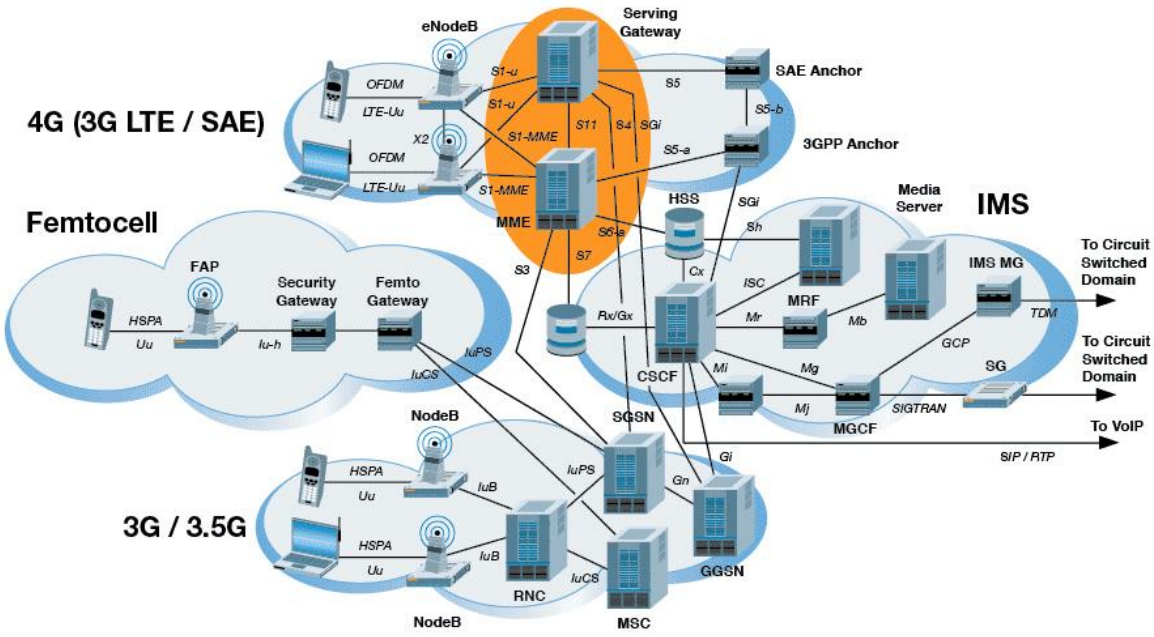


Figure 32: Mobile Network the whole image [51].

What is missing from this image is the 5G mobile networks and its interconnection with 4G. This may be depicted by the figure below.

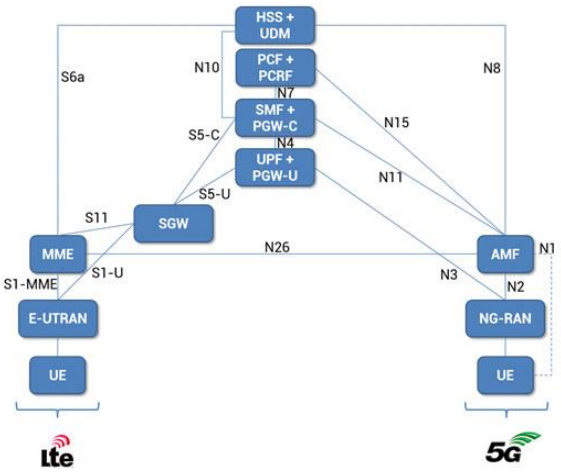


Figure 33: Common telecommunication network – 5G system added to the whole image [52].

Each of these systems has several nodes connected to each other. The particularity of mobile systems compared to other systems, like the internet, is that all services need an exchange of messages between a set of nodes to be established and performed. This makes the risk of failure to highly increase since failure of service may happen during exchange of any of these messages. To continue with this logic, these network nodes that are connected to perform a service, may be part of the same or a different network. For example, in 3G to 4G intersystem Tracking Area Update service, the nodes that may participate are from 4G, nodes eNodeB, MME, PGW, SGW, HSS and RNC/BNC, SGSN node from 3G network. This scenario may be examined to figure bellow. Additionally, nodes may be manufactured from different organizations which increases the risk of interoperability failures. As a result, with various nodes interconnected, new networks are formed adding new system and survivability requirements that must be considered through development of any new feature. The whole image may be depicted by the figures above.

The view of these interconnected systems that the current dissertation proposes for all stages of software development lifecycle for new features and for maintainace activities, is a multi-layered logic with the following levels:

Node level: Any node of mobile telecommunication network for which any new functionality or feature is to be developed. For example, MME should be considered to perform in node level.

System Level: As systems 2G, 3G, 4G and 5G systems, or any other that will follow, are considered. Nodes forming a system could be part of different PLMN operators. Any development task for a service that includes network nodes from same system should be considered in system level.

Intersystem Level: As intersystem the whole telecommunication system may be considered. Nodes forming a network for serving an inter-system scenario may be considered as an intersystem. For example, in the scenario below, an Intersystem TAU is depicted. The scenario includes nodes from 3G and 4G systems.

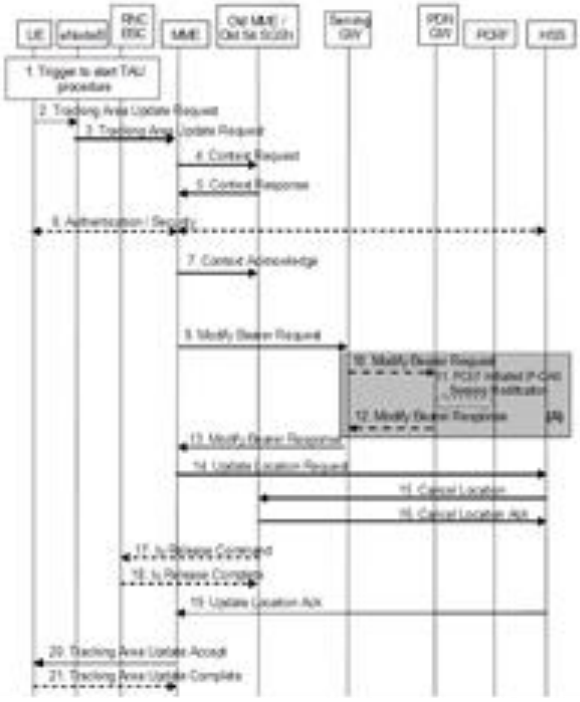


Figure 34: Intersystem TAU scenario [53].

To sum up, a service for a telecommunication system, is considered successful when each communication part between two nodes is successfully performed in order the path between terminal equipment and gateway to be established. For example, this virtual path is called Bearer for 4G networks and PDP Connection for 3G networks. This communication may be depicted by figure below:

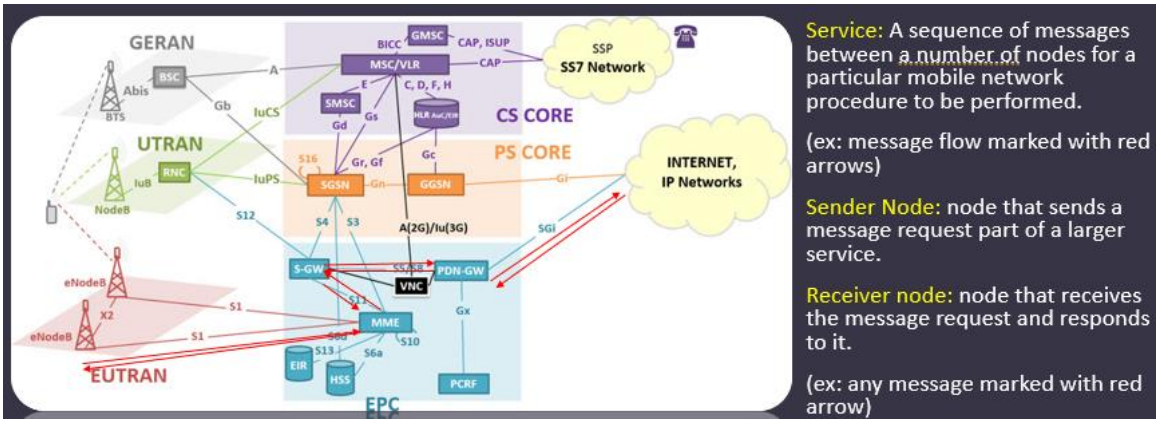


Figure 35: Service Establishment Example.

3.2.1 Mobile network from system perspective – Telecommunication Management System.

Mobile networks are considered to be very complex systems. Let's take as example the image bellow which compares the coverage of two providers of the mobile network in the U.S.A. We can see that the complexity of such a network is already overwhelming and it will be increased if LTE is spread to the point of older networks and if 5G or newer telecommunication technologies are added to the picture.

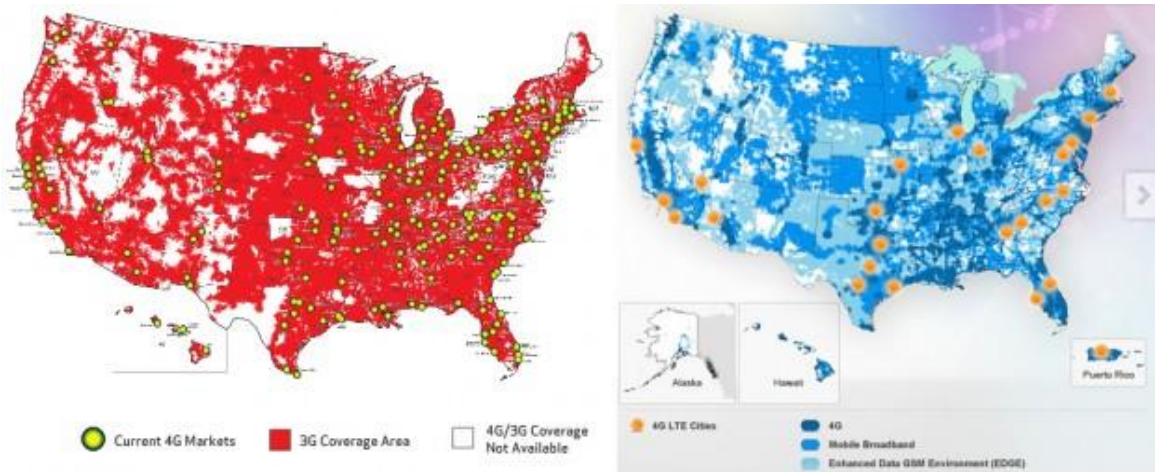


Figure 36: Compare of coverage maps of two providers in U.S.A. [54]

So, how could this complexity be controlled and what is the standard methodology (3GPP / ETSI) followed by telecommunication networks nowadays in order to manage such complex systems?

From system theory point of view a mobile system's characteristics, in accordance with telecommunication management rules according to 3GPP, may be listed as bellow:

1. **Components**: The components of a mobile system are:

    - Terminal devices (the mobile phone)
    - One or more access networks (ex. GSM, LTE radio network, PSTN, etc)
    - One or more core networks.
    - One or more intelligent node networks providing service logic and mobility management that extend the standard functionality of the system.
    - One or more transmission networks and physical means.

2. **Structure**: It would be safe to take as a rule that the mobile networks contrary to the internet have a hierarchical structure and are controlled by specific components of the system through control plane. What may emphasize this idea is that even if the mobile device needs to connect to the network for some service, it requests these services from mobility management device, and which then requests from the device to realize the service it requested.

   The components of a mobile network interact with each other mainly for serving signalling through control plane or for data signalling through user plane. Though these interactions are structured and controlled by the backend network. For example, the mobility management entity of the network controls certain number of antennas and it is connected with specific gateway for interaction of the network with other networks or the internet. Additionally, these interactions are supported through specific interfaces and their corresponding protocols.

3. **Behaviour**: the input / output and processing of the system as well as the characteristics of its behaviour.

4. **Interoperability**: the interaction of the system components and of the system with other systems and the environment. Interoperability also refers to the management of the emerging properties that arise from this interconnection.

   Mobile systems interact with other networks and are "exposed" to the internet. Additionally, the environment for a mobile network system has the characteristics of wireless and wired transportation mode. The interaction between the terminal devices and the radio network is wireless with the environment being the air and the interaction of the radio system with the core system and of the core system with other networks is affected by characteristics of wired networking.

   A telecommunication system may consist of different types of components and different technologies. In general, it will consist of access-, core-, transmission-, and service node networks and other components which follow various telecommunication management standardizations. The management including **monitoring** and **control of variety** of emerging properties resulting from the interconnection of these different parts is applied through **interfaces**.

5. **Functions**: the functionality of the system. System's **mission**. A mobile network's general functionality and system's mission is to provide in a timely manner and by specific predefined and even prepaid levels of quality of service, voice and data transmission services. More precisely, as it is presented by 3GPP:

   From the service perspective, the 3GPP system is defined to offer:

   - *Service support transparent to the location, access technique and core network, within the bearer capabilities available in one particular case.*

- *User to terminal and user to network interface (MMI) irrespective of the entities supporting the services required (VHE);*

- *Multimedia capabilities. [53]*

3.2.2. Threats to Telecommunication System Services

The following sub-chapter is part of the paper presented to "ICICM: International Conference on Information Communication and Management" ICICM 2019 with title "Fault Prediction Model for Node Selection Function of Mobile Networks" [55].

Threat to survivability is any attack, failure or disaster that may affect the system's critical services. If the system is survivable, the impact on critical services is minimal, no matter what the root cause of the failure is. Ideally the impact should be transparent to end users. Current research focuses on handling the complexity arising during the interconnection of network nodes, and specifically by choosing the always best, in terms of survivability, neighbouring node. This neighbouring node may be part of the same system or of another system in case of intersystem scenarios. An example of this may be an intersystem Handover that a UE travels from a 3G network to a 4G network. A central monitoring system that manages load and failures could always be useful by acting proactively to avoid service failure.

Since mobile networks are considered, the root cause of failures of critical services may be, due to fault management 3GPP [53], a result of a S/W failure, a H/W failure, a Path failure, a System State related failure like system load or overload situations, or Functional Failures (failure of a functional resource [53] in a node and no H/W failure).

1. S/W error in sender's or receiver's node may cause:
   - Malformed messages sent by sender node, causing the service failure in the receiver node or denial of a service by the receiver node if the message is not recognized at all. (Probability (P) of failure = 1)
   - Wrong processing of the messages at receiver's node causing any failure to the service. Any bug to the SW of receiver node. (P = 1)
   - Lack of robustness measures or failure of them in receiver's node, for example handling malformed messages or wrong information elements in messages, ignoring duplicate messages, handling of collision scenarios, handling synchronization issues like message delays by resending messages etc. (P < 1 for collision scenarios and synchronization issues and 1 for rest of the cases). Collision scenarios is when two messages arrive at the same time.
2. H/W failure of sender node may cause total denial of service for some time. Usually, restoration procedure defined by 3GPP [57] will be performed but there are cases that this is not succeeded. (P = 1)

3. Overload of sender node may also cause total denial of service for some time. Additionally, here overload control defined by 3GPP [58] / [59]) may be performed unsuccessfully or may not be performed at all. (P = 1)
4. Path failure / Communication Failure is another suspect of denial-of-service failure. (P = 1). Another failure regarding communication path may be delays on messages delivery or messages discarding which may cause any critical service failure (P < 1). Finally, loss of integrity of messages may be the result of a path failure (P < 1). As path between the two nodes could be considered any physical mean or even a whole network.
5. Finally, as functional failure may be considered any failure that may not be categorised as nothing has been reported, no alarms of failure or failure thresholds reached etc. For example, a large system load that has not yet reached overload thresholds may cause sometimes failures to services or some hanging resources may increase node's CPU or memory load that may also cause failures to critical services but the root cause most of the times is unknown. (P of failure < 1)

Any additional root causes identified during operation, could be added to the list of failures in order to be taken into account and thus prevent future failures of critical services.

3.2.3 Mobile network survivability measurements already implemented.

All constructors of mobile networks have already implemented various survivability mechanisms and have applied various measurements to the network. During this subchapter an attempt to gather most of these mechanisms and measurements is to be made. All these will be categorized to key survivability properties as described from R. J. Ellison et all [60], which are Recognition, Resistance, Recovery and Adaptation.

Recognition of failure - Telecommunication management system.

The telecommunication management system is the system that manages any event coming from any network node. It provides configuration capabilities, communication between network nodes and central management system, interoperability between PLMNs even if they are provided by different operators, fault and performance management through common measurements.

Management of telecommunication systems:

A management infrastructure according to 3GPP [53] should:

- provide services to the customer, through a suitable infrastructure.
- assure them (QoS, Fault Management etc.)
- bill them.

As a result, a telecommunication management system according to 3GPP [53] should provide:

- an architecture [OS-Operating system / NEs- Network Elements / Interfaces between them – Q within operator domain, X between different operators.] which will provide services with respect to business needs. The management of the equipment should support supply from different vendors.
- methodology to define interfaces between OS, NEs and different operators.
- tools to further define and refine Management Architecture (ex. Logical Layered Architecture LLA).
- number of generic / common management functions to be applied to interfaces with respect to business needs.
- flexible configuration capabilities for rapid deployment of services.
- allowance of remote control by reporting events and reactions in a common way.
- scalability and applicability to large and small deployments.
- cost effective solutions to operator's short-term needs.

Telecommunication management system presented by 3GPP [61] may be depicted by figure below:

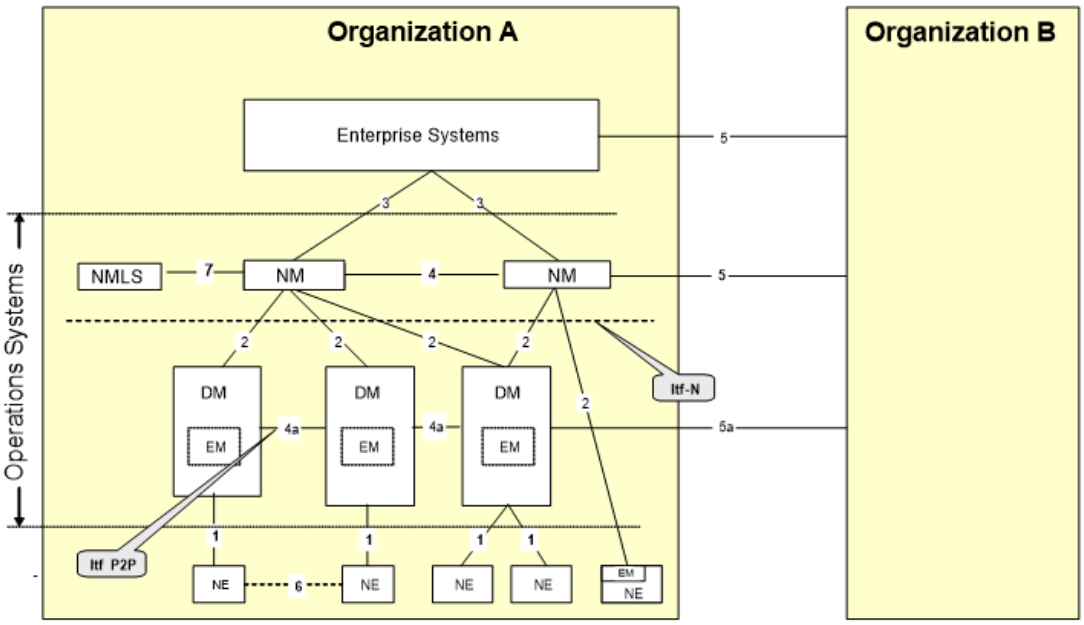


Figure 1: Management reference model

Figure 37: Management reference model [61]

"A number of management interfaces in a PLMN are identified in figure above, namely:

1. between the Network Elements (NEs) and the Element Manager (EM) of a single PLMN Organisation.

2. between the Element Manager (EM) and the Network Manager (NM) of a single PLMN Organisation;

NOTE: In certain cases, the Element Manager functionality may reside in the NE in which case this interface is directly from NE to Network Manager). These management interfaces are given the reference name Itf-N and are the primary target for standardization.

3. between the Network Managers and the Enterprise Systems of a single PLMN Organisation;

4. between the Network Managers (NMs) of a single PLMN Organisation;

4a. between the Domain Managers (DMs) of a single PLMN Organisation.

5. between Enterprise Systems & Network Managers of different PLMN Organisations;

5a. between the Domain Managers (DMs) of different PLMN Organisations.

6. between Network Elements (NEs).

7. between the Network Management Layer Service (NMLS) and the Network Manager (NM).

IRPs (Integration Reference Point) may be implemented at interfaces 2, 3, 4, 5 and 7." [61]

Through these interfaces, communication of information related to network elements management is been conducted. The structure proposed is a bottom – up structure and this is how information travels. Starting from NEs information travel to network manager entity. Information, that the current dissertation investigated, and is sent to network management has to do with alarms or various KPIs that measure the status of a NE. We will have a closer look of these later through this literature review.

Focusing on PLMN management which is related to the core mission of telecommunication systems, the management functions may be listed as below:

- *Performance management;*
- *Roaming management;*
- *Fraud management;*
- *Fault management;*
- *Security management;*
- *Software management*
- *Configuration management;*

- *Accounting management;*
- *Subscription management;*
- *Quality of Service (QoS) management*
- *User equipment management. [61]*

From these functions, focus will be paid on those related with survivability and these will be analysed below.

Performance Management

Performance management is firmly related to utilization of network resources and are based on load metrics gathered in NEs. The data may be transferred to Operations System for further evaluation. Then re-configuration may be an option that could be followed.

Fault Management

Fault management as it is indicated by 3GPP [61] includes fault detection, fault localization, fault reporting, fault correction, fault repair etc. This is depicted by table below taken from 3GPP [59]

ITU-T TMN Service Component TS 32.111-x [56]	**Telecommunication Operation Management Network Management Assurance Activities**
Alarm Surveillance	Detect Fault
Fault Localisation	Isolate Root Cause
Fault Correction	Decide Repair / Allocate Resources
Testing	Test

Figure 38: Fault Management [59]

In order to minimize the degradation of QoS the network users (network operators) must be able to:

- *detect failures in the network as soon as they occur and alert the operating personnel as fast as possible.*
- *isolate the failures (autonomously or through operator intervention), i.e. switch off faulty units and, if applicable, limit the effect of the failure as much as possible by reconfiguration of the faulty NE/adjacent NEs;*

- *if necessary, determine the cause of the failure using diagnosis and test routines; and*
- *repair/eliminate failures in due time through the application of maintenance procedures. [61]*

As we may see the proposal of the 3GPP standard is to provide network users with the ability to handle failures through detecting them by alarm system and through using reconfiguration of the system. Based on this logic, the current dissertation proposes an automatic way to handle failures that will be described in next chapter. The framework proposed is based on detecting failures and using reconfiguration proactively based on possibility of failure of an NE.

Most of these functions are located in the NEs and EM layers which have the self-healing capabilities. If events like alarms or KPIs from different NEs or EMs need to be correlated, this is realized through NM entities. In the first case, the fault management is reactive while in the second case it is proactive. Fault management process may be depicted by figure bellow:

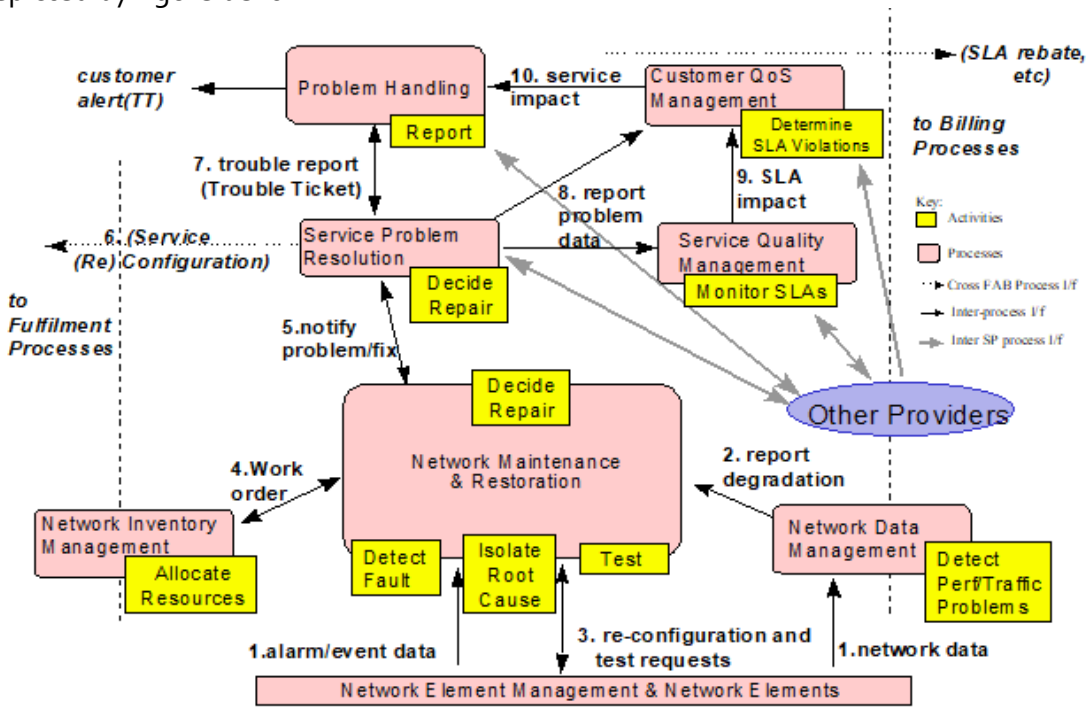


Figure 39: Service Assurance Process Flow [61]

Especially for software, the standard 3GPP [61] proposes a process for handling such faults. The steps are:

1. Detecting the software malfunctions in the network.
2. Perform corrective actions which may be reversion to an earlier software version, load and activation of correction software, re-activation of current software.

Though, we could say that this approach has some disadvantages because the failures that may happen because of loading the earlier revision may not be accepted. So, it is not something that is always possible to adopt as solution. What is proposed by the current dissertation is a framework for automatic detection of the failure and localization of the exact function in the source code that raises this failure. Then the reporting and correction of the error will be done much faster.

Key Performance Indicators and Alarms.

Starting with **Recognition**, 3GPP [62] **fault management requirements** indicate that it should be conducted by using and observing **measurements**, **counters** and **thresholds**. Based on these measurements, incidents could be characterized as ADAC (Automatically Detected and Automatically Cleared) for example inconsistencies of databases may be cleared by signalling between them, or ADMC (Automatically Detected and Manually Cleared) for example when a SW component must restart.

Counters and measurements may be an indicator of failure like a counter that indicates attach failures or call drops. For example, if the attach failures are more than a certain threshold then this may indicate that the system is overloaded or under failure and should be for example isolated.

Fault detection should be announced through an **alarm** system to operating system, to network or system operator who is a physical person. Such alarms should contain all necessary information of failure for further actions to be triggered like details about the faulty unit, the reason for overloading, the severity of fault, the cause of fault if known, etc.

For triggering corrective actions for the alarms, a **root cause analysis** should be performed. Network Element should have a management system for alarms to be able to clear alarms when resolved, to list active alarms and keep an alarm history to be possible for alarms to be retrieved and to organize buffering and storage of alarms. It should be noticed that a not efficient design of alarm management system could be a threat of survivability itself.

As a result, what is of vital importance for a fault management system is to be efficient enough so as the **root cause** of the failure to be discovered as soon as possible and be limited as far as possible. Especially in cases where multiple alarms are raised and most of them are correlated as they are results of the same root cause. The main functionality of Network Management system for this is the transformation of received resource alarms to service impact alarms. What is expected from this functionality is to depict the most critical and catastrophic alarms that should not be delayed or lost and must trigger immediate reaction of the system.

Alarms should be categorized according to criticality of faults. This would help the operators to separate alarms and act accordingly.

The characteristics that an alarm should have as described by 3GPP are:

- ***Relevance*** *i.e. not spurious or of low operational value;*
- ***Uniqueness*** *i.e. not duplicating another alarm;*
- ***Timeliness*** *i.e. not long before any response is needed or too late to do anything;*
- ***Importance*** *i.e. indicating the importance that the operator deals with the problem; alarms that only appear once and then disappear are not needed to be announced.*
- ***Explicability*** *i.e. having a message which is clear and easy to understand;*
- ***Recognizance*** *i.e. identifying the problem that has occurred;*
- ***Guidance*** *i.e. indicative of the action to be taken;*
- ***Prioritization*** *i.e. drawing attention to the most important issues.*

Additionally, alarms should identify clearly the management entity so as the faulty units to be maintained or replaced. Alarms that are self-healed, like stack overflow or insufficient memory could also not be announced. These could just be at log history. [62]

Fault management is performed through Integration Reference Points (IRP) the interface indicated by 3GPP for management of Telecommunication services. The system contexts of such an interface may be examined through figures bellow. These systems consist of IRP Agents or Entity Manager that generate alarms, notifications etc and the IPR Manager or the Network Manager that collects these data.

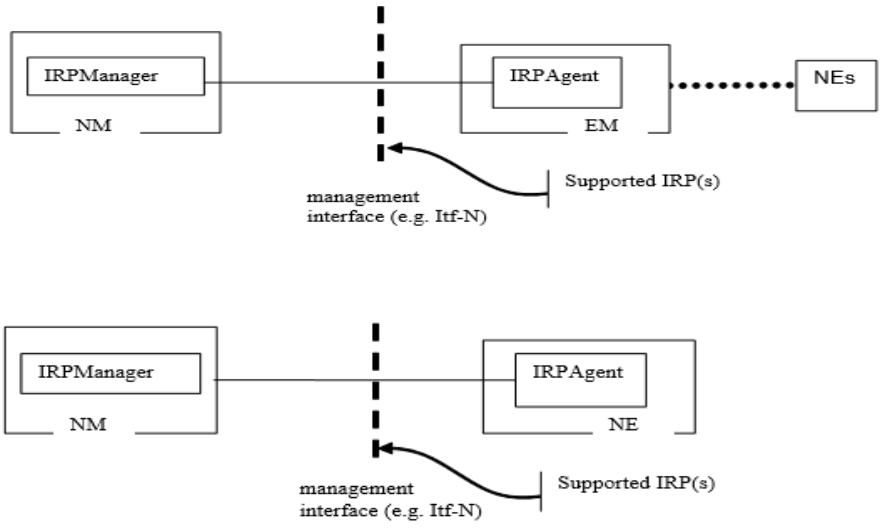


Figure 40: Integration Reference Points (IRP) [63]

As it may be observed, Itf-N interface connects Network Management system to Element Managers or to Network Elements directly. Through this interface alarms are arriving at NM element. There is also a need for a synchronization procedure of the alarm information arriving at Network Management element.

The supervised entities have to provide information about events and failures of the entity or the connections it retains and the network configuration. Additionally, the ltf-N should provide storing capability and later evaluation of information. Alarms should be available in a dynamic form describing the momentary alarm condition and allowing NM operator to monitor the system and in a history form so as alarms and logs of the past to be retrieved. It is also important that alarms for the same issue are raised only once and for this reasons alarms and logging information could be the same across Network Elements of the system.

For translating the alarms or state change event reports there should be a mapping of functional object generating these alarms or state change event reports and a corresponding object of the Itf-N. This mapping results to an alarm report of a logical object that models the equipment resource.

Finally, logs of **counters**, **states changing,** and **alarms** could have different log levels. Usually for testing purposes log level should be big enough to show as much information as possible. While operating the log level of the Network element is always lower for storing resources availability reasons.

The next step after alarm is raised, is to be acknowledged. This means that it is undertaken for treatment by an operator or the system.

Resistance mechanisms:

A list of resistance mechanisms available in 3GPP [53] are described below.

Dual Connectivity:

By this feature a UE may be connected to two eNBs at the same time. The advantage of this is the throughput and load balancing capabilities that are provided to the UE. Mobility Management though is been performed by the eNB that has the S1 Connection with MME.

Idle mode signalling reduction function (ISR):

This function provides a mechanism to limit signalling (TAU/RAU) during inter-RAT cell-reselection in idle mode. By this feature, the UE in idle state can reselect between UTRAN and E-UTRAN without any need to perform TAU or RAU procedures. For this to be succeeded, UE is registered to both MME and SGSN and has security contexts for

both networks. This means that it maintains the contexts for both networks as long as it is at the same TAI/LAI.

Load Balancing and Overload features:

- *GTP-C based load and overload control:*

By load control feature, Load Control information are sent to GTP control nodes in order to be taken into consideration on "PDN GW selection" and "Serving GW selection" procedures. The way that a node computes its load is implementation dependent. This information is communicated through any GTP control messages so as not to trigger extra signalling for loading mechanism.

By overload control feature, Overload Control Information are signalled through GTP-C in order to communicate overload situation to other nodes and actions to be taken. Overloading is reflected by nominal capacity in performance. This information may be used to reduce or throttle the amount of GTP-C signalling traffic between nodes. The overloading may be connected to specific APNs. In this case the resolution of the overloading may be applied by activation of "congestion control" by PGW or BY "PDN GW back-off time" feature.

If MME/SGSN receives overload information about an SGW or PGW it may reject any Session Management requests or Mobility management requests from UE, reduce or throttle messages towards overloaded GWs or other implementations may be assigned. If PDN GW receives overload indication for an MME/SGSN, it shall limit or completely block non-GBR bearer establishment, limit or completely block all dedicated bearer establishment, or other implementations may be applied.

What should be pinpointed in this case is that overloading detection and handling processes should not add any more "loading" to the overloaded node.

A conclusion that could be extracted from this mechanism is that the overloading is not transparent to the UE. As we will see later during restoration and resistance mechanisms, continuity of services to the UE is not guaranteed. They are only used to protect the mobility and data communication network and not the critical services. This is against survivability which is a mission driven capability as the mission of this system is to provide services to the final user.

- *Load balancing and re-balancing between MMEs:*

By load balancing feature, a UE that enters into an MME pool area, may be directed to an appropriate MME from load balancing perspective. For this to be succeeded every MME has a Weight Factor and the probability of the eNB to select an MME is

proportional to this factor. This is communicated to the eNB through S1-AP interface. Another policy could be based on UE configuration of low or high access priority which is transferred through RRC establishment signalling. For example, a UE with low access priority may have different load balance handling than other UEs.

By load re-balancing feature, UEs that are registered to an MME within an MME pool may be moved to another MME. Though if the MME is overloaded this feature should not be applied since other MMEs of the pool will probably be overloaded too. For the same reason, the balancing should be performed gradually.

In more details, when a UE in in connected state, MME initializes an S1 release procedure with cause "load balancing TAU required". In case UE is during TAU or Attach procedure, it completed this procedure and starts an S1 release procedure to proceed as described before. The same applies for a UE that is in idle state. MME pages the UE so as it becomes in connected state and proceed as described above.

After S1 and RRC release, UE initates a TAU without S-TMSI or GUMMEI which forces eNB during RRC establishment to select an MME from the pool based on Weight Factors. An example value of this factor is zero value if all subscribers have to be moved to other MMEs.

H/W or S/W failures of the MME may reduce its load capacity. In this case the Operation and Management mechanisms could detect this incident by certain alarms. Then operation and network management mechanism after ensuring that other MMEs will not be overloaded could trigger this feature.

- *MME control of overload:*

By this feature, MME is able to handle overload situation. For example, by rejecting NAS requests from UE. This may be applied by S1 overload procedure triggered for the suitable number of eNBs, that MME has S1 connection with, and reflect the amount of load that the MME needs to reduce. The selection of eNBs is random so as if two MMEs of the pool perform the same mechanism they do not both release the same eNBs. Though, random picking does not seem to be the better way to ensure such a situation as it is very possible that the same eNB may be selected after all.

By the "OVERLOAD START" message MME may request from eNB to reject RRC connections for non-Emergency and high priority services, reject new RRC connections (ex. TAU Update), reject new RRC connection requests from UEs that access the network with low priority etc. When rejecting a connection, eNB indicates a timer to the UE to not send further indications for a while. In any case support for emergency bearers should be maintained if possible. By this process there is a risk that if a voice service is marked as low priority not to be able to be served until the timer expires. So, these services should not be marked as low priority. After recovering, MME may send a

new percentage value in order to gradually permit more traffic or send "OVERLOAD STOP" message to some or all eNBs.

From eNB side, the "Extended Access Barring Feature" may be applied, in case all MMEs connected request to restrict the load for the UEs with "low access priority.

- *PDN GW control of overload.*

This feature provides mechanisms for avoiding and handling overload situations which may include rejection of requests from UE. PDN gateway may perform this functionality based on maximum number of active bearers per APN and/or maximum rate of bearer activations per APN. During this overload control PDN GW rejects requests from MME which rejects PDN connection requests from UE. Additionally, PGW may request the control of overload for a specific APN "PGW back-off time". Then the MME may select another PGW for that APN instead of rejecting PDN connections at once. If this is not possible, MME should reject the request.

Throttling of Downlink Data Notification Requests:

With this feature MME may restrict load that is coming from SGWs based on pre-configured threshold. MME may reject downlink data notification requests for non-priority traffic for idle UEs or request SGWs to selectively reduce the DDNs by sending according to a throttling factor, DDN Ack messages with throttling delay.

SGW may throttle bearers due to ARP priority and operator's policy and priority levels. MME will decide whether a DDN is priority or non-priority traffic based on ARP priority received from SGW or operator's policy. If ISR is not active, the SGW drops downlink data arrives for throttled bearers. If ISR is active, it only sends DDN to SGSN except if the SGSN has also requested a load reduction. Then it drops downlink data received.

The SGW resumes normal operation when throttling delay is expired.

NAS Congestion control:

By this feature, congestion of signalling for mobility management is handled. There are two modes of operation; "APN based congestion control" and "General NAS level Mobility Management control".

By APN congestion control the MME detects NAS signalling congestion associated with APN and starts / stops APN based congestion control based on:

-maximum number of active EPS bearers per APN,
-maximum rate of EPS Bearer activation per APN,
-one or more PGWs are not reachable or indicated congestion to the MME,

-maximum rate of MM signalling requests associated with a particular APN,
-setting in network management.

High priority contexts and emergency contexts should be excluded from this feature. This feature may be activated by MME or by network management system, or by restart or recovery of PGW, or by a partial failure or recovery of a PGW for a particular APN. With the use of a back-off timer, the deactivation of the feature is managed. While timer is running, MME rejects NAS requests and UE shall not initiate any NAS requests for Mobility Management procedures.

By General NAS level mobility Management control, MME may reject NAS level mobility management signalling requests. The functionality is pretty much the same as before.

UE Power saving:

This mechanism is used mostly for machine-to-machine devices like sensor devices. The mechanism provided focuses on changing idle state of the UE. Instead of been reachable, the UE is totally unreachable not even with paging mechanism and it may transit to connected state by periodic TAU,

Security functions:

The security functions include:

- *Guards against unauthorised EPS service usage (authentication of the UE by the network and service request validation).*
- *Provision of user identity confidentiality (temporary identification and ciphering).*
- *Provision of user data and signalling confidentiality (ciphering).*
- *Provision of origin authentication of signalling data (integrity protection).*
- *Authentication of the network by the UE.* [53]

-Authorization – authentication of UE. EPS AKA authentication and key agreement procedure between UE and MME.

-User identity confidentiality. Use of M-TMSI for identification of UE and MEE. This identity is only known in the UE and MME.

-User data and signalling confidentiality.

- RRC and UP security associations between UE and E-UTRAN for integrity protection and ciphering, and user plane encryption. Triggered by MME by sending AS security mode command to eNB. This enables ciphering of the UP traffic and RRC signalling, and integrity protection of the RRC signalling.

- NAS security functions between UE and E-UTRAN for integrity protection and encryption of NAS signalling. Triggered by MME by sending NAS security mode command message in order to protect NAS signalling. It is also used to change security association like security algorithm. It is part of Attach procedure, ME Identity Check procedure, and TAU procedure. MME sends NAS security mode command to the UE Including NAS algorithm, eKSI, ME Identity request and UE Security capability. UE responds with NAS Security Mode Complete (NAS – MAC, ME Identity).

Collision handling.

Part of almost all 3GPP documents that describe signalling on different interfaces, reader may come across description of collision scenarios. These scenarios include cases that two messages requesting something from a node arrive at the same time. Then a handling of which service to be served first should be designed to each node.

Implicit Detach procedure:

Implicit detach is the procedure followed by network to detach a UE that is idle for a long time period. This is a feature that may be used to eliminate signalling load and clean hanging connections that are not used anymore from a UE.

Mobile Network Nodes Recovery mechanisms:

Restoration procedures for GERAN, UTRAN and IMS networks (3GPP [57]).

The main objective of restoration procedures is the protection of data related to subscribers:

- Information stored in location registers.
- Subscriber data required to handle traffic. (mobility management / routing / security).
- Subscriber data required for supplementary services.

Recovery functionality is focussing on database nodes HSS, VLR, HLR and the rest of the nodes that perform mobility management services, data services and charging services.

The recovery mechanisms described in 3GPP [57] may be gathered to the following categories:

1. **Data corruption**: Inconsistency in subscriber data: For database entity VLR, this might mean that information is received for an MS or UE with no record. This will trigger mechanism to restore data in VLR from HSS. For MME or SGSN a message will

be received from a UE or MS that IMSI is unknown. Then again, these nodes should trigger the mechanism to restore data from HSS.

2. **Restart**: In case a node restarts, the rest of entities should be informed or detect the failure on their own in order to release any connection with the "restarted" entity. The same is true if VLR database entity restarts. All associations with this VLR should be deleted and re-established. If HSS is restarted, a non-volatile back-up should be used to restore all data and all associations with other nodes should be re-established.

3. **Failure**: A failure is detected by neighboring nodes since the failed node has no valid connection for a certain period. All associations with the failed node should be released. Though, there is the option to keep a healthy part of the path in order to avoid signaling for re-establishment. For example, in case eRab part of an established bearer fails, the part of connection between MME and SGW may be kept.

4. **Path failure**: path failure is handled to each node as the neighboring node is under failure.

A very detailed description of these mechanisms is presented in APPENDIX B' of current document.

MS Restoration Mechanisms – UE in idle mode for a long time:

- *Periodic Location Update*

Network requires from MS to periodically (due to an MS timer) establish radio contact in order to confirm its location. By this way, after a location register failure, the location data will eventually be restored to the Network.

- ***Periodic Routing Area Update***

All GPRS-attached perform routing area update for restoration of location data.

Self - configuring and self - optimizing network (SON) – 3GPP.

This concept has been first introduced at 4G networks and it seems that it will be continually evolving to the future. Self – organization in general is used during pre-operational state and operational state in order to reach the objectives bellow:

- Improvement of system operability under multi-vendor environment.
- Measurements and performance be common to different vendors.
- Ease network performance analysis and problem finding.

- Reduce effort for maintaining network at accepted levels.
- Increase network performance and quality reacting to dynamic processes of the network.
- Reduction of operational efforts and complexity.

SON can be divided into three categories; self-configuration, self-optimization, self-healing. SON solutions serve functions related to optimization of the network according to coverage and capacity by use of measurements as call drop rates or traffic counters which are used for pinpointing capacity problems. More precisely self-configuration and self-optimization are defined by 3GPP [64] as:

- *"**Self-configuration** process is defined as the process where newly deployed nodes are configured by automatic installation procedures to **get the necessary basic configuration for system operation**. This process works in pre-operational state. **Pre-operational** state is understood as the state from when the eNB is powered up and has backbone connectivity until the RF transmitter is switched on."*
- *"**Self-optimization** process is defined as the process where **UE & eNB measurements and performance measurements are used to auto-tune the network**. This process works in operational state. **Operational state** is understood as the state where the RF interface is additionally switched on. "*

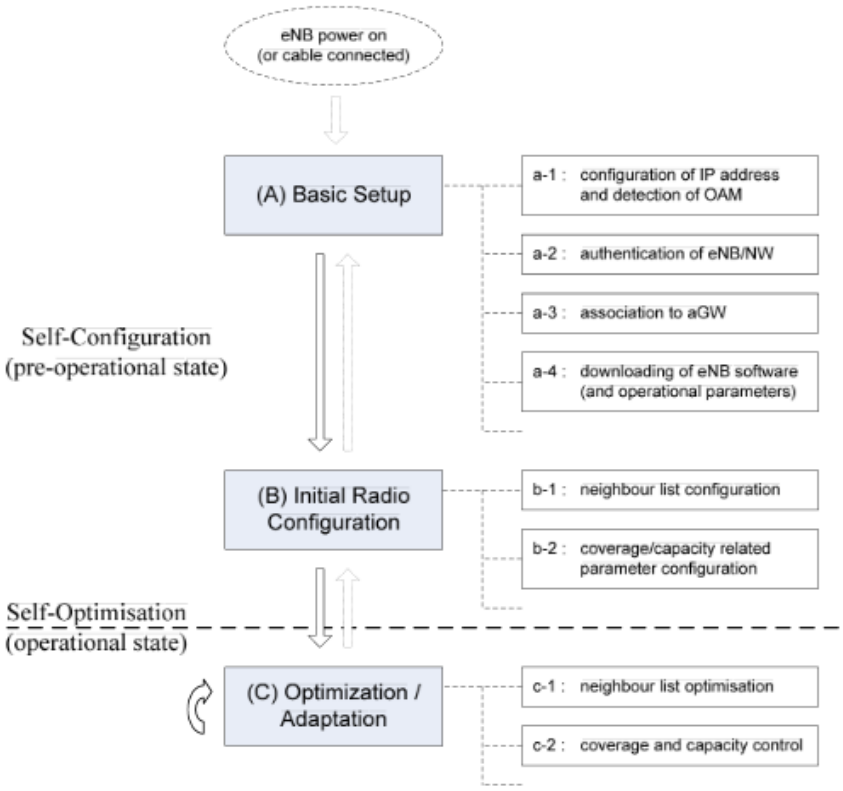


Figure 41.: Architecture of logical self-configuration / optimization functionality [64]

For self-configuration but most importantly for self-optimization it is very important to include the UE into analysis as it should be able to support measurements focussing for example on location or normal system operation and report them to the system and it should be able to be a self- optimized and self-configure unit of the system itself.

- ***Self-configuration***

Starting with self-configuration, the functions that take place are basic setup and initial radio configuration as it may be observed from figure 41 above.

With basic configuration the new node eNB is installed into the network as it is configured with an IP address, it is authenticated, it is associated with a gateway and the corresponding necessary software is installed to it.

With initial radio configuration the new eNB becomes a functional unit of the certain network it participates to as it is informed about its neighbours by the **ANR (Automatic neighbour relations)** and it is configured with certain information for its coverage area like the layer1 identifier, the physical cell identity, global cell identity and the transmission frequency and power. As a result, S1 and X2 interfaces are automatically configured. Dynamic configuration includes Layer1 id, PCI (Physical Cell Identities) and Cell global ID.

Starting with **PCIs**, there are two ways of assignment; centralized or decentralized. With centralized assignment the OAM (Operation Administration and Maintenance) system will have a complete reference and control for all PCIs. With distributed solution OAM assigns list of possible PCIs to eNB which is under installation and the eNB itself is the one that will adopt a PCI. This is applied by requesting a report of which PCIs are already in use, from UE (air interface) or by other eNB (X2 interface), and it will select a remaining PCI. For LTE there are 504 different PCIs which if mapped to eNBS contribute to a **collision and confusion free communication**. PCIs are included in any measurement report and combined with CGID they are used for handovers. The mapping between PCIs and CGIs can be done by OAM information or by UE reports.

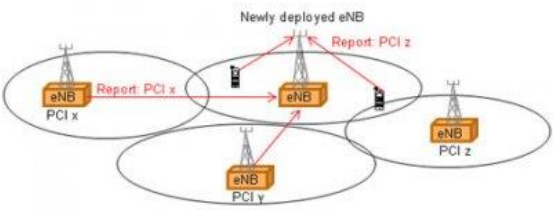


Figure 42: PCI Reporting [65]

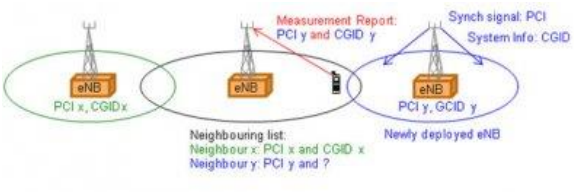


Figure 43: UE supported reporting of CGID. [65]

ARN is used for configuration of the eNB under installation and during operation. Maintenance of neighbouring lists will increase the network performance by increasing the **number of successful handovers** and minimizing the **number of dropped calls**. An ARN list may be used from an operator to block certain handovers for example from indoor to outdoor cells. Finally, it is applied to both inter-frequency and inter-RAT functionalities.

- ***Self-optimization.***

By self-optimization processes, an eNB node may be updated with network information that have changed such as the neighbouring list, coverage and capacity control information, handover information and interference details.

Starting with **Mobility Load Balancing (MLB)** based on periodic reports for load levels and available capacity, cells suffering congestion can transfer load to other cells with spare resources. For this to be succeeded, periodic reporting of load is applied between eNBs.

Such a report may contain:

- H/W load
- S1 transport network load.
- Radio resource status. (UL/DL)
    - Total allocation guaranteed and non-guaranteed bit rate traffic
    - Percentage of allocated PRB (Physical Resource Block)
    - Percentage of PRBs available for load balancing.

In case of inter-RAT load (use of MRB between different technologies), reporting is served by RAN information management protocol (RIM). Then a cell capacity class value set by OAM system will compare the different technology reports and weigh to apply an optimal solution.

Handover because of load balancing reasons will be conducted as a typical handover. Though, some parameters to the UE must be set, so that the UE does not return to the previous cell in case of being near to the limits of two cells. Additional configuration must be applied to source and target cell which might include shift to cell borders to avoid quick return of UE to source eNB.

Mobility Robustness Optimization (MRO) is used for automatic detection and correction of errors in the mobility configuration such as Radio Link Failure (RLF) as a result of too late or early handover or handover to an incorrect cell.

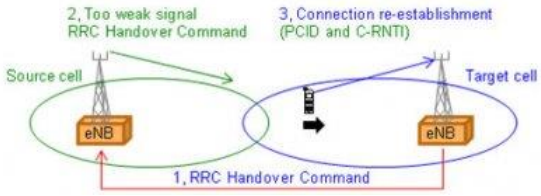


Figure 44: Late Handover, the UE does not receive the RRC [65]

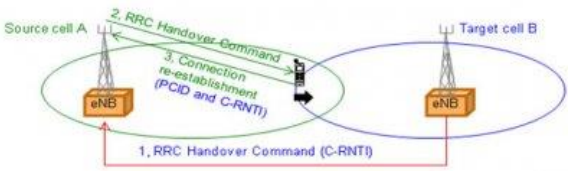


Figure 45: Handover too early, the signal strength in the target cell [65]

To avoid this, eNB should estimate a cell border shift so as the UE does not return.

Additionally to load balancing, **energy savings** are taken into consideration by the use of a mechanism that switches off cells when capacity is not needed. A cell though must be traceable to the network so as to be switched on by neighbouring cells when needed by wake-up call.

Finally, **Random Access Channel (RACH)** optimization is used to minimize interference coming from many attempts over the RACH. The report used by the eNB contains number of preambles sent until successful RACH, and number of failures. Additional information that may be used by a number of eNBs is zero correlation, root sequence, high speed flag, PRACH frequency offset etc.

- ***SON Use Cases as described by 3GPP.***

Firstly, it would be very useful to pinpoint the **metrics** that may be used to monitor the LTE system, represent its state and which are used to depict any threat or system malfunction. Some of these metrics are call drops, handover failures; call setup delays, handover delays, throughput, and traffic counters etc. These metrics are presented in accordance with the corresponding use cases of self-configuration and self-optimization.

- *Coverage and Capacity Optimization.*

These two objectives should be also considered as a general scope of the system and are presented below as defined in 3GPP [66] which is the specification for SON.

"Providing optimal coverage

*This objective requires that in the area, where LTE system is offered, users can establish and maintain connections **with acceptable or default service quality, according to operator's requirements**. It implies therefore that the coverage is continuous, and **users are unaware of cell borders**. The coverage must be therefore provided in both, idle and active mode for both, UL and DL.*

Providing optimal capacity

While coverage optimization has higher priority than capacity optimization in Rel-9, the coverage optimization algorithms must take the impact on capacity into account. Since coverage and capacity are linked, a ***trade-off between the two of them*** *may also be a subject of optimization." [66]*

For providing these, network planning tools must be considered. Based on theoretical models and measurements these tools may support network optimization. Examples of measurements are call drop rates for coverage and traffic counters for capacity problems.

- *Energy Savings.*

For increasing the capacity requirements in an area, capacity booster cells may be distinguished from cells and be switched off during low traffic periods or switched on due to traffic demands. Capacity booster cells are limited coverage cells providing the basic coverage. This may be applied though X2 interface.

- *Interference Reduction.*

This feature is based on switching on and off cells according to the needs of the network. This increases the network capacity and quality through interference minimizing.

- *Automatic configuration of Physical Cell Identity.*

This feature is based on an unavoidable reuse of the physical Ids to different cells so as the network to be collision and confusion- free. This means that the PID will be unique in the area that the cell is included in and that the cell will not have neighbours with the same PID.

- *Mobility Robustness Optimization.*

The main scope of this feature is to reduce the number of HO- related radio link failure and to avoid degradation of performance of services by for example ping-pong handovers. HO-related failures may be listed as follow:

- Failures due to too late HO triggering.
- Failures due to too early HO triggering.
- Failures due to HO to a wrong cell.
- Cell- resection parameters out of HO may result to these three.

Detection of HO - related failures is a main objective of this feature. For too-late HO the Radio Link Failure (RLF) occurs short time after the UE has connected to the target cell. Then the UE should re-establish the connection to a cell other than the source cell. For the too-early HO, the RLF occurs short time after the UE has successfully connected to the target cell. The UE must re-connect to the source cell. Finally, for the handover to the wrong cell the Cell Individual Offset (CIO) should be set incorrectly. The RLF occurs after the UE has connected to the target cell and it should re-establish a connection to a different cell from source or target cells.

Finally, HO parameter optimization function should be applied to the network in order to avoid unnecessary HO coming from false or miss-set user mobility patterns or cell coverage layouts. On the other hand, incorrect handover parameters configuration may lead to miss of HOs that should be executed. Those optimization parameters must be aligned with cell reselection parameters in order to avoid unwanted HOs coming from connection setup.

- *Mobility Load Balancing Optimization.*

The main objective of the mobility load balancing functionality is the optimization of cell reselection/handover parameters to handle unequal traffic and number of handovers and redirections needed to achieve load balancing. With respect to QoS parameters, service capabilities of RATs and network capacity limits, the cell-optimization of load parameters in a cell and in adjacent cells improve system capacity and minimize human intervention in the network management.

The functionality required for balancing the load, is based on an **algorithm** to distribute the UEs of a cell (idle or connected). The eNB monitors the load of the cell and informs neighbouring eNBs over X2 or S1 interfaces. The algorithm by comparing the load among the cells and taking into consideration the cell configuration or the type of ongoing services, decides that the load needs to be distributed to adjacent, co-located cells or even cells from other RATs. In cases of intra-LTE, negotiation for configuration of HO parameter trigger settings between involved eNBs takes place.

The load balancing information exchanged between neighbouring eNBs includes:

- Current radio resource usage (UL/DL GBR PRB usage, UL/DL non GBR PRB usage, UL/DL total PRB usage.)
- Current HW load indicator (UL/DL HW load: low, mid, high, overload)
- Current TNL load indicator (UL/DL TNL load: low, mid, high, overload)
- Composite available capacity indicator (UL/DL)
- Cell capacity class indicator (UL/DL).

- *RACH optimization*

RACH is used for initial synchronization between the device and the base station. It is not a collision free mechanism though and may cause call setup delays, data resuming delay, HO delays and affects the call setup success rate and the handover success rate. As a result, any network configurations (transmission power, antenna tilting, handover thresholds etc.) may also affect RACH (Random Access Channel) configurations. The solution is an automatic RACH optimization function to monitor the environment and determine and update appropriate parameters.

- *Inter-cell Interference Coordination:*

Self-configuration and self-optimization of control parameters for uplink and downlink lead to effective coordination of resource usage in related cells.

➢ ***Self-healing***

Self-healing includes automatic detection and removal of failures and automatic adjustments. By coverage and capacity optimization, problems of capacity in accordance with variations of the service demands, due to seasonal variations for example, may be solved. Finally, by minimization of drive tests (MDT) UE may retrieve and report information and parameters from indoor environments.

The concept of self-healing is the nearest concept to survivability as it presents the automatic or manually solution of faults by certain recovery actions (ADAC /ADMC). Triggering of self-healing mechanisms is applied by appropriate alarms which are continuously monitored. If any alarm is triggered, then there are two possible actions. The first is gathering more information in order to conduct deeper analysis by measurements, testing results etc and the second focuses on cases that need more rapid response and have to be healed automatically by appropriate self-healing process.

After the end of self-healing functionality, the IRPManager must be informed for the results and information about recovery actions must be logged.

Threats that 3gpp may be healed by self-healing process are divided to software faults, hardware faults and other faults. And the recovery actions are presented bellow:

For software faults the recovery actions are:

a) system initializations (at different levels),
b) reload of a backup of software,
c) activation of a fallback software load,
d) download of a software unit,

e) reconfiguration etc. [67]

For hardware faults the recovery actions depend on the existence of redundant resources. If no redundant resources are available, the recovery actions may be:

a) Isolate and remove the faulty resource from service so that it does not disturb other working resources;
b) Remove the physical and functional resources (if any) from the service, which are dependent on the faulty one. This prevents the propagation of the fault effects to other fault-free resources;
c) State management related activities for the faulty resource and other affected/dependent resources;
d) Reset the faulty resource;
e) Other reconfiguration actions, etc. [67]

In case there is a backup system, the steps a,c,d, and a specific recovery sequence are applied. In cases of other faults, the recovery actions are for further study.

In 3GPP the self-healing process is separated to two parts; the monitoring part and the healing process part. The state diagram of the process may be depicted from figure 46 bellow. The steps included are:

[Monitoring part:]

[SH01] The Self-healing Function monitors the TCoSHs (Trigger Condition of Self-Healing) continuously.

[SH02] When a TCoSH is reached, then an appropriate Self-healing Process shall be triggered.

[Healing process part:]

[SH03] The Self-healing Function gathers ***more necessary information*** *(e.g. measurements, CM data, testing result, etc).*

[SH04] Based on the TCoSH and gathered information, the Self-healing Function does ***deep analysis and diagnosis****, and gives the result. If the result includes recovery action/s, then go to next step, if not, go to End.*

[SH05] The ***configuration data*** *prior to the executing of the recovery action/s is* ***backed up*** *if needed.*

[SH06] If necessary, the Self-healing Function triggers the executing of the ***recovery action/s****.*

*[SH07] The Self-healing Function **evaluates the result** of the self-healing recovery action/s:*

*If the fault **hasn't been solved** and the **stop condition/s is not reached**, then the self-healing runs again, i.e. go to SH03.*
*If the fault **has been solved**, then go to [SH09].*
*If the **stop condition/s is reached**, then:*

*[SH08] If necessary, **fallback is executed**. Go to [SH09].*

*[SH09] The Self-healing Function emits a **notification to report the result** of the Self-healing Process.*

*[SH10] If necessary, the Self-healing Function **logs the information of the performed recovery actions and the occurrence of important events** during the self-healing process.*

Remark :

The detailed healing process part of the individual self-healing use cases may differ from this general description, for example:

1) The order of the bullet points in the list does not imply any statement on the order of execution.

2) In [SH05], whether the backup of the configuration data is needed and which configuration data should be backed up shall be decided on a use case by use case basis.

3) In [SH08], whether a fallback is needed shall be decided on a use case by use case basis.

4) In [SH10], whether log is needed and the detail of the logged information shall be decided on a use case by use case basis [67]

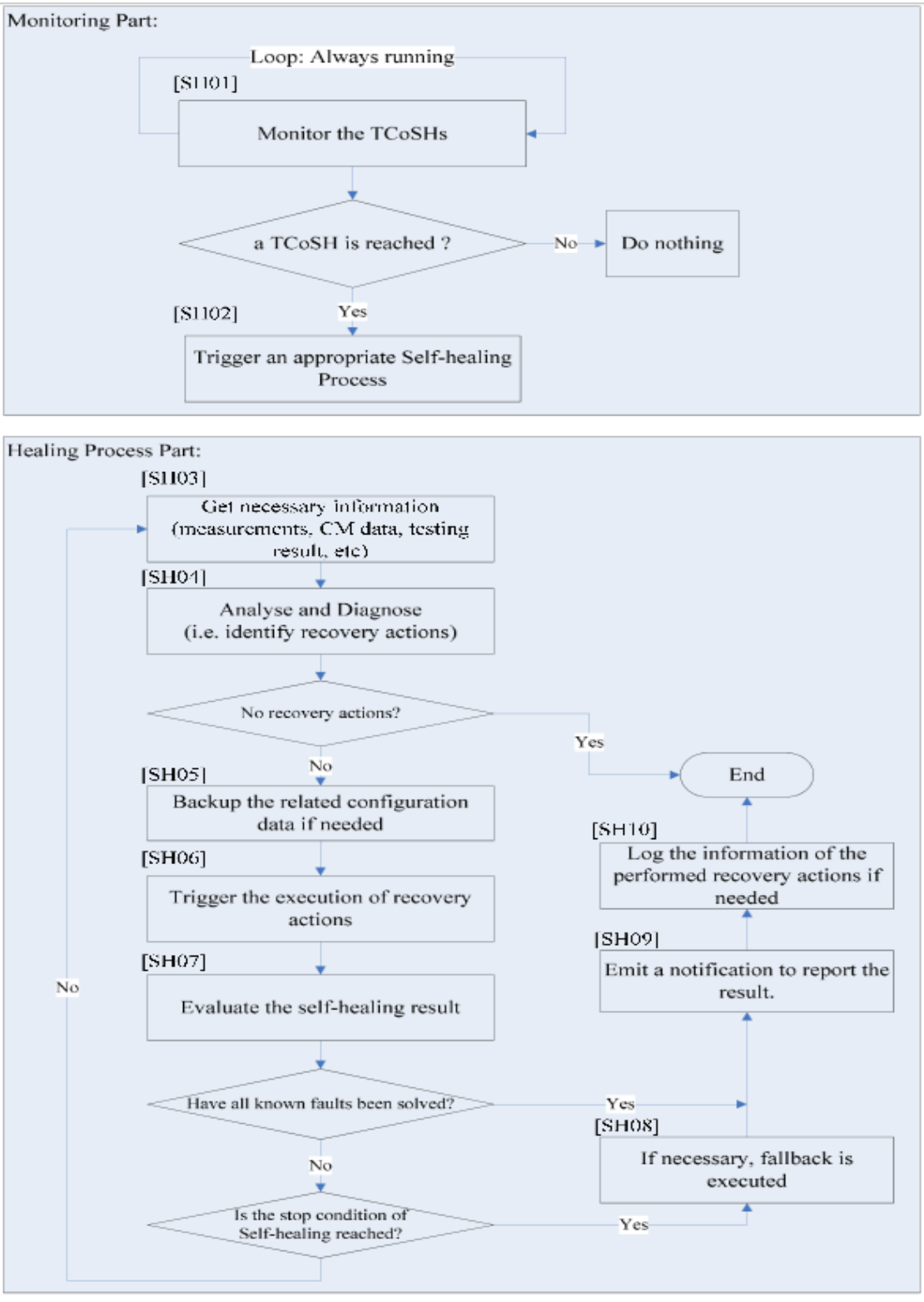


Figure 46: Self-Healing Process [67]

To sum up, recovery and self-healing processes described by 3GPP, have some disadvantages for which current dissertation will propose some solutions:

1. Solution proposed in case of software error is providing a fallback version. This seems not to be practical though since older version may have major issues for which it was replaced. So, in practice this is not something that would be used as solution. Another solution on this could be to wait by avoiding faulty node until the issue is solved by development team. This is feasible since operators hold nodes from different companies for back up and restoring processes.

2. In case of load or H/W failure, the solution proposed by 3GPP is to release all connections which means a huge loss of services. Survivability is focussing on protecting critical services. So, the solution proposed by current research is network management entities to be able to predict this event and avoid nodes that will cause service failure.

3.3 Software Development Lifecycle – Software Testing Lifecycle

3.3.1 Software Development Lifecycle

Before leaving literature review, some information about software development lifecycle and testing will be presented. Many software development lifecycles are available to be used for project management purposes in order to organize and assure that the product that will be produced is compliant to customers' and legislation requirements, the number of defects is low, the quality is assured to be high etc.

Starting with **waterfall model,** which is the most representative model of a linear-sequential life cycle model, development is divided into separate phases which are followed linearly. The output of one phase is input for the next phase.

Phases of waterfall model are:

Requirements Specification and Analysis: Gathering of all possible requirements for the system under development. In case of a survivable system, survivability requirements should be part of this phase. All requirements should be documented and used in next phases.

System Design: During system design, the design of these requirements takes place. During this phase, H/W and system requirements, and system architecture are defined.

For example, in case of a new feature after the design phase, the development team should know exactly what changes should be applied to the code.

Implementation: The actual implementation of changes described during the system design phase.

Integration and Testing: All units developed are tested and integrated to form a system.

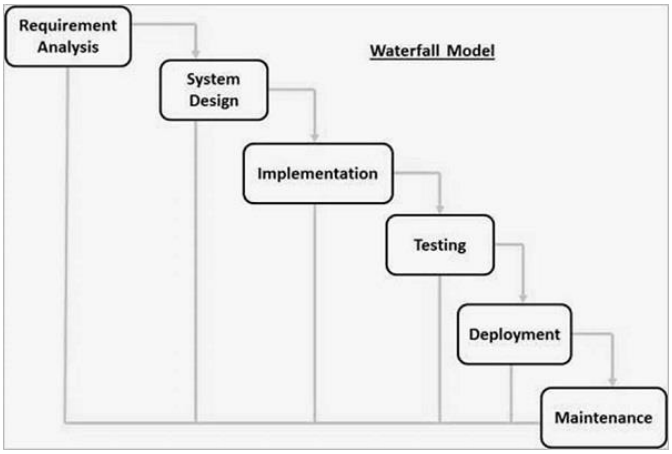


Figure 47: Waterfall Model [68]

The test process for waterfall model starts after the requirements specification, the design and the implementation phases. This means that testing is not part of the whole procedure. Testing phases according to Chris Brown et al [69] are:

Activity	**Inputs**	**Outputs**
Requirement's analysis	Requirement's definition, requirements specification	Requirement's traceability matrix
Test planning	Requirement's specification, requirements trace matrix	Test plan—test strategy, test system, effort estimate and schedule
Test design	Requirement's specification, requirements trace matrix, test plan	Test designs—test objectives, test input specification, test configurations
Test	Software functional	Test cases—test

implementation	specification, requirements trace matrix, test plan, test designs	procedures and automated tests
Test debugging	"Early look" builds of code, test cases, working test system	Final test cases
System testing	System test plan, requirements trace matrix, "test-ready" code build, final test cases, working test system	Test results—bug reports, test status reports, test results summary report
Acceptance testing	Acceptance test plan, requirements trace matrix, beta code build, acceptance test cases, working test system	Test results
Operations and maintenance	Repaired code, test cases to verify bugs, regression test cases, working test system	Verified bug fixes.

Table 2: Levels of testing for waterfall model [69]

Deployment of system: product is deployed and released to the market.

Maintenance: After product is released any bugs or faults are fixed and patches or update versions are released to the market.

Continuing with **V-model**, testing phases, depicted at figure bellow [70], are running simultaneously with development phases. So, when user requirements are defined, acceptance tests that will assure these requirements are developed. Additionally, when system requirements are defined, system test execution is taking place etc leading to unit test that will test the modules developed during implementation.

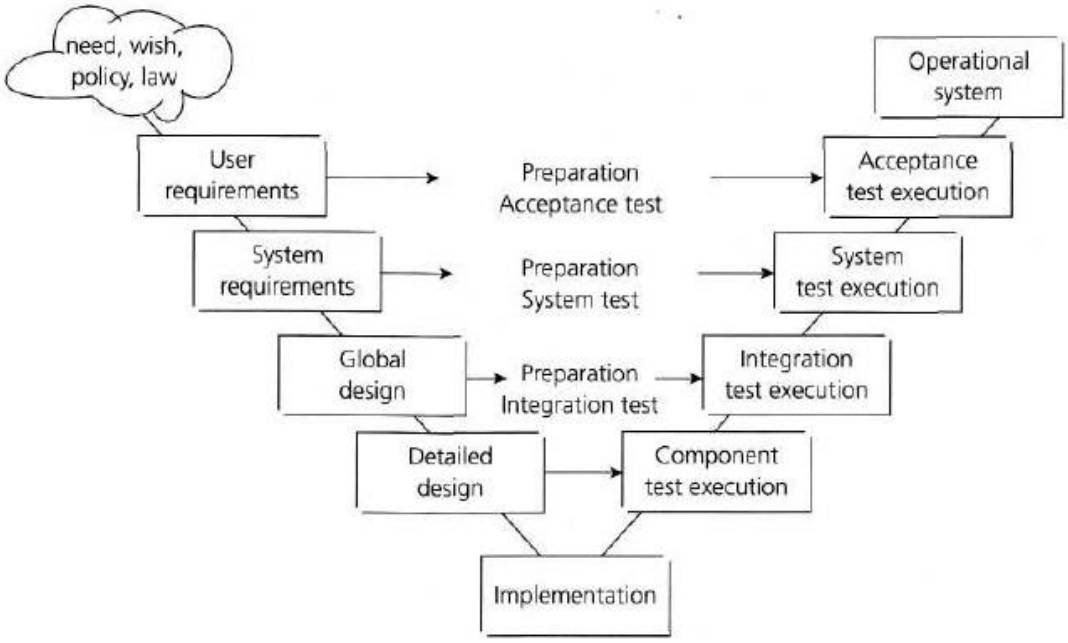


Figure 48: V-Model [70]

Another model that worth examination is the i**terative model,** and the proposal of Born-Yuan Tsai et all [71] for a testing framework called "Test Design Stages Processed Model". The iterative model is based on iteration of software development lifecycle stages in order to produce modules of the project at each iteration which then should be combined to provide a prototype.

"Iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented. At each iteration, design modifications are made, and new functional capabilities are added. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental)." [72]

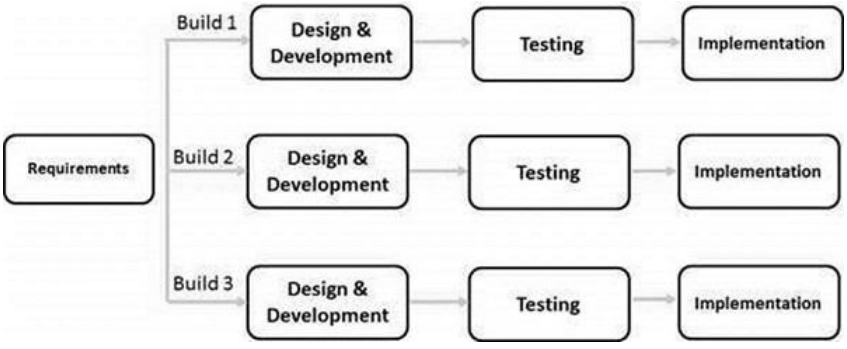


Figure 49: Iterative Model [71]

Focussing on testing, the main points of Born-Yuan Tsai et al [71] for testing of iterative model SDLC are:

1. TSP Lifecycle has a testing design stage. The output of this process will be test criteria, test plan, test cases and results expected.
2. Testing design phase is obligatory for a process to proceed.
3. Every time a module is ready is integrated to the whole system and the system is tested. If faults are found, the recently integrated modules are gone back to developers and in extend back to designers if the problem is there.

The model is horizontally divided into 3 phases:

(1) requirements specification, system test design and system testing,
(2) architectural design, integration test design and integration testing, and
(3) detail design, unit test design and unit testing.

The output of each process is the input for the next phase. For example, after module development unit testing takes place for this module when at the same time developers work on the next module. Depending on the phase the system is, other test methodologies should be used. For example, in requirements specification system testing of real system should be applied, in architecture design phase simulated testing that should test specific scenarios could be applied and during development process unit testing that tests the particular module could be applied.

Very interesting points to this methodology are:

- when test fails requirements specifications should be re-considered or test case and test design should be re-considered.
- testers should develop tests to evaluate interactions between subsystems.
- testers should develop tests to evaluate integration of different modules.
- at each iteration, testers may replace simulated parts with real modules in order to test integration of system.

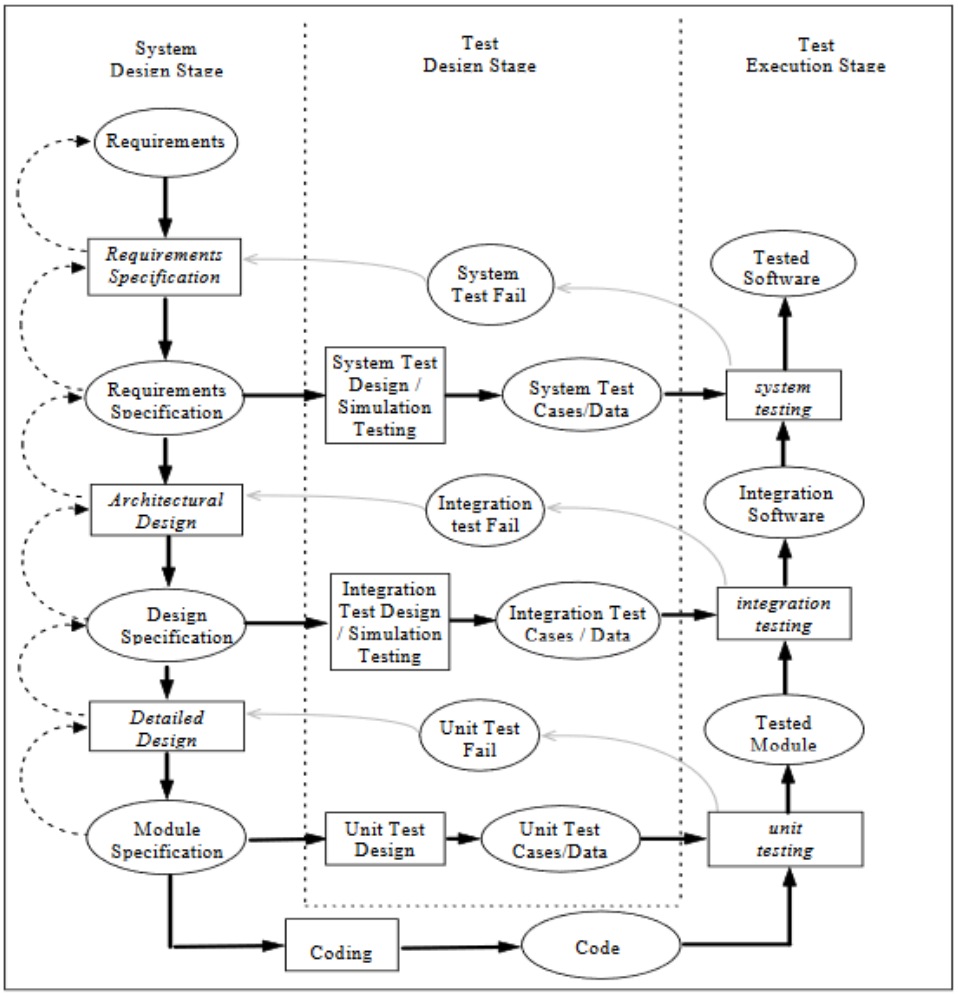


Figure 1 Detailed Overview of Testing in the Software Development Life Cycle

Figure 50: Testing Process [71]

It should be mentioned that all latest models of SDLC, like agile, followed by most of organizations, are based on iterative model of SDLC. **Agile,** is a representative approach of iteration model. Tasks are divided to time slots and at the end of the SDLC, specific features are delivered for a release. Each release is a new software build that contains all features that were planned to be added. A detailed definition of agile methodology is described below:

"*An iterative and incremental (evolutionary) approach to software development which is performed in a highly collaborative manner by self-organizing teams within an effective governance framework, with "just enough" ceremony that produces high-quality solutions in a cost effective and timely manner which meets the changing needs of its stakeholders."* [73]

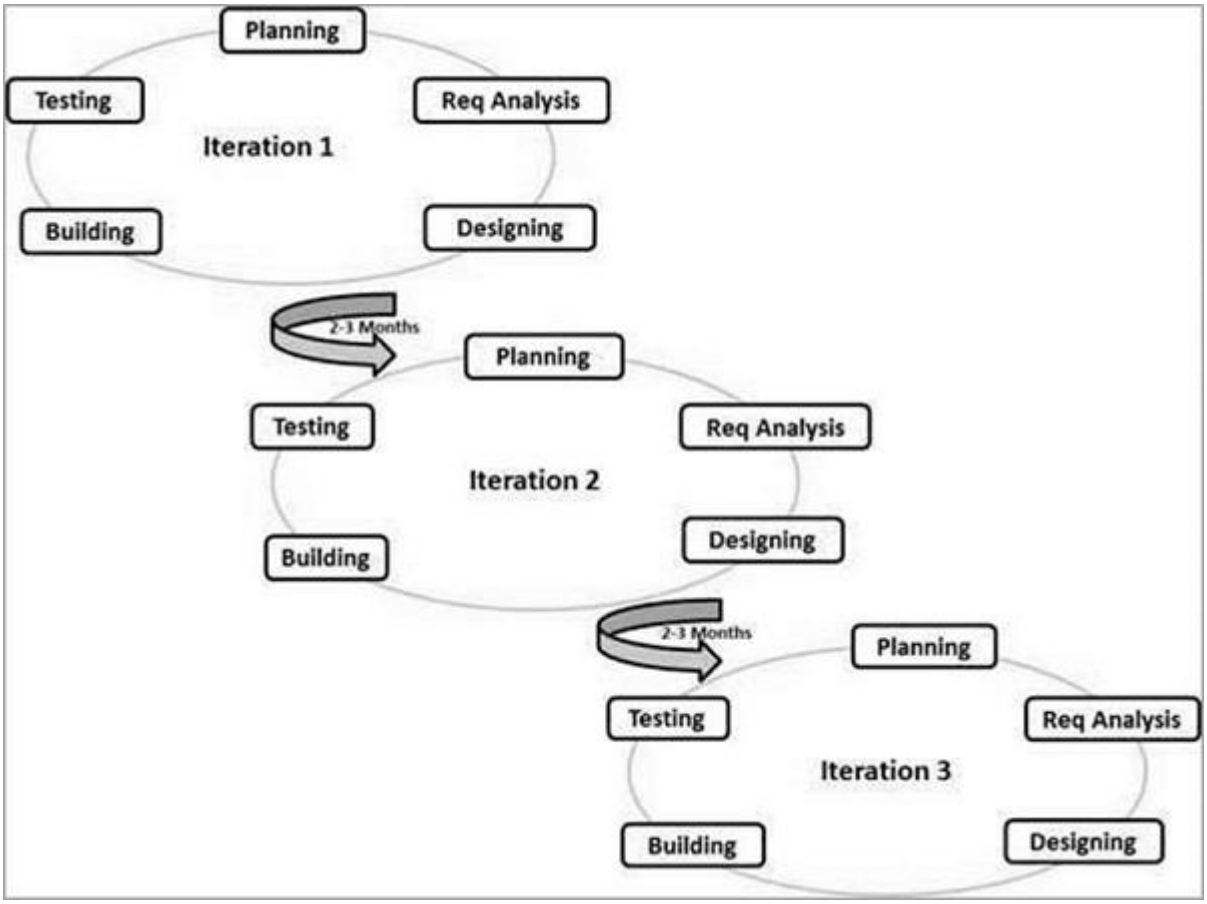


Figure 51: Agile model. [74]

The investigation of how resources will be allocated and how projects will be split to iterations in order to comply with development and testing activities will not be analysed for the current dissertation.

Focussing on Agile Testing methodology, there are some main principles:

1. **Continuous testing by continuous integration (CI) and continuous deployment (CD) tools** to ensure continuous delivery of defect free and quality assured product.

2. **Test Driven Development methodology (TDD)** which means testing is done while implementation and after requirements specification. By this methodology, requirements are depicted by test scenarios which should be fulfilled by code tasks that are continuously delivered. Tests should be written immediately after requirements specifications because what usually happens is testers and developers co-operate by pair development and tests tend to decline from requirements.

3. **Acceptance TDD:** Predefined test suites agreed with customer and conforming with clients' requirements specifications are included to delivery so as customer accepts the product.

4. **Everyone tests**. Apart from test teams, development teams also test and testing is as left shifted as possible so as to find defects to the earlier phases of the development of a new release of a product. Testing at early phases is conducted by Unit testing and new release is always tested against acceptance criteria by customer teams.

5. **Parallel Independent Testing**: An independent testing team could perform some more advanced testing based on software changes of the new release.

   An example of why this testing is needed, is the non-functional requirements that a system should emerge like **performance, usability or security**. These are missed when User stories, which are small parts of code that implement a functionality, are tested. Another example is production readiness or **deployment and upgrading testing** which tests the ability of the new release to integrate with other systems and to be installed. Additionally, methods like **exploratory testing** and n**on-functional testing** could be used. Exploratory testing is used to discover where product breaks specifications instead of testing where system conforms to specifications and non-functional testing or "test to the risk", is testing of attributes emerging from a system, some examples are memory leaks testing, performance or robustness testing, security testing etc. This methodology may also be used when regulatory compliance should be considered, when cost of testing is exponentially expensive, when the cost of finding a defect is raised the longer you wait, the technical environments are complex or large geographical distributed and finally when outsourcing teams are used to the product implementation.

6. **Left-shifting defect finding** for saving costs. This may be examined to the following curve:

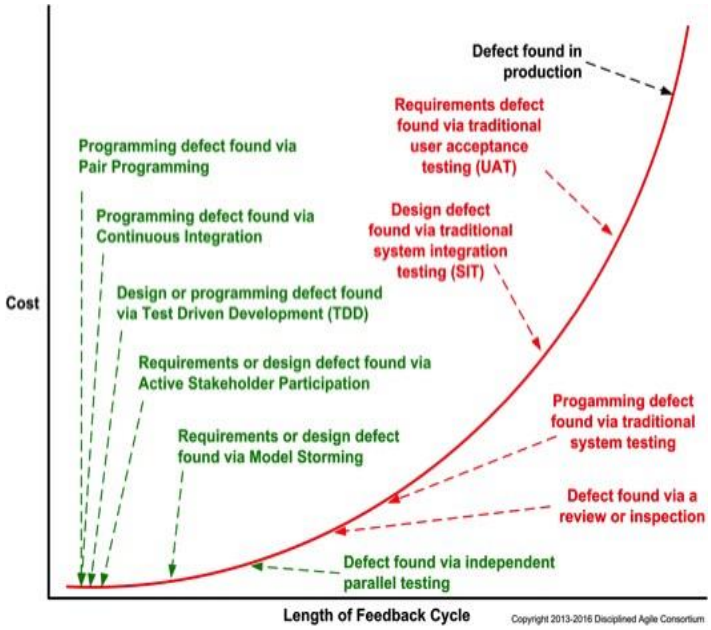


Figure 52: Left Shift Defects. [75]

7. **Defect Management:** Defects in an agile system are reported to development team most of the times through a tool. They are estimated and prioritized and dealt with as another new requirement. Which means that should be planned and tested as it was a new functionality.

8. **End of lifecycle testing:** Test the complete solution before it is ready to go into production.

Continuing with **spiral model,** there are a lot of approaches that consider this model as the most suitable for robust and survivable systems. An approach supporting this comes from Rick Linger et al [33] and it is described thoroughly to the current dissertation. Spiral model is considered as the most suitable model for survivability as it is a risk-driven development process to which survivability as a characteristic that system emerges should be added at all stages of spiral model lifecycle. What this characteristic is exactly, depends on the system's nature.

Key items of this theory are:

1. Survivability requirements that system should emerge are resistance, recognition, recovery and adaptation focusing on essential services which should be depicted.
2. Survivability requirements should be added at the "definition of done" list for all development processes of the lifecycle.
3. Among test scenarios, intrusion and fault scenarios should also be included to a

documented test strategy.
4. All stages of system functionality should be examined even those with degraded performance of system services.

Focusing on testing of this model proposed, system shall be tested against **functionality**, **performance**, and other system attributes. Additionally, **penetration testing** and **statistical usage-based** testing are two useful approaches for testing survivability. **Penetration** testing is focused more on testing the system's efficiency against security threats. Testers may use any method for gaining access to the system with any possible impact that this could have.

During **Requirements specification**, usage / intrusion scenarios should be defined so as survivable-system testing evaluates the performance of **essential and non-essential system services** for normal use of the system and for use of the system under intrusion. These usage scenarios should be derived from usage models which are described and derived by legitimate or intrusion usage scenarios of end user. Additionally, for defining requirements of system usage focussing on essential system services, survivability requirements (resistance, recognition, recovery, adaptation, evolution) should also be considered. These may be examined during intrusion usage phases described as Penetration Phase, Exploration Phase, Exploitation Phase. These types of usage may be examined as a spiral of increase intruder authority and this theory could also be extended so as spiral method as risk driven methodology to be used in order to develop a lifecycle model based on risk increasing in each cycle. By using **statistical** usage-based testing, usage models are expressed in testing of formal grammars of Markov chains by marking the probabilities that these usage scenarios may be realized. The model then can be used to be sampled to identify a set of test case and determine a certain probability distribution. The outcome of execution of these test cases (fail / pass) can predict eventual field experience with the software.

Finally, Secure SDLC is presented, which is the closest approach to the one that the current dissertation proposes for providing a survivable system. The SSDLC model that follows is from Microsoft since it is the most representative one found through literature.

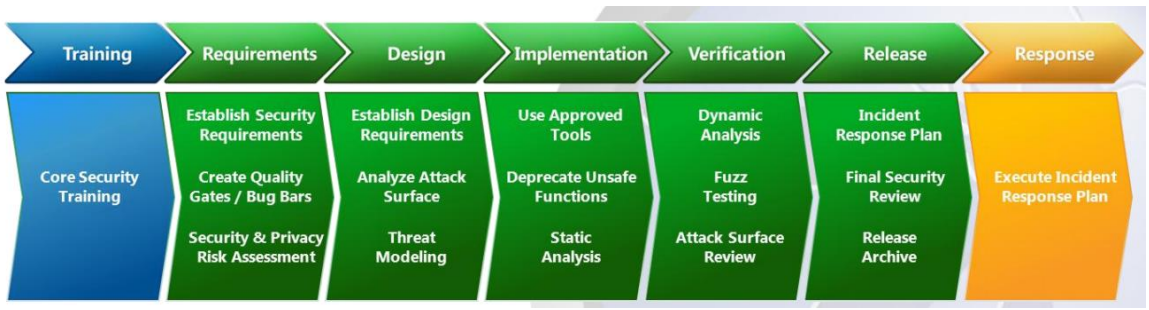


Figure 53: Secure SDLC [76]

Figure [53] above describes how security is embedded in all stages of the SDLC like requirements definition, design, development, testing and implementation phases. Additionally, training regarding security is considered of vital importance. So, during requirements specification phase, security requirements shall be established, and corresponding risk assessment should follow to mark security risks. During design phase, design requirements related to security should be considered and threat modelling should be used to identify security vulnerabilities, determine the risk and identify possible mitigations. Then the Implementation phase follows where secure coding principles should be followed and Static Analysis or Dynamic Analysis tools may be used to reveal possible vulnerabilities. During verification phase, testing of security requirements shall take place. Finally, release and response phases close the SDLC with Incident Response Plan to take place in order new threats to be reported and mitigation practices to be adopted to the next SDLC cycle.

The current dissertation will be based on life cycle for survivable systems described by [33] and the secure development lifecycle described above.

3.3.2 Testing Methodology:

Testing as a process is vital for discovering defects, preventing defects and getting measurements of the product's quality. Faults of software are divided to categories bellow:

Error: incorrect result by action of a **person**. Examples are wrong usage wrong design wrong building of software etc. Stringent timelines complex work situations and misinformation are main factors that cause an error.

Defect: **flaw** in system that causes the component to fail to perform its required function. Examples: incorrect statement, data definition.

Failure: deviation of a system from its expected delivery. A **defect** may cause a system to fail. Examples: system does not perform as expected by the end user, system executes an action which it should not, etc. Failure happens only when system is executed and may also be caused by environmental factors like climate change, pollution, magnetic fields etc.

The connection of these terms is: "Error can lead to a defect that may cause a failure."

For reducing the likelihood of an error to occur or re-occur the methodology used is Root Cause Analysis. This is the methodology used by a system to learn from previous faults.

Starting with software product lifecycle, a defect may arise at analysis, design, implementation phase or deployment phase. If the issue arises during development, then it may be considered as internal, detected during testing and corrected easily. If the issue arises at deployment phase, it is not corrected by an easy way. Therefore, customer requirements must be mapped to design as for example agile methodology supports. In other case, errors will be found when deliverable is given to customers. The most difficult time for an error to be found is during customer requirements translation. Maybe everything has been executed successfully and acceptance tests pass but the requirement fulfilled was wrong.

Testing is a procedure that should focus on:

Functional attributes: Testing ensures the product performs an expected task.

Non-functional attribute: How well or fast a task is performed. Needs metric such as time.

Verification: evaluates a product to determine if it meets the requirements.

Validation: evaluates if the product meets the needs of users that it was built for.

Software Testing Life Cycle (STLC):

Defect Lifecycle:

New: Defect is reported.
Assigned: Defect is assigned to a development team in order to be resolved.
Active: Investigation is ongoing.
Test: Fix is ready for testing.
Verified: Verification by testing.
Closed: The issue is closed.
Reopened: When defect found again. It was never fixed.
Deferred: When the defect will be available, on which future release.
Rejected: A defect may be rejected if is a duplicate one, or if it is not a defect, or if it is not reproducible.

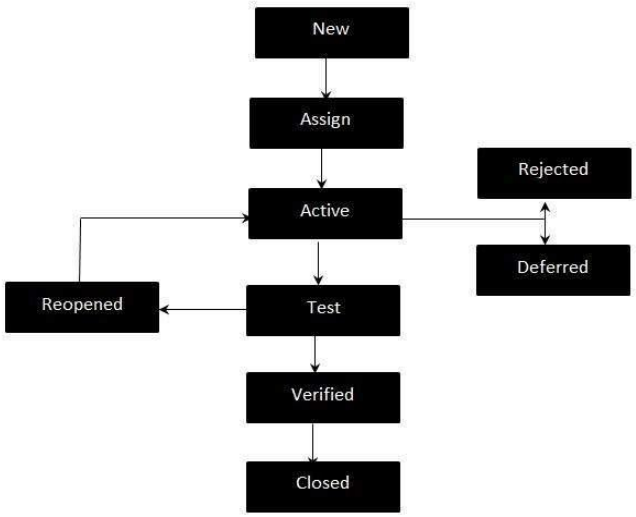


Figure 54: Defect Life Cycle ***[77]***

Test techniques

Different testing techniques and test approaches will be presented below. Some of these techniques will be used on the proposed SDLC to evaluate system survivability.

Some techniques or 'testing types' used to test the correct functionality of system are presented below. These testing types are categorized in accordance with the threats that would try evaluating if could be avoided.

Requirement's specification Failures:

- **Conformance / Compliance testing:** General term referring to testing if system functionality complies with requirements. For example, a telecommunication system like a 4G system should comply with 3GPP specifications. Furthermore, used to ensure that system is performing due to regulations.
- **Exploratory testing:** Cem Kaner, who determined the term in 1984 defines exploratory testing as "*a style of software testing that emphasizes the personal freedom and responsibility of the individual tester to continually optimize the quality of his/her work by treating test-related learning, test design, test execution, and test result interpretation as mutually supportive activities that run in parallel throughout the project*." [78] This technique is based on tester's competence which should be developed continuously.
- **Exhaustive Testing:** Test approach in which all possible data combinations are tested.
- **Scenario Testing:** Testing system against certain scenarios by various techniques. For example, by enumerating possible users and their actions, by evaluating the system

against legitimate usage, studying customer bugs and complaints by customer. Use Case testing could be considered as a testing mechanism of scenario testing, where test cases are based on usage cases and real scenarios.

Software Failures / Data Failures:

-Ad hoc testing: The tester seeks to find bugs in any way by creating tests that will run only once. These tests are per feature and are not added to a suite of tests that will be used in the future. It could be considered as an approach of error guessing or exploratory testing.
-Pair Testing / Buddy Testing: When people from development and testing work together occasionally as they have depicted some possible issue to the system. This is an out of the 'process' or development life cycle activity but happens very often mostly because specifications or requirements are not clear, the way the product functions is not known by the team, the dead line is near and there are still doubts about the quality of the product or in case that an error has been found and people solving it feel that they need to extend their research so as to find more bugs. This is a very important form of testing and it should be included to the process as it would improve the test coverage of the product.
-Boundary testing: Testing focussing on boundary values using extremes of input domain (ex. Maximum / minimum values, error values, etc.)
- **Data Integrity Testing:** Data quality, integrity and reliability are tested.
- **Data Driven Testing:** Testing based on data used from data files to test a variety of inputs to the System Under Test.
- **Anomaly testing:** Testing inputs or configurations that are out of ordinary and not as expected or as supposed by standards
-Statistical Testing: statistical methods are used to determine system's reliability or how faulty cooperating system 's can affect its operations.
- **Negative Testing**: Test that the system will not fail to an unexpected input.
- Fault-injection testing: used to test the coverage of system to fault inputs.
- Fuzz-testing: Test system by giving random data as input to the system. Types of fuzz testing inputs may be number or character fuzzing, application fuzzing, protocol fuzzing, file format fuzzing etc.
- Safety Testing: Adding system safety in SDLC so as S/W not to generate hazards and monitoring systems perform flawlessly when system needs to be supported by back-up systems. This term is very similar to survivability.

Hardware Failures:

- Destructive Testing: Testing based on making the system to fail in order to test system the robustness. System is tested until the application brakes in order to determine the service life of the product or the time that it will be available until it breaks.
- Scenario Testing may also be used to test impact from H/W failure to the system.

- **Failover Testing**: Testing of the ability of the system to allocate more resources and be able to move operations to back up System.
- **Recovery Testing:** Testing if the system is able to recover after failure in a timely manner.

Failures from Load increase:

- **Durability Testing**: Determines the characteristics of a system loaded with various load over time. These characteristics tested may be memory leaks, database resource consumption, I/O activity levels etc.
- **Performance Testing**: Testing system responsiveness, scalability, reliability, stability and resource usage under various workload. Load Testing, Stress Testing, Soak Testing and Spike testing are methodologies used during performance testing.
- **Load Testing**: Test the system under specific load.
- **Soak or Endurance or Stability Testing:** System performance under continuous expected load. It may reveal memory leaks, failure to close connections like database connections that may result to module stall or system crash, gradual degradation of response time.
- **Spike Testing**: Increasing the number of users suddenly in order to measure system performance.
- **Scalability Testing**: Test system's ability to grow by increasing the workload for users, the number of users or size of database etc. It may be measured by response time, throughput, requests per seconds, performance measurements while changing the attributes described above, CPU usage, Memory usage, Network usage etc.

Failures from overload circumstances:

- **Stress Testing:** Test the upper limit capacity of the system (load at maximum and beyond that). By stress test what may also be tested is if the system can monitor its behaviour so as to print meaning messages before and while crashing and if it has saved data needed before crashing.

Failures arise from intersystem connection:

- **Intersystem Testing:** Testing the integration points for a single application hosted at different environments.

Failures related to product installation:

- **Backward Compatibility Testing:** ensures that the product is working with new version of platform.
- **Configuration Testing:** Test the system with various configurations for S/W and H/W.
- **Compatibility testing:** Testing based on application's compatibility with the H/W that

this system will be installed to.
- Portability Testing: Test the ability of the product to move from one environment to another. Attributes of this testing are adaptability, installability, replaceability, co-existence.

General techniques:

- Vulnerability testing: Performed to evaluate the risks of system in order to reduce probability of the event.
- Disaster Recovery Plan testing: Testing step by step the disaster recovery plan or disaster recovery features in case of a telecommunication system. This testing will refer to know threats.
- Security Testing: Testing based on system's confidentiality, integrity, availability, authentication, authorization, non-repudiation. Techniques for security testing may be:

- Injection
- Broken Authentication and Session Management
- Cross-Site scripting
- Insecure Direct Object References
- Security Misconfiguration
- Sensitive Data Exposure.
- Missing function level access control.
- Cross-site request forgery
- Using components with known vulnerabilities
- Unvalidated redirects and forwards.

All of the above techniques could be part of penetration testing of web interfaces of access network devices for telecommunication systems.

Software Testing Levels that are associated with development lifecycle or testing life cycle:

Dynamic Testing: It involves testing of software for input and output values. Techniques of dynamic testing are functional and non-functional testing, and some levels of dynamic testing are Unit Testing, Integration Testing, System Testing, Acceptance Testing.

Functional Testing / Component Testing: This level of testing focuses on ensuring the correct functionality against certain predefined requirements. Most of the time it consists of test suites that include all possible outcomes of a system function. Error scenarios and **t**esting of integrated units as a group should also be considered in order to verify that system functions as expected.

Non-Functional Testing: Software Testing techniques verifying attributes of the

system like memory leaks, performance, robustness etc.

Some of these techniques are:

Baseline testing, Compatibility testing, Compliance testing, Endurance testing, Load testing, Localization testing, Internationalization testing, Performance testing, Recovery testing, Resilience testing, Security testing, Scalability testing, Stress testing, Usability testing, etc. These are described above.

Unit Testing: This level of testing is closer to development procedure since code units are tested. Functions, procedures or individual programs may be considered as units and they are tested against functionality requirements and coding issues like exceptions. It is quick as it may run right after a piece of code is changed and it is flexible in order to test all coding errors apart from logical errors.

System Integration Testing: Testing focused on units and modules integration. Functionality, Performance and Reliability between modules that are integrated are tested. Hybrid integration used to test standalone modules from interface and communication point of view may use two approaches, top down and bottom up in order to cover all possible combinations of functionality. Intra, inter system testing and pairwise testing are part of integration testing.

System Testing: Complete Integrated system as a whole is tested against requirements and quality standards.

Acceptance testing: System tested for acceptability, against certain business requirements that may be predefined by customers. After this process is completed the product is ready for release.

(User) Acceptance Testing: Test of requirements of specifications or contracts. It Is called 'acceptance' as it is mostly used by the end user to specify if system functionality satisfy the acceptance criteria. With acceptance testing system functionality is tested against several pre-defined scenarios. The testing system is very similar to end user's system or sometimes it is the actual user's system. Acceptance testing is used at Agile methodology as high-level testing defined by business customers.

Static Testing: S/W testing technique for testing without code execution. It is separated to two parts:

-Review: Used to find and fix errors and ambiguities in documents like requirements, test plans etc.

-Static Analysis: Code is analysed by tools for structural Defects. By this defects that

may be found are a variable with an undefined value, Inconsistent interface between modules and components, Variables that are declared but never used, Unreachable code (or) Dead Code, Programming standards violations, Security vulnerabilities, Syntax violations etc.

Deployment Testing: Testing project deployment to a new system or installation of upgrades of the project to existing systems is of vital importance as usually most downgrades and crashes of system happen during this procedure. This is a testing level that should take place after testing the system.

Sanity / Smoke Testing / Build Verification / Basic Acceptance Test: Basic tests executed when a new build is loaded to the system under test in order to ensure that after changes, basic functionality of system performs as expected.

Alpha / Beta testing: The terms are related to testing levels of system during any development life cycle. Alpha testing is the first testing phase of the system after development and unit testing. Starting with white-box techniques, the next step may include grey-box or black – box testing techniques in order to produce the alpha release of product. Beta testing is the next phase of testing of the system including more sophisticated testing like usability or stress testing. The result of beta testing is a release that could be available for demonstration.

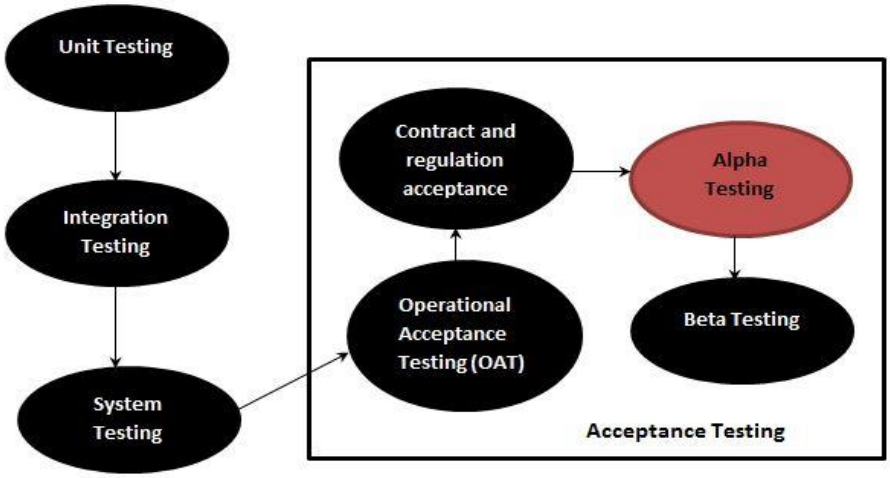


Figure 55: Defect Life Cycle [79]

Code Driven Testing: testing is the initiator of development as test cases that test the requirements specified are first design. They are executed to fail and drive the development to create code for the test to pass.

Feature Testing: Test new functionality added to the system at a particular release cycle. Firstly, requirements of feature should be understood, then test scenarios should

be extracted and documented, tests should be executed and if the outcome is successful, feature is able to be released.

Operational Testing (OAT): Testing readiness of System Under Test (SUT) to be released. Techniques that may be used for this SDLC testing part may be:

Operational Documentation Review, Code Analysis, Installation Testing, End-to-End Test Environment Operational Testing, Service Level Agreement Monitoring Test, Load & Performance Test Operation, Security Testing, Backup and Restore Testing, Fail over Testing, Recovery Testing

Methods used for Test Coverage:

Basis Path Testing: Tests are created for all possible paths of execution.

Basis Test Set: Test derived from internal structure of a component in order to achieve 100% conerage. Used before unit testing to test memory leaks, statement testing and path testing.

Branch Testing / Decision coverage: Ensures that each one of the possible branches from each decision point are executed and all possible combinations of variables that are parts of the decision should be tested as well.

Statement Coverage: Each code statement should be included to some test.

Breadth testing /Depth testing: Test suites that validate the full functionality of a system and not focusing only on the product features. The algorithm used for the top-down method in order to test new modules by this hierarchy, is breadth-first or depth first.

Cyclomatic Complexity: Source code complexity measurement connected with a number or coding errors. It is calculated by Control Flow Graph that measures the number of possible paths through a module. Lower this complexity, the vulnerability of the system becomes smaller.

Linear Code Sequence and Jump (LCSAJ) Testing: Identifies the code coverage starting from the start of the program or branch and ends at the end of the program.

Logic Coverage Testing: Verifies the subset of total number of truth assignments to the expressions. [Decisions in programs, Finite State Machines and StateChatrs, Requirements etc.]

Loop testing: Validates loops (simple, nested, concatenated, unstructured etc.)

Model-based testing: Test cases are derived from model describing functionality of SUT.

Monkey Testing: Randomly testing of the system with random data as well.

Parallel testing: Using same inputs to different versions of the product to see how it performs.

Penetration Testing: Used for security purposes, where an authorized attempt is made to violate the security or integrity of a system.

3.3.3. Process Performance Measurements:

Before closing literature review, it is important to present how development process may be measured. This will provide an indication regarding survivability and safety. A very important objective of a product or project is to be a **six-sigma project**. This is a very important indication for survivability since six sigma means that **99.99966%** of all opportunities to produce a feature will be free of defects. Since 6sigma is referring to linear data, the corresponding value for discrete data is Defect Per Million Opportunities (DPMO).

$$DPMO = \left(\frac{\text{total number of defects found in a sample}}{\text{total number of defect opportunities in the sample}}\right) \times 1{,}000{,}000$$

Figure 56: DPMO Value [80]

For achieving a six-sigma level 6, the corresponding DPMO value, as it is shown in the figure bellow is 3.4 defects per million opportunities. This is the target value that could be reached for a survivable system.

Sigma Level	DPMO
6	3.4
5	230
4	6,200
3	67,000
2	310,000
1	700,000

Figure 57: Sigma Level and corresponding DPMO Value [81]

4. Proposed Methodology

During this chapter, a software development lifecycle for survivable telecommunication systems will be described. Before that, two approaches that could enrich survivability requirements already defined by 3GPP, will be presented. The first one is related to recognition of failure, and the second one is related to resistance to failure. The first one is part of paper with title "Self-Diagnosis Framework for Mobile Network Services" published in JACN Journal [82], and the second one is from a paper presented in ICICM 2019 Conference with title "Fault Prediction Model for Node Selection Function of Mobile Networks" [55].

4.1. Approaches beyond 3GPP to enhance survivability.

As it has already been described by research of Richard C. Linger et al. [33], the three main requirements of survivability are recognition, resistance and recovery of failure. Additionally, an extended research on all related measurements described by 3GPP which is the official standard for mobile telecommunication systems, has been presented through literature review. What may be observed is that even though there are a lot of recovery and resistance mechanisms described by 3GPP, when it comes to recognition of failure, the standard gives a lot of detailed instructions on how data should be gathered and reported. Though it does not indicate what to report when a failure happens and what actions should follow a failure, leaving it to the constructing organization to specify. During the current dissertation two related approaches will be presented.

4.1.1. Recognition of failure.

Root Cause Analysis.

The framework that is proposed below, is a service – centric approach that automatically performs a detailed root cause analysis of failure, contrary to traditional monitoring and self-diagnosis sub-systems that are focused on general system performance indicators like failure counters, or general alarms. The analysis proposed to be provided is very detailed, reaching message information elements (IE) level, comparing to traditional Key Performance Indicators (KPIs) that are just connecting the failure with a cause which most of the times is misguiding for the root cause of failure to be investigated. Besides that, the root cause analysis of service failure is performed as soon as the failure happens. In this way, if an operator or a developer or a tester of responsible vendor has this information, the time needed for the analysis of the problem will be minimized. Additionally, the framework proposed is based on existing

monitoring methodology described by 3GPP standard for telecommunication management (3GPP [61], 3GPP [83]), and fault management (3GPP [62]), to provide a self-diagnosis framework in node level. Service monitoring tasks will be performed at each node and any failures accompanied with a possible root cause, based on root cause analysis, will be reported to the network management entity (NM) to be used for other processes of SON, like self-configuration or self-healing, or to be handled manually.

The current research is a result of a few years of working experience in testing and development of 4G networks and of bug fixing for 4G networks.

The issue that needs to be addressed is the misleading information that is provided by monitoring systems in form of Key Performance Indicators (KPIs) related to fault causes that are very difficult to localize and understand. For example, a GTPV2 protocol failure cause is "No resources available". Though, someone who manually investigates this error or an automate self-configuration or self-healing system, needs more information in order to process the failure, like to which node there are no resources. Another example is the cause "System failure". This is the most misleading one since no real cause of failure is provided. We have absolutely no information about what went wrong. Though, there are other causes that are very clear but again we cannot localize which node has failed. For example, the cause "Mandatory IE incorrect". We understand that there was something wrong with one or more Information elements of the received GTP message. Though is really the sender node that has introduced the fault? If this message was sent to another node would also be rejected? These are questions that may be used accompanied with gathered failure KPIs to automatically localize and analyse the failure so that to be fixed manually or automatically.

As it has already been presented, services of mobile networks consist of many messages forming a message flow between network nodes.

The framework that is presented is focusing on node-level diagnosis (NE level) so that the root cause of a service failure to be depicted. This means that the scope is the service to be monitored at all levels and connections between nodes, message by message in order to reach the final goal of fault analysis as described by 3GPP "Fault Management" standard [62] which is: "to minimize the effects of failures on the QoS as perceived by the network users it is necessary to detect failures in the network as soon as they occur and alert the operating personnel as fast as possible". Here we could add: "and with as much detailed root cause analysis as possible."

So, in case of a service failure, the topology of the network nodes for which management system may extract some diagnostic evidence, may be described by figure bellow. A "sender" node (for example MME) is sending a message (for example Create Session Request) and the "receiver" node (for example SGW) receives and processes it to continue with communication with another node (for example PGW), or

to just answer back to the "sender" node. Any node is responsible for monitoring the message that it sends and report any failure to the whole message transaction, which is part of a larger service flow.

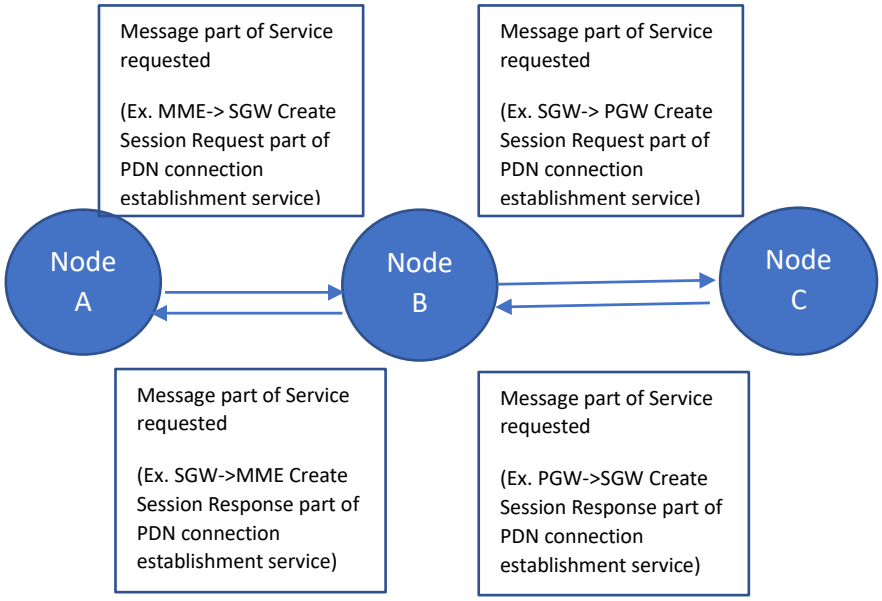


Figure 58: Topology of NE fault management system

The outcome of sending a message, may be summed up by three cases, two of which may be considered as service threats since they will cause degradation of QoS service under the acceptable limits, no matter what the root cause of failure is. These 3 cases are: the message is successfully answered; the message is never answered; or the message is rejected. Usually, most of monitoring systems reach this level of analysis using KPIs of attempt, success or reject of a message, in form of counters. Sometimes, rejection metrics are accompanied with a possible cause of rejection provided by the rejection message. The current paper tries to formulate a more sophisticated way of finding the root cause that may be used additionally to the current network management system established to operator's environment. The diagnosis framework proposed will use a top-down level approach for reaching the root cause, following or forming a tree structure (figure 3.1.1.4). Diagnosis of the root cause will be based on several questions that a developer or a tester would make in order to find the root cause of a failure and may be automated for taking self-configuration decisions by the SON. These questions or levels of root cause analysis are:

1) **First level** of root cause analysis has to do with general metrics regarding the failing node or the overall system. Are there any other failures of this kind happening at the same time? For example, when the system is loaded, all services failure indicators tend to increase since the system is not providing the QoS expected. Or is there any alarm raised when the failure occurred? These questions may give us information regarding

the overall system status. For example, by this we may know that a node connected to the examined sender node is under restoration and an increase to a service failure is expected. In case this is true then there is no need to send any alarm of service failure. This is the reason why this level of analysis is first, in order to avoid increased load of alarms. Finally, is there any other service or services that their failing KPI may indicate the failure of the service examined? For this level analysis, regression analysis may be used.

2) **Second Level**: Is this failure related to a rejection message or the message sent was never answered and what is the frequency of this failure? So, the case that a message is never answered or is rejected, may be valid occasionally, under certain circumstances, or may happen any time the service is requested. This information is very important for root cause analysis. As a result, the categorization of threats to a service that are proposed and examined through current paper may be listed below:

a) Total Denial of Service causing service failure. Message is not answered. This may be permanent in cases of S/W failure or for some time in case of Overload or H/W failure until the failure is restored.
b) Permanent Service Rejection. Message signalling between a pair of nodes always results to a failure by receiver node rejecting messages.
c) Occasional Failure causing degradation of service QoS. This means that failure is not permanent, but it leads to unacceptable levels of service QoS.

3) **Third Level**: Is this failure happening with all similar connected nodes? Third Level of root cause analysis is related to locating the node that is responsible for failure and it is based on statistics gathered for any nodes that the examined sender node is connected to. For example, an MME may select certain SGWs to serve different UEs, based on topological closeness of the UEs and SGW, or based on SGWs load state as 3GPP [84] DNS procedure preserves. If the failure is happening always with one SGW then the diagnostic system may propose that this SGW is the most likely node to be the root cause of the failure. If the failure is happening with all SGWs, then the sender node is the most likely to be the responsible node for the failure.

4) **Fourth Level**: What seems to be the probable cause of failure due to "3GPP Fault Management Standard"? For the fifth Level of root cause analysis what will be used is the classification of fault management standard by 3GPP. The "faults", as they are described by the "Fault Management" standard of 3GPP [61] are grouped into one of the following categories:

1. "*Hardware failures, i.e. the malfunction of some physical resource within a NE.*
2. *Software problems, e.g. software bugs, database inconsistencies.*
3. *Functional faults, i.e. a failure of some functional resource in a NE and no hardware component can be found responsible for the problem*
4. *Loss of some or all of the NE's specified capability due to overload situations.*

5. Communication failures between two NEs, or between NE and OS, or between two OSs." [61]

The service impacts related to these failures, and the fault analysis that could follow may be depicted by figure 59 below. What should be mentioned here is that there are already many standard procedures that deal with such failures documented by 3GPP. For example, 3GPP [57] which is describing the restoration procedures in case of node failure or the 3GPP [59] that handles overload mechanisms. Though, there are cases that these resistance and recovery mechanisms are not performed since not all organizations have implemented them, or there is a failure in their implementation or other root cause has led to these faults. An example of the last case is overload of a node because of CPU or Memory load as result of hanging resources, and not because of service requests load. Finally, even if they exist, they are not designed to predict the failure. They only indicate it after it has been realized. This root cause analysis may go on until the level of detail the fault management system is designed to be handled.

The proposal is the Network Element (NE - the sender node), to report these levels of root cause analysis in an XML-form and send the report to the Network Management (NM) entity through ltf-N interface in order further actions to be applied. XML form is chosen since nowadays is used by most programs and programming languages. The architecture of the fault management system is described in 3GPP [62] and may be observed in figure below:

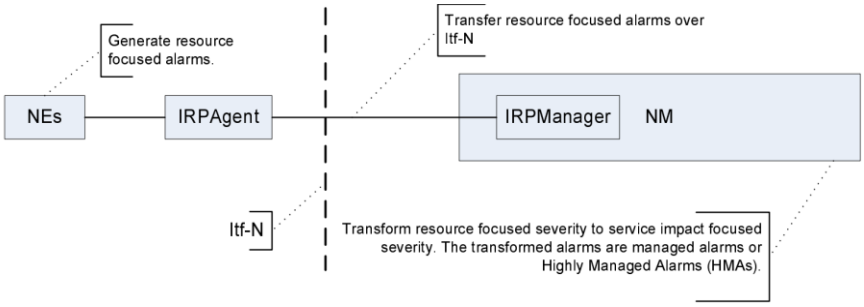


Figure 59: Fault Management System Architecture [62].

The reference standard for XML schema alarms is 3GPP [85]. For any of the causes described under existing tag "ProbableCause" the current solution proposes a more detailed root cause analysis to be provided by an alarm that is already triggered or by a new alarm produced for the purposes of the current framework. This XML structure is presented in below.

```
<element ref="xai:ServiceFailureRCA"/> <!--NEW element proposed by current document additionally to
                                probableCause below-->
<element ref="xai:probableCause"/><!--Fourth Level of RCA already exists in 3GPP [61]-->
                                        <!--Some Examples as defined at 3GPP [61] are:
                                                Loss Of Synchronization, Out Of Memory, Software Error, Link
```

```
                                        Failure, Delayed Information, Denial Of Service, Message Not
                                        Expected etc-->
  <complexType name="ServiceFailureRCA">
      <sequence>
          <element ref="xai:MessageFingerprint"/><!-- ServiceRequested -->
          <element ref="xai:ServiceFailureKPI"/><!--KPI of Service Failure -->
          <!--First Level of RCA:
          Start of list of KPIs
          that have been proved to be
          statistical important for
          calculating the probability of
          service failure.-->
          <element ref="xai:StatisticalRelatedKPI"/>
          ...
          <element ref="xai:StatistcalRelatedKPI"/>
          <!--END of list-->
          <element ref="xai:TypeOfFailure"/><!--Second Level of RCA -->
          <element ref="xai:FailedNode"/><!--Third Level of RCA -->
      </sequence>
  </complexType>
  <complexType name="ServiceFailureKPI">
      <sequence>
          <element name="ServiceFailureKPIName" type="string"/>
          <element name="TimesOfFailure" type="integer"/>
          <element name="ServiceNumberOfAttempts" type="integer"/>
          <element name="RejectionCause" type="string"/>
      </sequence>
  </complexType>
  <complexType name="StatistcalRelatedKPI">
      <sequence>
          <element name="RelatedKPI" type="string"/>
          <element name="ProbabilityOfServiceFailure" type="float"/>
      </sequence>
  </complexType>
  <simpleType name="TypeOfFailure">
      <restriction base="Integer"><!-- Second Level of RCA -->
          <enumeration value="0"/> <!-- 0: request never answered -->
          <enumeration value="1"/> <!-- 1: request always rejected -->
          <enumeration value="2"/> <!-- 2: request rejected occasionally but exceeded acceptable levels of
      failure-->
      </restriction>
  </simpleType>
  <complexType name="FailedNode"><!--Third Level of RCA -->
      <sequence>
          <element name="FailedNodeIP" type="string"/><!-- IP of sender/receiver node -->
      </sequence>
  </complexType>
  <complexType name="MessageFingerprint">
      <sequence>
          <element name="ReceiverNode" type="String"/><!-- Receiver's IP -->
          <element name="MessageId" type="integer"/><!-- message sent (ex. CSR id=0) -->
          <element name="Fingerprint" type="String"/><!-- values of message IEs that represent this type
          of message-->
      </sequence>
</complexType>
```

Source Code 1: Fault Management System Architecture

Self-diagnosis framework:

Any failure should be somehow related with the message sent in order the root cause analysis framework to provide more accurate results. Furthermore, it is very important to indicate which is the exact message sent since there are many messages of the same category. For example, there are many types of Create Session Requests like emergency request or IOT request depending on the values of the Indication flags of the CSR message. By using any message IEs, a fingerprint of the message containing those IEs that characterize the type of the message should be kept. The current research has chosen to use a search tree to represent messages sent by each node to the receiver node. The search tree will be kept to the sender node to extract conclusions regarding the failure of the message sent. This search tree will have as parent node the sender node. The next level will be the nodes that this sender node is connected to and then the messages sent to these nodes figure 2. Each message could be extended to the IEs that it includes. Of course, we cannot use all IEs in our example because we would end up with a huge tree. But every combination that provides a failure should be kept in this search and reported in "messageFingerprint" field by the XML report file presented in "Source Code 1".

	Hardware Failure	Software Problems			Functional Faults	Overload/Increased Load	Communication failures
		Sending Node	Receiving Node	Database Inconsistencies		Request Load	
SYMPTOM: Total Denial of Service	Node B H/W has failed, resources like CPU or memory are extremely loaded (other cause than increased requested load ex. hanging resources).	Malicious message is sent from node A and ignored to node B	Node B S/W bug result to wrong processing and ignorance of message sent by node A		Configuration issue of node A or B: ex. wrong IP addresses are configured.	Node B is overloaded by message requests.	Path between node A and node B has failed. (Path may be a physical path or a whole network)
Fault Analysis 5	Increased DOS failure indication - all messages sent to a certain node are not answered.	When sender node is sending this message to all nodes that is connected to and service is always failing	When node is sending this message to one node or a family of nodes that is connected to, and service is always failing.		Increased DOS failure indication - all messages sent to a certain node are not answered.	Increased DOS failure indication - all messages sent to a certain node are not answered.	Increased DOS failure indication - all messages sent to a certain node are not answered.
SYMPTOM: Permanen	Resources of receiver node or any other node	1 Malicious message is sent from node A to	Node B S/W bug result to wrong processing of message received by node A which may result to malicious message sent to	Corrupted data may cause failure to	4 Configuration issue of node A or	Node B is overloaded by message requests and services are	

t Failure by Rejection message	connected to the receiver are unavailable	node B causing service failure	another node or sent back as an answer to node A causing service rejection Lack of robust mechanism to ignore malicious components of message in case it is possible.	service request. (3GPP 23.007 provides the recovery from this failure)	B: service is not supported by node B	rejected by overload mechanism [3GPP 29.807].	
Fault Analysis	Usually certain rejection causes (ex. Resources Unavailable) will indicate this.	When sender node is sending this message to **all nodes** that is connected to and service is always failing. Certain cause may also indicate this ex. mandatory IE incorrect.	When node is sending this message to **one node or a family of nodes** that is connected to and service is always rejected. 2	Usually certain rejection causes will indicate this failure (ex. Unknown UE) no need for extra root cause analysis.	Usually certain rejection causes will indicate this failure (ex. service is not supported).	Certain rejection causes will indicate this failure (ex. Overload indication in answering messages).	
3 **Occasional Failure by Rejection Message**	CPU / Memory load causing message processing delays		Lack of robust mechanism to ignore re-sending messages Lack of robust mechanism to handle collision scenarios			Increased Load causing message processing delays	Synchronization inconsistencies in communication of the two nodes causing message delays, message re-sending or collision scenarios.
Fault Analysis	Occasional rejection of same messages.		Occasional rejection of same messages.			Overall messages rejection KPIs increase.	Occasional rejection of same messages.

Table 3: Root cause analysis for 5th level of analysis.

In our example of figure 60, message Create Session Request (CSR) sent by a MME to a SGW, contains several information elements that indicate different types of service. Of course, IEs like IMSI or MSISDN are changing depending on the UE that requests the service. So, they are not good indicators for categorizing the different CSR messages.

Though, if IEs like indication flags, RAT type, APN etc are used, the categorization of the request is easiest. For example, the system may conclude that any time a CSR with indication flag "Control Plane Only PDN Connection Indication" for a certain APN is

requested the SGW replies with a rejection message instead of just counting the number of Create Session Responses with general rejection cause.

Some examples of root cause analysis that may be extracted from the tree of figure 60 may be listed below:

- For the example (1) what may be concluded is that any time the message Create Session Request with RAT Type = 0000 0000 and indication flag 0000 0101 0011 0001 for the APN = 0000 0001 is sent from MME to any SGW (SGW1, SGW2), the service fails. So most probably, the failure is on MME (or UE) side. Number (1) is also marked on table of figure 8 with service possible root cause analysis.
- Example (2) shows a case that no matter the value of Indication IE that follows, if the RAT Type IE has value 0001 0010, then the service is failing. Though, this is not true with all SGWs that MME is connected to. So, it is safe to conclude that SGW1 seems to be the node that causes the failure.
- Example (3) is a representation of synchronization issue of node SGW2 in certain CSR messages. We may also see that failure / attempt ratio is less than 1 which indicates that this is a random failure and not a failure that happens every time.
- Example (4): Possible miss-configuration of SGW3. The cause of failure "no resources available" indicated that.
- Example (5): Permanent DOS failure indicating a H/W failure or overload or communication error of SGW3 etc. If the error is not yet reported, the NE should do so.

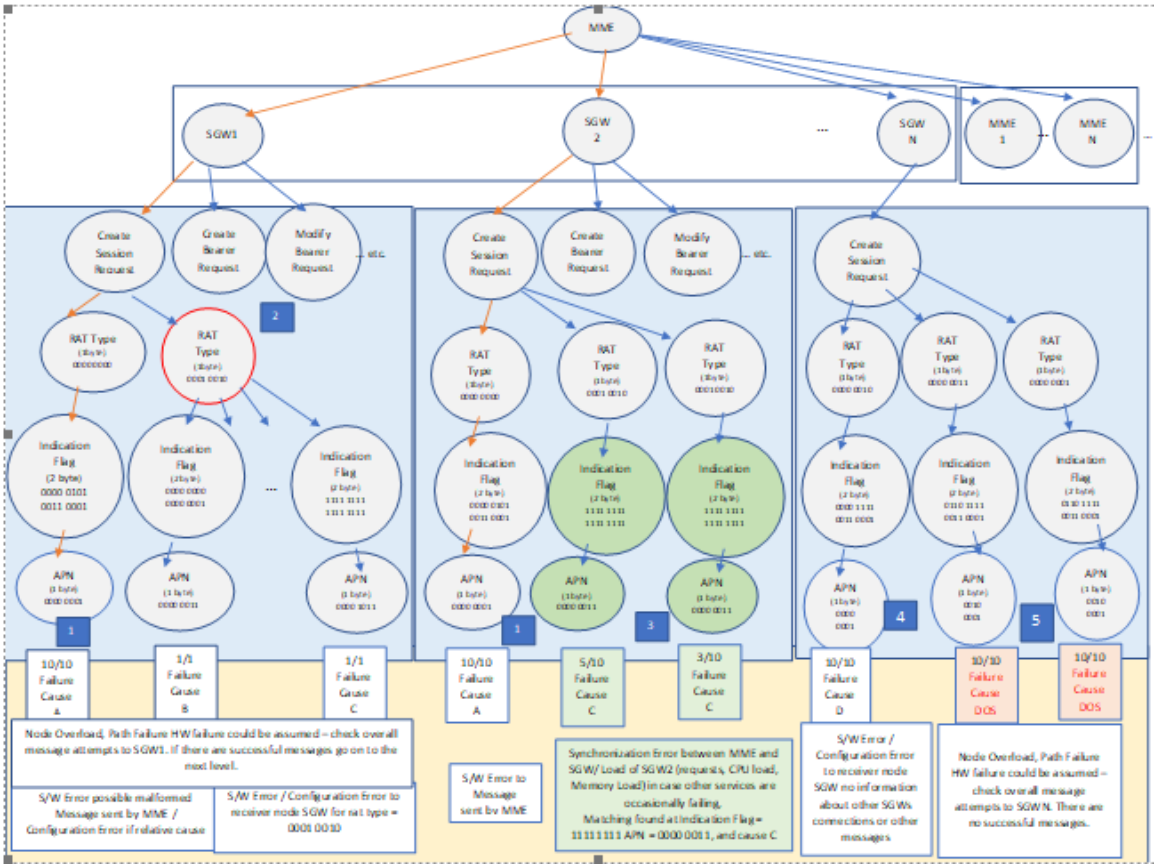


Figure 6o: Search Tree of MME node.

4.1.2. Resistance to failure.

Another source of variation for telecommunication systems, is the different recognition, resistance, recovery, security or business continuity measures that each mobile network system operator employs. It is therefore important for network planning not to care about what these measures are, but what is the actual survivability of the critical services when they are applied to network nodes. Current research supports that this may be measured through the probability of critical services' failure and by avoiding nodes with increased probability of failure. Apart from the different appliance of these measures, the state of the node, or the system, each time, is of vital importance when considering critical services failure.

The focus of the proposed methodology, contrary to methodologies found through literature review that focus on availability, is not on systems failure but on services failure, which is part of the definition of survivability. It is worth mentioning here that a service for a mobile telecommunication system, is a whole flow of messages between network nodes pieced together. An example of such a service is the Attach procedure of 4G networks, requested by the UE by sending Attach request, and for which

messages between eNB, MME, SGW, PGW, HSS, PCRF, etc. nodes should be sent until the UE successfully registers to the 4G network. As "critical" could be considered any service that an operator wants to be protected from failure, additionally to traditional network management techniques provided by standards for developing a mobile telecommunication network.

For providing a survivable system to the end users, the nodes that are probable to cause less failures to critical services should be chosen. These nodes are not always the same for all services or for all users. There are many factors that affect this choice and should be considered if the architecture of the system supports self-organization services. As a result, no matter what the system architecture is, what is the system's failure level or what survivability, security or business continuity measures are provided, the current research focuses on investigating the ways of providing the **"always best" nodes** to the end user, for the service being requested, based on failure probability. This is opposing to the selection criteria 3GPP DNS standard [84] describes, which are the shortest distance, or less load. If each time the "always best" node is selected from a pool of available neighbouring nodes returned by DNS query, to perform its part of the critical service, then the end-to-end system will be the "always best" in serving the critical service requested. After all, choosing the nearest node or the node with less load is a measure for decreasing the possibility of failure. So, it is safe to assume that failure possibility, as an additional selection indicator, includes these two attributes and will have many potentials on providing a more reliable system.

Why same systems differ.

Before presenting survivability and selection model, what should be explained is why same systems may differ from each other. In other words, what makes same nodes perform differently even if they were developed under the same standards. Firstly, same network nodes that an operator uses, may be manufactured by different companies. Operators tend to buy same nodes from different companies for cost issues, or in the context of their business continuity plans. Even though the same nodes should be developed by any manufacturer confronting to 3GPP standards, their actual implementation may differ causing variation of their behaviour. Especially when it comes to "implementation specific" requirements as they are described by 3GPP, which are not clearly defined, like collision scenarios handling, synchronization issues handling, survivability (recognition, resistance, and recovery - 3Rs), security, robustness, fault tolerance, reliability or business continuity mechanisms implemented. Apart from different implementations of the same node, there is a difference in behaviour even between replicants of the same product from the same manufacturer. These differences may be the result of different software versioning, different configuration, or different state of performance. For example, a loaded system reacts differently from a system working under a low load. It may for example add a delay to the network which results to critical service failure. So, it is safe to conclude that

choosing a node that minimizes the possibilities of service failure is a better criterion in order to handle this complexity than choosing the topologically closest node.

Fault Monitoring

Threats to survivability of mobile telecommunication systems, that the upcoming framework tries to cope with, have been described in chapter (3.2.2). There the probability of failure for each threat is also described. Threats to any communication between two nodes which are part of a mobile telecommunication system, and the corresponding probability of failure, may be summarized by the figure below:

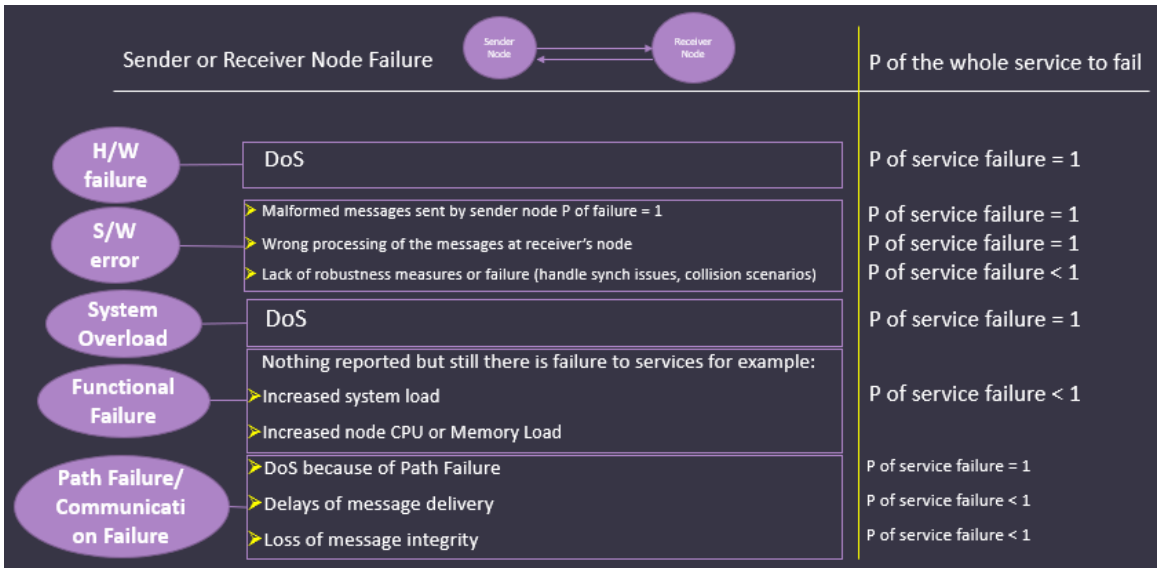


Figure 61: Threats to mobile services survivability

Evaluation of the probability of failure involves real time monitoring of the system after being released to production, against various threats to critical services. Monitoring model proposed will be based on already implemented monitoring measurements of the system which usually are Key Performance Indicators (KPIs), in form of counters, that are measuring critical services failures (ex. An Emergency Attach Failure Counter may measure the times that the service has failed in node MME). These measurements will facilitate the estimation of the probability of a critical service to fail if a node is chosen at each time interval. The need for new failure KPIs could follow each time a new threat arises, to enrich the prediction system.

Starting with actual failures of critical services, what should be measured and monitored in node level, no matter the root cause, is the node's performance against defects which lead to failures. The most common and useful measurement that shows how a certain system performs over time, is the 6Sigma value for continuous data or the DPMO (Defects per million opportunities) value for discrete data as the number of service failures. If the system performs near the 6Sigma value, then the process

capability is within user acceptance limits and it produces near to zero defects [86]. Of course, this, among others, is a measurement that will have already been used by the manufacturer of the product before releasing it to the operator. However, as we already presented, network nodes with real traffic, tend to deviate from laboratory results, since the system is exposed to several variations. The DPMO value may be calculated through the following formula:

DPMO of critical service = (Total Number of Defects of Critical Service / Total Number of Attempts of Critical Service) x 1,000,000.

Results for the monitoring of critical service will be gathered in each node, per attempt until 1,000,000 attempts are reached. The "sender" node may keep track of failures, for each node with which it is connected, in sliding windows of 1,000,000 attempts (staring at point i and finishing at point i+1,000,000). **The node with a DPMO value larger than 3 should be avoided, even if the value of 1,000,000 attempts has not been reached yet, since when it is reached, it will exceed the corresponding 6Sigma value which is 3 failures per million opportunities [9].** This is the **first mode of operation (1)**. The **second mode of operation (2)** is not to exclude the node from been selected but continue with test data until the failure is restored (until we have 1,000,000 successful attempts with failures lower than 3). If the 1,000,000 attempts are reached, then the sliding window should be shifted by one (starting at i+1) in order statistics for next 1,000,000 attempts to be gathered.

Let us assume that the sender node chooses for a critical service a certain node from an available pool of receivers. The calculation of the DPMO value starts from the first attempt of the critical service request for that node. Let n be the number of attempts, f the number of failures and i the starting index of the sliding window (i=0 for the first attempt). Then the pseudo-algorithm that could be followed may be described below:

Function Monitoring DPMO value:

- Sender node, requests for a critical service from the receiver node. This may be a real traffic request or a test request depending on what mode the algorithm is being executed.
- A pool of receiver nodes is returned by DNS query for the sender node to choose.
- Increase the number of critical service attempts (n) by 1 (n=n+1).

IF a critical service fails
- Count failures of current window (f) (a table may be kept with positions that failed, and failures could be counted from i to n).
- Increase the number of failures (f = f+1)
- Optionally **Call Execute Statistical Analysis**. (may be found bellow)

END
IF the number of failures (f = 3) (6Sigma DPMO value)
- Mark the node as "**failed**".
- Send a report to node monitoring entity for investigation of the root cause and fault fixing.

ELSE IF (f < 3) and node was already marked as "failed"*

- Mark the node as "**active**".
- Send a report to node monitoring entity for failure to be fixed.

END

IF (n >= 1,000,000) **

- Shift window of 1,000,000 to point (i+1), where i is the starting position of the previous window of 1,000,000 attempts.

END

* See figure 63 (window from i= 15 to i=26), node was marked as failed but is not anymore.

** This is performed in order not to shift statistical window if 1,000,000 attempts have not yet been reached – used for the first 1,000,000 attempts.

Let's see an example of monitoring DPMO value. Instead of a window of 1,000,000 attempts we will use 10. The result is depicted by figures 62,63 below. What may be observed is the statistical window of 10 attempts from position i=3 to 12 marked with black box.

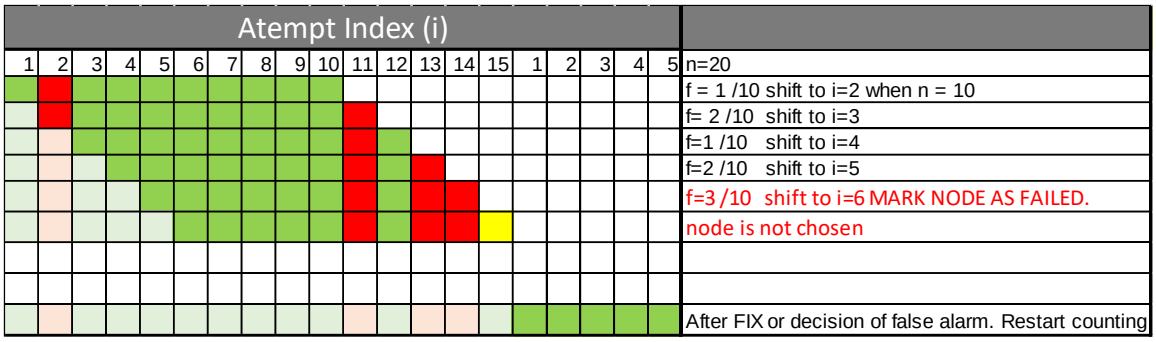


Figure 62. Example of monitoring system with n=10 and 1st mode of operation.

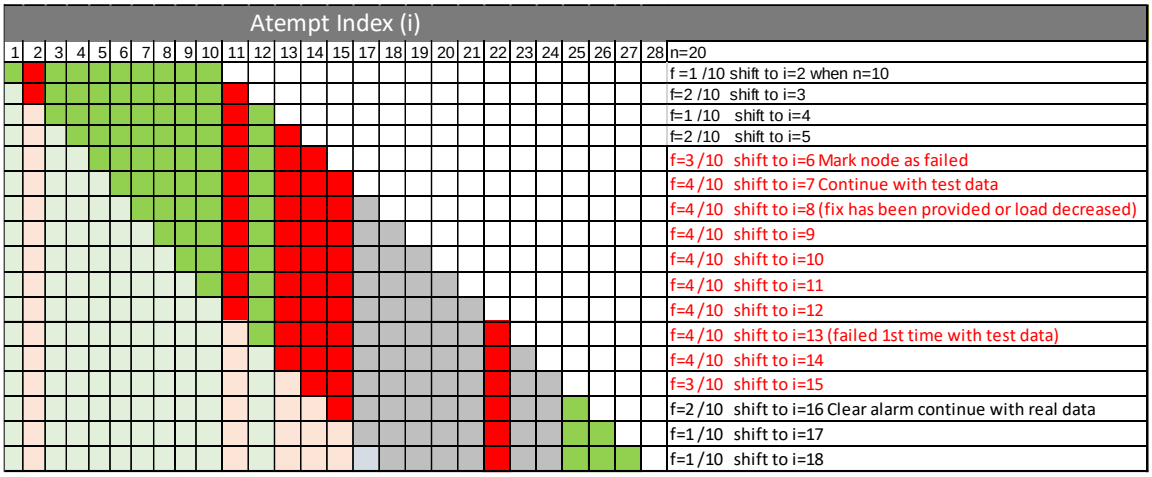


Figure 63. Example of monitoring system with n=10 and 2nd mode of operation.

What is worth mentioning to the examples above is that in 1st case the execution will stop when 3 failures are found, until problem is fixed, while in the 2nd case, the

execution will continue with test data (marked with grey colour starting from i = 17) and failure may be cleared on its own (for example when i = 26) if node state changes or node is self-healed. In this case we exclude the node, but we give it a chance to self-heal. Finally, other variations of this model may be considered in order for the system to react. For instance, after 2 consecutive failures the node may be marked as "failed" without waiting for the 3rd. One failure in any case should not be consider as a threat, until the next comes. So, the distance between failures could also be considered as a factor for excluding a node from available pool. A first restriction of this model is that sometimes, critical service failures are expected to happen, and it is not a matter of system's fault. These cases are false alarms for the fault management system proposed and the node may stop considering them as criterion for choosing the next node.

Root Cause Analysis

The monitoring of KPIs is proposed to be measured by each node and root cause analysis may follow in order the failure to be fixed. Though, results of root cause analysis may be forwarded to central monitoring system or telecommunication management system as described by 3GPP [61] in order a more centralized solution to be applied to the whole system. Then analysis of all data gathered may be also processed, for overall results on nodes QoS. Then configuration management actions may follow.

Apart from counting the actual failures of critical services, there are many other KPIs related to other services, that may indicate the root cause of the failure of a critical service. To see if these KPIs are statistically important in order to extract conclusions on critical service failures, it is proposed by the current research to use regression analysis of predictive analytics. Then selection function may act proactively and nodes with those KPIs indicating increased probability of failure, to be avoided. For example, there may be a statistical connection between a Detach Failure KPI and Attach Failure KPI. This may mean that no matter what the root cause is, each time detach fails with a certain cause, attach fails too. Then we may extract conclusions about the possibility of a new attach request to fail, by checking Detach Failure KPI.

There are many proposals in the literature for using data analytics or predictive analytics in telecommunication systems. Most of them focus on providing network management solutions, resource allocation solutions, offload mechanisms, customer experience management, etc. [87, 88]. Additionally, self-optimization and self-adaptation proposals are based on big data analytics [88]. Though, current research is not focusing on providing overall system centric solutions for fault avoidance or load management. It focusses on proposing a model that by taking into consideration metrics from each node, it can provide a reliable and robust solution any time a critical service is requested. No matter what the security, reliability, business continuity etc. measures are applied to the system. This is one of the main principles of survivability. Critical services must survive even if the system is under attack or failure.

KPIs usually are gathered per second, a time interval that is suitable for a mobile network that performs thousands of transactions per second, or whenever a new failure arises in order to investigate its root cause. If the time interval is bigger, then the monitoring system may lose critical to failure events (ex. a certain system load) and may not have the proper time to react.

The first type of regression that will be used is the linear regression on each independent variable X (or 'predictor') according to the following formula:

$$y = a + b*x$$

Where a and b may be calculated according to the gathered values in order to give a straight line that best fits the samples gathered. The existence of connection between failure to critical services, represented by the dependent variable Y (or 'criterion variable'), and any other KPI, represented by the independent variable X (or predictor), should be proved to be statistically important, by using correlation coefficient R. Then, a threshold of any other KPI (X) to a node that indicate a critical service failure, could be depicted. This threshold can be used to avoid nodes that suffer from failures that reach this threshold. For the calculation of the correlation coefficient the following formula can be used:

$$r = \frac{\sum(x - \bar{x})(y - \bar{y})}{(n-1)s_x s_y}$$

Where Sx and Sy represent the sample standard deviations of the x and y data values, respectively. The correlation coefficient takes values between (-1,1). If R is near to 0 then, there is no relationship between the variables. If it is near 1 there's a perfect positive relationship. Finally, if R is near -1 there is a perfect negative relationship. If there is a positive relationship, the two variables tend to move in the same direction and the opposite is true if there is negative correlation. An example is value of R to be equal to 1 which means that each time a certain KPI (X) is increased, because of a certain service failure, the critical service KPI (Y) also fails. This gives us a probability of failure of a critical service equal to 1 and, if possible, the node should be avoided.

Let's consider this through the following example (figure 3). Let's assume that the x-axis represents a certain failure given by KPI1, and y-axis the number of failures of a critical service. We proved by DPMO value that the least accepted number of failures of a critical service from cascade effects for a certain node, is lower than 3. It would then be safe to assume that if X reaches the value of 3, then the node should not be chosen for critical service Y.

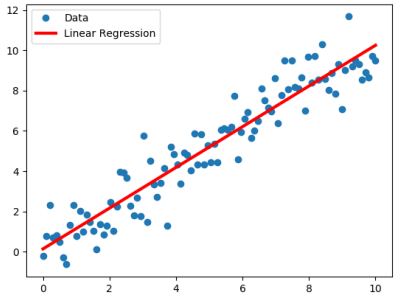


Figure 64. Example of Linear Regression diagram. [89]

A second restriction of the model proposed is that DPMO rule and single regression analysis cannot act at the same time since by DPMO rule the service attempts will not be more than 3. This is not enough attempts for safe conclusions on statistical relationship to be extracted. Though, in cases root cause analysis of failure is needed, regression analysis may provide very promising results. So maybe more than 3 failures could be allowed for some time in order root cause of cascade failure to be reveal, or after 3 failures the system may continue with testing messages in order for root cause to be depicted, or to give the node a chance to heal itself. Using test data is a common practice that fault management 3GPP [62] proposes to use for checking the fix provided for a certain fault, and current paper proposes that test data could also be used for root cause analysis or for ensuring that a failure has been self-fixed.

What is important to stress before closing this example, is that if this cascade failure is corrected, then this correlation may not exist anymore, and R will be calculated near to 0. Regression analysis will show this fix next time sample with same coverage is gathered and this threshold should not be considered anymore as a rule for prediction of failure. In this way the provided monitoring system is dynamic and adapts to any changes in system topology or to any side effects from S/W upgrades, load changes etc. Evidently, some failures cannot be prevented in order the system to gather sufficient metrics for regression analysis, and this is the third restriction of the model proposed.

Apart from cascade effects that are not clearly correlated to critical service failures, regression with multiple independent variables can be used to depict the lowest threshold of combination of failure KPIs that historically has been proved to indicate critical services' failure increase. Usually, the root cause of critical services failures in these cases are related to system's state and probability of failure is lower than one. For example, critical service or any service failures may have been caused because the system is under resistance phase, due to big load (but not yet overload), CPU increase, memory increase etc. In other words, is about to fail but failure thresholds have not yet been reached for restoration mechanisms to react. If many critical services are requested at that time and failures exceed the acceptable limits, then the DPMO value will immediately exhibit 3, and the certain node will be avoided. Though, there may not be enough attempts for the DPMO to be calculated or the system may just need to act proactively without waiting for 3 failures since the possibility of critical service failure is increased. Usually, in load situations, the most frequently used procedures are those

which may be used to get an idea of the possibility that a critical service may fail as their failure KPIs are the first to increase.

The formula of multiple regression of predictor variables X1, X2, …, Xk and a response Y, can be written as:

$$Yi = B0 + B1Xi1 + \cdots + BkXik + ei, i = 1...n$$

Null hypothesis should be used, to calculate P values, and to prove the contribution of each variable to the model each time. Usually any P-value < 0.05 for a certain KPI (Xi) would mean that its contribution is statistically important. If there is no contribution of a KPI, then it should be discarded from the model. This should be checked every second or less in order for the monitoring system to be able to track any new KPIs that should be considered as contributors or discard any KPIs that are not contributing any more. Additionally, it could be checked any time a new failure arises. Any increase to critical service failure (Y) should be marked as suspicious of failure and if it is discovered at next time critical service is requested, the node should be avoided.

To sum up, any statistically important connection between a service failure KPI, and critical service failure KPI, should be reported to the node monitoring system in order to be handled. The same analysis may be applied through multiple regression since sometimes more than one failure indicators may predict an increase of critical service's failure. This statistical analysis may be performed periodically, or when a failure appears as in the algorithm before. For performing this statistical analysis, and reporting it to the node monitoring system, an array with a critical service failure KPI (Y) and the list of other KPIs (X1, X2,…,Xn) should be kept [KPIY, [KPIX1, KPIX2…KPIXN]] each time the critical service fails.

Function Execute Statistical Analysis:

- Calculate Single Regression function of KPI_Y with any "increased" failure KPI_{Xi} (not zero values for example).

IF the correlation coefficient **(R)** of a certain KPI is near to one

- Create a report of this root cause and send it to the monitoring system. [KPI_Y, KPI_{Xi}]

END

IF monitoring system does not response with false alarm

-mark node as "**failed**"

-store value KPI_{Xi} that failure has been revealed.

END

- Calculate Multiple Regression function of KPI_Y with any "increased" KPI_{Xi} (not zero values for example).

IF overall null Hypothesis gives a p-value < 0.05

- Create a report of this root cause and other k statistically important factors and send it to the monitoring system. [KPI_Y, [KPI_{X1}, KPI_{X2}…KPI_{Xk}]]

END

IF monitoring system does not response with false alarm

-mark node as "**vulnerable**"

-store values [KPI_{X1}, KPI_{X2}…KPI_{Xk}] that failure has been revealed.

END

So, at the end, the monitoring system will have nodes marked as failed that should be avoided since P of failure is 1, nodes that are marked as vulnerable, that could be avoided since P of failure is near to 1 and active nodes that may be selected.

Selection Algorithm

When 3G, 4G systems are considered, each sender node receives a certain pool of nodes by DNS network element and has to decide which one to use. Currently, the criteria are based on distance, on certain features support, on roaming policies and on avoiding overloaded nodes. Though, this logic of selecting the "always best" node may be applied on top of these criteria to provide a survivable network for critical services. For 5G systems there is a whole entity that performs selection function but still the criteria are not based on failure. The current research supports that by monitoring the system's key performance indicators or by just monitoring the number of critical services failures, the possibility of critical service failure may be predicted, and the system may act proactively by choosing the safest nodes each time.

The final restriction of this model comes from the system architecture, as it has been described by standards, which sometimes indicate that if a node is chosen for an end user then he or she will be using this node until the equipment leaves the network or moves to another area. For example, an SGW chosen by an MME for a UE will be used until the UE detaches or until SGW does not cover the tracking area that UE moves to. In these cases, even if the SGW has increased possibility of failure for procedures that will follow it cannot be changed. A proposal to this one though is that for some exceptional cases, in order not to add extreme load to the network, and if the possibility of failure is 1 then relocation of nodes may be performed. An example of such a case may be an emergency service. In any case, by use of other software architectures, like cloud systems introduced in 5G, there is no such restriction since nodes are "services", for example in a microservice architecture, that may be easily turned off and replaced or used in a more efficient way.

So, the Selection algorithm based on prediction of failure of critical services may be described by the figure 64 below and may be applied by any node that should select a receiver node to perform a certain critical service.

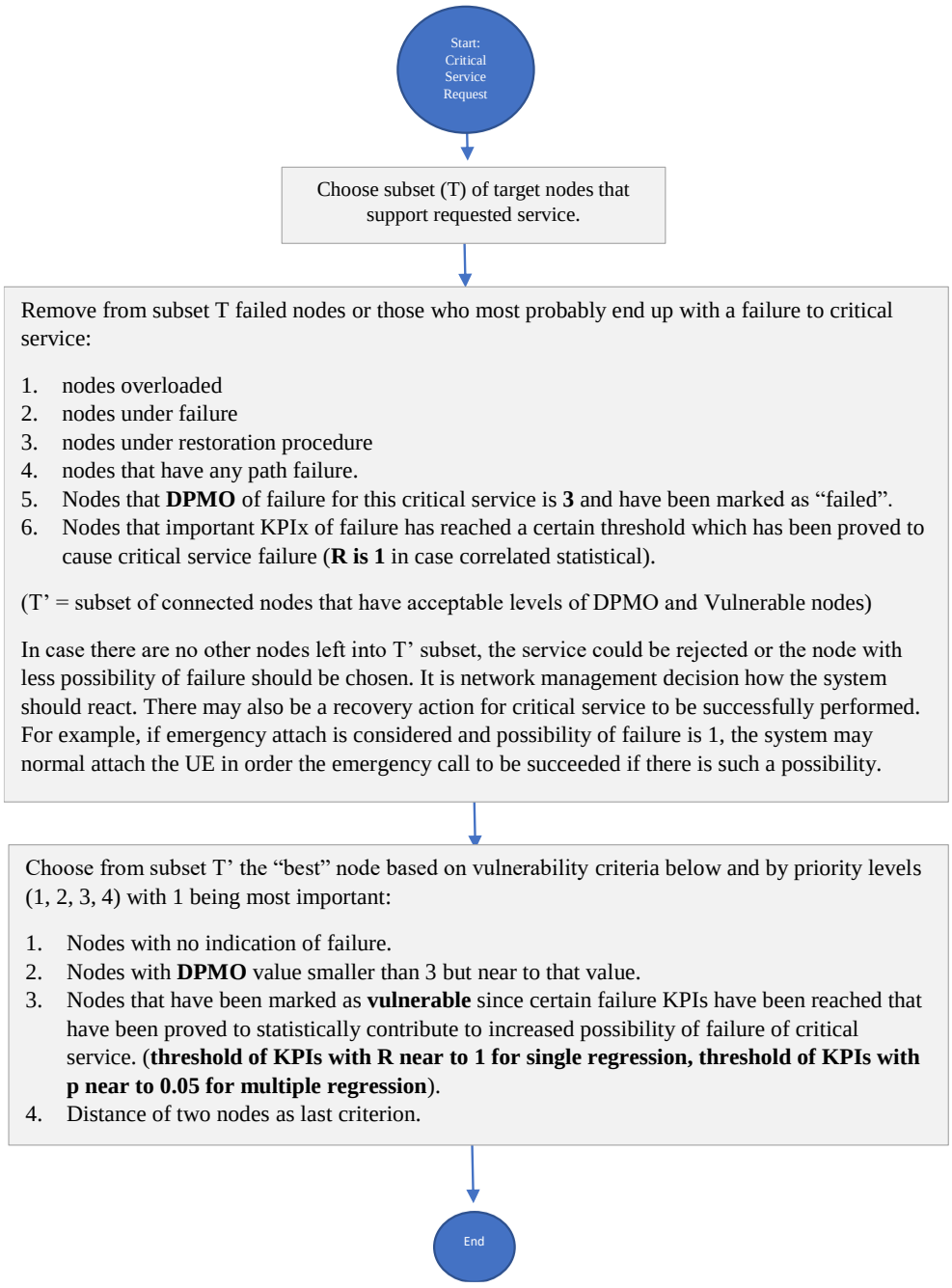


Figure 64. Selection Algorithm

4.2. SDLC for survivable telecommunication systems.

Nowadays, telecommunication systems are mostly based on Agile, or other iterative models, and spiral model, to provide continuous delivery of new functionality. At the end of all iterations, a new updated system, or a new release, has been developed and should be tested against overall functionality in order to be delivered to the customers. The main disadvantage of all proposed methodologies is that in cases of complex multi-layered systems with various components the proposed methodologies miss the requirements and threats to the system arising from the interconnection of different sub-systems. In these cases, the upper-level system is the environment that the inner system functions in. For example, telecommunication systems, have the particularity to consist of multiple nodes and to serve functionality at multiple levels; node level (ex. MME, GGSN, MSC/VLR etc.), system level (2G, 3G, 4G, 5G) and intersystem level (interconnection of systems). The result of not taking into consideration system inter-operability is a very important increase on the number of defects connected to these threats. Additionally, the testing methodologies concerning such systems are not organised and focus mostly on providing correct functionality. Sometimes, there are methodologies proposed concerning fault scenarios, but they are not documented and embedded to system development lifecycle. Usually anything that is out of the documented process tends not to be followed by development or testing teams.

Current research is based on improving this process so as the **main requirement** of the system development to be the **survivability** of the critical services and the output of the overall process after system evaluation to be a **survivable system**. The main idea of the current methodology in order survivability to be emerged by the designed system, is to consider the whole (inter)system as a deliverable of any new functionality. In this way, all requirements against survivability to all system levels will be considered and tested. The inputs to this methodology or process, that will be described are a new functionality (feature) and a defect that should be developed to an already developed, up to some level system, but it could be used for starting to develop a system from scratch. The main idea is when a new functionality is to be developed to treat it as a critical service, if it is one, and at the same time to treat it as a threat to already developed critical services of the system.

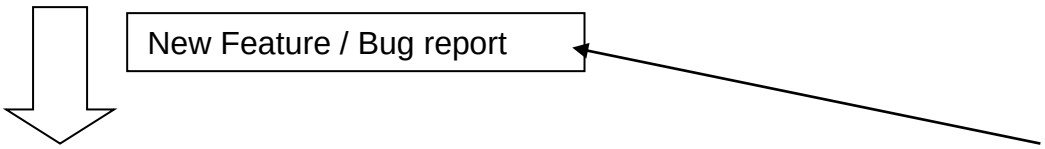
With respect to literature review regarding survivability principles, any survivable system design methodology should start by defining the **system's mission** and the **critical services** that serve system's mission. These should be documented and dealt with as requirements to any new functionality. As it has been described, the system should be managed at multiple layers so as interconnection properties that emerge to be examined. This also applies to definition of system's mission or critical services. Every system layer has its own mission and critical services that should be documented.

Starting from the inter system, its mission is identified with the scope and use of the whole system. The definition of critical services of the inter - system is abstract to this

level but clearly defines the way that these services serve the system's mission. If the new functionality that is to be developed is a critical service, then it should be added to documented critical services and clearly define in what way it contributes to system's mission. Going over to the next level, sub – systems mission should be defined in such a way to clearly define in what way each sub-system contributes to inter-system's critical services. Functional areas of this contribution should be marked as critical services for the system. This should continue until reaching the lowest level. Since these systems are networks, the lowest level is a single node. The mission of a node and its contribution to critical services should also be described.

The solution proposed by the current dissertation is Survivability by Design, meaning that survivability should be part of the software development lifecycle (SDLC) of the telecommunication system. The idea comes from Richard C. Linger et al [33] with the paper "Life-Cycle Models for Survivable Systems", that proposes survivability to be part of the SDLC phases and describes how this could be achieved. This is the theory that the current research is based on to describe how survivable telecommunication systems shall be developed. As it has already been mentioned, the proposed system that is being described by the current dissertation, apart from [33] it is also based on [90]. Secure SDLC gives an organized framework on how to design a secure system and security is of vital importance for survivability. Therefore, the current system is based on SSDLC as it is described by [90].

So, the proposed system may be schematically described by the figure 65 below.



Requirements Specification	Design	Implementation	Testing	Release	Maintenance
Survivability Requirements Compliance Analysis Survivability Risk Analysis	Survivability Threat Modelling Establish Design Requirements related to Survivability	Survivability Code Best Practices Unit Testing	Survivability Requirements Testing -Functional Testing -Integration Testing -System Testing	Survivability Incidence Response Plan	Any incident or discrepancy of KPIs may trigger a new cycle after being analysed and reported as bug.

Figure 65: Survivable SDLC.

Input to the Survivable SDLC above is a new feature, or a maintenance task like a bug fixing. An example of such a feature may be something complicated like developing the voice bearer procedure, that is most common in early stages of system development, or something less complicated like adding specific Information Elements to a message.

Requirements Specification Phase

The first step of each iteration of the SDLC proposed is the establishment of **survivability requirements**. Of course, this will be part of the overall requirements specification step of SDLC. Other requirements that may be defined are functional requirement, security requirements, performance requirements, etc.

Survivability requirements are focused on resistance, recognition, and recovery. The current proposal supports that the system under development should be always compliant with survivability requirements defined by 3GPP standards or any other rule the organization has defined. Additionally, taking into consideration that a system update may happen at node level, since code for the new feature or bug fixing will be inserted, relevant survivability requirements more specific to the new feature or bug should be defined. So, there are two levels of requirements that the system should be compliant with. General survivability requirements defined by 3GPP and software update specific requirements resulted from requirements analysis of any new feature or maintenance task.

3GPP General Survivability Requirements - Compliance Analysis:

A detailed list of 3GPP survivability requirements for telecommunication systems, that should be fulfilled by the whole system after any change by new feature implementation or maintenance task (bug fixing), may be depicted by the tables below. Requirements are grouped to service level requirements that are related to services and network level requirements that are related to network availability in order to support the operation of the services. Service level requirements are separated to recognition, resistance and recovery requirements as theory of survivability preserves.

Focusing on network level requirements, for each group, requirements related to 3Rs (recognition, resistance, recovery, adaptation) methodology are presented. Apart from 3Rs, as it has already been described, requirements definition should be based on system level. So, survivability requirements specification may be organized to Intersystem level requirements, system level requirements and node level requirements.

Survivability Requirements classification:

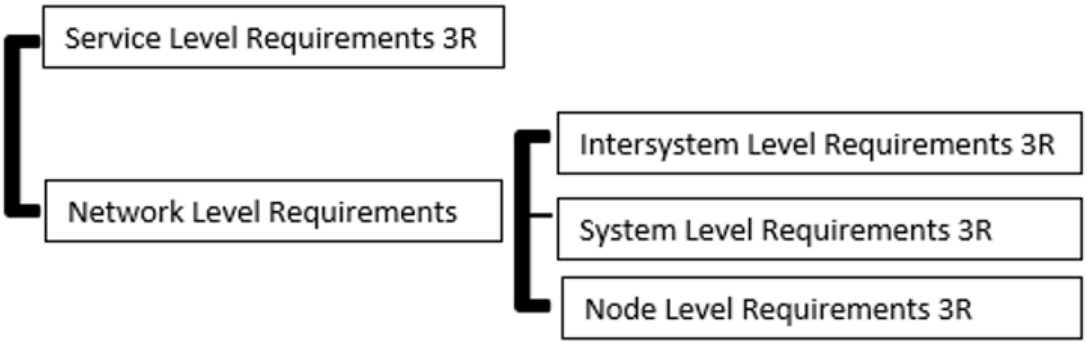


Figure 66: Survivability Requirements classification

A detailed analysis of survivability telecommunication system requirements follows. All 3GPP document references are added in the table in column 3GPP Doc Num, so they will not be included to references chapter unless there are other point in the document that are referred to. Any 3GPP document may be found in official site of 3GPP [91]

- ***Service Level Requirements:***

3GPP Title	3GPP Doc Num	3GPP Service Survivability Requirements related to failure Recognition
(1) UMTS Terrestrial Radio Access (UTRA) system	**2101-301**	*Set of attributes to describe UMTS bearer service (**delay variation tolerance, maximum transfer delay, maximum bit error rate**) information transfer rate attributes **(peak bit rate, mean bit rate, occupancy)**. *Performance: inherent transmission delay and level of traffic blocking
(2) Performance Management (PM)	**32401**	Data gathered through telecommunication management system are gathered to support performance evaluation on: - **Quality of Service** (e.g. delays during call set-up, packet throughput, etc) QoS can indicate the network performance expected to be experienced by the user.
Found across multiple 3GPP documents		
(3) Error Causes **Please refer to certain interface 3GPP document for more details**		Specific error causes may be returned to the request message each time indicating a certain failure. For example, in GTP messages error cause "Mandatory IE incorrect" may be returned. From this the root cause of failure may be depicted and corrected by development team in case it can be corrected. Otherwise, there may be causes like "network failure" with root cause some failure to the network where all connections of the node with the node that returned this value, should be deleted.
3GPP Title	**3GPP Doc Num**	**3GPP Service Survivability Requirements related to failure Resistance**
(4) UMTS	**2101-301**	Handover should be transparent. In case of speech call loss of information may be tolerated but handover should be quick to **avoid connection break.** In case of data service temporary break is tolerable but not loss of information. Handover between terrestrial environments should be seamless within the same network
		Handovers should not increase the load on the fixed network significantly
		The level of security should not be affected by handovers
		Bearer services cannot be handed over between two environments if they are not supported in both. However, handover to an alternative bearer **offering reduced capabilities** should be possible where this is supported by the service in use. The radio

		interface should have the capability to provide for handover and roaming between networks run by different operators
(5) Services and System Aspects;	**22 101**	Any handover required to maintain an active service while a user is mobile within the coverage area of a given network, shall be seamless from the user's perspective.
		The 3GPP system shall be able to provide **continuity between CS voice services** and the full duplex speech component of IMS multimedia telephony service with no negative impact upon the user's experience of the voice service. The same should be true for IMS services.
		The system shall support either - transparent relay of the IP signaling and traffic; - service aware interconnection
(6) 3G security; Security threats and requirements	**21 133**	Service Integrity: "It shall be possible to protect against unauthorized modification of user traffic" Service availability: It shall be possible to prevent intruders from restricting the availability of services by logical means
(7) Security Objectives and Principles	**33 120**	Security Objectives: 1. to ensure that the security features standardized are compatible with world-wide availability 2. to ensure that the security features are adequately standardized to ensure world-wide interoperability and roaming between different serving networks;
(8) Security architecture	**33 401**	The standard presents: - user identities confidentiality: MSIN, the IMEI, and the IMEISV should be confidentiality protected - user data signaling confidentiality: All S1 and X2 messages carried between RN and eNB shall be confidentiality-protected. Synchronization of the input parameters for integrity protection shall be ensured for the protocols involved in the integrity protection. - Integrity protection, and replay protection, shall be provided to NAS and RRC-signaling. - authentication and key agreement procedure between the mobile device and the core network, - security interworking of mobile networks (EUTRAN-UTRAN-GERAN),
(9) Technical Specification Group Services and System Aspects;	**23 401**	Authentication: NAS security mode control procedure is to take an EPS security context into use, and initialize and start NAS signaling security between the UE and the MME with the corresponding EPS NAS keys and EPS security algorithms
(10) 5G; Security architecture and procedures for 5G System	**33.501**	The standard presents: - network access security: enable a UE to authenticate and access services via the network securely, including the 3GPP access and on-3GPP access, and in particularly, to protect against attacks on the (radio) interfaces -network domain security secure exchange of signaling and user plane data between networks. - User domain security: user access to mobile equipment. - Application domain security: enable applications in the user domain and in the provider domain to exchange messages securely
		As it is presented to the current standard part of network life-cycle includes: the PLMN network is being adjusted to meet the long-term requirements of the network operator and the customer, e.g. with regard to performance, capacity and customer satisfaction through the enhancement of the network or equipment up-grade

Found across multiple 3GPP documents

(11) Error Handiling	Some error causes indicate failures that can be handled in order to avoid dropping the service. Sometimes these handlings may be found across 3GPP documents or there may be implementation specific approaches that each organization implements during development of the device. To the example above "Mandatory IE incorrect" if we assume that the mandatory IE that is not correct is bearer ID. And the message causing this error is an answer to a previous message, then we may conclude which is the correct bearer id and ignore the error instead of dropping the service. The same may happen with network errors if we use relocation through selection functions to relocate the service that may be dropped in case it is critical (voice bearer for example)
(12) Collision Handling	Collision is the case where two messages requesting a service arrive at a network and at the same time or one request arrives before the whole process of messages of the previous one has been completed. Then a handing of these requests should take place. This handling may be for example to serve both requests by a priority sequence, or to drop one of the two. For example, in case a request arrives for a UE that is already in process of a handover there is no meaning in processing it since the UE will leave from current Tracking area. Though there are cases that the service should continue to the Tracking area the UE will move to.

(13) Error Handiling		Apart from error causes defined by 3GPP documents and robust measurements that should be developed in order such cases to be handled, here we introduce some other error handline requirements: 1. The system should be able to resist to failures related to loss of messages. The failure should be ignored if this is possible. For example, if an acknowledgement message has not arrived, the service could be considered as established to avoid dropping it. If it could not be ignored, then the system should consider if there is a failure of neighboring node. In this case, the node should inform network management system and release any connection associated with this node. 2. The system should be able to react to messages arriving later or earlier than expected. This should not have any impact to the service or to any other following services. 3. The system should be able to resist to failures related with duplicate messages sent to the nodes. 4. Any new functionality should be considered as a threat to the **critical services** already developed and any possible failure should be handled.
(14) Hanging Processes		As "hanging processes" we mean a service that fails, and leaves resources reserved causing failure to future services. For example, if a PDN Connection fails to be released and it is found as "already established" when a new PDN Connection is requested. This PDN Connection may be a critical service like voice bearer.
3GPP Title	**3GPP Doc Num**	**3GPP Service Survivability Requirements related to service Recovery from failure and adaptation.**
(15) Restoration procedures	23 007	The data stored in location registers are automatically updated in normal operation; the main information stored in a location register defines the location of each mobile station and the subscriber data required to handle traffic for each mobile subscriber. The loss or corruption of these data will seriously degrade the service offered to mobile subscribers; it is therefore necessary to define procedures to limit the effects of failure of a location register, and to restore the location register data automatically
(16) Services and Systems Aspects;	22 101	The voice call continuity user's experience shall be such that, to the greatest degree possible, a consistency of service is provided regardless of the underlying communication infrastructure and technology
(17) UMTS	2101-301	Flexibility: Negotiation of bearer service attributes (bearer type, bit rate, delay, BER, up/down link symmetry, protection including none or unequal protection), parallel bearer services (service mix), real-time / non-real-time communication modes, adaptation of bearer service bit rate
		UTRA should adapt flexibly into changes and should have the capability to serve a variety of traffic densities (up to very high densities) and a variety of traffic mixes in an economical way.
		Flexibility and dynamic reconfiguration: minimum set of bearer capabilities, operating modes and features to ensure that inter-operability is always possible; continuity of operation during dynamic updating of terminal capabilities.
(17) Self-Organizing Networks (SON); Self-healing concepts and requirements	32541	In the case of software faults, the recovery actions may be : a) system initializations (at different levels), b) reload of a backup of software, c) activation of a fallback software load, d) download of a software unit, e) reconfiguration, etc. In the case of hardware faults, the recovery actions depend on the existence and type of redundant (i.e. back-up) resources. If the faulty resource has no redundancy, the recovery actions may be: a) Isolate and remove the faulty resource from service so that it does not disturb other working resources; b) Remove the physical and functional resources (if any) from the service, which are dependent on the faulty one. This prevents the propagation of the fault effects to other fault-free resources; c) State management related activities for the faulty resource and other affected/dependent resources; d) Reset the faulty resource; e) Other reconfiguration actions, etc. If the faulty resource has redundancy, the recovery action shall be changeover, which includes the action a), c) and d) above and a specific recovery sequence. The detail of the specific recovery sequence is out of the scope of the present document

Table 4: Service Level Survivability Requirements.

➢ *Network Level Requirements:*

3GPP Title	3GPP Doc Num	3GPP Network Survivability Requirements related to failure Recognition		
		Node Level	**System Level**	**Intersystem Level**
(1) Telecommunication management; Principles and high-level requirements	32.101	Telecommunication management system consists of an architectural framework or management reference model, that is used to collect measurements for management functions. Some of which are related to survivability like performance management, fraud management, fault management, security management, etc. With the use of performance measurements, configuration of system due to load needs may be executed. Additionally, for fault management, alarms or events may also imply a needed re-configuration for avoiding failures. Failure may be detected; isolated and root cause may be depicted.		
(2) Performance Management (PM)	32401	Data sent at node level are gathered through telecommunication management system to support performance evaluation on: - traffic levels within the network, including the level of both the user traffic and the signaling traffic - verification of the network configuration: evaluation of effectiveness of changes of network plan related to traffic levels. - resource access measurements - resource availability (e.g. the recording of begin and end times of service unavailability)	Network Operators are informed of PM - related events through alarms and may act accordingly.	
(3) Fault Management;	32.111-1	If the faulty resource has no redundancy, the recovery actions shall be: - Generate and forward appropriate notifications to inform the **OS** about all the changes performed.		
3GPP Title	**3GPP Doc Num**	**3GPP Network Survivability Requirements related to system Recovery from failure and adaptation**		
		Node Level	**System Level**	**Intersystem Level**
(4) Restoration procedures	23 007	The data stored in location registers are automatically updated in normal operation; the main information stored in a location register defines the location of each mobile station and the subscriber data required to handle traffic for each mobile subscriber. The loss or corruption of these data will seriously degrade the service offered to mobile subscribers; it is therefore necessary to define procedures to limit the effects of failure of a location register, and to restore the location register data automatically. The document describes data restoration procedures for VLR, HLR, HSS, GGSN, SGSN, MME. Triggering point is receiving a request for unknown IMSI in cases when the failing node has not detected the failure or receiving a message with restoration indicator set to not confirmed. These indicators show data corruption and procedure for restoring of these data through message exchange follows.		
		Node restart. If a node restarts it sends a reset indicator to the neighboring nodes. Upon receiving such an indicator, the neighboring node shall inform its neighbors about the failure and release and re-initiate any PDN connection associated with failing node.		

(5) Fault Management	**32.111-1**	After a fault has been detected and the replaceable faulty units have been identified, some management functions are necessary in order to perform system recovery and/or restoration, either automatically by the NE and/or the EM, or manually by the operator. If the faulty resource has no redundancy, the recovery actions shall be: a) Isolate and remove from service the faulty resource so that it cannot disturb other working resources; b) Remove from service the physical and functional resources (if any) which are dependent on the faulty one. This prevents the propagation of the fault effects to other fault-free resources; c) State management related activities for the faulty resource and other affected/dependent resources.		
(6) Self-Organizing Networks (SON); **Self-healing concepts and requirements**	32541	In the case of **software faults**, the recovery actions may be : a) system initializations (at different levels), b) reload of a backup of software, c) activation of a fallback software load, d) download of a software unit, e) reconfiguration, etc. In the case of **hardware faults**, the same as line of fault management above plus this: a) **Reset** the faulty resource; b) Other reconfiguration actions**, etc. If the faulty resource has redundancy, the recovery action shall be changeover. **Here we see that reconfiguration is something proposed by 3GPP but not a "must have" attribute.		
3GPP Title	**3GPP Doc Num**	**3GPP Network Survivability Requirements related to failure Resistance**		
		Node Level	**System Level**	**Intersystem Level**
(7) (UMTS); **protocol description and error handling**	25.921	The error handling shall be specified in the protocol for the cases when the requirement for presence or absence of an IE indicated by the condition is not followed.		
	23401	SGW-MME / SGW-PGW **GTP-C Load Control** feature is an optional feature which allows a GTP control plane node to send its Load Control Information to a peer GTP control plane node which the receiving GTP control plane peer node uses to augment existing GW selection procedure	**APN level load control** may be supported and activated in the network. If this feature is activated, the PDN GW may convey the Load Control Information at APN level (reflecting the operating status of the resources at the APN level), besides at node level.	

(8) Technical Specification Group Services and System Aspects;		SGW-MME / SGW-PGW **GTP-C Overload Control** feature is an optional feature. Nodes using GTP control plane signaling may support communication of Overload Control Information in order to mitigate overload situation for the overloaded node through actions taken by the peer node(s)	**NAS Level Congestion control**: The MME may detect the NAS signaling congestion associated with the APN and start and stop performing the APN based congestion control based on criteria: (max number of EPS bearers and EPS bearer activation per APN, one or multiple PDN GWs of an APN are not reachable or indicated congestion to the MME, Maximum rate of MM signaling requests associated with the devices with a particular subscribed APN, Setting in network management)	
		MME-Enb The **MME Load Balancing** functionality permits UEs that are entering into an MME Pool Area to be directed to an appropriate MME in a manner that achieves load balancing between MMEs.	PDN **GW control of overload** by rejection of PDN connection requests from UE.	
		MME-Enb The **MME Load Re-balancing** functionality permits UEs that are registered on an MME (within an MME Pool Area) to be moved to another MME		
		MME The MME shall contain mechanisms for avoiding and handling overload situations		
		SGW-MME **Throttling of Downlink Data Notification** Requests. MME may restrict the signaling load that its SGWs are generating on it, if configured to do so.		
		MME-UE UE Level **NAS congestion**: The MME may detect the NAS signaling congestion associated with the UEs belonging to a particular group. The MME may start and stop performing the group specific NAS level congestion control based on criteria (maximum rate of MM and SM signaling requests associated with the devices of a particular group, Setting in network management)		
(9) Configuration Management (CM);	32.600	Configuration Management (CM), in general, provides the operator with the ability to assure correct and effective operation of the PLMN network as it evolves. CM actions have the objective to control and monitor the actual configuration on the Network Elements (NEs) and network resources, and they may be initiated by the operator or by functions in the Operations Systems (OSs) or NEs. CM actions may be requested as part of an implementation program (e.g. additions and deletions), as part of an optimization program (e.g. modifications), and to **maintain the overall Quality of Service (QoS)**. The CM actions are initiated either as single actions on single NEs of the PLMN network, or as part of a complex procedure involving actions on many resources/objects in one or several NEs.		

Table 5: Network Level Survivability Requirements.

New feature / maintainace task at Node Level Survivability Requirements:

These survivability requirements are result of analysis of the above general requirements of survivability defined by 3GPP and are more focussed on node level requirements that later will be translated to development tasks for node implementation. Most of the times these are the implementation specific requirements as described by 3GPP, which means that each organization is responsible to decide for their implementation. At node level, all survivability requirements related to service unavailability may be summarized into the communication between two nodes. A detailed analysis of this concept is described in chapter 'Recognition of Failure' of approaches beyond 3GPP. With respect to this analysis, requirements that should be extracted at node level may be listed as follow:

1. Node should be able to recognise any failure to the service of neighbouring network element nodes and report it to the centralized network management system. Additionally, the root cause of failure should be defined and communicated to the network management system. So, if for example the neighbouring node does not answer to messages and there is no restart counter indicating node failure defined for specific message flow, the node under development should report this to the network management system. Additionally, the node under development should try to investigate the root cause of failure at node level.
2. The same should happen in case of timing issues which may reveal a load situation of the neighbouring node.
3. Node should report any failure detected to its own system, to the network management system. The failure may be related to H/W failure, CPU or Memory Overload or S/W failure that the system detects it comes from itself.
4. Node should try to self-heal from failure to any possible level.
5. Requirements related to code robustness should be defined. New functionality should provide resistance to failure which is related to any of the following reasons:
    - Collision scenarios: Definition of possible collision scenarios and system requirements at node level that could be extracted.
    - Synchronization issues: Definition of requirements arising from scenarios to which answers to already sent messages arrive later than expected or duplicate answers arrive. Duplicate messages sent by the node under development should be recognised and reported.
    - Hanging resources: Definition of possible impact to the new functionality of any service that left as 'hanged' should also be extracted.

Finally, part of requirements specification should be the definition of Key Performance Indicators related to service failure. More precisely, thresholds regarding service failures should be defined after which recovery actions should be triggered. More

details on this mechanism may be found to the chapter 'Resistance to failure' of approaches beyond 3GPP proposed by the current research.

To sum up, during requirements specification, requirements related to compliance with the standard should be reported and implementation specific requirements defined by the current dissertation should also be extracted in order the system to be designed and implemented as a survivable system.

The next step is **Compliance Analysis** and may be conducted by development team in order to decide if the system is compliant with standard's requirements and which requirements will not be fulfilled with the corresponding justification. Additionally, any legislation requirements should be included.

Survivability Risk Analysis

Then, **Survivability Risk Analysis** should follow. The main objective of survivability risk is the avoidance of critical services failure. So, the risk analysis from survivability perspective that should be conducted is a detailed **Failure Mode and Effects Analysis (FMEA)**. FMEA is a tool that may be used in order to discover failure in project early stages. By this tool, practitioners may discover what may go wrong with the developed project. During FMEA, for survivability to be accomplished, the first thing that should be done is to define if a critical service for system mission is directly affected by the new feature or maintenance task and in what way. Additionally, it should be defined if the feature is a critical service on its own. FMEA may be conducted by filling the table below:

Process Step	Potential Failure Mode	Potential Failure Effect	SEV[1]	Potential Causes	OCC[2]	Current Process Controls	DET[3]	RPN[4]	Action Recommended
What is the step?	In what ways can the step go wrong?	What is the impact on the customer if the failure mode is not prevented or corrected?	How severe is the effect on the customer?	What causes the step to go wrong (i.e., how could the failure mode occur)?	How frequently is the cause likely to occur?	What are the existing controls that either prevent the failure mode from occurring or detect it should it occur?	How probable is detection of the failure mode or its cause?	Risk priority number calculated as SEV x OCC x DET	What are the actions for reducing the occurrence of the cause or for improving its detection? Provide actions on all high RPNs and on severity ratings of 9 or 10.

1. **Severity**: Severity of impact of failure event. It is scored on a scale of 1 to 10. A high score is assigned to high-impact events while a low score is assigned to low-impact events.
2. **Occurrence**: Frequency of occurrence of failure event. It is scored on a scale of 1 to 10. A high score is assigned to frequently occurring events while events with low occurrence are assigned a low score.
3. **Detection**: Ability of process control to detect the occurrence of failure events. It is scored on a scale of 1 to 10. A failure event that can be easily detected by the process control is assigned a low score while a high score is assigned to an inconspicuous event.
4. **Risk priority number**: The overall risk score of an event. It is calculated by multiplying the scores for severity, occurrence and detection. An event with a high RPN demands immediate attention while events with lower RPNs are less risky.

Figure 67: Failure Mode and Effects Analysis (FMEA) [92]

As it may be observed, FMEA as risk analysis gives a quantifiable method of calculating the risk of any failure.

To sum up, during requirement specification phase, requirements of survivability should be defined and added to general requirements of system under development.

From these requirements a compliance analysis will reveal if the system is compliant with standards or legislation. Finally, a survivability risk analysis should be conducted via an FMEA procedure in order possible threats to services to be revealed.

Design phase

Part of Secure Software Development design phase is **Threat modelling**. This should be enriched with survivability threats found by previous steps (FMEA). According to OWASP [93], a threat model typically includes:

- Description of the subject to be modelled
- Assumptions that can be checked or challenged in the future as the threat landscape changes
- Potential threats to the system
- Actions that can be taken to mitigate each threat
- A way of validating the model and threats, and verification of success of actions taken

So, for each threat, as it was defined during previous steps, mitigation actions should be extracted. Here is where decisions are made on how to mitigate each risk in order the system to be able to recognise, resist and recover from survivability threats.

What is important to pinpoint here is that threat modelling is a process that goes on during all steps of SDLC in case new threats arise later than design phase.

Accompanied with Threat modelling, **design principles** related to survivability are established during design phase.

More precisely, recognition, resistance and recovery mechanisms should be defined, with respect to survivability requirements, to mitigate any threat to services survivability including:

- failures related to system's availability at any level (node, system, inter-system level) (including overload situations)
- software failures
- path failures (including overload situations affecting the network between system nodes)
- configuration failures
- any other incident that could cause service failure.

Implementation phase

During implementation phase, the actual feature implementation takes place, with respect to requirements and design principles. During this phase, from survivability

perspective, it is important to take into consideration certain **code practices** (secure programming, robust programming etc), that will ensure avoidance of software error, and enrich Unit testing with scenarios related to survivability requirements.

Starting with code practices, from survivability perspective, apart from producing secure and robust code, in order to avoid software failures, it is important to catch and report any event related to survivability to the network management system. Examples of such events may be exceptions, memory load or overload cases and security incidents like DoS attacks, buffer overflows etc. Additionally, any service failure should be accompanied with a trace of the exact point of failure for diagnostics reasons. This should be reported to the network management system as solution proposed by the current dissertation proposes (4.1.1).

Apart from code practices, **unit testing** should include scenarios related to survivability requirements (Recognition, Resistance, Recovery) as module testing level. More details on these scenarios as well as a draft test plan will be described in the next chapter.

Testing phase

To continue, the testing phase of the proposed SDLC is presented. Testing is the way to evaluate a system's survivability. Testing phase should also follow the same model and test cases should be designed for node, system, and intersystem level. In this way the whole system will be tested each time.

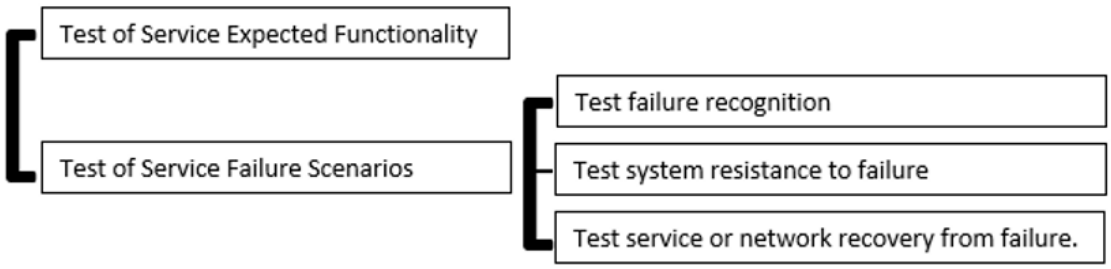


Figure 68: Testing mobile systems survivability requirements

Additionally, test cases should include tests against services' correct functionality, and they should be extended to also test any resistance, recognition or recovery survivability requirement to all testing levels (node, system, intersystem). For this to be achieved test-driven development is the most appropriate approach. Modern SDLC approaches are test-driven which is what is also proposed for the current SDLC.

Test-driven means that the tests are designed according to the requirements and are constructed even before the development of new features or maintenance tasks like bug fixing. Additionally, through this work we propose another approach that is related

to test-driven development and has to do with **failure impact** evaluation. In other words, testing may be also used to evaluate the impact of any failure to critical services, and having this information available, new tasks may be extracted for the next iteration cycle regarding failure recognition, resistance or recovery. So, in this case tests are indeed driving the development and are a tool to discover many issues or threats that may occur from any combination of services. So, any time a new service is to be developed or updated, testing any possible combination of it with critical services will reveal any threats to critical services from the newly inserted code.

Impact analysis could be applied in any iteration and phase of SDLC providing new requirements related to survivability requirements. Tests related to impact analysis may be:

1. Executing critical services before and after newly developed or modified service.

2. Executing critical services after failure of newly developed or modified service.

3. Executing critical services in collision with newly developed or modified service.

Additionally, another proposal is to test all survivability requirements for each new or modified functionality. So apart from just testing failure scenarios, recognition of failure and recovery from failure or resistance to failure should be also tested in order testing procedure to be considered complete.

All tests related to survivability evaluation and corresponding test approaches that could be used, are depicted in the following table below. Test scenarios are also related to corresponding threat to survivability and impact of realization of this theat. Finally, any test case should be added to regression testing in order to ensure that future changes will not affect the existing functionality.

Survivability Threats		Root Cause of Failure	Failure Impact in Node Level	Test Scenarios	Testing Methods	Examples from 4G network
	Hardware failures, i.e. the malfunction of some physical resource within a NE.	Device damage	Messages sent from one node to neighboring node may not be answered.	Testing of scenario where device is forced out - no information of the event to management system. The NE should be able to track the issue and report to management system. The impact from this failure to service under development and the restoration time should be defined.	Functional testing	Unplug of the device.
				Testing of scenario where device fails and sends alarm to management system. Service under development should be released or served by alternative resources after	Functional testing	Enforcement of the NE to send a failure alarm to the management system

3GPP Fault Management; 32.111-1 Categories of faults for which an NE (network element) may raise alarms are:				system re-configuration.		
		CPU / Memory Overload	Messages sent from one node to neighboring node may not be answered or answered with delay.	Testing of scenarios that the message is not answered from neighboring NE in all phases of service establishment and test requirements related to handing of this situation.	Functional testing Unit or Module Testing Static Analysis,	Test scenarios where message is not answered.
		System missconfiguration				
			Faulty messages may arrive to NEs.	Testing of scenarios that the message arrives with wrong configuration information.	Functional testing Unit or Module testing Static Analysis Fuzzy-testing Fault-injection testing	Test message with wrong information about MMEs capability of supporting IOT devices.
	Software problems, e.g. software bugs, database inconsistencies	Any S/W bug that results in wrong functionality of service or non-compliance with standards	Service rejection or faulty service establishment.	Test-driven development with tests that are designed due to 3GPP standards requirements.	Functional testing Unit or Module Testing Static Analysis Fault-injection testing	Test all scenarios that reflect 3GPP requirements.
		S/W Bug lead to hanging processes	Future service requests may be rejected.	Enforce processes to be hanged and see if system reacts according to requirements. Test critical services impact if attempted.	Functional testing Unit or Module Testing Static Analysis,	Test if after deletion of a voice bearer it can re-established.
		Missing of robustness measurements like handling collision scenarios or handling of wrong Information Elements in messages	Service rejection or faulty service establishment.	Testing of all possible collision combination, especially with critical services, and test scenarios during which messages have wrong IEs that could be handled by robustness measurements.	Functional testing Unit or Module Testing Static Analysis,	PDN connection consists of a series of messages. A test case could include the modification of bearer id to a wrong one and see if system is robust enough to handle this error.
3GPP Fault Management; 32.111-1		S/W bug that may lead to unanswered messages	Service rejection.	Testing of scenarios where messages of process under development are not answered.	Functional testing Unit or Module Testing Static Analysis,	A test case could be the PDN establishment and testing if service is properly rejected.
				Test the impact to critical services. Test cases with critical services already established and the above scenario following should be tested. The opposite is also valid scenario and should be tested. In this case failures from hanging	Functional testing Unit or Module Testing Static Analysis,	Testing of the above scenario after and before voice bearer handover.

				processes will also be tested.		
		S/W bug that may lead to message sent twice	Service may be re-established if there is no mechanism for ignoring repeated messages	Testing of scenarios where messages of service under development are sent twice.	Functional testing Unit or Module Testing Static Analysis,	A test case could be sending PDN request twice for the same bearer.
	Functional faults, i.e. a failure of some functional resource in a NE and no hardware component can be found responsible for the problem.	Any other failure that may lead to service unavailability	Service Failure	Any related test	Functional testing Unit or Module Testing Static Analysis,	Any related test
	Loss of some or all of the NE's specified capability due to overload situations.	System Overload of requests	Messages sent from one node to neighboring node may not be answered or answered with delay.	Testing of service impact after increasing system load. Testing service impact after increasing load of service.	Stress testing Load testing Stability testing	Try to establish a voice bearer in a loaded system and an overloaded system. And try to see the impact to the system and voice bearer when system is loaded by voice bearer requests.
	Communication failures between two NEs, or between NE and OS, or between two OSs.	S/W failures, H/W failure, Overload situations, Path / Link failures, Network timing issues.	Messages sent from one node to neighboring node may not be answered or answered with delay.	Testing of scenarios of scenarios that the message is not answered from neighboring NE in all phases of service establishment and test requirements related to handing of this situation.	Functional testing Unit or Module Testing Static Analysis,	Testing of scenarios of scenarios that the message is not answered from neighboring NE.
Security Testing	Any security threat should be considered and tested. Details on security testing will not be provided to current document.					
Failure Recognition	In all possible errors, network management system should be tested. Network management system should be informed about any kind failure and should be able to trigger system resistance or recovery mechanisms. So, any NE that is under development should be tested against this functionality also.					
System Recovery	In all possible failure scenarios, recovery mechanisms following should also be tested.					

Table 6: Evaluation of Survivability Requirements through testing.

Release

During release phase, a new version of a product is released. After all iterations and end to end testing, a new version of the product is ready for release. In telecommunication systems, a new release may include software of more than one node. For example, there may be a new version of SW of MME node SW, of SGW node and of telecommunication management system.

Part of release phase is the incidence response plan. Due to references, *"an incident response plan is a set of instructions to help IT staff detect, respond to, and recover from network security incidents"* [94]. In case of a survivable systems, the system by itself is built in a way to be able to recognise, resist and recover from attacks, failures, etc. Though, any survivability measurements may also fail, and IT staff should be able to perform the required survivability measurements based on an Incident Response Plan.

"An incident response plan often includes:

- *A list of roles and responsibilities for the incident response team members.*
- *A business continuity plan.*
- *A summary of the tools, technologies, and physical resources that must be in place.*
- *A list of critical network and data recovery processes.*
- *Communications, both internal and external." [96]*

An incidence from survivability perspective is a service failure of any kind. Several thresholds of service failures may raise alarms for incidence response. These values may be defined as Survivability Key Performance Indicators and are defined during requirements specification. Certain actions regarding survivability requirements "recognition", "resistance" and "recovery" should be defined for each incident.

Maintenance

Any failure of survivability requirements or any exceed of threshold of Survivability KPIs may trigger a new SDLC cycle in order software changes that fix any issues to be analysed, implemented, and tested.

After analysis of proposed SDLC, an application of each step on an example feature will be described in the next chapter.

Survivability evaluation metrics

Any service failure related KPI shall be considered as survivability evaluation metric for the developed system.

5. Application of proposed methodology.

So far, the methodology for building a survivable telecommunication system has been described. During this chapter, appliance of the proposed methodology will be presented. As a case study the feature that will be used is a 4G Voice Bearer Establishment.

Before the analysis of the SDLC, some assumptions should be presented. The system is considered as "developed" up to some point. It will be assumed that:

1. Features attach, tracking area update, and default bearer establishment are already implemented. This should be a precondition for a voice bearer to be established.
2. 4G network system is already implemented.
3. Testing environment at all levels is already defined. Unit testing is executed by development team, functional, system, performance, and end to end testing is executed through QA (Quality Assurance) teams.
4. Test cases of all levels are executed in regression mode, through Continuous Integration – Continuous Deployment (CI/CD) system, each time code is committed to the system. By this way, it is ensured that there will be no failure to the already implemented system.

5.1. Case Study: 4G Voice Bearer Establishment.

Before starting with the SDLC proposed, the 4G voice bearer establishment process is to be presented. Bearers in 4G network is a virtual tunnel established between two end points for traffic management. For example, between two mobile phone subscribers or a mobile network user and a web server service a web page on the internet. A bearer indicates how data travelling between the two end points are to be treated. Some bearers guarantee the maximum bit rate that the traffic will experience, and some others do not. There are two kinds of bearers. The default bearer and the dedicated bearers. A default bearer is the first bearer created when a subscriber connects to the network for the first time. A dedicated bearer is created on top of an existing default bearer, with which it shares the IP address, in order to provide a dedicated tunnel to specific traffic (Voice over IP, video, etc.). A voice bearer is a dedicated GBR (Guarantee bit rate) bearer with QCI (QoS class of identifier) 1 as it may be depicted by the table below:

QCI	Bearer Type	Priority	Packet Delay	Packet Loss	Example
1	GBR	2	100 ms	10	VOIP Call
2		4	150 ms	10	Video Call
3		3	50 ms		Online Gaming (Real Time)
4		5	300 ms	10	Video Streaming
5	Non-GBR	1	100 ms		IMS Signaling
6		6	300 ms		Video, TCP based services e.g. email, chat, ftp etc.
7		7	100 ms	10	Voice, Video, Interactive gaming
8		8	300 ms	10	Video, TCP based services e.g. email, chat, ftp etc.
9		9			

Table 7: QCI and corresponding IP level packets characcteristics. [96]

A dedicated bearer starts as "UE-initiated", meaning that a UE is calling another UE, and continues as "Network-initiated" when the network is trying to reach (through paging process) and establish a connection with the other UE. For the current dissertation we will focus on "UE-initiated" dedicated bearer establishment. This is the establishment of the dedicated bearer from the caller UE to the network.

The flow of "UE-initiated" dedicated bearer activation may be depicted by the figure below:

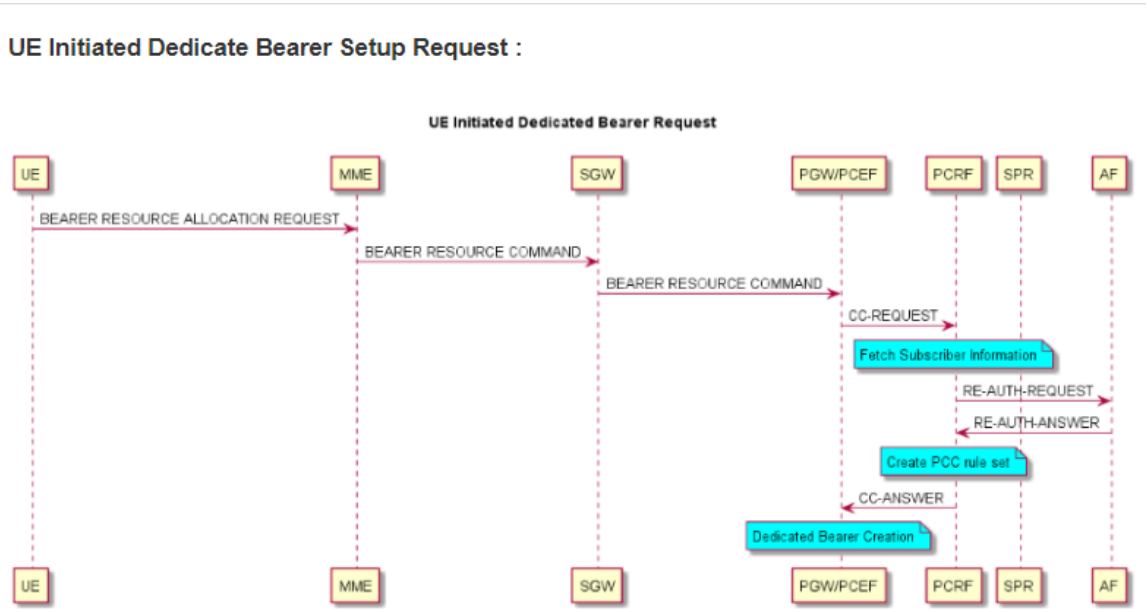


Figure 69: UE-initiated barer setup request. [97]

Additionally, the current dissertation will focus on the Mobility Management Entity (MME) as network node under investigation. MME provides the required data for this process and delivers information regarding data radio bearer like min/max bandwidth, Quality of Service etc. Furthermore, MME is the device that will initiate paging process and authentication of the mobile device and select the appropriate SGW to complete the communication between the two devices. So, MME is playing a role of vital

importance for survivability. Therefore, the current dissertation with focus on this device.

A short description of the procedure provided by 3GPP 24301 (NAS Protocol) may be:

"*The purpose of the UE requested bearer resource allocation procedure is for a UE to request an allocation of bearer resources for a traffic flow aggregate. The UE requests a specific QoS demand (QCI) and optionally sends a GBR requirement for a new traffic flow aggregate. If accepted by the network, this procedure invokes a dedicated EPS bearer context activation procedure (see subclause 6.4.2) or an EPS bearer context modification procedure (see subclause 6.4.3). If there is a PDN connection for emergency bearer services established, the UE shall not request additional bearer resources for this PDN connection*." [98]

So, during this procedure MME will receive message bearer resource allocation request from the eNB and will send message bearer resource command to SGW. In case of an error MME will send a Resource allocation reject message to the eNB indicating the root cause of failure.

Detailed description of the procedure:

1. UE sends through eNB a Bearer Resource Allocation Request message to the MME and starts timer T3480 waiting for an answer. If an answer does not arrive before the timer expires, the UE will re-send the message. "*This retransmission is repeated four times, i.e., on the fifth expiry of timer T3480, the UE shall abort the procedure, release the PTI allocated for this activation" [98].*

2. The Bearer Resource Allocation request message should include the following fields:

Information Element	
Protocol discriminator	(M) Protocol being used
EPS bearer identity	(M) EPS bearer identity: EPS bearer identity is used to identify a message flow
Procedure transaction identity	(M) procedure transaction identity (PTI) is an identity which is allocated by the user equipment for the UE-requested bearer resource activation
Bearer resource allocation request message identity	(M) Message type - identifies that this is a bearer activation procedure
Linked EPS bearer identity	(M) Default bearer that dedicated bearer that will be established with connect to.
Spare half octet	(M) Filled with spare bits for description of EMM/ESM messages
Traffic flow aggregate	(M) The packet filters determine the traffic mapping to EPS bearer contexts. Uplink packet filters shall be applied by the UE (3GPP 23401) UE shall set this to "Create new TFT" since there is no active bearer yet.
Required traffic flow QoS	(M) EPS quality of service. The UE should define a required QCI (QoS Class Identifier) which is 1 in out case and GBR required
Protocol configuration options	(O) Included in the message when the UE wishes to transmit (protocol) data
Device properties	(O) include this IE if the UE is configured for NAS signaling low priority

Figure 70: Bearer Resource Allocation Request message parameters [98]

3. MME checks whether resources requested by message received by verifying existence of default bearer provided in Linked Bearer ID field of the message. If there is a failure, then MME will send a "RESOURCE ALLOCATION REJECT" message with a relevant cause. PTI value may be used to define which is the exact procedure that the failure message is coming for.

4. MME sends Bearer Resource Command Message to the SGW which will forward the message to PGW. The Bearer Resource Command message may be depicted by figure below:

Information elements	P	Condition / Comment	IE Type	Ins.
Linked EPS Bearer ID (LBI)	M		EBI	0
Procedure Transaction Id (PTI)	M		PTI	0
Flow Quality of Service (Flow QoS)	C	This IE shall be included on the S4/S11 interface if the "Requested New QoS"/"Required QoS" is included in the corresponding NAS message (see section 9.5.10 and section 9.5.15a in 3GPP TS 24.008 [5]) or the "Required traffic flow QoS" is included in the corresponding NAS message (see section 8.3.8 and section 8.3.10 in 3GPP TS 24.301 [23]). If SGW receives this IE, SGW shall forward it to PGW across S5/S8 interface.	Flow QoS	0
Traffic Aggregate Description (TAD)	M	The TAD consists of the description of the packet filter(s) for a traffic flow aggregate. MME shall include this IE over S11 interface.	TAD	0
	CO	If S4-SGSN receives this IE from the UE, it shall include it over S4 interface.		
	CO	If SGW receives this IE, the SGW shall forward it to PGW over S5/S8 interface.		
RAT Type	C	This IE shall be included for MS initiated PDP Context modification procedure and Secondary PDP context activation procedure.	RAT Type	0
Serving Network	O	This IE may be included in the MS initiated PDP Context modification procedure.	Serving Network	0
User Location Information (ULI)	O	This IE may be included in the MS initiated PDP Context modification procedure.	ULI	0
EPS Bearer ID	C	This IE indicates the EPS Bearer that needs to be modified. It shall be included for MS initiated PDP Context modification procedure. For EUTRAN this IE shall be present if it is triggered by the NAS Bearer Resource Modification Request message and its value shall be set to the value of the "EPS bearer identity for packet filter" IE received in that NAS message.	EBI	1
Indication Flags	O	This IE shall be included if any one of the applicable flags is set to 1. Applicable flags: - Change Reporting Support Indication: this flag may be included in the MS initiated PDP Context modification procedure. - Direct Tunnel Flag: this flag may be included in the MS initiated PDP Context Modification procedure.	Indication	0
S4-U SGSN F-TEID	C	This IE shall be included on the S4 interface when direct tunnel is not established in the MS initiated PDP Context modification procedure	F-TEID	0
S12 RNC F-TEID	C	This IE shall be included on the S4 interface when direct tunnel flag is set to 1 in the MS initiated PDP Context modification procedure.	F-TEID	1
Protocol Configuration Options (PCO)	O		PCO	0
Signalling Priority Indication	CO	The SGSN/MME shall include this IE on the S4/S11 interface if the UE indicates low access priority during the procedure. The SGW shall forward this IE on the S5/S8 interfaces if received from the MME/SGSN.	Signalling Priority Indication	0
Private Extension	O		Private Extension	VS

Figure 71: Bearer Resource Command message parameters [99]

In case of a failure (which may have been received from PGW), SGW will answer with a Bearer Resource Indication message to the MME which will then send a "RESOURCE ALLOCATION REJECT" message to e-NB. PTI value may be used to define which is the exact procedure that the failure message is coming for. If UE receives such a message it should stop the timer and release traffic flow aggregate description associated to this PTI value.

If there is no rejection message, it could be supposed that the dedicated bearer between UE and Network has been established. Then the procedure will continue by SGW indicating to the MME if it will send Activation or modification message to the UE. The UE will handle the response by stopping the timer T3480 and continuing the procedure. Part of the continuation is paging the end users on the other side of the line in order to establish the call connection. The current dissertation will focus on this part of the whole procedure since the main objective of presenting this case study is to show how the presented methodology can be used to design a procedure to be survivable.

What should also be underlined here is that as part of a development and testing team of telecommunication networks over that past 6 years, my experience is that requirements specification of this procedure will end by implementation of these requirements defined by 3GPP and presented above. So, the contribution of the current research is the definition of possible threats to survivability and the enrichment of requirements specification with best practices that will result in a secure, robust and survivable system.

5.2. Application of methodology proposed on Voice Bearer Establishment.

The following chapters will describe the methodology proposed through a case study which is the voice bearer establishment. Voice bearer establishment is considered by 3GPP as a critical service as it has already been described by the general requirements specification that the designed system should comply with. For more details, please refer to table 3.

5.2.1. Requirements Specification of Voice Bearer Establishment.

The first step of the SDLC that will be analysed is the requirements specification. Here the focus will be paid on depicting survivability requirements and on compliance with requirements defined by the prototype (3GPP standards).

Survivability Requirements.

Survivability requirements defined by 3GPP and requirements that are result of the current research may be depicted by table below:

Service Level Requirements:

- A dedicated voice bearer is a Guaranteed Bit Rate (GBR) bearer which means that the bit rate provided should be as requested. Any degradation of the guaranteed bit rate

levels shall be recorded in order suitable actions to be performed. Degradation levels of this GBR should be defined in accordance with user experience of this degradation in order acceptable quality of service levels to be defined. Survivability measurements should ensure that service will not fall under these QoS levels. Number in front of any requirement is the related 3GPP requirement defined in table 3.

Failure Recognition Requirements	
Rule Number	**Description of Requirement**
(1) Bearer delay variation tolerance, maximum transfer delay, maximum bit error rate **(16) Services and Systems Aspects;**	A dedicated voice bearer is a Guaranteed Bit Rate (GBR) bearer which means that the bit rate provided should be as requested. Any degradation of the guaranteed bit rate levels shall be reported in order suitable actions to be performed. This is part of SGW node and not MME though. MME must ensure that the bearer has been established with the expected Guaranteed Bit Rate (GBR). Main objective of voice dedicated bearer is described by number 16 requirement: The voice call continuity user's experience shall be such that, to the greatest degree possible, a consistency of service is provided regardless of the underlying communication infrastructure and technology
(2) Performance Management	Any delay in call set-up or degradation of acceptable Guaranteed Bit Rate (GBR) levels shall be reported. General metrics related to performance management may be also reported. If these are available network management system could investigate the relation of this failure with performance issues. So, any time MME detects a call set-up delay should report the event and include performance metrics as well.
(3) Error cases	Any of error indications shall be reported to the network management system in order statistics to be gathered and possible mitigation actions to be performed. Key Performance Indicators shall also be reported as expected. Among KPIs that should be reported are: - Procedure Attempt - Procedure Success - Procedure Failure - Procedure Failure with specific cause, information regarding the neighboring node - Procedure Collision scenario.
Failure Resistance Requirements	
Rule Number	**Description of Requirement**

(4, 5) Handover Requirements	After dedicated bearer setup what should be tested is scenarios related to the handover requirements: - Handover should be transparent. In case of speech call loss of information may be tolerated but handover should be quick to avoid connection break. - Handover should be transparent. In case of speech call loss of information may be tolerated but handover should be quick to avoid connection break. - Handover to an alternative bearer offering reduced capabilities should be possible where this is supported by the service in use
(6, 7, 8, 9) Security objectives (10 – for 5G systems)	Dedicated bearer should not be established if the UE requesting it is not authenticated through authentication procedure. A specific error cause value "User authentication failed" will be returned from SGW in this case. An attach procedure should be triggered by the network in order to re-authenticate the UE.
(11) Error Handling (3GPP defined error causes)	Error cases will be investigated for messages that arrive at MME and messages that are sent by MME and are part of the dedicated voice bearer establishment procedure. 1. Error cases NAS Message **Bearer Resource Allocation Request** (eNB -> MME) defined by 3GPP. The ESM cause value typically indicates one of the following: #26: insufficient resources; (network decides that there are no sufficient resource to serve the UE's request) #30: request rejected by Serving GW or PDN GW; #31: request rejected, unspecified;(ex. in case request is for an Emergency PDN connection) #32: service option not supported; #33: requested service option not subscribed; #34: service option temporarily out of order; #35: PTI already in use; #37: EPS QoS not accepted; #41: semantic error in the TFT operation; (message sent by **MME** in case TFT operation is other than "Create a new TFT") #42: syntactical error in the TFT operation; (If TFT operation is "Create a new TFT" but packet filter list is empty) #43: invalid EPS bearer identity; - (sent by MME - the ue should deactivate existing default bearer. #44: semantic error(s) in packet filter(s); (packet filter consists of conflicting packet filter components) #45: syntactical error(s) in packet filter(s) (two or more packet filters among all TFTs associated with the PDN connection would have identical packet filter precedence values); #56: **collision** with network-initiated request; (when request for a

<table>
<tr><td></td><td>dedicated bearer establishment comes at the same time that the UE requested for a dedicated bearer establishment. Priority to nt initiated procedure). Drop call should not be reported.
#59: unsupported QCI value;
#60: bearer handling not supported; (if the request is for an already established LIPA PDN Connection – sent by the network)
#95 – 111: protocol errors.

1. Error cases GTPV2 Message Bearer Resource Allocation Command (MME -> SGW) defined by 3GPP. The ESM cause value typically indicates one of the following:
- "User authentication failed".
- "Semantic error in the TAD operation".
- "Syntactic error in the TAD operation".
- "Semantic errors in packet filter(s)".
- "Syntactic errors in packet filter(s)".
- "Collision with network-initiated request".
- "Service denied".
- "Bearer handling not supported".

MME should also have an error matching mechanism in order to translate error cause received from SGW to the error cause that it will be sent to e-NB. (some matching has already been performed – marked with same colors)</td></tr>
<tr><td>(13) Error Handling other than 3GPP documents</td><td>MME could be robust to the following errors in message Bearer Resource Allocation Request received:

<table>
<tr><th>Information Element</th><th></th></tr>
<tr><td>Protocol discriminator</td><td>(M) Protocol being used</td></tr>
<tr><td>EPS bearer identity</td><td>(M) EPS bearer identity: EPS bearer identity is used to identify a message flow</td></tr>
<tr><td>Procedure transaction identity</td><td>(M) procedure transaction identity (PTI) is an identity which is allocated by the user equipment for the UE-requested bearer resource activation</td></tr>
<tr><td>Bearer resource allocation request message identity</td><td>(M) Message type - identifies that this is a bearer activation procedure</td></tr>
<tr><td>Linked EPS bearer identity</td><td>(M) Default bearer that dedicated bearer that will be established with connect to.</td></tr>
<tr><td>Spare half octet</td><td>(M) Filled with spare bits for description of EMM/ESM messages</td></tr>
<tr><td>Traffic flow aggregate</td><td>(M) The packet filters determine the traffic mapping to EPS bearer contexts. Uplink packet filters shall be applied by the UE (3GPP 23401) UE shall set this to "Create new TFT" since there is no active bearer yet.</td></tr>
<tr><td>Required traffic flow QoS</td><td>(M) EPS quality of service.
The UE should define a required QCI (QoS Class Identifier) which is 1 in out case and GBR required</td></tr>
<tr><td>Protocol configuration options</td><td>(O) Included in the message when the UE wishes to transmit (protocol) data</td></tr>
<tr><td>Device properties</td><td>(O) include this IE if the UE is configured for NAS signaling low priority</td></tr>
</table>

1. Protocol discriminator – 3GPP defines that in case of receiving a non-expected Protocol discriminator, then message shall be ignored by MME. Though, for survivability reasons, MME could suppose that it has received correct Protocol discriminator value and continue the flow, since the values are fixed and related to message received. The flow may fail in the following messages for another reason.</td></tr>
</table>

	2. EPS Bearer Identity for the dedicated bearer is provided by MME so, the value of this IE should not be taken into consideration for this message. (No wrong value here) 3. PTI value is UE defined to track messages of the same Procedure requested by the UE. It is very important from MME side to include the right value for PTI 4. Message type (and spare half space that may be used for message type identification) should also have a default value. So, if QCI requested from UE is 1 then MME may imply what should be the correct value for message type and not immediately reject the message for having wrong value in this IE. "For EPS; the default value for bit 7 is 1. The value for bit 8 is 0 for the EMM protocol and 1 for the ESM protocol." [100] 5. Linked EPS Bearer Id is one of the default bearers of the UE. If there is only one default bearer available, then this MME could tolerate a wrong value since it may know the default bearer id for this UE from other resources. 6. Traffic Flow aggregate is again a fixed value that the MME may suppose that the value is "Create new TFT" instead of ignoring the message as indicated by 3GPP. QCI value should be set to 1 and this cannot be omitted by the UE because it is the service that is requested. The rest IEs have optional values so no need to apply any robustness measures. Any of the above failures should be reported to the network management system in order relevant metrics to be analyzed and handled appropriately. For example, if MME is constantly receiving messages with wrong "Linked EPS Bearer Id" from a particular UE (or from a number of UEs that may be defined by a specific pattern – ex. constructed by the same company), it may suppose that this certain UE is constantly sending a wrong value and may contact the responsible team to fix it (maybe this means opening a ticket another company that constructs the UEs). In case all messages arrive with this error, then the problem might be in the MME. For example, during message decoding. 1. MME should be able to detect an error to its own implementation in cases of receiving an error as an answer from Bearer Resource Command message sent by SGW. If MME is constantly receiving a specific error message that leads to service failure from that particular SGW, it should be reported to the network management system in order to be investigated further. In case this is happening with all or more than one SGWs

	that MME is connected to, then MME should report that also since the error may be in the message sent by MME itself. 2. Message replay. If MME receives a message Bearer Resource Allocation request twice, then there are two possible root causes. The first is that UE's timer waiting for the next message has been expired. Then it is normal for UE to re-send the message. The second is that an issue of the network between MME and UE could have caused this by faulty replaying the message. Finally, UE may have an issue it is resending the message before the timer started on it expires. In any case when there is an on-going process of bearer resource allocation, then MME should ignore the message replay and continue with the ongoing process. Then MME should inform the network management system about possible error of the UE. MME would know if the timer expiration is reasonable by advising its own timers since MME is also waiting for messages from the SGW, or by starting a dummy timer when receiving the message from the UE. In this case, only an estimation of when the timer will be expired in UE is possible since the delay of the message to actually reach MME should be also considered. In case there is no ongoing process then MME will initiate the process again.
(12) Collision handling	For handling collision cases, MME shall keep a matrix with process priorities in case two request messages arrive at the same time. The handling of the collision may be: • Dropping the ongoing process and continue the incoming process • Dropping the incoming and continue the ongoing process • Keep ongoing on hold continue incoming and then complete the ongoing • Keep incoming on hold, continue the ongoing and then complete the incoming. • Continue both procedures in parallel. Priorities of the procedures should depend on two factors: 1. The criticality of the service. For example, critical services shall not be dropped. 2. If one of the two services should be completed first as a prerequisite for the other. Finally, analysis of requirements specification team should be based on logical representation of the scenario in order to decide which handling is the best to keep in any case.

	During the current dissertation it is not possible to list all possible combinations of collision scenarios that could happen and the handling from MME side. Though some possible cases that should be considered will be included as part of requirements specification. In our example, voice bearer is a critical service. So, it should have a bigger priority in cases of collision. For example, in case of another dedicated bearer establishment (possibly not a voice bearer), UE initiated dedicated bearer request with QCI 1 should have priority. Though, there are some cases that it should be dropped. For example: 1. While dedicated voice bearer establishment is ongoing, UE initiates a detach procedure or a default bearer deletion procedure of the bearer that ongoing dedicated bearer that is to be established, will be linked to. 2. Any request for emergency voice bearer establishment. 3. As it is indicated by 3GPP in case of network initiated dedicated bearer request procedure comes, then UE initiated should be dropped. It is very important here to pinpoint that 3GPP has only defined this single resolution and all the other combinations should be derived by searching through 3GPP documents or would be implementation specific. With implementation specific handling, the problem is that in case the network devices are not implemented from the same organization, there is no standard way for collision handling, and this would lead to many service failures. So, one of the contributions of the current dissertation is showing how important is any possible failure scenario to be documented, standardized and taken into consideration during requirements specification. Additional to these cases, some scenarios could be resolved by holding the dedicated bearer initiation procedure, continue the incoming procedure and then continue the dedicated bearer initiation procedure. For example, if a handover request comes at the same time as the voice bearer request, then the second procedure could be on hold. In this case, new TEIDs will have been exchanged during HO and the voice bearer will be successfully established with the correct addresses for the two communication end points. If the UE has left MME before bearer resource command, then HO procedure shall have priority against voice bearer procedure since UE is moving, and communication will be lost otherwise.

	Apart from these cases, there are cases that both procedures could continue. These are the cases that ongoing process would not be affected in any case. For example, a simple Tracking Area Update procedure could continue in parallel since it does not affect the UE initiated voice bearer establishment procedure. This is how the requirements for collision scenarios should be specified in general.
(14) Hanging Processes	During the requirements analysis what should be depicted is which processes if left hanging would affect the dedicated bearer initialization procedure. This should be also part of threat analysis during design phase. Some examples of the procedures that could cause failure to the dedicated bearer process if left hanging would be: - Default bearer establishment. If this procedure was left in the middle, then the UE may suppose that there is a default bearer, and the network side may had failed to establish it. Since there is no physical experience in default bearer creation, (as for example in voice bearer that a human interacts with the procedure), the dedicated bearer establishment may fail. There is an error code for this example defined by 3GPP. - Default bearer deletion. The same as before may happen for default bearer deletion. If UE sends such a request, then it considers the bearer as non-existing. On the other hand, the network may have not deleted the bearer and may consider it as established. Though this may be a threat for default bearer establishment or for network initiated dedicated bearer establishment. - Dedicated bearer establishment. If an old, dedicated bearer has not successfully been deleted, then the network may suppose that the bearer is still established and deny establishing a new dedicated bearer. In this way a call may fail to be established. - Hanging bearers could also be result of restoration procedures. MME should have a mechanism established that is executed periodically and recognizes and releases hanging bearers. Though this mechanism may fail since another message request may arrive in the middle and being rejected. So, it is important to extend this mechanism by MME maintaining a timer that would indicate when the procedure shall be released assuming it is left as "hanging". This timer should be equal to the time the message Bearer Resource

	Allocation arrived plus timer the UE is waiting for an answer until it resends Bearer Resource Allocation Request message. This should be part of any procedure that other part is waiting for a message by the node under development.
Failure Recovery Requirements	
Rule Number	**Description of Requirement**
(15) Restoration procedures	The current dissertation will not extend to provide network related survivability controls since it is supposed that these controls are already designed and working as expected. **MME restart:** Though as indicated by 3GPP [57] in case of Mobile originated Service request: *"For service request, where the UE is unknown in the MME (i.e., the MME has no MM context for the UE), the MME shall reject the service request with an appropriate cause. In order to remain attached, the UE shall then perform a new attach and should (re-) activate its EPS Bearer contexts."* [57] The same could be performed in case of UE initiated dedicated bearer request. MME could reply with an error that would indicate release of the current default bearer and re-establishment (or re-attach) in order the call to be established. **SGW Failure** Additionally, in case SGW has failed and MME receives an indication for that, then MME shall drop the UE initiated dedicated bearer request procedure and follow restoration procedures indicated in 3GPP [57].
(17) Flexibility and dynamic reconfiguration;	This control is part of end-to-end dedicated bearer establishment process. In this part of the process, the UE asks for bearer services attributes and it is part of SGW and PCRF nodes to provide to UE the closest values of these attributes possible, based on system availability and UEs charged features available. In case of emergency call though, the connection shall be established in any case.
(18) SON	In case MME realizes that there are constant failures in current procedure, then recovery actions as described in table 3 "Service Level Survivability Requirements" shall be performed. The ideal situation would be MME to be able to reload a backup of the procedure code in case this is compliant with modifications of the latest code in other nodes. It should be marked at the network management system of MME if it is ok to reload a backup system and it will not result to service failures. This could be analyzed during analysis phase and possible failures could be marked during

	threat modelling.

Table 8: Voice Bearer Survivability Requirements.

As it has already been described, network level requirements will not be analysed during dedicated bearer establishment feature, since it is supposed that they are already implemented and working. In case something is not implemented, it should be considered as another feature. Though, during dedicated bearer establishment procedure, MME may depict and should report to the network management system, delays in response messages by SGW, failed messages received by UE or SGW which should be reported to the corresponding node in order to be fixed or finally, failure of SGW system in case SGW is not answering to messages sent by MME. MME shall compare the responds of other nodes to other procedures to depict such an error as it has already been described in the proposal of failure recognition (4.1.1).

Compliance Analysis.

The whole process of requirements specification shall be considered part of compliance analysis since requirements are part of 3GPP standard.

Though, cases that are not described by 3GPP standards or are implementation specific should be marked here. The risk of specific implementation should be calculated and decisions of what is the most suitable solution to continue with, shall be taken.

Survivability Risk Analysis.

As it has already been defined, tool chosen to conduct **risk analysis** is failure mode and effects analysis FMEA. FMEA will be used as **risk** analysis tool based on failure, since there is an already implemented system up to some point, and the process that is to be developed is already defined in 3GPP. So, there is a need to specify which are the potential failures from the already standardised process, how this process (which is a critical one) may be threatened from the already implemented processes that are executed at the same time, and what are the potential failures that the to be developed process may cause to the already implemented system.

Additionally, to FME, a root cause analyses method could be used to define root cause of failure. For the current dissertation this will be skipped. Any **design principles** related to survivability that should be followed are part of action recommended.

Process Step	Potential Failure Mode	Potential Failure Effect	SEV	Potential Causes	OCC	Current Process Controls	DET	RPN	Action Recommended
	Possible failure	What is the impact of failure	Severity on scale of 1 to 10	Possible Root cause of failure	How frequently is the cause likely to occur?	Existing controls that prevent / detect the failure.	How probable is detection of the failure ?	Risk priority number (SEV*OCC*DET)	Actions to reduce occurrence of the cause or improve detection
MME receives Resource Allocation Request	message arrives with wrong **protocol discriminator.** Other than "EPS mobility management messages"	MME will discard the message due to 3GPP 24.007 [11] .	8	There might be a S./W error on UE or message integrity failure due to connection path issues.	1*	None proposed by 3GPP. The proposal for detection of failure in order the central management system to be informed will be chosen.	10	80	The system that will be developed will ignore this IE since this is a NAS message and all NAS messages need to have a fixed value. Though in case this value is different, the system will report the error to the network management entity.
	message arrives with wrong **EPS Bearer Identity**	No impact since MME will decide which will be the value for this field and will return it to the UE	1	There might be a S./W error on UE or message integrity failure due to connection path issues.	1	None. No impact of failure.	10	10	No action needed.
	message arrives with wrong **procedure transaction identity**	Due to 3GPP 24301[7.3.1] the network shall respond with a BEARER RESOURCE ALLOCATION REJECT message including ESM cause #81 "invalid PTI value"	8	There might be a S./W error on UE or message integrity failure due to connection path issues.	5	None proposed by 3GPP. The proposal for detection of failure in order the central management system to be informed will be chosen.	8 (MME needs to cross check if the PTI is used by other processes.)	320	Reporting of message failure should be performed. Failed value could not be ignored because MME is not able to suppose which should be the correct value that UE should send.
	message arrives with wrong **request message identity**	Due to 3GPP 24301[7.4] network react in such a failure is implementation dependent. Though, MME may respond with STATUS message and cause #97 "message type non-existent or not implemented and discard the request.	5	There might be a S./W error on UE or message integrity failure due to connection path issues.	5	None proposed by 3GPP. The proposal for detection of failure in order the central management system to be informed will be chosen.	3 (MME should check if QCI requested = 1 and message IEs, in order to imply that this is a resource allocation request messa	75	MME should check if QCI requested = 1 and message IEs, in order to imply that this is a resource allocation request message. Reporting of message failure should be performed.

							ge else there is no easy way to know what the messa ge since this IE is a faulty one)		
	message arrives with wrong **linked eps bearer identity**	Due to 3GPP 24301[6.5.3.5] MME will reject the request with cause #43 "invalid EPS bearer identity". Then the UE shall deactivate existing default EPS bearer context locally.	10 (oth er reso urce s may also be relea sed)	There might be a S./W error on UE or message integrity failure due to connection path issues Additionally, here seems that UE requests a bearer that MME does not know it exists. MME may have released the default bearer, or it is left hanging (not properly released).	3	None proposed by 3GPP. The proposal for detection of failure in order the central managemen t system to be informed will be chosen.	10	300	MME should check if other requests with this linked bearer ID have been served and if it received any similar message with wrong linked bearer id. Additionally, it should check if any bearer deletion process took place recently. If there is only one default bearer available, and MMME has evidence that this bearer is active, then MME could tolerate a wrong value since it may know the default bearer id for this UE from other resources and try to establish the bearer.
	message arrives with wrong **traffic flow aggregate**	**Due to 3GPP 24301 Message with wrong traffic flow aggregate shall be ignored or rejected with cause #41 or #42**	8	There might be a S./W error on UE or message integrity failure due to connection path issues	3	None proposed by 3GPP. The proposal for detection of failure in order the central managemen t system to be informed will be chosen.	10	240	Traffic Flow aggregate is again a fixed value that the MME may suppose that the value is "Create new TFT" instead of ignoring the message as indicated by 3GPP. Detection of error and informing NM entity should follow.
	message arrives with wrong **required traffic flow QoS**	**Due to 3GPP 24301 Message with wrong traffic flow QoS shall be rejected with cause: #37: EPS QoS not accepted;**	8	There might be a S./W error on UE or message integrity failure due to connection path issues	3	None proposed by 3GPP. The proposal for detection of failure in order the central managemen t system to be informed will be chosen.	5 in case UE reques ts a failing but accept able an accept able QoS, MME is not able to detect the issue. The issue	120	In case UE requests a failing but acceptable an acceptable QoS, MME is not able to detect the issue. The issue may be detected only in cases that the value is wrong. In this case, there is no way MME to assume what UE may requested but it may assume that the default value was requested and continue the flow with this value.

							may be detected only in cases that the value is wrong.		
	message arrives with wrong **protocol configuration options**	No impact since this is an optional IE.	1	There might be a S./W error on UE or message integrity failure due to connection path issues	1	None proposed by 3GPP. The proposal for detection of failure in order the central management system to be informed will be chosen.	10	10	No actions since there is no impact
	message arrives with wrong **device properties**	No impact since this is an optional IE.	1	There might be a S./W error on UE or message integrity failure due to connection path issues	1	None proposed by 3GPP. The proposal for detection of failure in order the central management system to be informed will be chosen.	10	10	No actions since there is no impact
	message **arrives twice**	Message arrives twice at MME	3 (there is a mechanism to ignore duplicate messages. Though, there is a possibility that this does not work)	There might be a S./W error on UE or network issue.	3	There is a mechanism to ignore duplicate messages. This message shall be added to this logic.	8	72	This message shall be added to this logic of rejecting duplicate messages already implemented at MME.
Bearer resource allocation request mesage	Message receives causing integrity/ciphering failure	Due to 3GPP 24301 MME shall discard this message leading in failure of the procedure.	10	Connection issues or security attack may cause this failure.	1	MME shall discard this message leading in failure of the procedure.	10	100	The issue shall be reported to the Network Management Entity of MME for further analysis
	message **delays** to reach MME.	In this case, timer of UE may expire, and UE may re-send the	3	There might be a network issue, CPU or memory load issue or an increased traffic	3	There is a mechanism to ignore duplicate messages.	8	72	This message shall be added to this logic of rejecting duplicate messages already

MME receives Resource Allocation Request		message. This will be the above case,		load issue.		This message shall be added to this logic.			implemented at MME.
	Collision with network-initiated request	The call will be dropped	5	No root cause that could be eliminated	5	3GPP indicates that network-initiated procedure has priority to call shall be dropped with error cause #56.	8	400	There is no resistance mechanism that could be performed. Detection of the collision should be reported.
	Any other collision scenario	The call may be dropped or not depending of the procedure	3 (the procedure is considered as critical and should have priority)	No root cause that could be eliminated	5	Collision matrix used by MME to resolve such cases.	8	120	Collision matrix shall be updated with all possible scenarios. Any collision instance shall be reported to NM Element.
Message Bearer Resource Command	Error in any message IEs	Possible rejection message from SGW.	5	S/W Error in MME Code, network integrity issues	5	3GPP has defined specific error codes that may be returned to the MME as a rejection message	5	125	MME shall investigate the root cause of failure as described in chapter 3.1 and prepare a report for the NM Element. W. Maybe a concept of correcting some parameters if possible, could be realized. For example, if MME receives error "Semantic error in the TAD operation" it may re-check the message sent and correct any information if possible and resend the message.
	Message is not sent by MME	In this case, timer of UE may expire, and UE may re-send the message. This will be the above case,	3	There might be a network issue, CPU or memory load issue or an increased traffic load issue.	3	There is a mechanism to ignore duplicate messages. This message shall be added to this logic.	8	72	This message shall be added to this logic of rejecting duplicate messages already implemented at MME.
	Message Bearer Resource Command is "not answered" In this case, SGW does not answer in any case. Though	If SGW did not receive the message because of failure, it means that his procedure and all other connections of eNBs with this SGW	10	SGW failure	3	Restoration procedure as indicated by 3GPP indicates if a failure is recognised in SGW all related connections shall be released.	8	240	MME shall implement a detection mechanism by waiting the next procedure in order to be sure that message has arrived and SGW is not down. Additionally, MME shall consult its internal counters

	MME could wait for the next procedure to start in order to assume that everything is ok. There is no other way MME knows that something is wrong with SGW but in case it is informed by other process or by counter included in echo message sent by SGW, MME shall stop this procedure.	will fail. The failure shall be tracked and reported to network administration in order to be fixed.							related to failed messages by SGW to be sure that it should raise an alarm that SGW possibly has failed (Described in 3.1.2)
	All collision scenarios of message Bearer Resource Allocation Command shall be checked.	The call may be dropped or not depending of the procedure	3 (the procedure is considered as critical and should have priority)	No root cause that could be eliminated	5	Collision matrix used by MME to resolve such cases.	8	120	Collision matrix shall be updated with all possible scenarios. Any collision instance shall be reported to NM Element.
Bearer Resource allocation procedure as a threat for other processes	The first case is bearer resource allocation procedure to be incoming procedure in a collision scenario	Other procedures may be dropped	5	No root cause that could be eliminated	5	Collision matrix used by MME to resolve such cases.	8	200	Collision matrix shall be updated with all possible scenarios. Any collision instance shall be reported to NM Element.
	bearer resource allocation procedure left hanging.	Other requests may be rejected by MME	10	S/W Error in MME Code	2	None.	2	40	MME shall check for hanging procedures and release any resources kept for this dedicated bearer.

Table 9: FMEA Table.

* 1: failure is marked as unlike to occur since usually this IE is derived from a central point and the failure would be general, which means that more than one message would fail.

The actual design requirements should be dedicated to the source code indicating were exactly the new code should be added. Since this is a case study, this is out of scope for the current dissertation since no source code is available.

After the above analysis the risk priority number (RPN) indicates the areas of focus for the development team. The riskiest failures regarding survivability are marked with red colour.

5.2.2. Design Phase.

Part of design phase is threat modelling and design survivability requirements. Threat modelling that will be executed for security reasons shall be enriched with threats depicted by risk analysis of the previous step. Development team shall work together to depict the exact points in the code that should be changed.

5.2.3. Implementation Phase.

During implementation phase, the actual feature implementation takes place, with respect to requirements and design principles.

Survivability Code Best Practices.

Survivability Code Best Practices should have already been decided until now since the product is already implemented up to some point. Though, the current dissertation proposes that the ideal architecture could be the use of a service-oriented architecture like microservices architecture. In this way each service is a discrete unit that can be updated – restored independently without need of updating the whole node. So, any time MME or any other node detects a failure to bearer resource allocation procedure, the bug may be fixed, and the service may be updated without affecting the rest of the services. Additionally, returning to previous software versions for this particular service is easier until the issue is fixed.

Unit Testing.

This testing level will be described during testing phase that follows. Any test in module level that would test the source code in function level should be included to ensure the correct functionality of the code.

4.2.4. Testing Phase

During testing phase, a test plan will be described, and testing levels will be reported. In any of test cases, recognition of failure shall be also tested as already described in FMEA

Test case Description	Unit	Functional	System	Integration
Message Bearer Resource Request arrives with error in protocol discriminator	x	x		
Message Bearer Resource Request arrives with error in EPS Bearer Identity	x	x		
Message Bearer Resource Request arrives with error in procedure transaction identity	x	x		
Message Bearer Resource Request arrives with error in request message identity	x	x		
Message Bearer Resource Request arrives with error in linked eps bearer identity – non existing error identity	x	x		
Message Bearer Resource Request arrives with error in traffic flow aggregate	x	x		
Message Bearer Resource Request arrives with error in required traffic flow QoS	x	x		
Message Bearer Resource Request arrives with error in protocol configuration options	x	x		
Message Bearer Resource Request arrives with error in device properties	x	x		
Message Bearer Resource Request arrives twice	x	x		
Test all possible rejection messages that could arrive from SGW and test if the procedure stops when it should without leaving hanging procedures. A collision scenario could be used to test hanging procedure.	x	x		
All collision scenarios of message Bearer Resource Allocation Request shall be tested.	x	x		x
All collision scenarios of message Bearer Resource Command shall be tested.	x	x		x
Leave bearer resource allocation procedure as "hanging" (active even after it has finished) and test the behaviour of MME for all critical services. Any unwanted behaviour should be marked as bug and reported as requirement.	x	x		x
Send message bearer resource allocation with integrity /ciphering failure.		x		
Test all possibilities of Handover Scenarios. HO arriving before and after dedicated bearer establishment.		x		
Teesta all critical services before and after the dedicated voice bearer resource service in order to identify possible hanging procedures that would cause voice bearer failure.		x		
SGW has restarted, or failed, and MME receives a restart counter in GTPv2 Echo message.		x		
SGW has failed and MME recognises the failure if failure of messages received from that SGW is increased MME shall send a report to Network Management entity about the possible failure.				
MME has restarted and does not recognise UE requesting for a dedicated bearer,		x		
Test as many of the above scenarios as possible with real nodes (UE, Enb, SGW).			x	x
Load network and see if MME may handle the message (CPU / Memory of MME shall be loaded with requests) (critical message has priority)			x	
Overload the network and see if MME can handle the message (critical message has priority)			x	
Stress MME with failure Bearer resource allocation messages from the UE. Analyse the impact to MME and other critical services.			x	

Table 10: Test plan for survivability requirements of Voice Bearer establishment.

Some requirements could not be specified without the procedure been implemented. So, after implementation of procedure and by using specific test cases, new requirements may arrive. This is something often in cases of complicated systems, using the test cases to test the systems behaviour and take decision about the required-correct behaviour afterwards. So, after first release of the procedure, testing phase with all possible combinations that are not standardized yet shall follow. In the next iteration of the SDLC, these requirements will be integrated to system design. An example is the test case: "Test all possibilities of Handover Scenarios. HO arriving before and after dedicated bearer establishment."

5.2.4. Release Phase

After all iterations and end to end system testing, the new version of the product is ready for release. During this phase, an incidence response plan should be prepared. The current dissertation will not concentrate on this since no special technology has been defined for the already implemented up to some point product and there is no team to define roles and responsibilities.

5.2.5. Maintenance Phase

As is has already been described, maintenance is based on system monitoring and analysis of several key performance indicators (KPIs) that indicate the existence of a possible error and may trigger the team to search for a bug. If a bug in the code is depicted, the bug is reported and a new SDLC starts to fix it.

Maintenance phase may be performed periodically, after several non-critical bugs have been gathered, to trigger one new cycle of software development. Or it may be performed as soon as any critical bug arises to the system.

KPIs that may be used for the current scenario may be:

1. Bearer resource allocation procedure attempt
2. Bearer resource allocation procedure success.
3. Bearer resource allocation procedure failure.
4. Bearer resource allocation procedure failed by rejection to bearer resource allocation reject. Each error code should have a different KPI.
5. Bearer resource allocation procedure failed by rejection to bearer resource allocation command message. Each error code should have a different KPI.
6. Bearer resource allocation procedure failed because timer T3480 of UE has expired.
7. SGW Failure indicator should be increased in case MME's alert indicating SGW failure is correct.

6. Results and analysis of proposed methodology.

The current dissertation is focussed on proposing a software development lifecycle for building a survivable mobile telecommunication system. Being part of an R&D team for developing and testing mobile telecommunication systems, I discovered that the main focus for handling system failure was paid on 3GPP. Anything out of scope of the standard was not really considered, including those solutions marked by the standard as "implementation specific". So, after investigation of current implemented survivability measurements defined by 3GPP, the current research focussed on extending the system's survivability by covering as many as failure scenarios as possible.

A quantitative and qualitative analysis on the case study examined through the current research follows in order to examine if and how the current research has contributed to improving developed system's survivability.

For the analysis, failure scenarios described during FMEA will be used as failure cases that the system may experience. The data that are presented are hypothetical since there are no real data available. Though, they are based on my 6 years working experience in R&D team of LTE telecommunication system. The analysis will be based on comparing traditional SDLC based mostly on 3GPP standard, with the proposed SDLC which is based on survivability requirements. The analysis will be quantitative and qualitative.

Quantitative Analysis:

The first table that follows shows the scores of the two SDLCs on recognition of failure, resistance to failure and recovery from failure, which are the key properties of survivability. Additionally, the table depicts the actual failures. The sample examined is 10 failures that could happen on communication between UE, MME and SGW.

	Failure Scenario	**SDLC based on 3GPP**				**SDLC proposed.**			
#	**Suppose that each failure arrives 10 times.**	**Recognition**	**Resistance**	**Recovery**	**Failed**	**Recognition**	**Resistance**	**Recovery**	**Failed**
1	Message Bearer Resource Request arrives with error in protocol discriminator	**10**	**0**	**0**	**10**	**10**	**8**	**0**	**2**
2	Message Bearer Resource Request	**10**	**0**	**0**	**0**	**10**	**0**	**0**	**0**

	arrives with error in EPS Bearer Identity								
3	Message Bearer Resource Request arrives with error in procedure transaction identity	**10**	**0**	**0**	**10**	**10**	**0**	**0**	**10**
4	Message Bearer Resource Request arrives with error in request message identity	**0**	**0**	**0**	**10**	**3**	**2**	**0**	**8**
5	Message Bearer Resource Request arrives with error in linked eps bearer identity – non existing error identity	**10**	**0**	**0**	**10**	**10**	**2**	**0**	**8**
6	Message Bearer Resource Request arrives with error in traffic flow aggregate	**10**	**0**	**0**	**10**	**10**	**4**	**0**	**6**
7	Message Bearer Resource Request arrives with error in required traffic flow QoS	**10**	**0**	**0**	**10**	**10**	**6**	**0**	**4**
8	Message Bearer Resource Request arrives with error in protocol configuration options	**10**	**0**	**0**	**0**	**10**	**0**	**0**	**0**
9	Message Bearer Resource Request arrives with error in device properties	**10**	**0**	**0**	**0**	**10**	**0**	**0**	**0**

10	Message Bearer Resource Request arrives twice	**10**	**0**	**0**	**0**	**10**	**0**	**0**	**0**
11	All possible rejection messages that could arrive from SGW.	**10**	**0**	**0**	**10**	**10**	**0**	**1**	**9**
12	All collision scenarios of message Bearer Resource Allocation Request shall be tested.	**10**	**4**	**0**	**6**	**10**	**6**	**0**	**4**
13	All collision scenarios of message Bearer Resource Command shall be tested.	**10**	**4**	**0**	**6**	**10**	**8**	**0**	**2**
14	Bearer resource allocation procedure is left as "hanging" (active even after it has finished)..	**0**	**0**	**0**	**10**	**5**	**5**	**0**	**5**
15	Send message bearer resource allocation with integrity /ciphering failure.	**10**	**0**	**0**	**10**	**10**	**0**	**0**	**10**
16	SGW has restarted, or failed, and MME receives a restart counter in GTPv2 Echo message.	**10**	**0**	**0**	**10**	**10**	**0**	**2**	**8**
17	Failure to SGW is depited since the next procedure that will complete Dedicated Bearer Request never starts. It is the same with SGW not answering	**0**	**0**	**0**	**10**	**10**	**0**	**4**	**6**

	case.								

Table 11: SDLC proposed process capability compared to existing SDLC.

Some Comments for each failure:

- **Failure 1: Message Bearer Resource Request arrives with error in protocol discriminator.**
  The score of resistance to failure, is marked with a high value (8/10) since value is fixed and the probability the rest of the flow to succeed if MME supposes that it has received it correctly, is large. On the other hand, 3GPP indicates that the procedure should be rejected with a special cause.

- **Failure 2: Message Bearer Resource Request arrives with error in EPS Bearer Identity.**
  This is a case that MME should ignore any error since it is MME that will decide for this value after the flow has been completed. This is the same for both systems.

- **Failure 3: Message Bearer Resource Request arrives with error in procedure transaction identity.**
  No way MME knows what the correct value could be since this value is produced by UE. So, this is a failure for both systems in any case.

- **Failure 4: Message Bearer Resource Request arrives with error in request message identity.**
  In this case, if QCI value is 1, and if any other filed in the message is compatible with Bearer Resource Allocation Request, it may be assumed that the UE is requesting for a voice dedicated bearer. Since this is a critical service, it is important to resist to the failure if possible. So, MME may give it a try by ignoring the error in request message identity field and suppose it has received the correct one. The possibility of resistance to such failure is low (2/10) for the proposed system since the rest of the message should be correct.

- **Failure 5: Message Bearer Resource Request arrives with error in linked eps bearer identity – non existing error identity**
  In this case, if UE has only one default bearer, then MME may try to establish dedicated bearer on this one. The possibility to actually resist to that failure is low (2/10).

- **Failure 6: Message Bearer Resource Request arrives with error in traffic flow aggregate**
  Resistance is low, (4/10) since MME may resist only in case of failure with cause #41 which means that TFT operation value is other than "Create a new TFT". For this to

be succeeded error with cause value #42 "syntactical error in the TFT operation; " where packet filter list is empty should not be erroneous. Though, it is more possible to save cases with error cause #41 since value "Create a new TFT" is fixe and easy for MME to assume it.

- **Failure 7: Message Bearer Resource Request arrives with error in required traffic flow QoS.**
  In this case the message is rejected with cause value #37 "EPS QoS not accepted". Though since this is a critical service, MME may try to establish a connection with lower QoS values, which may be modified afterwards during continuation of dedicated bearer establishment procedure part of which is the "modify bearer request" process.

- **Failure 8, 9: Message Bearer Resource Request arrives with error in protocol configuration options or device properties.**
  These fields shall be ignored by both SDLCs since the IE is optional.

- **Failure 10: Message Bearer Resource Request arrives twice.**
  There is a mechanism indicated by 3GPP standard for ignoring replay messages and should have been implemented by now. So, it is supposed that no failure is expected from this case. In case this recognition mechanism fails, then MME will consider the failure as a collision scenario and the incoming message will be rejected since UE is on call already.

- **Failure 11: All possible rejection messages that could arrive from SGW.**
  MME has no control on what is received from SGW. Maybe a concept of correcting some parameters if possible, could be realized. Therefore, recovery from failure is calculated as 1 (very low possibility of surviving the failure). For example, if MME receives error "Semantic error in the TAD operation" it may re-check the message sent and correct any information if possible and resend the message.

- **Failure 12, 13: All collision scenarios of message Bearer Resource Allocation Request. All collision scenarios of message Bearer Resource Command.**
  Number of resistances is more for the proposed system, since all possible combinations will be considered and tested. Bearer Resource Allocation procedure is critical, and it has greater priority compared to other procedures.

- **Failure 14: Bearer resource allocation procedure is left as "hanging" (active even after it has finished).**
  Number of resistances is more in the proposed system since resistance mechanism of clearing bearer resource allocation procedure has been designed and implemented. Though, there are still some cases of failure that should be considered.

- **Failure 15: Send message bearer resource allocation with integrity /ciphering failure.**
  In both cases, the failure is inevitable, and no corrective actions should be considered since such a failure may mean that an attack has been performed to the system.

- **Failure 16: SGW has restarted, or failed, and MME receives a restart counter in GTPv2 Echo message.**
  MME may choose another gateway to serve the requests, as it has been proposed by the current research, and this may have a positive impact to failures.

- **Failure 17: Failure to SGW is recognised since the next procedure that will complete Dedicated Bearer Request never starts. It is the same with SGW not answering case.**
  MME will recognise the failure much faster this time and some of the requests may be successfully completed by MME choosing another SGW when realises that KPI of failures related with the connected SGW is increased dramatically. Though, some of messages will be rejected until the incident is recognised by the system, or by the current procedure.

Some results based on the above calculations may depict a progress in recognition, resistance, and recovery from failures by the proposed system (orange colour).

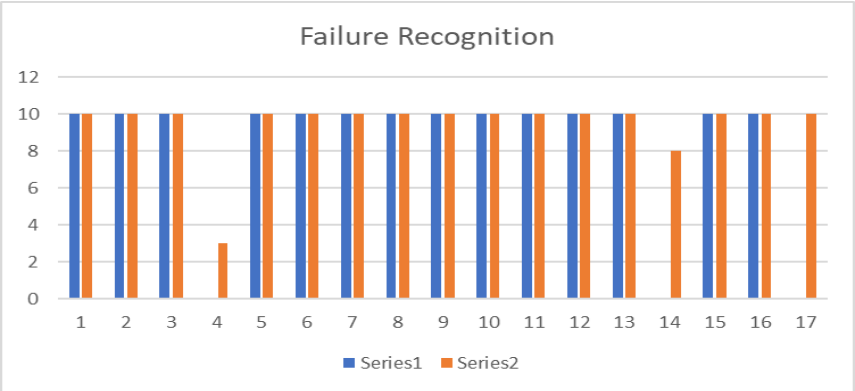


Figure 71: Comparison of processes capability regarding failure recognition.

In case of failure recognition, both systems are pretty efficient. The proposed system though improves recognition to some cases.

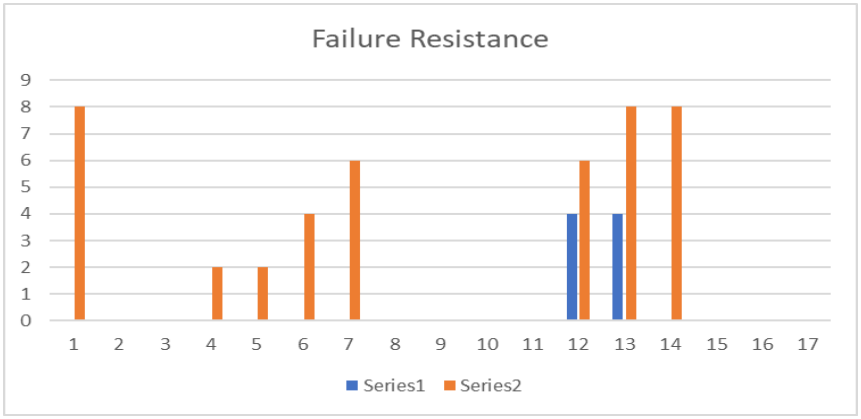


Figure 72: Comparison of processes capability regarding failure resistance.

The proposed system seems to make a major impact on resistance to failure compared to the old system which is based only in 3GPP standard to resist to failures.

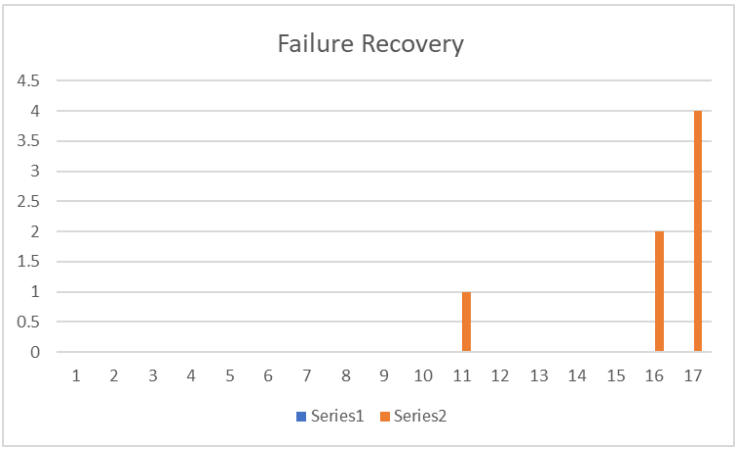


Figure 73: Comparison of processes capability regarding failure recovery.

Recovery from failure has been proved part of survivability that needs improvement for both systems. This could be part of future work. Though, the proposed system seems to have some increased capability of recovering from failures against the old system.

To sum up, it has been discovered that the proposed system has the potential to decrease number of overall failures and provide a more survivable system.

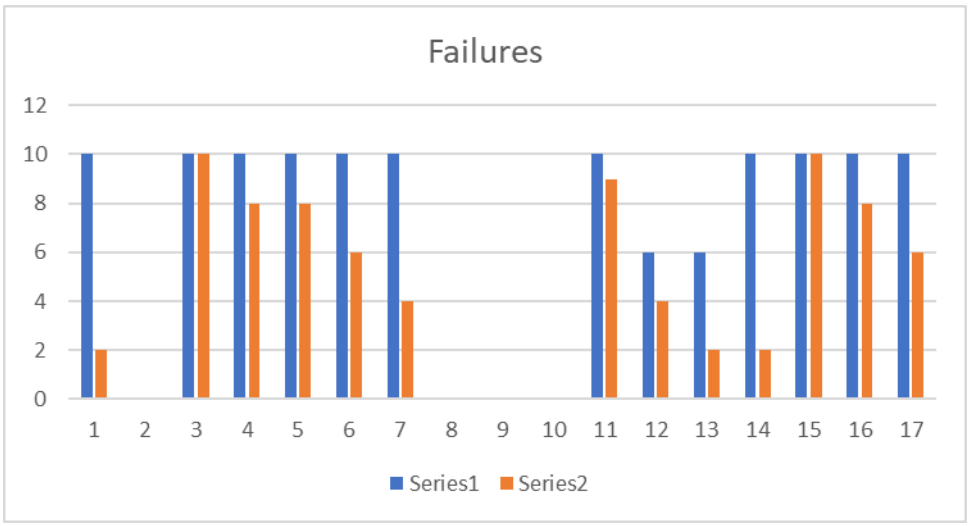


Figure 74: Comparison of processes capability regarding number of service failure.

Qualitative Analysis:

Apart from metrics indicating the proposed process improved capability, a short description of quality improvements shall follow. These could also improve the recognition, resistance and recovery KPIs of the whole system.

In the table that follows what may be depicted, are the KPIs that will be increased in the two systems and description of improvement of the proposed system to the analysis of failure and possible reaction to failure by the central management system as proposed to the current methodology. What should be pinpointed here is that the old system, following 3GPP, would keep metrics and KPIs only for known error causes and a general failure KPI. On the other hand, the proposed system, will extend these KPIs by indicating all possible error causes and marking the node that caused the error. This is useful for the network management entity (NME) as there could be a special handling that could provide overall resistance to failure.

	Failure Scenario	Recognition KPI		Quality Improvement Description
#	Suppose that each failure arrives 10 times.	SDLC based on 3GPP	SDLC proposed.	
1	Message Bearer Resource Request arrives with error in protocol discriminator	-Bearer Resource Request Failure	-Bearer Resource Request Failure-custom cause. -Bearer Resource Request Resistance to error with custom cause), UE	In the first case, there are two KPIs, a general failure error and an error with cause related to the failure. The second KPI shall be related to the UE produced this error (via IMSI). If this KPI is increased, then NME shall report a possible S/W error for that UE.
2	Message Bearer Resource Request arrives with error in EPS Bearer Identity	-Bearer Resource Request Success	-Bearer Resource Request Success	In this case, the error shall be ignored.

3	Message Bearer Resource Request arrives with error in procedure transaction identity	-Bearer Resource Request Failure -Bearer Resource Request Failure with cause #81	-Bearer Resource Request Failure -Bearer Resource Request Failure with cause #81, UE	The case is the same as (1)
4	Message Bearer Resource Request arrives with error in request message identity	-Bearer Resource Request Failure -Bearer Resource Request Failure with cause #97	-Bearer Resource Request Failure -Bearer Resource Request Failure with cause #97, UE -Bearer Resource Request Resistance to error with cause #97)	The case is the same as (1)
5	Message Bearer Resource Request arrives with error in linked eps bearer identity – non existing error identity	-Bearer Resource Request Failure -Bearer Resource Request Failure with cause #43	-Bearer Resource Request Failure -Bearer Resource Request Failure with cause #43, UE -Bearer Resource Request Resistance to error with cause #43)	The case is the same as (1)
6	Message Bearer Resource Request arrives with error in traffic flow aggregate	-Bearer Resource Request Failure -Bearer Resource Request Failure with cause #41, #42	-Bearer Resource Request Failure -Bearer Resource Request Failure- cause #41, #42. -Bearer Resource Request Resistance to error with cause #41, UE.	The case is the same as (1)
7	Message Bearer Resource Request arrives with error in required traffic flow QoS	-Bearer Resource Request Failure -Bearer Resource Request Failure with cause #37	-Bearer Resource Request Failure -Bearer Resource Request Failure with cause #37, UE -Bearer Resource Request Success. (in case of successful resistance)	The case is the same as (1)
8	Message Bearer Resource Request arrives with error in protocol configuration options	-Bearer Resource Request Success	-Bearer Resource Request Success	The case is the same as (2)
9	Message Bearer Resource Request arrives with error in device properties	-Bearer Resource Request Success	-Bearer Resource Request Success	The case is the same as (2)
10	Message Bearer Resource Request arrives twice	-Bearer Resource Request Success	-Bearer Resource Resistance to failure with custom cause value, UE -Bearer Resource Request Success	In this case, the improvement of the proposed methodology is that an increment in KPI with cause of failure and relation of it with the UE producing the error, may reveal a failure to the handling of the replay timer of the UE. If fixed, this will decrease the system load and decrease the probability of system failure because of increased load.
11	All possible rejection messages that could arrive from SGW.	-Bearer Resource Request Failure -Bearer Resource Request Failure with cause returned	-Bearer Resource Command Failure -Bearer Resource Command Failure- cause <number of causes returned> , SGW	In this case, KPI for the proposed system is related to the SGW that produced the failure cause. Increment of this counter may have a meaning for MME and action items that it should perform: - MME may discover an error in its own system if this failure happens to more than one SGWs that it is connected to. If

				needed, returning to an older S/W version may solve the issue. In this case, the older version would not include the procedure at all so this is not an option. - MME may discover an error in SGW if this failure happens only with one SGW node. - MME may discover an error in UE if the error is related to information coming from the same UE.
12	All collision scenarios of message Bearer Resource Allocation Request shall be tested.	-Bearer Resource Request Failure -Bearer Resource Request Failure with cause #56	-Bearer Resource Request Failure -Bearer Resource Request Failure with cause #57 and any other custom causes regarding the colliding procedure.	There are no further actions in this case.
13	All collision scenarios of message Bearer Resource Command shall be tested.	-Bearer Resource Request Failure -Bearer Resource Request Failure with cause "Collision with network-initiated request"	-Bearer Resource Request Failure -Bearer Resource Request Failure with cause "Collision with network-initiated request" and any other custom causes regarding the colliding procedure.	There are no further actions in this case.
14	Bearer resource allocation procedure is left as "hanging" (active even after it has finished)..	-Bearer Resource Request Failure	-Bearer Resource Request Failure -Bearer Resource Request Failure with cause "hanging procedure"	In this case, if MME is designed following the rules of the proposed system, it may depict an error in its implementation. If needed, returning to an older S/W version may solve the issue. In this case, the older version would not include the procedure at all so this is not an option.
15	Send message bearer resource allocation with integrity /ciphering failure.	-Bearer Resource Request Failure	-Bearer Resource Request Failure -Bearer Resource Request Failure with cause "integrity/ciphering failure", UE	There are no further actions in this case. An alarm to warn about a possible attack will be raised in both cases.
16	SGW has restarted, or failed, and MME receives a restart counter in GTPv2 Echo message.	-Bearer Resource Request Failure	- Bearer Resource Request Failure - SGW Failure	There are no further actions in this case.
17	Failure to SGW is depited since the next procedure that will complete Dedicated Bearer Request never starts. It is the same with SGW not answering case.	-Bearer Resource Request Failure	- Bearer Resource Request Failure - SGW possible Failure	In this case, MME of the proposed system is able to recognize a possible failure earlier since messages are not "answered" and choose another SGW for the service. In case this KPI is increased dramatically, MME may follow the procedure as it is indicated by the restoration procedures 3GPP standard and choose another SGW.

Table 12: Qualitative analysis of SDLC proposed.

7. Conclusions and Future Work

The current research has been conducted to investigate how to build a survivable mobile telecommunication system which will provide survivability to its critical services. The criticality of a service is already defined by general requirements defined by 3GPP which is the organization that produces the standardization of mobile telecommunication systems. The main objectives of the current research and corresponding actions may be summarized as follow:

1. Theoretical research on definition of survivability and on methods of providing survivability of system's mission.

An extended research has been conducted and presented through literature review in order to investigate the rules and requirements of a survivable system. The most promising method that was followed was the one from R. J. Ellison et all [60], who described recognition, resistance and recovery from failure, as key capabilities of a survivable system.

2. Extending already defined survivability measurements by 3GPP or literature, to cover self-configuration and self-diagnosis deficiencies.

The research continued by investigation of already defined survivability controls by 3GPP. For this investigation most of the 3GPP standardization documents were examined and controls related to survivability were presented in a table form. Then any deficiencies on survivability controls were depicted and two proposals were presented. The first was based on self-diagnosis of failure, and it was published in JACN Journal [82]. The second one was based on failure resistance by self-re-configuration of the system in order to prevent service failure based on failure possibility. This proposal was presented in ICICM 2019 conference [55].

3. Constructing a framework for developing survivable mobile telecommunication systems.

Based on survivability requirements gathered and the ones the current research proposed, a Software Development Lifecycle was constructed in order to provide a mobile telecommunication system with increased service survivability. The method was published in ASTES Journal [101].

4. Apply methodologies proposed to 4G Voice bearer establishment procedure, that will be used as a case study.

A detailed presentation of the proposed methodology to the 4G Voice bearer establishment has been presented in the current document.

5. Gather results and analyse them in order to ascertain that proposed process's capability is improved against service and system failure.

Finally, results have been gathered and a quantitative and qualitative analysis of these results have been conducted. Analysis showed that the proposed methodology has many potentials to improve the process of constructing survivable mobile telecommunication systems with more emphasis on recognition and resistance of failure.

The overall results of the current research indicate that recognition, resistance and recovery of failure should be of main focus in the analysis of requirements and testing of telecommunication systems, contrary to mentality of development teams that tend to focus only on functional requirements.

As future work, it would be very interesting to investigate solutions that would improve the recovery from failure. Additionally, application of proposed system to 5G systems could be used to depict the potentials of the SDLC proposed by the current research.

8. References

1. Wikipedia, "Survivability"
[http://en.wikipedia.org/wiki/Survivability]

2. Howard F. Lipson & David A. Fisher, (1999), "Survivability – A New Technical and Business Perspective on Security", *CERT® Coordination Centre Software Engineering Institute.*

3. Vickie R. Westmark, (2004) "A Definition for Information System Survivability", *Proceedings of the 37th Hawaii International Conference on System Sciences.*

4. John C. Knight, Kevin J. Sulliva, (2000), "TOWARDS A DEFINITION OF SURVIVABILITY", *Department of Computer Science University of Virginia.*

5. M. Al-Kuwaiti, N. Kyriakopoulos, S. Hussein, (2008), *"A Comparative Analysis of Network Dependability, Fault-tolerance, Reliability, Security, and Survivability",* IEEE.

6. Matthew G. Richards, Daniel E. Hasting, Donna H. Rhodes, (2008), *"Empirical Validation of Design Principles for Survivable System Architecture",* Massachusetts Institute of Technology 77 Massachusetts Ave., Building NE20-388 Cambridge, MA 02139.

7. http://www.cert.org/archive/html/analysis-method.html

8. http://en.wikipedia.org/wiki/List_of_system_quality_attributes

9. Pentti Tarvainen, (2004), "Survey of the Survivability of IT Systems", T*he 9th Nordic Workshop on Secure IT-Systems, Helsinki University of Technology, Finland.*

10. Matthew G. Richards, Daniel E. Hasting, Donna H. Rhodes, Annalisa L. Weigel, (2007), *"Defining Survivability for Engineering Systems",* Stevens Institute of Technology, PROCEEDINGS CSER 2007, March 14-16.

11. Wikipedia, "Dependability"
[http://en.wikipedia.org/wiki/Dependability]

12." *The System of Systems Approach for interoperable Communications"*, Department of Homeland Security.

13. Ian Sommerville, (2004), "Critical Systems", Software Engineering, 7th edition. Chapter 3.

14. Trevor Hilder, *"The Viable System Model"*, June 1995

15. John C. Knight, Elisabeth A. Strunk, Kevin J. Sullian, (2003), "Towards a Rigorous Definition of Information System Survivability", *Department of Computer Science University of Virginia, Proceedings of the DARPA Information Survivability Conference and Exposition, IEEE.*

16. Wikipedia, "Robustness"
[http://en.wikipedia.org/wiki/Robustness_%28computer_science%29]

17. Wikipedia, "Dynamic Systems Theory"
[http://en.wikipedia.org/wiki/Dynamical_systems_theory]

18. Robert Whitcher, BCI Webinar, (June 2009)
[http://www.efectus.cl/upload_files/documentos/271020090850 25-1413811 39.pdf]

19. Spyros Kokolakis, (November 2009), Presentation*: "IS Security: The Systems approach".*

20. *Ludwig Von Bertalanffy," Perspectives on General System Theory: Scientific Philosophical Studies"*

21. W. Ross Ashby, (1957), *"An Introduction to Cybernetics"*, Chapman and Hall LTD, 1957.

22. Critical Systems Labs, Software Safety and Risk Experts
[http://www.criticalsystemslabs.com/]

23. Wikipedia, "Systems Theory"
[http://en.wikipedia.org/wiki/Systems_theory#cite_ref-17]

24. Srinidhi Boray, "GENERATIVE TRANSFORMATION – ECONOMIC MODEL – SYSTEM DYNAMICS", 2009
[http://ingine.wordpress.com/2007/08/22/enterprise-architecture-economic-model-system-dynamics/]

25. http://en.wikipedia.org/wiki/Anthony_Stafford_Beer

26. Stafford Beer, (June 1970), *"Managing Complexity".*

27. Wikipedia, "Cybernetics"
[http://en.wikipedia.org/wiki/Cybernetics]

28. Richard A. Schmidt,Craig A. Wrisberg, "Motor learning and performance: a situation-based learning approach", 2008.

(Old 28). Survivability-Over-Security: Providing Whole System Assurance

29. Robert J. Ellison, David Fisher, Rick Linger, Howard F. Lipson, "Survivable Network Systems: An Emerging Discipline", 1997

30. Robert J.Ellison, David A. Fisher, Richard C. Linger, Howard F. Lipson, Thomas A. Longstaff, Nancy R. Mead, (1999), *"Survivability: Protecting Your Critical Systems"*, Carnegie Mellon University, IEEE

31. Wikipedia, "System of Systems"
[http://en.wikipedia.org/wiki/System_of_systems]

32. Douglas Lancaster, (2002), *"Systems Survivability"*, SANS Institute 2002.

33. Richard C. Linger, Howard F. Lipson, John McHugh, Nancy R. Mead, Carol A. Sledge, (2002), *"Life-Cycle Models for Survivable Systems"*, Carnegie Mellon University.

34. Nancy R. Mead, *"Requirements Engineering for Survivable Systems"*, Carnegie Mellon University, 2003.

35. Robert J. Ellison, Carol Woody, (2010), *"Survivability Analysis Framework"*, CERT Program, Carnegie Mellon.

36. Wikipedia, "Viable system model"
[http://en.wikipedia.org/wiki/Viable_System_Model]

37. Raul Espejo and Antonia Gill," *The Viable System Model as a Framework for Understanding Organisations*".

38. Maria Karyda, Spyros Kokolakis, Evangelos Kiountouzis, *"Redefining Information Systems Security: Viable Information Systems*".

39. Partha Pal, Franklin Webber, Richard Schantz, Joseph Loyall, Ronald Watro,(2002), *"Survival by Defence – Enabling "* , BBN Technologies.

40. John, Knight, Dennis Heimbigner , Alexander Wolf, Antoinio Carzaniga, Jonathan Hill, Premkumar Devanbu, Michael Gertz, (2009), *"THE WILLOW SURVIVABILITY ARCHITECTURE"*, University of Virginia, University of Colorado, University of California.

41. Wang Li, Li Zhi-Shu, Yin Feng, (2008), *"A Dynamic Survivability Reconfiguration Framework Based on QoS",* IEEE Computer Society

42. Matti A. Hiltunen, Richard D. Schlichting, Carlos A. Ugarte, and Gary T. Wong, (2000*), "Survivability through Customization and Adaptability: The Cactus Approach."*, The University of Arizona.

43. Dongyan Chen, Sachin Garg, Kishor S. Trivedi, (2002), "Networks survivability Performance Evaluation: A Quantitative Approach with Applications in Wireless Ad-hoc Networks", MSWiM'02, September 28, 2002, Atlanta, Georgia, USA.

44. Alex Hai Wang, Su Yan, Peng Liu, (2010), "A semi-Markov Survivability Evaluation Model for Intrusion Database Systems", 2010 International Conference on Availability, Reliability and Security

45. Abdul Jabbar Mohammad, David Hutchison, James P.G. Sterbenz, (2006), Resilinets web page. Available: http://www.ittc.ku.edu/resilinets/index.html

46. Yun Liu, Kishor S. Trivedi, "A GENERAL FRAMEWORK FOR NETWORK SURVIVABILITY QUANTIFICATION", AFOSR MURI grant no. F49620-1-0327

47. Ming Liang, Zhao Gang, Wang Dongxia, Huang Minhuan, Li Xiang, Miao Qing and Xu Fei, 2014, "A Novel Method for Survivability Test Based on End Nodes in Large Scale Network", KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS VOL. 9, NO.

48. Chunlei Wang, Liang Ming, Jinjing Zhao, Dongxia Wang, 2011, " A General Framework for Network Survivability Testing and Evaluation ", Journal of Networks VOL.6, NO.6.

49. Ming Liang, Huang Minhuan, Wang Dongxia, Kuang Xiaohui, Wang Chunlei, Feng Xuewei, (2011), "Research on Survivability Metrics Based on Survivable Process of Network System".

50. Le-Jun Zhang, Wei Wang, Lin Guo, Wu Yang, Yong-Tian Yang, 2007, "A Survivability Quantitative Analysis Model for Network System Based on Attack Graph", Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong.

51. Motorola, "4G LTE Key Features (EPS to EPC)" , (2011)
[http://gsmcommunications.blogspot.gr/2011/04/lte-key-features-eps-to-epc.html]

52. Marcin Dryjanski, "5G Core Network – Architecture, Network Functions, and Interworking", (2019)
[http://rfglobalnet.com/doc/g-core-network-architecture-network-functions-and-interworking-0001]

53. 3GPP 23401, "*General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access"*, Version 16.12.0

54. Verizon Community, (2013)
[http://www.zdnet.com/blog/apple/four-reasons-why-the-verizon-ipad-3-beats-the-at-and-t-model/12722]

55. Mykoniati Maria, Costas Lambrinoudakis, (2019) "Fault Prediction Model for Node Selection Function of Mobile Networks", ICICM: International Conference on Information Communication and Management

56. 3GPP 32111, "Universal Mobile Telecommunications System (UMTS); Telecommunication Management; Fault Management; Part 3: Alarm Integration Reference Point: CORBA solution set", Version 3.2.0

57. 3GPP 23.007, "Restoration procedures", Version 16.1.0

58. 3GPP 29809, "Study on Diameter overload control mechanisms", Release 12

59. 3GPP 29807, "Study on General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTP-C) overload control mechanisms", Release 12

60. R.J. Ellison, D.A. Fisher, R.C. Linger, H.F. Lipson, T.A. Longstaff, N.R. Mead (1999), "*Survivable Systems: An Emerging Discipline*," Proceedings of the 11th Canadian Information Technology Security Symposium (CITSS), Ottawa, Ontario Canada, Communications Security Establishment.

61. 3GPP 32101, "Telecommunication management; Principles and high-level requirements", Release 1999

62. 3GPP 32111-1, "Telecommunication management; Fault Management; Part 1: 3G fault management requirements", Release 1999

63. 3GPP 32150, "Telecommunication management; Integration Reference Point (IRP) Concept and definitions", Release 6

64. 3GPP 36300, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2", Release 8

65. Magdalena Nohrborg," Self-Organizing Networks"
[http://www.3gpp.org/technologies/keywords-acronyms/105-son]

66. 3GPP 36902, "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Self-configuring and self-optimizing network (SON) use cases and solutions", Release 8

67. 3GPP 32541, "Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; Telecommunication management; Self-Organizing Networks (SON); Self-healing concepts and requirements", Release 15

68. Tutorialspoint, "SDLC - Waterfall Model"
[https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm]

69. Chris Brown, Gray Cobb, Robert Culbertson, (2002), 'Introduction to Rapid Software Testing'.

70. Dorothy Graham, Erik Van Veendaal, Isabel Evans, Rex Black, (2006), "Foundations of Software Testing: ISTQB Certification"

71. Bor-Yuan Tsai, Simon Stobart, Norman Parrington, Barrie Thompson, (1997), "Iterative Design and Testing within the Software Development LifeCycle".

72. Tutorialspoint , "SDLC Iterative Model",
[https://www.tutorialspoint.com/sdlc/sdlc_iterative_model.htm]

73. Scott W. Ambler," Disciplined Agile Software Development: Definition", (2005)
[http://www.agilemodeling.com/essays/agileSoftwareDevelopment.htm]

74. Tutorialspoint , "SDLC Agile Model"
[https://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm]

75. Tutorialspoint, "Independent Testing"
[https://www.tutorialspoint.com/software_testing_dictionary/independent_testing.htm]

76. Alex Thissen, Microsoft technologies and products
[https://owasp.org/www-pdf-archive/SDL_in_practice.pdf]

77. Tutorialspoint ,"Defectr Life Cycle"
[https://www.tutorialspoint.com/software_testing_dictionary/defect_life_cycle.htm]

78. Cem Kaner, "A Tutorial in Exploratory Testing", (2008)

79. Tutorialspoint ,"Alpha Testing"
[https://www.tutorialspoint.com/software_testing_dictionary/alpha_testing.htm]

80.SixSigmaDaily, "Six Sigma Tools: DPU, DPMO, PPM and RTY,
[https://www.sixsigmadaily.com/dpu-dpmo-ppm-and-rty/]

81. J.Ravichandran,"Calculate overall sigma level",
[https://www.isixsigma.com/new-to-six-sigma/sigma-level/using-weighted-dpmo-calculate-overall-sigma-level/]

82. Mykoniati Maria, Costas Lambrinoudakis, (2019) "Self-Diagnosis Framework for Mobile Network Services", JACN 2019

83. 3GPP 32102, "Telecommunication management; Architecture", Release 1999

84. 3GPP 29303, "Domain Name System Procedures; Stage 3", Release 8

85. 3GPP 32111-5, "Telecommunication management; Fault Management; Part 5: Alarm Integration Reference Point (IRP): eXtensible Markup Language (XML) definitions", Release 6

86. Amar Sahay, "Managing and Improving Quality: Integrating Quality, Statistical Methods and Process Control", Business Expert Press © 2016

87. YING HE1, FEI RICHARD YU2, (Senior Member, IEEE), NAN ZHAO1, (Member, IEEE), HONGXI YIN1, HAIPENG YAO3, AND ROBERT C. QIU4,5, (Fellow, IEEE), "Big Data Analytics in Mobile Cellular Networks", IEEE Access, 2016

88. Mirza Golam Kibria, Kien Nguyen, Gabriel Porto Villardi, Kentaro Ishizu and Fumihide Kojima, "Big Data Analytics and Artificial Intelligence in Next-Generation Wireless Networks".

89.https://medium.com/coinmonks/linear-regression-withtensorflow-canned-estimators-6cc4ffddd14f

90. Alex Thissen, Microsoft technologies and products
[https://owasp.org/www-pdf-archive/SDL_in_practice.pdf]

91. 3GPP Organization, "Specification Numbering"
[https://www.3gpp.org/specifications/specification-numbering]

92. George Forrest, "FMEA (FAILURE MODE AND EFFECTS ANALYSIS) QUICK GUIDE"
[https://www.isixsigma.com/tools-templates/fmea/fmea-quick-guide/]

93. Victoria Drake, "Threat Modeling"
[https://owasp.org/www-community/Threat_Modeling]

94.CISCO, Products and Services – Security, "What Is an Incident Response Plan for IT?"
[https://www.cisco.com/c/en/us/products/security/incident-response-plan.html]

95. CISCO, Products and Services – Security, "What Is an Incident Response Plan for IT?"
[https://www.cisco.com/c/en/us/products/security/incident-response-plan.html#~how-to-create-a-plan]

96. Bectechnologies, "Quality of Service (QoS) in LTE"
[https://bectechnologies.net/wp-content/uploads/2019/12/QoS.pdf]

97. Arindam Ghosh, "Dedicated Bearer setup in LTE and impact on VoLTE Precondition", (2007)
[https://netmanias.com/en/post/blog/11789/lte-volte/dedicated-bearer-setup-in-lte-and-impact-on-volte-precondition]

98. 3GPP 24301, "Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3", Release 8.

99. 3GPP 29274, "Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C); Stage 3", Release 8.

100. 3GPP 24007, "Mobile radio interface signalling layer 3; General Aspects", Release 1999

101. Mykoniati Maria, Lambrinoudakis Costas, "Software Development Lifecycle for Survivable Mobile Telecommunication Systems", 2021

102. Brough Turner, Marc Orange, NMS Communications, "3G Tutorial"
[http://www.slideshare.net/mabongi/3-g-tutorial-7149386]

103.Webex Sunday Session, "Introduction to Mobile Core Network", (2013)
[[http://services.eng.uts.edu.au/~kumbes/ra/Wireless_Networks/GPRS/GPRS.htm]

104.CISCO, "Cisco GGSN Release 9.0 Configuration Guide, Cisco IOS Release 12.4(22)YE1"
[https://www.cisco.com/c/en/us/td/docs/ios/12_4/12_4y/12_4_22ye/ggsn9_0/cfg/12422ye_cfgbk/ggsnover.html]

105. 3GPP 23857, "Study of Evolved Packet Core (EPC) nodes restoration"

APPENDIX A'

Definitions of Survivability and security available according to literature.

Reference	Definition of survivability is…
[1]	To "provide quantitative measures for the network's capability to tolerate failures and to provide continuous service."
[2]	Defined in terms of network survivability where it is "1) the ability of a network to maintain or restore an acceptable level of performance during network failure conditions by applying various restoration techniques and 2) the mitigation or prevention of service outages from potential network failures by applying preventative techniques."
[4]	The "quality of a system to handle all essentially critical operation instances successfully."
[3] [5] [6] [7] [8] [9] [10] [11] [12]	The "capability of a system to fulfill its mission in a timely manner in the presence of attacks, failures, or accidents."
[13]	The "ability of a system to continue operation despite the presence of abnormal events such as failures and intrusions."
[14]	A "network's ability to perform its designated set of functions given network infrastructure component failures, resulting in a service outage, which can be described by the number of services affected, the number of subscribers affected, and the duration of the outage."
[15]	"Robustness under conditions of intrusion, failure, or accident."
[16]	The "ability of a system to maintain a set of essential services despite the presence of abnormal events such as faults and intrusions."
[17]	"That a system can be made robust to partially successful attack through general architecture features, through adaptability (flexible response to unanticipated changes) and flexibility (ability to adapt to a range of adverse events without having to anticipate the particular response in advance)."
[18]	To "provide network design and management procedures towards minimizing the impact of failures on multi-networks."
[19]	The "ability of a system to tolerate intentional attacks or accidental failures or errors."
[20]	Defined in terms of information survivability where it is "the ability of an information system to continue to operate in the presence of faults, anomalous system behavior, or malicious attack."
[21]	The "ability of a system to provide service (possibly degraded) when various changes occur in the system or operating environment."
[22]	Where network systems "continue functioning even when under attack."
[23]	The "ability of a system/network to be maintained in the working state, given that a deterministic set of failures occurs to the system/network; therefore, the survivability is always "yes" or "no" for a given failure scenario."
[24]	"Phases of survivability are attack detection, damage confinement, damage assessment and repair, and attack avoidance focusing on continued service and recovery."
[25]	The "capacity of a system to provide essential services even after successful intrusion and compromise, and to recover full services in a timely manner."
[26]	"The availability within a crucial time period"
[27]	"Network design and management procedures to minimize the impact of failures on the network."
[28]	Defined in the terms of a telecommunications network where it is "the ability of the network to maintain or restore an acceptable level of performance in the event of deterministic or random network failures, such as link failures and node failures."
[29]	Defined in terms of performance where it will "ensure that, under given failure scenarios, network performance will not degrade below predetermined levels."
[30]	The "ability of a network to cope with facility outages, capacity overloads, and natural disasters."
[31]	The "robustness of communication networks vis-à-vis events that affect a significant portion of the network topology."
[32]	Where "integrity is not compromised at the occurrence of unexpected disasters."
[33]	The "measure of the degree of keeping the performances of a kind of military weaponry or equipments or other military forces, which undergoing enemy's attacks."
[34]	The "ability of an item to perform a required function at a given instant in time after a specified subset of components of the item to become unavailable."
[35]	The "measure of a network's endurance in the presence of possible component failures (of the measure of the magnitude of attack needed to render a network nonfunctional)."
[36]	Where "survivable network must achieve an acceptable level of performance under demanding conditions."
[37]	The "assurance of stored information's integrity, confidentiality, and continuous availability guaranteed over time."
[38]	Defined in terms of survivable information systems through adaptation where it is "allowing a system to continue running, albeit with reduced functionality or performance in the face of reduced resources, attacks, or broken components is often preferable to either complete shutdown or continued normal operation in compromised mode."
[39]	Defined in terms of a survivable system where it "must be adaptable, able to respond to attacks and achieve its goals."
[40]	The "capability of a system to complete its mission in a timely manner, even if significant portions are incapacitated by attack or accident"
[41]	A "certain percentage of traffic can still be carried immediately after a failure."
[42]	The "degree to which a system has been able to withstand an attack or attacks, and is still able to function at a certain level in its new state after the attack."
[43]	The "capability of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures, or accidents." And was also defined as "preserving essential services in unbounded environments, even when systems in such environments are penetrated and compromised."
[44]	Defined in terms of a survivable system where it is "available to fulfill its mission in a timely manner, in the presence of attacks, failures, or accidents."
[45]	Defined in terms of a survivable system where it "satisfies its survivability specification of essential services and adverse environments."
[46]	The "capability of a system to fulfill its mission in a timely manner despite intrusions, failures, or accidents."
[47]	The "capability of an enterprise to continue to fulfill its mission by preserving essential services, even when systems are penetrated and compromised."
[48]	"Service stream over time"
[49]	"Assured continuity of essential infrastructure services under defined adverse conditions: natural, accidental, or hostile."
[50]	"Certain path-connectivity is preserved under limited failures of network elements."
[51]	Where systems "must continue to perform adequately in the face of various kinds of adversity."
[52]	The "capability of a network system to complete its mission in a timely manner, even if significant portions are incapacitated by attack or accident."
[53]	The "Extent to which the software will perform and support critical functions without failures within a specified time period when a portion of the system is inoperable."

Figure 76: Survivability Definitions [3].

APPENDIX B'

Restoration mechanisms

Partial failure handling procedures.

Apart from restoration procedures, there are also partial failures handling procedures. Partial failure procedures focus on LTE mobile networks and on cleaning up hanging PDN connections after a full failure of a remote node (HW / SW failure on MME, PGW, ePDG etc.). Though a significant number of PDN connections remain unaffected.

The recovery of PDN connections is applied by informing the peer nodes about them by using a common identifier for the set of PDN connections. The common identifier is called CSID – Connection Set Identifier and represents a set of PDN connections within a node. If the node fails, then the CSID is signalled to peer nodes. The identifier that globally uniquely identifies a set of PDN connections is the fully qualified CSID (FQ-CSID) identifier which is a combination of the node identity and CSID. This ID is used in order to prevent unwanted removal of PDN connections in case two nodes use the same CSIDs. Id FQ-CSID is stored for each PDN connection to peer nodes (nodes that are providing this PDN connection) and is used later in order to group same PDN connections. FQ-CSIDs are established to peers by Attach or PDN connection procedures. By this every node knows the association of its own FQ-CSIDs and the FQ-CSIDs of other peers so as to delete related PDN connections.

To continue, the most basic restoration procedures are presented below:

Location Register Restoration (VLR / HLR / HSS).

Avoiding loss of data stored to location registers when part of registers fail, demands a certain strategy which could contain back up units to replicate vulnerable units, periodic back up of data etc. Additionally, integrity of data should be ensured so as services not to be impacted. Failure is when integrity of data cannot be ensured.

VLR and SGSN must delete all IMSI records affected by failure when it restarts after the failure. GGSN must delete all non-static PDP records affected and restore static PDP when it restarts. HSS/HLR or CSS require periodic back up of data.

HLR (Home location register) is a database containing details of each mobile subscriber (MSISDN) authorized to a PLMN who uses the GSM network. This information may be related to services of the subscriber requests.

VRL (Visitor Location Register) is a database used for roamers subscribers. Each BTS is connected with one VLR so as the subscriber can be registered to only one VLR. Data

stored to VLR contain subscriber info and the HLR of the subscriber.

HLR and **VLR** are connected in order to communicate and serve roaming functionality.

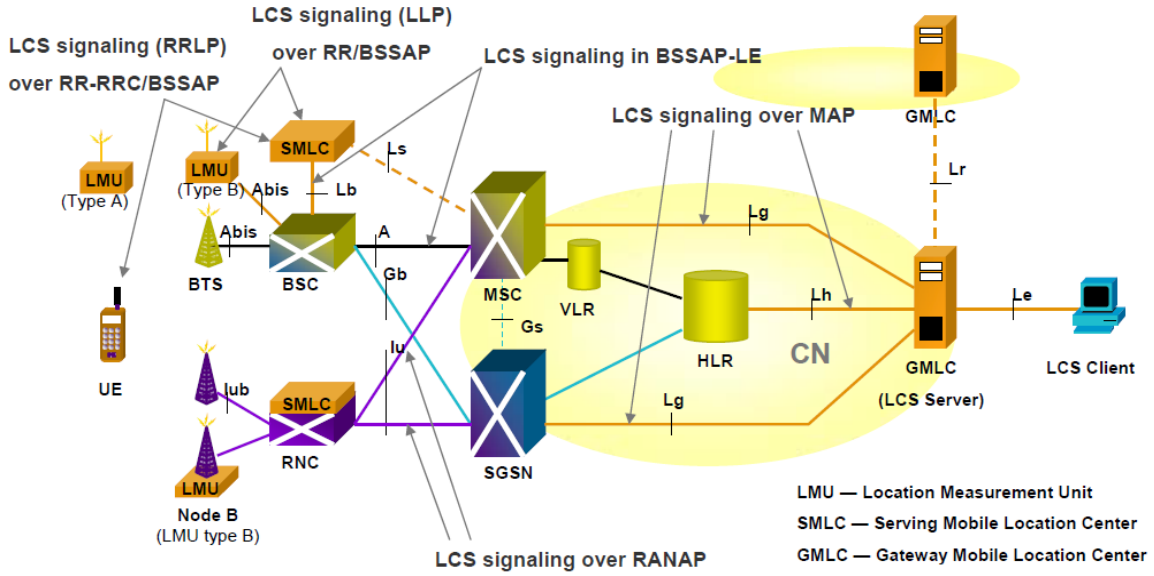


Figure 77: Location Registers [101]

HSS (Home Subscriber Server) is the database containing subscriber data for 4G networks. HSS consists of HLR / AuC. HSS stores information about user identification, numbering and addressing information, user security, user location and user profile. HSS participates in functionalities as mobility management, call and session establishment, authentication and access authorization, identification handling etc.

VLR Restoration:

- Scenario 1: VLR receives service messages for an MS with IMSI with no record in VLR.

Fault recognition:

Restoration to VLR Data may be triggered from three messages:

- **Location Update.**
- **IMSI Attach**.
- **Provide roaming number**

For restoration procedures to be able to handle all traffic for each MS subscription data and location information in VLR **must be consistent** with data or location information of HLR.

Example: VLR provides roaming number in dialogue terminating message, retrieve authentication data from HLR and trigger the restore data procedure. Simultaneously, for the particular subscriber the process followed is as the MS registers for the first time to the VLR.

Procedures that this failing scenario may be true are:

Incoming Call, Mobile Terminated Short Message, Mobile Terminating Location Request, Incoming LCS Information Request, Outgoing MS request, Outgoing LMU Request, Location Updated or IMSI Attach.

Set:

- 'Confirmed by Radio Contact',
- 'Subscriber data confirmed by HLR'
- 'Location Information Confirmed by HRL'
- 'Location Information Confirmed by SMLC'

restoration indicators to 'not confirmed'.

Restoration:

Restoration to VLR Data may be triggered from messages:

- **Location Update / IMSI Attach**. Then the VLR retrieves subscriber data from the HLR by sending "Update location request" which triggers "Insert Subscriber Data" from HLR.

- **Provide roaming number**: The VLR may send "Restore Data" request to HLR or "Send Parameters Request" which trigger "Insert Subscriber Data" from HLR.

When subscriber data are restored, indicators value is changed to "confirmed".

'Confirmed by Radio Contact' is set to 'confirmed' after authenticated radio contact. 'Location information Confirmed by HLR' is set to 'confirmed' after successful 'Update Location' procedure. 'Location Information Confirmed by SMLC' is set to 'confirmed' if HLR subscriber data indicates an LMU.

If Update location Response message contains an error different than "Roaming not allowed" or "Unknown Subscriber", or if there is a parameter issue, then no error should be sent to MSC and IMSI should not be deleted from VLR since indicator "Subscriber

Data Confirmed by HLR" is in "Confirmed" Status.

- Scenario 2: VLR receives a "Reset Indication" from SGSN or MME when UE served is also attached to non-GPRS services or non-EPS services.

Fault recognition:

When VLR receives such a message, assumes that information about UEs registered to MME /SGSN is no longer reliable.

Example: During a CS Fallback procedure VLR maintains data for the roaming UE subscriber. VLR receives a "Reset indicator" from MME.

Set 'Confirmed by Radio Contact' restoration indicator to "not confirmed."

Restoration:

VLR may send paging on both SGs and A/Iu interfaces.

- Scenario 3: VLR receives an HLR reset message.

Fault recognition:

This message may be sent after HLR restart after failure.

Example: MS is registered to an HLR for which VLR has received an HLR reset message.

Set 'Location Information Confirmed to HLR' restoration indicator to "not confirmed."

Restoration:

A reset message from HLR will trigger to VLR an update location for all subscribers affected. Then HLR, sends a "Forward Check SS Indication" request message to VLR and sets "Check SS" to "Check Not Required".

- Scenario 4: VLR receives a Location Services LCS Reset message.

Fault recognition:

When VLR receives 'LCS Reset' message it indicates failure of SMLC or MME and loss of location service transactions.

Example: SMLC failure.

Set 'Location Information Confirmed to SMLC' restoration indicator to "not confirmed."

Restoration:

After successful 'Update Location' procedure triggering a successful LCS Registration, indicator 'Location Information Confirmed to SMLC' will be set to 'confirmed'.

- Scenario 5: VLR receives an 'IMSI Detach' message from an LMU registered with an SMLC.

Fault recognition:

When VLR receives 'IMSI Detach' message it indicates that LMU is to be deactivated.

Example: Offline maintenance of LMU.

Set 'Location Information Confirmed to SMLC' restoration indicator to "not confirmed."

Restoration:

After successful transfer of LCS information message between SMLC and LMU, indicator 'Location Information Confirmed to SMLC' will be set to 'confirmed'.

- Scenario 6: VLR Failure with restart.

Fault recognition:

VLR informs the SGSNs affected, that the connections with the failed VLR are not valid and they should be deleted.

Example: VLR failure that ends to VLR restart.

VLR sends BSSAP and Reset message.

Restoration:

Case of VLR Connected with SGSN: SGSN marks all associations with VLR invalid and sets the indicator "VLR-Reliable" to false. Associations will be re-initiated one by one by RAU or combined RA/LA update from each UE.

For an MS which is IMSI Attached and GPRS attached (combined attach):

1. If combined RAU is received then SGSN proceeds with update location for non-GPRS services towards VLR.

2. If periodic RAU is received, Then Detach Request should be sent as an answer, so as MS re-performs a combined attach. Another way is the location update process for non-GPRS services to VLR. In second case the MS seems that will be attached only to GPRS.

Case of VLR Connected with MME: MME marks all associations with VLR as invalid and sets the indicator "VLR-Reliable" to false. Associations will be re-initiated one by one by TAU or combined RA/LA update from each UE.:

1. If combined TAU is received then it should perform update location process towards VLR.

2. If periodic TAU is received, UE is requested to re-attach to non-EPS services or sends location update for non-EPS services procedure.

When VLR restarts, all IMSI records affected by failure have been erased (subscriber data and location information). These will be restored after a "Provide Roaming Number" or "Update Location Area" individually for each subscriber.

Additionally, all TIMSIs and LMSIs are invalid and not used again. Instead IMSI is used to the first radio contact of MS during restoration.

- Scenario 7: VLR Failure without restart.

Fault recognition:

VLR is not in service for a given period.

Example: VLR does not answer to messages for a pre-defined time.

VLR is not in service and is marked in such a way.

Restoration:

The same as in case of VLR restarts. The additional service is that update location or reattach messages are sent to an alternative available VLR and not to the same as before.

- Authentication.

In 2G authentication triplets may be reused when no triplets are available. In 3G shall not be reused.

HLR/HSS Restoration:

The corruption or loss of subscription data to HLR has impact to its own PLMN and other PLMNs that it serves. Additionally, data must be restored to VLRs too as there are roaming subscribers associated with failed HLR.

- Scenario 7: HLR Restart.

Fault recognition:

Reset message to other nodes.
"Location information Not Confirmed in HLR" is set to SGSN, VLR and "Location information Not Confirmed in HSS" MME when receiving Reset message.
"Check SS" indicator is set to "Check required" when HLR restarts after failure.
"Check SS" indicator is set in case of "Forward Check SS Indication" service implementation.

The "Forward Check SS" procedure is used for corruption of supplementary services of HLR for informing the user to verify his supplementary services data. The restoration consists of creating new network-initiated transaction.

Restoration:

HLR/ HSS must restore all data from non-volatile back-up. HSS/HLR must reset all MS Purged flags which are flags that indicate that certain subscribers are unreachable. Each subscriber must be marked as "SS Check Required".

When HLR restarts, it sends a Reset message to SGSN. SGSN then marks mobility management contexts for MSs affected as invalid and if there is a connection with MSC-VLR it sets the Non-GPRS Alert Flag (NGAF).

When SGSN receives a valid LLC frame (A/Gb) or a valid GTP-U packet (Iu mode) then it restores data to HLR through Update Location Request and restore NGAF flag. These procedures may be delayed by a given period of time so as to avoid high signalling.

A Reset message is also sent to VLR, MME and relevant MMs are marked as "Location Information not confirmed in HLR /HSS". The actions following node, include an update location process so as HLR to store appropriate data.

SGSN Restoration.

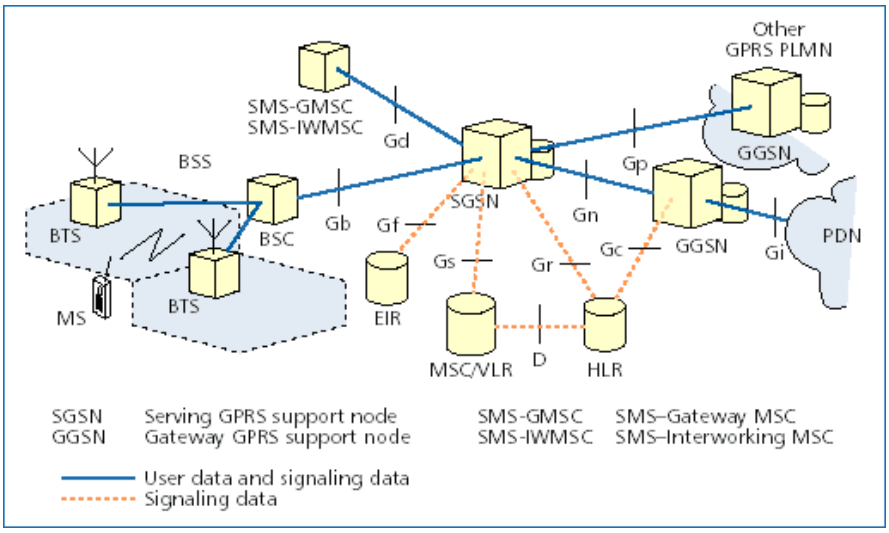


Figure 78: SGSN Interfaces [103]

- Scenario 1: SGSN Failure – Gn/Gp Interface.

Fault recognition:

When SGSN fails, it deletes all MM and PDP contexts affected.

Example: SGSN Restart.

Reset message to VLRs

Restoration:

When VLR receives the reset message it marks all associated with failed SGSN as unreliable. The rest process is described at VLR session.

- Scenario 2: Routing Area Update Request / Attach Request arrive to SGSN for an MS (IMSI) that there is no MM context.

Fault recognition:

"Subscriber Data Confirmed by HLR" indicator is set to "Not Confirmed" to indicate the inconsistency of subscriber data between the SGSN and HLR. The indicator "Location Information Confirmed in HLR" is also set to "Not Confirmed" to indicate the

inconsistency of location data between HLR and SGSN.

Restoration:

SGSN will reject the message with appropriate cause. The MS then may re-attach to re-activate PDP contexts. When SGSN successfully performs an update GPRS location to HLR indicator is set to "Confirmed".

- Scenario 3: Service Request arrive to SGSN for an MS that there is no MM context.

Fault recognition:

Service request message for MS with no PDP context.

Restoration:

SGSN will reject the message with appropriate cause. The MS then may re-attach to re-activate PDP contexts.

- Scenario 4: PDU Notification Request message arrive to SGSN for an MS that there is no MM context.

Fault recognition:

PDU Notification Request message for MS with no PDP context.

Restoration:

SGSN will return a PDU Notification Response to GGSN with appropriate cause. Then the SGSN will page the MS so as to reattach and reactivate the PDP Context.

- Scenario 5: SGSN receives mobile-terminated SM from SMS-GMSC for unknown IMSI.

Fault recognition:

SGSN receives mobile-terminated SM from SMS-GMSC for unknown IMSI.

Restoration:

SGSN will reject the request.

- Scenario 6: SGSN receives paging request over Gs for unknown IMSI and SGSN has

not completed recovery.

Fault recognition:

SGSN receives paging for unknown IMSI.

Restoration:

SGSN may page with IMSI to the location provided by MSC/VLR or to routing areas of MSC/VLR if no location provided. After a combined GPRS attach, the SGSN will continue serving the paging request.

- Scenario 7: SGSN failure of S4 interface.

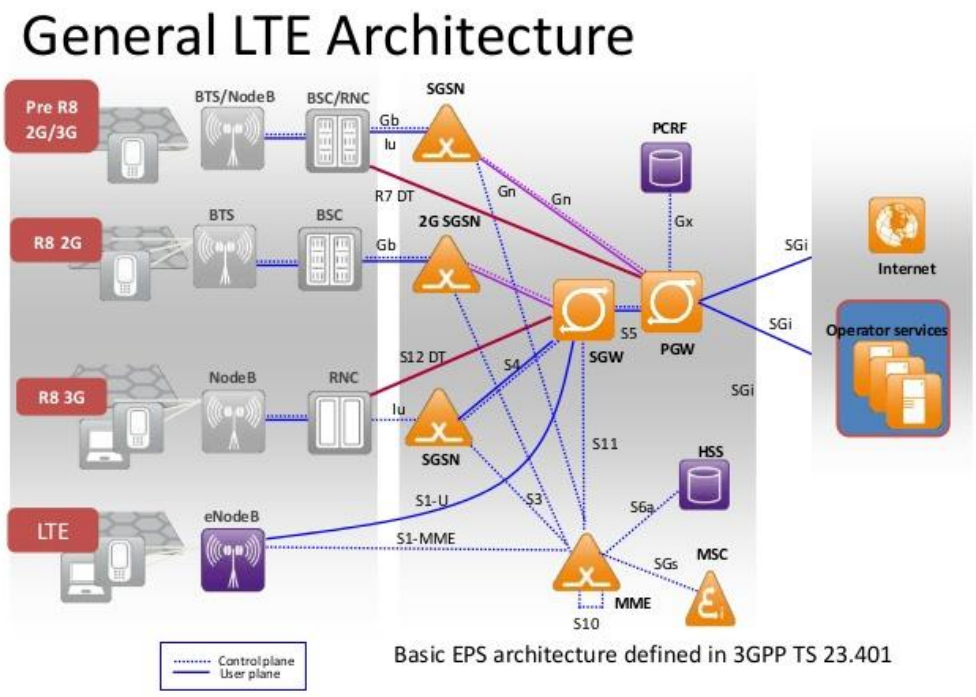


Figure 79: 4G, 3G Interfaces [103]

Fault recognition:

SGSN receives a GTP-U PDU from a serving GW for which there is no bearer context.

Restoration:

If SGSN receives a GTP-U PDU from a serving GW for which there is no bearer context SGSN shall discard the message and send a GTP error indication to SGW.

In case of S4-SGSN, if the feature of network triggered restoration is not supported, if a Downlink Data Notification message is received for an MS with no MM context, then a Downlink Data Notification ACK message is returned with suitable cause. The SGW will delete the bearer context and if there is no ISR to MME associated for the particular bearer context, the SGW should notify PGW to delete bearer context.

➢ Scenario 8: SGSN restart.

After a restart, SGSN shall delete all MM, PDP, MBMS UE, MBMS Bearer contexts (Multimedia Broadcast Multimedia service) affected by the restart.

Fault recognition:

SGSN restarts. "SGSN Reset" indicator is set to true.

Restoration:

When GGSN detects the restart, it deletes all PDP contexts or MBMS Bearers associated with the restarted SGSN. If it has only MBMS contexts and more than one SGSN connected, it shall not delete these contexts.

The MBMS SGW that detects the SGSN has restarted, should re-establish the bearers associated to the restarted SGSN. This is succeeded by MBMS Session Start procedure to the restarted SGSN or an alternative from the pool. Appropriate actions follow to re-establish the session to another SGSN.

"SGSN Reset" indicator is set to false after a certain period of time determined by the operator which should be longer than the timer of periodic RAU.

All associations with VLRs affected should be deleted. A Reset message is sent through Gs interface and indicator of VLR "Confirmed by radio contact" is set to "Not Confirmed" for the MSs affected. After successful RAU or combined RA/LA update for each MS affected, associations will be re-initated.

➢ Scenario 9: HLR restart.

After a restart, SGSN will receive a Reset message by HLR

"Location Information Confirmed in HLR" indicator is set to "not confirmed"

Restoration:

When GGSN detects the restart, it will set the "Location Information Confirmed in HLR" indicator to "not confirmed". After successful completion of Update GPRS Location procedure, it is set to "Confirmed".

➢ Scenario 10: VLR restart.

After a restart, SGSN will receive a Reset message by HLR

"VLR - Reliable" indicator is set to "false"

Restoration:

The scenario concerned is an MS attached both to GPRS and non-GPRS services. After successful performing of update location to VLR the indicator is set to "true".

➢ Scenario 11: SGW failure.

Fault recognition:

SGW failure is indicated by restart counter or path failure timer in case of restart.

Example: Combined attach.

Restoration:

When SGSN detects a failure to SGW, it shall delete all PDN connections and MM bearer contexts and release all internal resources associated with them or restore PDN connections if this is supported.

➢ Authentication.

In 2G authentication triplets may be reused when no triplets are available. In 3G shall not be reused.

After SGSN restart and before any authenticated radio contact of MS, P-TIMSI and TLLI are invalid. These are temporary identifiers provided to MS so as IMSI is not transferred through radio communication. In this case SGSN will request MS to identify itself with IMSI to make a relationship with old TLLI and allocate a new P-TIMSI.

Additionally to these restoration procedures, 3GPP TS 23857 proposes some other solutions for restoration so as mobile terminated services to be delivered to the UE in case of SGSN failure with restart. Until now, UE will re-attach to the network only when the UE has uplink signalling or data to send.

The first proposal is by called Downlink Data Triggered Attach procedure and it indicates that the UE is enabled to make the re-attach to the EPS on the arrival of downlink packet after SGSN restart.

The second proposal includes the HSS interaction in order to make the previous proposal more efficient. Instead of just DDN data with IMSI, the SGSN will check if the UE has already been attached to another SGSN by sending Update Location Request to the HSS. If it is rejected, then UE is attached to another S4SGSN.

The third proposal is called pro-active paging based. By this procedure, when S4-SGSN fails network initiates a paging procedure with IMSI (proactive paging). This paging is performed by an alternative S4-SGSN from pool selected by SGW or by the same S4-SGSN if there is no pool.

Finally, the last solution is by using S4-SGSN self-stored paging information. This solution includes storing of S-TMSI and TAI list to a non-volatile memory before S4-SGSN restarts. By the use of this memory, in the arrival of Downlink Data with IMSI, S4-SGSN uses this stored information to accomplish the paging procedure.

GGSN Restoration.

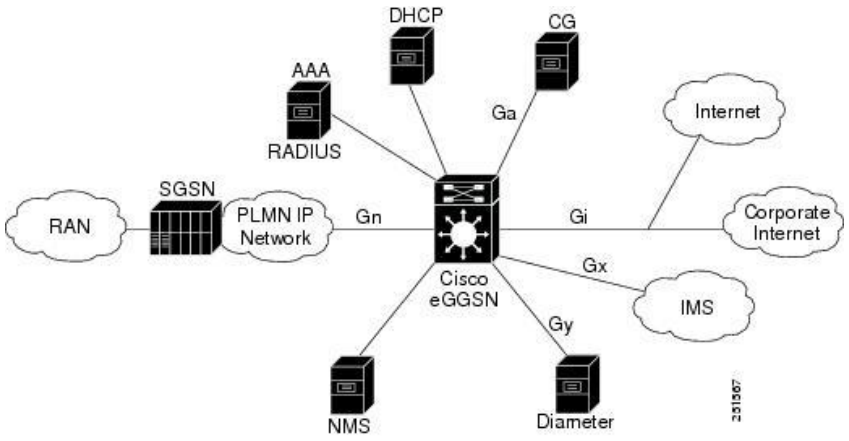


Figure 80: GGSN Interfaces [104]

- Scenario 1: GGSN Failure.

Fault recognition:

GTP-U PDU with no corresponding PDP context.

Example:

GTP-U PDU with no corresponding PDP context.

Restoration:

All PDP contexts affected by a GGSN failure should be deleted. If a GTP-PDU is received for which no PDP context exist, the PDU should be discarded, and an error indicator should be replied to SGSN.

- Scenario 2: GGSN Restart.

Restoration:

All PDP contexts and MBMS UE contexts affected by a GGSN restart should be deleted. When SGSN detects a restart, it should deactivate all associated PDP contexts or MBMS UE contexts. In case of PDP contexts SGSN could request from MS to reactivate them.

- Scenario 3: SGSN Restart.

Fault Recognition:

GGSN will recognise a restart to SGSN by a restart counter in GTPv1 echo response message.

Restoration:

If SGSN restart is detected, GGSN shall delete all PDP contexts, MBMS UE contexts and free all internal resources associated with these contexts.

- Scenario 4: PCRF Restart.

Restoration:

If GGSN needs to send a request to a restarted PCRF, the GGSN may discard the request and tear down all associated PDP contexts by sending a PDP Context Deactivation Request to SGSN, with cause "Reactivation requested". Then UE will request the same PDP context for the same APN. PDP emergency should not be affected by this failure.

MME Restoration.

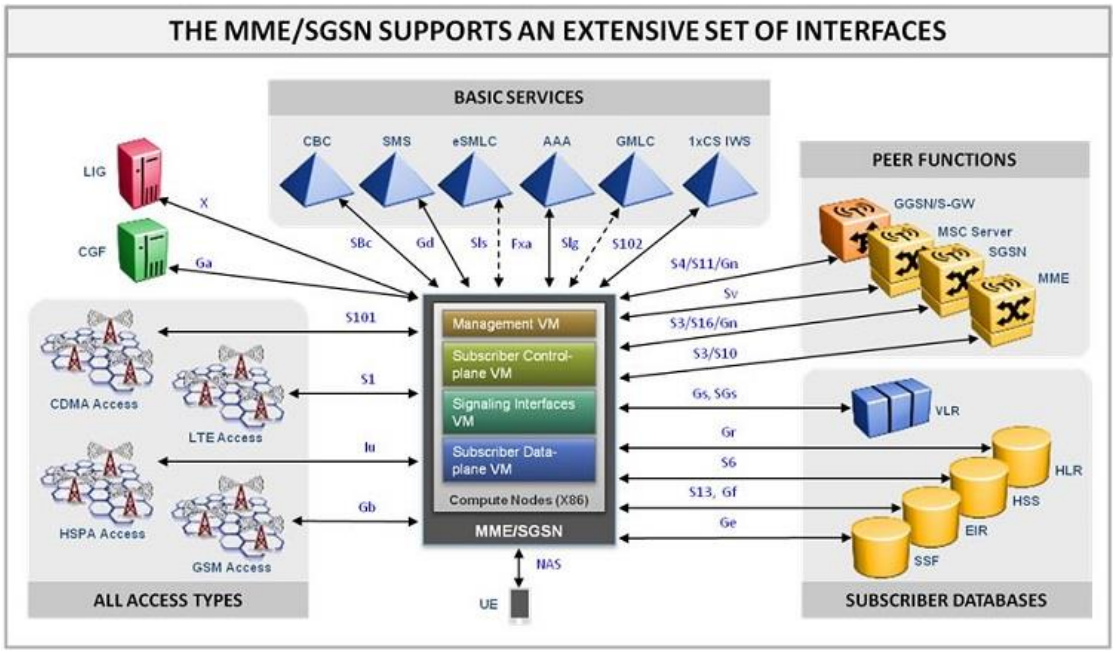


Figure 81: MME Interfaces [105]

➢ Scenario 1: MME Restart.

Fault recognition:

MME Restart

MME Restart, Reset message

Restoration:

After MME restarts, it deletes all MM bearer contexts affected that it may have stored.

When GW detects a restart of MME it will re-establish all MBMS bearer services affected by initiating a MBMS Session Start procedure to the restarted MME or another MME in the pool. The MBMS context should be the same as the previous one. It shall not delete the MBMS bearer contexts unless all connected MMEs and SGSNs have restarted.

If MME receives a Downlink Data Notification for non – existing MM context, then an ack message should be sent to GW with appropriate cause. The SGW should delete all related bearer context and should inform PDN-GW to delete bearer context too. In case network triggered services restoration is supported, the restoration procedure will be

performed.

All VLR associations with restarted MME are marked as un-reliable and may be deleted. After reset message over SGs interface restoration indication "Confirmed by radio contact" to VLR is set as "Not confirmed". After next successful performance of Combined TA/LA update from a UE association will be re-initiated.

- Scenario 2: TAU or Attach message for a UE that MME has no IMSI record.

Fault recognition:

TAU message with IMSI that MME has no record.

"Subscriber Data Confirmed in HSS", "Location Information Confirmed in HSS" indicators will be set to "Not Confirmed".

Restoration:

In case MME has an MM context for the UE, indicators will set to "Confirmed" after MME successfully performs an update location request to the HSS. In case of TAU If the MME has no MM context for this UE, it shall reject the requests. In case of attach, the MME will create MM context for the UE and continue with getting more information from the HSS by Update Location procedure.

- Scenario 3: Mobile terminating services requested by MSC/VLR.

Fault recognition:

Paging message with IMSI that MME has no record for.

"MME – Reset"

Restoration:

One way of restoration is by using an alternative MME from the pool if this feature is supported by the network. Otherwise, after receiving a paging for a UE with unknown IMSI, and "MME-Reset" indicator is set to true then MME sends paging request to TA that it may have information for or to all TA connected to this MME. In case "MME-Reset" indicator is set to false the paging is rejected. If "MME – Reset" indicator is set to false but the IMSI is known for the MME and the UE is in registered state, then paging shall be sent to the UE.

- Scenario 4: Mobile originated service request.

Fault recognition:

Service request message.

Restoration:

If the UE is unknown for the MME, service request shall be rejected. Then the UE should re-attach and re-activate bearers needed.

- Scenario 5: HSS Reset message is received.

Fault recognition:

HSS Restart

"Location Information Confirmed in HSS" indicator will be set to "Not Confirmed".

Restoration:

Indicator will set to "Confirmed" after MME successfully performs an update location request to the HSS.

- Scenario 6: SGW Failure.

Fault recognition:

SGW failure is indicated by restart counter in case of restart.

Example: Combined attach.

Restoration:

When MME detects a failure to SGW, it shall delete all PDN, connections and MM bearer contexts and release all internal resources associated with them or restore PDN connections if this is supported.

- Authentication.

In E-UTRAN EPS authentication vectors shall not be reused.
Apart from solutions described by 3GPP 27.003, there is a proposal for restoration of system from MME failure described at 3GPP TS 23.857. This proposal is based on restoration of MME failure or of failure link between MME and SGW without need for re-attach or re-establishment of bearer contexts by the UE, for example by the use of other nodes available in the pool. This provides continuity of service to the subscriber

and avoids the network overload which is the result of trial of thousands of UEs connected to fail MME to re-attach to the network after failure.

Alternative solutions for MME failure [107]:

1. Use of downlink data triggered attach feature:

By this feature, the UE has the ability to re-attach on the reception of a downlink packet. This is achieved by marking bearer contexts that should be maintained as DLDTA. In case of re-attach for example, the bearer with QCI 5 should be maintained. Additionally, TA information coordination between SGW and MME should be used. Then the SGW shall maintain S5/S8 bearer contexts for DLDTA bearers. When a downlink data message arrives, it is forwarded to the MME in order to page the UE and make it re-attach. If MME is changed, then the DDN message will be forwarded to the new MME.

2. MME failure by re-attachment with HSS interaction:

By this approach, the re-attachment to the MME will be more efficient. Again S5/S8 bearers are maintained for a certain period defined by a timer. If downlink data arrive during this period of time to SGW, will be rejected and DDN message with IMSI will be send to MME. MME will check if UE has been registered to another MME by Update Location Request procedure with IMSI and restoration indication to the HSS. Then HSS will send update location request with subscription data or will reject the update location request with appropriate cause. By this network overloading by paging messages is avoided if UE has already connected to a new MME.

3. Proactive paging-based approach:

By this approach, when MME fails, the network may initiate paging for EPS services by using IMSI. This process is a combination of two phases. During preparation phase, MME marks the UE and data that will be assisted with IMSI paging and communicates this information with SGW. During execution phase, the SGW starts a "Paging for EPS / GPRS services using IMSI" to trigger UE to perform attach procedure.

4. Using MME self-stored paging information.

UE is enabled to re-attach to EPS triggered by downlink data packets or signalling after MME restarts. Again, SGW maintains S5/S8 bearers for a predefined by a timer period of time, so as to be able to receive downlink data packets. If MME restarts, a Downlink data notification message with IMSI will be sent to the MME. The difference here is that MME has a non-volatile memory to which stores S-TMISI, TAI list, IMSI, SGW FQDN and PGW FQDN so as to page UE upon receiving DDN message with IMSI. Then a service request procedure will be triggered which will be rejected by MME so as UE to

re-attach to the network.

If **ISR is activated** there are some more proposals. With ISR, the MME maintains the SGSN control plane IP address and TEID, the SGSN maintains the MME control plane IP address and TEID, and the SGW maintains control plane IP addresses and TEIDs of the MME and the SGSN. [107]

SGW Restoration.

- Scenario 1: SGW Restart.

Fault recognition:

SGW restart.

Restoration:

After SGW restart, all bearer contexts affected by restart should be deleted. SGW shall place local restart counter to GTPV2 echo messages and PIMPV6 heartbeat responses. Previously used TEID values should not be reused after SGW restart to avoid inconsistences.

- Scenario 2: SGW Failure.

Fault recognition:

SGW receives a GTP-U PDU for which no bearer context exists.

Restoration:

SGW should discard the GTP-U PDU and return a GTP error indication.

- Scenario 3: MME Failure.

Fault recognition:

Restart counter to Echo request / response messages.

Example: SGW receives echo request from MME with restart counter incremented.

Restoration:

SGW shall delete all PDN connections associated with node failed and free any internal

resources for these PDN connections. Additionally, network triggered restoration techniques may be used.

If network triggered service restoration is supported, then SGW does not release PDP connections but retains them until a certain predefined timer expires. This time interval allows UE to re-attach to the network. Until then and since PDP context is not released, downlink data continue to arrive to SGW. Then after receiving the create session request message for a UE, SGW releases corresponding bearers and proceed with create session request.

If network triggered service restoration feature and Idle Mode Signalling Reduction features are both supported, then SGW shall maintain PDN Connection and bearer contexts for a predefined duration determined by certain timer. It is as if the failed node is still active. The same is applied for S4-SGSN as well. SGW in this way may still send DDN data to the restarted MME or S4-SGSN until the UE is informed for the restart. If the timer expires, then deactivation and release of resources follows. Additionally, in case of failure and not of restart, SGW may use another node from the pool that has not failed.

- Scenario 4: PGW Failure.

Fault recognition:

PGW Restart counter in GTPV2 echo request/response messages and PIMPV6 heartbeat responses that SGW receives from PGW. If PGW does not restart after failure SGW may realize that by a maximum path failure duration timer (messages are not answered).

Example: SGW receives echo request from PGW with restart counter incremented.

Restoration:

If SGW detects that PGW has restarted or failed but not restarted, it shall delete all PDN connections associated with node failed and free any internal resources for these PDN connections.

PGW Restart Notification – PRN- feature: This feature if supported by peer nodes indicates that SGW could inform MME with PGW Restart Notification message that PGW has restarted. Then MME should delete all PDN connections associated with SGW and the restarted PGW and freeing associated internal resources. The cause may be "reactivation required" so as PDN connections to be re-established or "explicit detach" with "reattach required" in case there are no other PDN connections available for a UE. PDN restoration may be based on operator's policy concerning QoS characteristics, roaming related policies, APNs characteristics etc. MME may also apply restoration to

radio part of the connection.

Other alternative solutions for SGW restoration are described by 3GPP TS 23857.

First proposal of the document is MME / S4-SGSN to re-establish resources with a new SGW or with the old after restart. For this to be succeeded MME / S4-SGSN and PGW should maintain bearers and MM contexts. Then if UE is in ECM-IDLE state, MME / S4-SGSN select a new SGW (SGW relocation) or the old one if it is recovered. This includes Create Session request procedure so as bearer contexts to be established to SGW and PGW. If the UE is in connected state, the MME / S4-SGSN, detects a new SGW but it should first delete S1 / Iu connections. The failure may be detected also from eNB which may initiate this procedure. If the UE is in IDLE state and Service Request is initiated, then SGW relocation should be performed first. If a TAU/RAU procedure is initiated, then TAU/RAU Request message MME / S4-SGSN are triggered to perform an SGW relocation. If MME / S4-SGSN change is also involved, then the old MME – S4-SGSN should indicate to target in GTPV2 Context Response message that SGW relocation is needed.

The second proposal involves PGW initiated paging request at receiving of Downlink data. Again MME/S4-SGSN and PGW maintain bearer contexts. When downlink data arrive at PGW, it selects an SGW to send paging with IMSI. Then MME/S4-SGSN performs a Network initiated Service Request procedure and an SGW relocation procedure

PGW Restoration.

- Scenario 1: PGW Restart.

Fault recognition:

PGW restart.

Restoration:

After PGW restart, all bearer contexts affected by restart should be deleted. PGW after restart, shall place local restart counter to GTPV2 echo messages and PIMPV6 heartbeat responses.

- Scenario 2: PGW Failure.

Fault recognition:

PGW receives a GTP-U PDU for which no bearer context exists.

Restoration:

PGW should discard the GTP-U PDU and return a GTP error indication to SGW. Previously used TEID values should not be reused after PGW restart to avoid inconsistencies.

- Scenario 3: SGW Failure.

Fault recognition:

PGW receives restart counters in echo request / response messages and PMIPV6 heartbeat responses.

Restoration:

PGW should delete all PDN connections associated and all internal resources for these PDN connections. Alternatively, procedures to restore connections between MME and PGW after SGW failure may take place. These will be examined later.

- Scenario 4: PCRF Failure.

Fault recognition:

PCRF restarts.

Restoration:

PGW should discard any IP-CAN Session Modification Request towards this PCRF and tear down all associated PDN connections by PGW initiated bearer deactivation procedures with cause "reactivation requested". Then UE will try to re-establish connection to the same APN.

An alternative proposed by 3GPP TS 23857 document, is that when MME detects the failure to release PDN Connections with cause "re-activation required". If there is only one PDN connection the cause should be "re-attach required".

PCRF Restoration.

- Scenario 1: PCRF Restart.

Fault recognition:

PCRF restart.

Restoration:

If PCRF fails, all PCC contexts and Diameter sessions affected are lost.

- Scenario 2: PCRF receives a non-initial message for Diameter session that does not exist.

Fault recognition

Non –initial message for unknown diameter session.

Restoration:

If PCRF fails, all PCC contexts and Diameter sessions affected are lost.

GERAN / UTRAN Restoration.

- Scenario 1: BSS Failure.

Fault recognition:

BSS Restart

Restoration:

All contexts affected should become invalid. The storage of data at BSS is volatile.

- Scenario 2: RNC/BSS Failure.

Fault recognition:

From SGSN the recognition is applied by message reset.

Restoration:

All contexts affected should become invalid. The storage of data is volatile. SGSN that recognises the fault deletes the RABs for all affected PDP contexts. Any affected PDP contexts of Direct Tunnel with GGSN will be recovered when SGSN receives an Iu connection establishment request from MS or when GGSN informs the SGSN that it has received an error indication message from RNC.

**TEID values should not be re-used right after a restart.

- Scenario 3: RNC/BSS receives GTP-U PDU for which no RAB context exists.

Fault recognition:

No data for this GTP-U PDU.

Restoration:

GTP-U PDU should be discarded and a GTP error indication should be returned to the GGSN or SGSN.

- Scenario 4: RNC/BSS Failure S4 mode.

Fault recognition:

From SGSN the recognition is applied by message reset.

Restoration:

All contexts affected should become invalid. The storage of data is volatile.

SGSN that recognises the fault deletes the RABs for all affected PDP contexts. If ISR is activated, S4-SGSN shall release all access bearers towards the serving SGW and remove downlink user plane address and TEID. Additionally, an S4-SGSN should maintain MBMS bearer contexts but delete RNC related information. Then S4-SGSN should re-establish MBMS sessions.

The restoration will be applied by MS trying to establish an Iu connection or if SGW initiates a Network Triggered Service Request process.

- Scenario 5: RNC/BSS receives GTP-U PDU for which no RAB context exists.

Fault recognition:

No data for this GTP-U PDU.

Restoration:

GTP-U PDU should be discarded and a GTP error indication should be returned to the GGSN or SGSN

- Scenario 6: Iu path failure.

Fault recognition:

No SCTP association.

Restoration:

RNC shall release all MBMS services and S4-SGSN shall maintain MBMS bearer contexts but delete RNC related information.

E-UTRAN Restoration.

- Scenario 1: eNodeB Failure.

Fault recognition:

eNodeB Restart / Reset message to MME / no more SCTP association in service

Restoration:

All contexts affected should become invalid and deleted.

MME that recognises the failure, shall delete all associated data with this eNB for S1 AP connection and all UE S1AP IDs. Then it should delete all S1 bearers towards SGW by release access bearer request procedure and should initiate a dedicated bearer deactivation for GBR bearers.

SGW that receives a release access bearer request should release all eNB related information (eNB addresses,TEIDs etc) but maintain all UE's serving contexts.

Recovery id succeeded through UE service request process or network-initiated service request process.

Enb should not immediately reuse TEIDs after restart.

- Scenario 2: Public Warning System Restoration after Enb failure.

Fault recognition:

eNodeB Restart

Restoration:

When Enb restarts it shall delete all warning data. Then it should inform CBC to re-load all warning message data by a PWS restart indication message. This should be done through two MMEs of the pool in order to ensure that CBC will receive the message. This message is also used for individual cell resetting since these warning message data

include TAI and EAI data which means that are associated with each particular cell.

- Scenario 3: S1-AP path failure.

Fault recognition:

No SCTP association

Restoration:

Enb shall release all RRC connection of affected UEs but keep sending warning messages. MME shall proceed with actions described at Enb failure.

When path has been recovered Enb shall continue as no failure had existed.
If Enb warning data are de-synchronised, which may be recognised if CBC tries to modify data during the failure, then "Write – Replace – Warning" procedure takes place.

Path Management Procedures (GTP – restoration of data connection).

In general path management failure detection for GTP-C and GTP-U involves messages Echo Request, Echo response and T3-Response timer. Messages Echo Request are sent periodically.

The T3-Response timer is used to determine how much time the node will wait for the Echo response message before implying that the other node is unavailable. A peer's IP address specific counter is set when echo response message arrives. The same counter is incremented each time T3-Response timer expires. If the counter exceeds N3-Requests the path should be considered unavailable.

After detecting the failure, the node should notify it to Operational and Maintenance system and either delete associated with failing peer's IP address PDP connections or maintain these connections for a maximum path failure duration. If this duration exceeds, the resources associated should be deleted. If control / user plane signalling is received other interfaces and timer has not expired yet, then the maintained resources shall be deleted.

- Scenario 1: S1/S4 Path failure SGW – MME/S4-SGSN.

Fault recognition:

Echo Request, Echo response and T3-Response timer

Restoration:

If SGW detects a path failure and supports the network triggered service restoration procedure, it should maintain the S5/S8 bearer contexts and continue with restoration procedure. During restoration, if there is no alternative path (another MME / S4-SGSN) to the pool or an alternative control plane IP address to the same MME / S4 –SGSN, then PDP contexts should be deleted.

For UEs in connected state, SGW should continue sending downlink packets to eNB for the PDN connections maintained regardless of whether it supports network triggered service restoration procedure.

- Scenario 2: S5 path failure SGW – PGW unavailable.

Fault recognition:

Echo Request / Echo response messages and T3-Response timer.

Restoration:

If SGW supports PGW Restart Notification, then it shall delete all PDN connections affected and inform MME / S4-SGSN for the failure by sending a PGW Restart notification if supported from those nodes too. By this MME will restore PDN connections earlier before receiving PGW Downlink Triggering Notification from PGW. This may also be considered as a backup notification in case PGW never sends anything. In case PGW Downlink Triggering notification message is sent, then part of restoration of SGW is to include PGW F-TEID and GRE Key for control plane to the message and forward it to MME / S4-SGSN.

MME which receives PGW Downlink Triggering notification message will check if there is any PDP connection containing this PGW F-TEID and GRE key. If there is no PDP connection it will not proceed with restoration. This may mean that it has already restored the PDP connections. Otherwise, it will proceed with PGW triggered SGW restoration procedure. If UE is connected, then it may restore PDN connections by Serving GW relocation procedure.

The same is happening for S4-SGSN node.

- Scenario 3: S5 path failure SGW unavailable - PGW.

Fault recognition:

Echo Request / Echo response messages and T3-Response timer.

Restoration:

If PGW detects a failure at S5 path, it should maintain S5 bearer for restoration and send a PGW Downlink triggering notification message with PGW F-TEID for control plane.

GTP - U Error indication.

- Scenario 1: Error indication message for PDP contexts.

Fault recognition:

Error indication message

Restoration:

If GGSN receives an error indication message and direct tunnel feature is on, which means that data could be transferred directly from RNC to GGSN, it should mark associated PDP contexts as invalid. It should discard downlink data for these PDP contexts and inform SGSN for the failure. Then the SGSN will re-establish the connection and GGSN will forward downlink traffic to SGSN. Otherwise, GGSN shall delete PDP context and notify Operation and Maintenance network element.

If Gn/Gp SGSN receives an error indication from GGSN, it shall delete all PDP contexts, inform Operation and Maintenance network element and request from MS to re-establish PDP contexts by sending a PDP Context request with cause "re-activation required". If it receives the error indication from RNC, it shall release RAB connection. It could keep the PDP contexts and send re-establish the RAB connection.

If S4-SGSN receives error indication from SGW it shall delete bearers, notify Operation and Maintenance network element and request from UE to re-establish bearers. If it receives the message from eNB,it shall release RAB connection, inform Operation and Maintenance and request from UE to re-establish RAB connection.

If RNC or NodeB receive GTP error indication, from SGSN it shall release RAB connection and resources. If the message is received from GGSN, RNC shall release RAB connection and delete PDP contexts with SGSN and request from MS to re-establish PDP contexts. If RNC receives the message from SGW, it shall release the RAB connection and inform SGSN to delete related bearers.

If ENB receives an error indication message from SGW, it shall release RAB connection and resources. Then MME proceeds with releasing bearer contexts depending on certain policy, QoS, APN, ARP and on the kind of bearer that is to be released. Then it may proceed with re-activation of bearer context or re-attach. If the ENB receives the message from another eNB or from SGW through indirect forwarding tunnel, it may

ignore the message or delete the forwarding tunnel contexts without deleting EPS bearers.

If SGW receives an error indication message it will handle it as follow:

- If it is from RNC or eNB, just the information concerning GTPU-U tunnel TEIDs must be deleted and not the bearers over this tunnel. Then the SGW sends DDN message to SGSN or MME and starts buffering downlink packets until connection is re-established.

- If it is from S4-SGSN for a bearer other than the default, SGW may delete bearer contexts, inform Operation and Maintenance and ask from S4-SGSN to re-establish the connection with a DDN message. If it is for the default bearer, all other bearers associated with the default may be erased and the process continue as before. The same is applicable if the message is received from PGW.

If PGW receives an error indication message, it will handle it the same way as SGW above.

If MME / S4-SGSN receives a DDN message from SGW because of the error indication message received from ENB / RNC, MME should act as follow:

If UE is idle, MME / S4-SGSN should proceed with Network Triggered Service Request procedure. If UE is connected, then S1 release or Iu release should take place before.

Restoration of data in PCRF.

➢ Scenario 1: PCRF Restart

Restoration: No restoration procedure. PCC contexts and Diameter sessions affected are lost.

➢ Scenario 2: PCRF receives a message for which no diameter session exists.

Restoration: Discard message and send diameter error indication to PCRF Client.