UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE INFORMÁTICA

A FAULT- AND INTRUSION-TOLERANT ARCHITECTURE FOR EDP DISTRIBUIÇÃO SCADA SYSTEM

Nuno André Carnido Medeiros

Mestrado em Segurança Informática

Novembro 2011

UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE INFORMÁTICA

A FAULT- AND INTRUSION-TOLERANT ARCHITECTURE FOR EDP DISTRIBUIÇÃO SCADA SYSTEM

Nuno André Carnido Medeiros

Orientador: Prof. Dr. Alysson Neves Bessani

Mestrado em Segurança Informática

Novembro 2011

Resumo

Nas últimas décadas tem havido um grande investimento na criação de uma infra-estrutura para distribuição de energia que ofereça uma maior qualidade de serviço e também uma maior cobertura de todo o território nacional Português. No entanto, a criação de novas instalações eléctricas e a própria expansão da rede não garantem por si só a qualidade de serviço exigida, apesar da maior robustez e confiabilidade das tecnologias mais recentes. É muito importante ter a capacidade de monitorizar toda a infra-estrutura a qualquer altura de modo a poder responder o mais rapidamente possível a quaisquer incidentes ou falhas que possam ocorrer na rede eléctrica. O objectivo principal de um fornecedor de energia é reduzir o tempo de indisponibilidade do serviço prestado aos consumidores de electricidade e aumentar a qualidade do serviço.

O papel de executar uma supervisão rápida e eficaz de toda a infra-estrutura de distribuição de energia é delegado ao operador de rede que fornece a rede de distribuição. O principal objectivo será optimizar o fluxo de energia através da gestão e operação da rede eléctrica, garantindo a qualidade do serviço técnico. Para além de estar alerta para qualquer incidente, os operadores de rede são também responsáveis para iniciar todas as medidas necessárias para resolvê-los, de acordo com a análise feita no momento.

Hoje em dia, a operação da rede de distribuição é assistida por ferramentas de gestão de alta tecnologia, os sistemas Supervisory Control and Data Acquisition (SCADA) que permitem a supervisão e controlo remoto da rede de distribuição de energia da EDP Distribuição. No entanto, esta supervisão não foi sempre apoiada por tecnologias sofisticadas. Antes do aparecimento dos sistemas de informação, a monitorização da rede eléctrica era realizada localmente, nas subestações de Alta Tensão, por operadores de rede que trabalhavam em turnos de 24 horas por dia, sete dias por semana, garantindo o controlo e gestão de incidentes a todo o momento.

Com o desenvolvimento das tecnologias da informação e de comunicações, os sistemas SCADA surgiram. Estes sistemas são ferramentas muito valiosas para a supervisão e operação da rede eléctrica em tempo real. Os sistemas SCADA têm uma arquitectura muito complexa composta por sistemas de informação com bases de dados relacionais, que recebem quase em tempo real informação dos diferentes componentes eléctricos monitorizados e que têm também a capacidade de executar comandos remotamente sobre esses mesmos componentes.

A implementação destes sistemas possibilita uma melhor gestão da rede eléctrica de distribuição, a redução de custos de operação, e permite também a automatização de procedimentos e a padronização dos processos a nível nacional. Na EDP Distribuição, o sistema SCADA representa apenas um dos sistemas core do Generation Network Information System (GENESys), um sistema que incorpora as funcionalidades SCADA com funcionalidade

de gestão da distribuição de energia, fornecidas por um sistema Distribution Management System (DMS).

O sistema GENESys representa o núcleo da gestão da rede eléctrica da EDP Distribuição, e a sua robustez, confiabilidade e disponibilidade é crítica, considerando as funcionalidades do mesmo. A arquitectura actual do sistema tem algumas debilidades e carece de capacidade para tolerar alguns tipos de falhas na sua estrutura.

Sendo assim, propomos uma arquitectura tolerante a falhas e intrusões para o GENESys, com o objectivo de criar um sistema mais confiável e seguro. A arquitectura é composta essencialmente por três camadas distintas que iremos endereçar com três soluções dedicadas, com a perspectiva de melhores resultados globais. Para as camadas inferiores do sistema, as instalações eléctricas monitorizadas e os *sites* dos Frontends, propomos mecanismos de tolerância a falhas baseado em redundância com gestão aplicacional. Para a camada dos sistemas SCADA e DMS, propomos a implementação de um protocolo de replicação tolerante a intrusões, uma vez que ambos os serviços são fundamentais para uma gestão confiável e segura da rede eléctrica da EDP Distribuição. O protocolo MinBFT irá fornecer uma camada extra de segurança para os sistemas uma vez que o algoritmo de replicação de máquinas de estado irá garantir que, se um invasor comprometer uma das réplicas do sistema, não será capaz de controlar e comprometer o funcionamento correcto da rede eléctrica.

Além disso, realizamos duas análises distintas sobre a arquitectura proposta, sempre dividindo-a pelas três camadas abrangidas. O objectivo da primeira análise é entender quais são as capacidades de tolerância a falhas introduzidas nas diferentes camadas do GENESys pelas soluções propostas. Na segunda, realizamos uma análise de custo-benefício para inferir sobre a viabilidade de nossa proposta, reconhecendo tanto os seus custos como os benefícios técnicos e operacionais.

Palavras-chave: Distribuição de Electricidade, SCADA, GENESys, Confiabilidade, Segurança, Tolerância a Falhas, Tolerância a Intrusões.

Abstract

Over recent decades there has been a great investment in creating an infrastructure for energy distribution that offers a higher quality of service and also a greater coverage over the Portuguese national territory. However, the expansion of facilities and the power grid do not guarantee by themselves the required quality of service, despite the increased robustness and reliability of the more recent technologies. It is very important to monitor the entire infrastructure at all times in order to respond as fast as possible to incidents and failures that occur in the power grid. The main objective is to reduce the downtime of the service provided to electricity consumers and to increase the quality of service.

The role of performing a quick and effective oversight of the entire infrastructure of power distribution is delegated to the utility providing the distribution grid. Its main objective is to optimize the flow of energy by managing and operating the power grid, ensuring quality of technical service. In addition to being alert to any incident, the network operators are also responsible to initiate all the necessary measures to solve them, according to the analysis made at the time.

Nowadays, these functions of great responsibility are facilitated by management tools, usually supervisory control and data acquisition (SCADA) systems that allow remote monitoring and control of the EDP Distribuição power grid. However, the monitoring of the power grid has not always been supported by sophisticated technologies. Before the appearance of information systems, the oversight of the facilities was carried out locally by grid operators who worked in shifts covering high voltage substations twenty four hours a day, seven days a week, ensuring control and incident management at all times.

With the development of information technologies and communications, the SCADA systems emerged. These systems are the most valuable tools on providing supervision and operation of the electric power system in near real-time [1]. The SCADA system have a very complex architecture composed by information systems and database applications which receive real-time information and execute commands over different electrical components on the several telemetry electrical sites, based on sensors, actuators and controllers thereby present.

The SCADA systems implementation results in better manageability of the power grid, the reduction of operation costs, and allowing the automation of procedures and the standardization of processes at national level. In EDP Distribuição, the SCADA system only represents one of the core systems of Generation Network Information System (GENESys), a system that incorporates the features and functionalities of SCADA but also the management of the distribution of energy, provided by a Distribution Management System (DMS).

The GENESys platform represents the core of the EDP Distribuição power grid management and its robustness, reliability and availability is critical, considering the functionalities it provides. The current architecture of the EDP system as some weaknesses and lacks on the ability of tolerating faults within its structure.

We propose a fault- and intrusion-tolerant architecture for GENESys, aiming on a more dependable and secure system. The architecture is mainly composed by three different layers which we address with three dedicated solutions with the view of better global results. For the lower layers of the system, the Telemetry Sites and the Frontend Sites, we propose fault-tolerant mechanisms based on redundancy with applicational management. For the backend systems layer we propose the implementation of an intrusion-tolerant replication protocol since both SCADA and DMS services are crucial for a dependable and secure management of the EDP Distribuição power grid. The MinBFT protocol will provide an extra layer of security to the backend systems since the state machine replication algorithm will guarantee that if an attacker compromises one of the system replicas he will not be able to control and jeopardize the power grid operation.

Furthermore, we perform two different analyses over the proposed architecture, always dividing it by the three covered layers. The first analysis objective is to understand which are the fault tolerance capabilities introduced to the different layers of GENESys by our solutions. In the second, we perform a cost-benefit analysis to infer about the viability of our proposals by acknowledging both their costs and the technical and operational benefits.

Keywords: Electricity Distribution, SCADA, GENESys, Dependability, Security, Fault Tolerance, Intrusion Tolerance.

Acknowledgments

This thesis represents the closure of the most challenging fifteen months of my life. It represents the hard work, the sleepless nights, the absence to my dearest, the absence at work, the multitasking, the creation of new friendships and my absence with the old ones, and everything I had to go through to reach this stage of completion. However, in truth, this adventure would have never been possible without the people who helped me to overcome all the obstacles, without their love and support.

First, I would like to thank the most important people in my life, those who gave me everything to become who I am today. My dear parents, thank you for your support and sacrifice over the years and I know that without your love things would have been a lot harder than they were.

I would also like to thank my sweet girlfriend Joana Magalhães, for all the support, patience and love throughout this endeavor. I want you to know that I am sorry for not having been able to support you more closely during the biggest professional challenge of your life, and that I love you.

A word of appreciation goes also to Carnegie Mellon University, Information Networking Institute, and for Faculdade de Ciências, Universidade de Lisboa, and all the professors I had, which made this master program a captivating and outstanding experience.

To my advisor, Prof. Dr. Alysson Bessani, whose guidance was preponderant throughout the development of my thesis. Thank you for your orientation, support and availability at all times.

A special word of appreciation goes to my employer, EDP Distribuição, for giving me the opportunity of attending this master degree. I would like to thank in particular Carlos Mota Pinto for proposing me this master and encouraging me to apply. I would also like to thank António Leitão, Fernando Rocha and Miguel Areias for their patience, guidance and priceless help that made this thesis possible. To Pedro Gama, thank you for understanding. Last, but not least, Aurélio Blanquet which I know it was ultimately responsible for this opportunity.

Finally, I would like to thank my friends without which I would never be as happy as I am. Thank you for your support, companionship and friendship throughout this master. I am sorry for my absence and I will be back soon.

Dedicated to Mom and Dad

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Electricity, as we all know it today, was a joint discovery by several scientists which studied the applications and all the required technicalities associated with it. The main objective at the time was creating a cheap and safe way to provide light, and electricity was seen as the best solution to do it. The English scientist Michael Faraday was the one responsible by developing the principles behind the electrical motor and generator, i.e., the conversion of electrical energy in mechanical energy, and vice-versa. Soon, processes for the massification of the production of electricity where established since the interest for such an energy source was growing and growing all around the world. Power plants where built, and energy supply networks began being installed on the main European and North American cities, what we today refer as power grids.

The electricity generation requires the conversion of other different types of energies and scientists where centered on those which they already identified for long as energy sources, usually used for heating or lighting. The whole electricity generation research and development started being focused on burning fossil energy sources such as oil, gas and coal, to produce steam to be then used to drive a steam turbine and generate electricity. This paradigm accompanied us until not so long ago, when the first concerns about the massive exploitation of such resources start being raised by the scientific community. Primarily, these concerns were about the resources availability since they start getting exhausted from nature. Furthermore, another concern has gain a lot of impact over the last few years, based on awareness campaigns over the fact that such energy resources have very bad consequences for the planet, because of the resulting intense toxic emissions.

Facing this problem has become one of the main challenges for society as for creating a new more sustainable paradigm for the future. It is critical to research and deploy more ecological ways of generating and distributing electricity without affecting its must required availability and dependability. Furthermore, the electricity system should include instruments to improve its efficiency and security.

The electrical utilities are giving their best on transforming and evolving their electrical system to answer the challenges that society is now facing.

- They are investing on alternative and clean energy sources, such as wind, solar and hydraulic, for the generation of electricity;
- Encourage consumers to become micro producers by facilitating and subsidizing the integration of microgeneration in the electrical grid;
- Investing in more efficient and sustainable power plants;
- Creating new and more efficient power grids to grant the network itself with some intelligence capabilities with the objective of providing services with higher quality and reliability in an autonomous way.
- Reinforce the power grid quality of service by deploying resilient and robust management systems to allow a near real-time monitoring and incident response with the utmost availability.

Regarding the last topic referred, utilities have been investing over the last few years on improving their quality of service. They have been implementing advanced technological solutions such as supervisory control and data acquisition (SCADA) systems aiming on remote monitoring and controlling the power grid infrastructure at all times. A robust and dependable system will allow the grid operators to detect power grid incidents promptly and execute the required corrective actions quickly and efficiently. This will obviously lead to a reduction in restoration time and consequently to the improvement on the quality of service. However, this type of technology although very beneficial, does not come without its downsides. The vital operation of managing the electrical power grid is now supported by a distributed system vulnerable to the fragilities of technology and the dangers of connectivity and the cyber world. The failure, or even worse, a cyber attack on these systems can lead to catastrophic scenarios since they provide full remote control capabilities over the power grid infrastructure. Moreover, concerns such as dependability and security should be accounted while implementing and managing such critical systems.

It is important for companies to understand the risks involved while using this type of systems and to acknowledge the available fault and intrusion tolerance mechanisms which can ensure a more reliable and secure service. It is also crucial to understand and evaluate the costs of the available approaches, and how they counterbalance the required improvements in dependability and security. These costs and ultimately the return on investment, related with the available solutions benefits, need to be carefully studied and clarified so that appropriate conclusions can be drawn about the viability of these investments.

1.2 Motivation

The consumers have become increasingly more demanding about the quality of service provided by the energy utilities, in this context, the distribution of electricity. In fact, it makes sense considering that each person is affected in the absence of electricity as well as their quality of life. Therefore, for the past few years it has been the scope of EDP Distribuição increasing the quality of service provided by investing both in field technology, i.e., better and more robust electrical equipments, and also in the supervision and remote control systems, to ensure the better responsiveness possible to any incident.

The supervision system implemented on EDP Distribuição is denominated GENESys. It is responsible for providing the monitoring and remote control capabilities over the electrical power grid. This system is decisive to guarantee the correct power grid management and, more importantly, the fastest response to any electrical incident.

Our main objective is improving the dependability and security of such a critical system. We present an alternative architecture for GENESys based on fault and intrusion tolerance which we will prove later on to be very effective.

1.3 Contribution

In this thesis we propose an alternative design for the GENESys system of EDP Distribuição. We are aiming to increase the dependability and robustness of the GENESys platform by proposing alternative implementations for the different layers of its architecture.

This work will analyze thoroughly the current architecture and propose an alternative fault- and intrusion-tolerant design to address the current fragilities. We will also evaluate the impact of our proposal on the GENESys system operation and EDP Distribuição.

1.4 Organization

The remaining chapters in this thesis are structured in the following way:

Chapter 2 - Literature review on fault and intrusion tolerance underlying the concepts which lead to its definition. We present the means to achieve dependability, more specifically, the fault and intrusion tolerance mechanisms from which we develop our work.

Chapter 3 - Thorough description of the GENESys system underlying the concepts, the operational and architectural details.

Chapter 4 - Presents a fault- and intrusion-tolerant architecture for the GENESys system together with its functional and technical details.

Chapter 5 - Concluding remarks and notes on future work.

Chapter 2

Intrusion Tolerance

Throughout the development of computer systems a new technological solution has risen to answer a new paradigm with special requirements. The distributed computing architecture may be described as a set of multiple autonomous computers physically distributed, communicating with each other through a computer network. The main benefits of such a solution are the fast sharing of data stored in different locations, as it happens in companies with various headquarters spread around the world, and creating a powerful, resourceful and with high level of performance system [3] without supporting it in a single high-end computer, but instead in several low-end computers, as a more cost-effective solution.

The dependability concerns in distributed computing are similar to those of basic centralized computing. These concerns are shared by any critical computer environment where maintaining the correct functioning despite the existence of vulnerabilities and faults, both malicious and accidental, is the main priority.

Historically, security research and development (R&D) has focused its scope on decreasing the vulnerabilities of systems and preventing and detecting faults or intrusions. This is still considered the best approach against most types of faulty behaviors, however, it is proven to be insufficient to withstand the threats of the cyber world.

For the past few years, a new approach on security as appeared and it is called intrusion-tolerant computing. An intrusion-tolerant system is one that maintains its security properties (i.e., confidentiality, integrity and availability) despite the presence of faults in one or more of its components [4]. Intrusion tolerance complements more classical security paradigms, acknowledging that intrusions may happen and we need to create mechanisms to handle them in an effective way so that the normal operation of a given system is not affected by it. Therefore, it relies in the assumption that systems are vulnerable and intrusions might occur. This new approach calls for triggering mechanisms to prevent an existing fault from leading into system failure, making the system fault-tolerant. These systems handle the faults by applying different reaction, counter-reaction, recovery and masking mechanisms which make faults tolerable, instead of catastrophic.

The fault-tolerant approach supports itself in the concept of dependability which is described as the ability to deliver service that can justifiably be trusted [5]. It is a global concept that subsumes the usual attributes of reliability, availability, safety, integrity and maintainability [6]. Therefore, to accomplish a dependable operation on a system one needs to rely not only on the combination of the more classical security paradigms, such as fault prevention, fault removal and fault forecasting, but also in this more recent, yet very important, fault tolerance approach. Since malicious failures can be the most dangerous adversary for the dependability of a system, we should also assume that a malicious user will also be able to find a way to attack a given system with malicious purposes. To answer this particular fault scenario, the concept of intrusion tolerance was defined [4]. It is a specific fault tolerance design approach which aims on preventing the failure of a system as a direct consequence of malicious or Byzantine faults [7], which might be caused by an intrusion [8].

It is imperative to create a robust defensive line so that even if the conventional security mechanisms are overcome, the presence of faults will not be reflected into a system failure since the system is able to coexist with them.

2.1 The Concepts behind Fault-Tolerant Computing

In this section we present and explain the main concepts and mechanisms for fault- and intrusion-tolerant distributed systems, for a better understanding of the section to come.

2.1.1 Faults, Errors and Failures

As it can be inferred about the concept of failure, it describes a behavior that is considered to be incorrect and abnormal. A failure occurs when an actual running system deviates to this specified behavior. These failures result from errors which are symptoms of existing system faults, which are invalid system states that should never be reached, considering its specification. Faults can be responsible for multiple errors and consequently multiple system failures.

The fault-error-failure sequence is presented in Figure 1, as well as a representation of the constructive guidance to build a dependable system. The fault tolerance mechanisms also illustrated will be described in the next sections. It is important to refer that it is impossible to build a completely dependable and protected system. The objective of these combined techniques is to provide the foremost security possible.

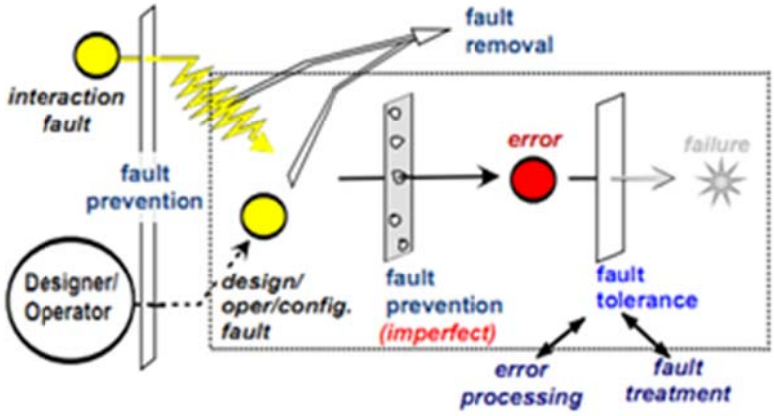


Figure 1 - Fault-Error-Failure sequence.

There is a classification for faults based on their characteristics and how the failed component behaves once it has failed [3].

Omissive faults occur when a component does not perform an interaction as it was specified to do. They occur essentially in the time domain.

- **Crash faults -** The component either completely stops operating or never returns to a valid state;
- **Omission faults -** The component completely fails to perform its service;
- **Timing faults -** The component does not complete its service on time.

Assertive faults occur when interactions are not performed as specified. They occur in the value domain.

- **Syntactic faults -** The interaction format is incorrect;
- **Semantic faults -** The interaction format is correct but the content is incorrect.

Byzantine or arbitrary faults result from the union of omissive and assertive faults. These are clearly the most dangerous types of faults since they have arbitrary behaviors that can reflect in many different ways and with different consequences, i.e., a process may crash, disobey the protocol, send contradictory messages, and collude with other malicious processes).

It is important to refer that there are interactions between different faults. These interactions indicate that some of the types of faults include others, meaning that if a system is able to tolerate one of the types, it might be confident that it tolerates some other. This relationship is depicted in Figure 2, and more detailed below.

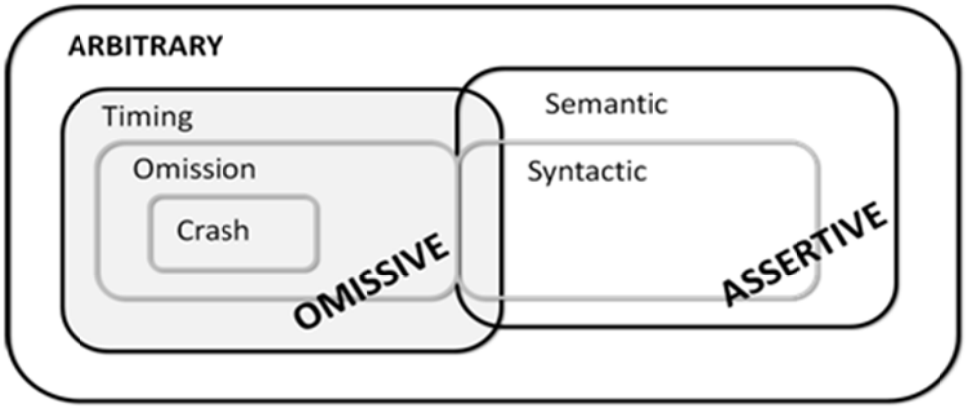


Figure 2 - Relationship between classes of faults.

Omission faults include crash faults - Since a crash fault corresponds to a stop of an interaction, if a system can tolerate omissions, in which an interaction never occurs, it can tolerate omissions that occur due to crash.

Timing faults include omission faults - Since an omission is an infinite delay, if a system can tolerate a delay that leads to the violation of a time limit it can also tolerate an infinite delay, due to an interaction that never occurs (omission).

Semantic faults include syntactic faults - Since a syntactic fault occurs when the format is incorrect, it can be also considered a semantic fault where the construction is incorrect, and therefore, if a system tolerates semantic faults it will also be able to tolerate syntactic faults.

Byzantine faults include all other types of faults - Since a Byzantine fault can be described as an arbitrary behavior, if a system can tolerate any of type of behavior it can also tolerate any specific type of faults.

System failures can range from different types of faults and even from the combination of some of the existing. In the case of an intrusion that leads into a component failure, it is the combination of an internal fault, usually called vulnerability, and an external fault, usually called attack. The intrusion occurs when a malicious user is able to attack (external fault) and exploit some existing vulnerability (internal fault) leading into a system or component failure.

The concepts to which we referred above can be defined as [3]:

- **Vulnerability -** Intentional or unintentional fault in a computing or communication system that can be exploited with malicious intention;
- **Attack -** Malicious intentional fault attempted at a computing or communication system, with the intent of exploiting a vulnerability in that system;
- **Intrusion -** A malicious operational fault resulting from a successful attack on vulnerability.

In Figure 3 we present an illustration of the AVI (Attack-Vulnerability-Intrusion) model which represents the occurrence of an intrusion and how it develops into a failure.

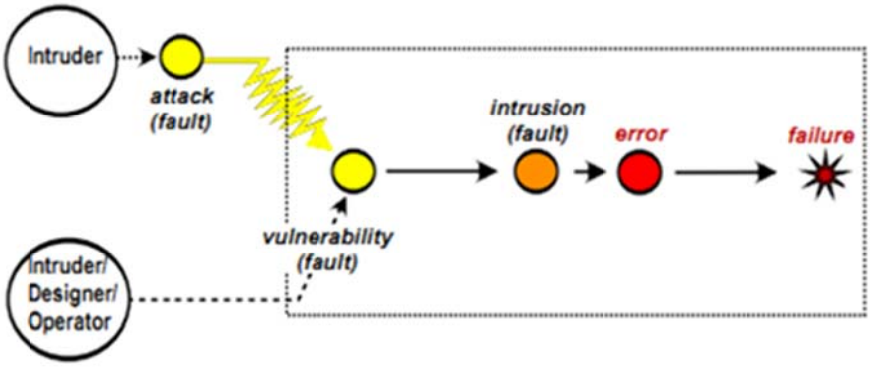


Figure 3 – AVI model.

It describes the intimate relation between an attack and vulnerability. In truth, an existing vulnerability is harmful without any matching attack and an attack is completely inconsequent without target vulnerabilities.

2.1.2 Trust and Trustworthiness

These two concepts are intimately connected with the dependability of a system. To be more precise, they relate strongly with the concepts of "dependence" and "dependability" which are general arguments for the security of a system [9].

Trust is the accepted dependence of one component, on a set of functional or non-functional properties of another component, subsystem or system [10]. Trustworthiness is the measure in which a component, subsystem or system meets a set of functional or non-functional properties [10]. Therefore, a system and its components must be both trusted and trustworthy for a perfect balance between the expectations of its operation and its effective coverage.

Components, systems, applications and even users have a vast set of trust relations between themselves. The inconsistency between the two concepts comes when one of the intervenients proves itself not trustworthy enough for the level of trust that it is granted with, from one, or more, of the other entities. Therefore, for a correct and dependable operation, it is critical for every trusted entity of the relation to be as trustworthy as the level of trust it is related to, thus, providing full coverage over its trust assumptions.

2.1.3 Dependability

Dependability is a very important system property when considering critical infrastructures such as the EDP Distribuição power grid. It can be described as the ability to deliver service that can justifiably be trusted [5]. Therefore, the biggest threats to dependability are faults, errors and failures.

There are several attributes of a system that can be assessed to determine the dependability of a distributed system:

- **Reliability -** The measure of the continuous delivery of correct service;
- **Safety -** The degree to which a system does not fail in a non-catastrophic way;
- **Maintainability -** The measure of the time to restoration of correct service;
- **Security -** Guaranteeing confidentiality, integrity and availability in service provision;
- **Integrity -** Absence of improper system alterations;
- **Availability -** The measure in which a service or piece of information is protected from denial of authorized provision or access;
- **Resilience -** Ability to recover from external faults to recover from external fault.

Clearly, the GENESys platform should be a system as dependable as possible. Since the consequence of a system failure deriving from one of the several types of faults presented can be catastrophic, means to attain dependability should be addressed:

- **Fault removal -** Detecting and removing faults;
- **Fault forecasting -** Predicting the existence of faults;
- **Fault prevention -** Eliminate the conditions for faults to occur;
- **Fault tolerance -** Continuous operation despite the presence of a fault/error.

The first three are related with more classical approaches which are already implemented in the EDP Distribuição infrastructure, or being addressed on projects underway. They focus in a more preventive way of providing a dependable system, since they intend to prevent or detect a fault or error from occurring.

As we referred before, the more recent approach that we are interested on refers to fault tolerance where the objective is to maintain a correct functioning of a system despite the

presence of faults and errors. Fault tolerance can be achieved by the implementation of different mechanisms for error and fault processing.

2.1.4 Fault Tolerance Mechanisms

As it was referred before, a fault-tolerant system is one which is able to behave correctly even in the presence of faults. There are several mechanisms that can be applied in a system to guarantee such capabilities. Below, we describe these mechanisms in detail [10]:

Error detection - these mechanisms aim at detecting the error immediately after it occurs so it is able to confine it to avoid propagation and also to trigger error recovery mechanisms and fault treatment mechanisms.

- **Reactive Detection -** Detects and reports errors at run-time;
- **Proactive Detection -** Detects and reports errors at development-time.

Error/Fault recovery - The objective is to restore a system state with errors/faults into one without them, thus, secure.

- **Backward recovery -** This mechanism takes the system back to a previous state known as correct, and resumes;
- **Forward recovery -** This mechanism takes the system forward to a state where correct provision of service can still be ensured;
- **Error/fault masking –** In this mechanism the system state has enough redundancy that the correct service can be provided without any noticeable glitch.

Fault Handling - The objective is to record the detected faults to avoid later reactivation.

- **Diagnosis -** This mechanism intends to identify and record the cause of an error in terms of location and nature;
- **Isolation -** This mechanism performs physical and logical exclusion of the faulty components from further participation in service delivery;
- **Reconfiguration -** Performs a switch to spare components or reassigns tasks among non-failed components;
- **Reinitialization -** This mechanism checks, updates and records the new configuration and updates system tables and records.

2.2 Byzantine Fault-Tolerant Protocols

The mechanisms presented in Section 2.1.4 are related with the occurrence of accidental faults. However, taking into consideration the reality in which systems are embedded today, they are vulnerable to intrusions from malicious users that may lead to any arbitrary (Byzantine) behavior. The Byzantine fault model is usually addressed by Byzantine fault-tolerant (BFT) replication mechanisms which aim on providing systems with intrusion tolerance capabilities. An intrusion-tolerant system is one that maintains its security properties (i.e., confidentiality, integrity and availability) despite some of its components being compromised by an adversary [4].

Most of the intrusion tolerance mechanisms are based in protocols that apply Byzantine state machine replication [11]. More specifically, to robust critical services, these techniques will replace a single server by a set of state machine replicas. The concept of Byzantine state machine replication is defined as a state machine that is replicated across different nodes in a distributed system. Each state machine replica maintains the service state and implements the service operations [12].

In this section we present different techniques that can be applied to provide intrusion tolerance capabilities to distributed system. The earlier work related with BFT lead to techniques designed to demonstrate theoretical feasibility but that were too inefficient and too expensive to be used in practice in asynchronous systems. However, in 1990 the Practical Byzantine Fault Tolerance (PBFT) protocol was proposed as a Byzantine fault-tolerant state machine replication algorithm designed to work correctly in an asynchronous system and improving the performance of previous algorithms by more than an order of magnitude [12]. More recently, other practical algorithms have been proposed which may be good solutions for our future proposal, such as Zyzzyva, which is very performant due to its speculative execution, and MinBFT which requires a smaller number of replicas and has a reduced number of communication steps.

Considering we are addressing a near real-time [1] distributed system and our interest is implementing the most performant and practical protocol, we will describe the referred algorithms in the next sections, however, other BFT protocols are available.

2.2.1 Practical Byzantine Fault Tolerance

During the 90s, some systems were proposed to support BFT replication. However, the protocols were too expensive to be even considered, since they required a high number of replicated servers. In 1999, Miguel Castro and Barbara Liskov introduced the Practical Byzantine Fault Tolerance [12] (PBFT) algorithm, which provides high-performance Byzantine state machine replication, processing thousands of requests per second with sub-millisecond increases in latency [13], and, more importantly, at lower cost, making it, as the name refers, “practical”.

The PBFT protocol is based on the following system model:

- Asynchronous distributed system where nodes are connected by a network;
- Byzantine failure model:
    a) Faulty nodes behave arbitrarily;
    b) Independent node failures.
- Cryptographic techniques to prevent spoofing and replays and to detect corrupted messages;
- Very strong adversary;
- 3f+1 number of replicas.

The protocol is based on state machine replication and may be applied to any deterministic replicated service with a state and a number of operations. By deterministic we refer to systems in which the same output state will always be produced from a given starting state and input, hence, no randomness is involved in the development of future states of the system [14].

PBFT provides safety [15] over an asynchronous network over the assumptions that only f replicas can be simultaneously faulty, for a *3f+1* number of replicas, and only guarantees liveness in synchrony systems:

- **Safety -** The system maintains state and looks to the client like a non-replicated remote service. Safety includes a total ordering of requests.
- **Liveness -** Clients will eventually receive a reply to every request sent. If a request from a correct client does not complete during the current view, then a view change occurs.

The protocol uses consensus and propagation of system views: state is only modified when the functioning replicas agree on the change.

The algorithm has two operations modes, the normal operation and view-change. The first represents the full procedure from the client request until it receives the correct result from at least *f+1* of the replicas. Figure 4 shows the operation of the algorithm in the normal case of no primary faults where replica 0 is the correct primary and replica 3 is faulty.

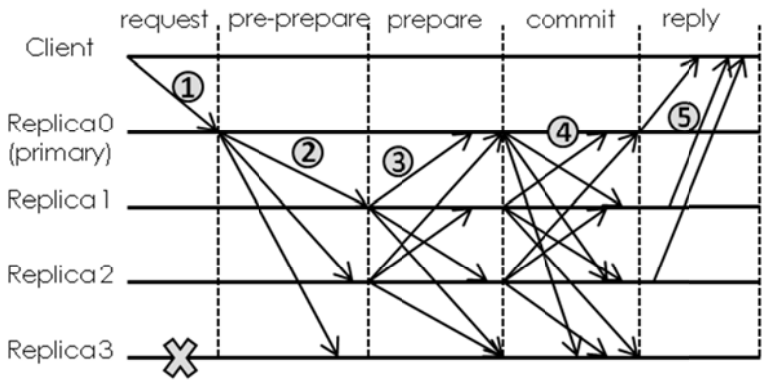


Figure 4 - Normal Operation Mode **[12]**.

1. ***Request*:** Client sends a request to the primary. The primary can then validate the message and propose a sequence number for it;

2. ***Pre-prepare*:** Primary sends pre-prepare message to all backups. This allows the backups to validate the message and receive the sequence number;

3. ***Prepare*:** All functioning backups send prepare message to all other backups. This allows replicas to agree on total ordering;

4. ***Commit:*** All replicas multicast a commit. The replicas have agreed on an ordering and have acknowledged the receipt of the request;

5. ***Reply:*** Each functioning replica sends a reply directly to the client. This bypasses the case where the primary fails between request and reply.

The second operation mode corresponds to the view-change protocol which provides liveness by allowing the system to make progress even when the primary fails. Figure 5 illustrates the view-change operation where the primary replica 0 is faulty.

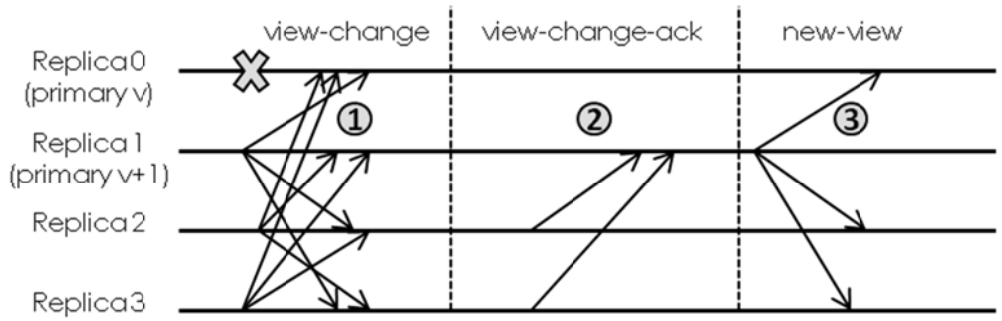


Figure 5 -View-Change Mode**[12]**.

1. ***View-change*:** A backup replica triggers the view change protocol if it stays with some pending message *m* for more than a certain time limit (request timeout expires);

2. ***View-change-ack*:** A replica sends a view-change-ack to the primary of the next view (*v+1*) but the new primary will only accept a view-change from a replica if it receives *2f-1* view-change-acks for it from other replicas;

3. ***New-view*:** Once *v+1* has seen *2f* view-change messages, it multicasts a new-view message. This message contains all the valid view change messages received for *v+1* as well as a set of all requests that may not have been completely ordered yet (due to primary failure).

For the algorithm to work correctly, the client needs to be BFT-aware. They must implement timeouts for view-change and must wait for replies directly from the replicas. Once the clients receive *f+1* replies, the results are accepted and guaranteed correct, even if in the presence of *f* failed replicas.

2.2.2 Zyzzyva

The Zyzzyva protocol [16] uses speculative execution to reduce the cost and simplify the design of Byzantine fault-tolerant state machine replication. By applying such a method it also improves latency and throughput, being arguably the fastest protocol one can devise for ordering requests under the Byzantine fault model. Inspite of that, the protocol continues to ensure the correctness properties of liveness and safety.

- **Liveness -** The protocol guarantees liveness only during periods of synchrony. If a primary is correct when correct client issues a request, the request is completed. However, if a request from a correct client does not complete during the current view, then a view change occurs. This guarantees that if a request is issued by a correct client it eventually completes.

- **Safety -** The agreement sub-protocol is safe within a single view and the agreement and view change protocols together ensure safety across views.

The main difference from its base protocol, PBFT, is that it does not require the expensive three-phase commit protocol to reach agreement on the order in which the request must be processed. The client sends the request to the primary which will immediately deliver it to the remaining replicas. They optimistically adopt the order proposed by the primary and respond directly to the client, who will then be responsible for detecting inconsistencies, reject inconsistent replies, help convergence of correct replicas to a single total ordering of requests, and only rely on responses that are consistent with total order.

There are two different behaviors during the protocol operation. First, gracious execution where there is a normal operation without any faulty replica, represented in Figure 6, and when there is one faulty replica, represented in Figure 7.

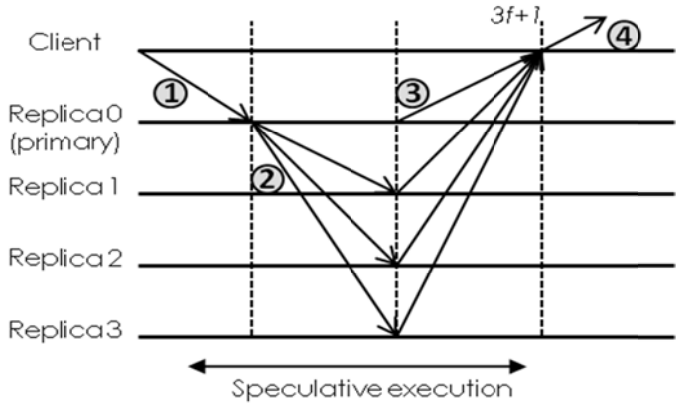


Figure 6 - Normal operation **[2]**.

In the Gracious execution, the following steps are taken:

1. The client sends the request to the primary;
2. The primary forwards the request to the replicas;
3. Each replica (speculatively) executes a request just after receiving the sequence number of this request by the primary. After executing the request the replicas send their response to the client;
4. The consistent state of *3f+1* lead the client to consider the request complete and it acts on it.

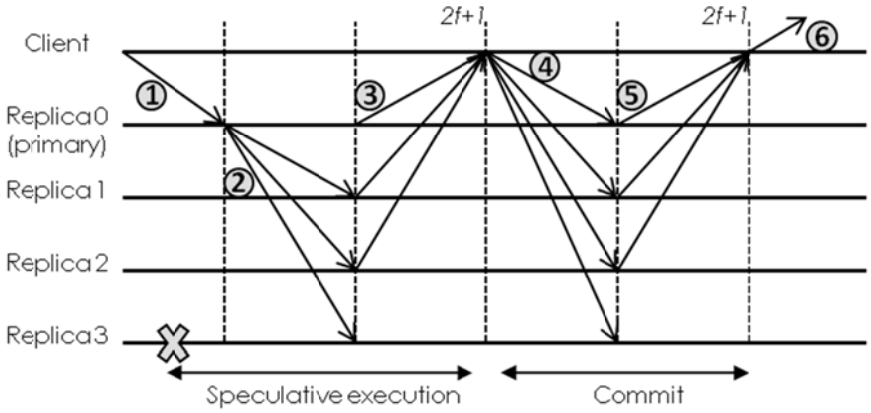


Figure 7 - Gracious execution **[2]**.

In the presence of a faulty replica or asynchrony, the following steps are executed:

1. The client sends the request to the primary;
2. The primary forwards the request to the replicas;
3. Each replica (speculatively) executes a request just after receiving the sequence number of this request by the primary. After executing the request the replicas send their response to the client;
4. The client receives between *2f+1* and *3f* mutually-consistent responses, then the client gathers 2f+1 responses and distributes a commit certificate to the replicas;
5. In this step two different scenarios may occur:
6. *2f+1* replicas acknowledge receiving a commit certificate;
7. If a sufficient number of replicas suspect that the current primary is faulty, then a view change occurs where the replicas roll-back into a safe state and a new primary is elected;
8. The client considers the request complete and acts on the corresponding reply.

Notice that Zyzzyva may require replicas to be able to rollback their states, which constrains the services that it can implement.

2.2.3 Minimal Byzantine Fault Tolerance

Some of the limitations for the popularization of the intrusion tolerance paradigm are the costs usually involved with its implementation. The high number of necessary replicas, the diversity among them, and the complexity of the protocols all contribute to this problem. The Minimal Byzantine Fault Tolerance (MinBFT) is a more recent algorithm, designed over PBFT, which comes as an effort to overcome such constraints, since it improves the previous algorithms in terms of several metrics, such as the number of replicas, trusted service simplicity and number of communication steps [2].

Number of replicas - Typically, the BFT protocols require *3f+1* replicas to be able to tolerate f faulty replicas. This algorithm is more efficient since it reduces the number of required non-faulty replicas to be applied for majority voting to *2f+1.*

Trusted service simplicity - The system requires only *2f+1* replicas because it leverages a local trusted/tamperproof component. The Unique Sequential Identifier Generator (USIG) is a local service that exists in every server. It assigns to messages unique, monotonic and sequential identifiers for each server. It is implemented as a trusted service that provides an interface with operations only to increment a counter and to verify if other counter values incremented

by other replicas are correctly authenticated. This way, a malicious replica will not be able to make different correct replicas execute different operations with a given sequence number assigned by its USIG.

Number of communication steps - Replicas should be deployed in different geographic sites to guarantee service in case of disasters and large scale DDoS. Thus, a replicated system operating in a WAN-of-LANs might have a delay in communication and consequently an increase in latency when compared with a system operating in a LAN. To reduce the limitations of such architecture, the MinBFT algorithm reduces the number of communication steps required for its normal operation and view-change. The primary uses the trusted counters in the USIG to assign sequence numbers to client requests. Furthermore, it produces a signed certificate, based in message authentication codes (MAC), to ensure that all non-faulty replicas can be sure that all messages with the same identifier carry the same content, and ultimately, agree on the same order for the execution of the requests.

As we referred, the message exchange pattern of MinBFT is similar to the one in PBFT, but, it will only require *2f+1* replicas to ensure safety. Liveness will only be guaranteed for synchronous systems. We will next present the sequence of events representing the normal case operation, in Figure 8.

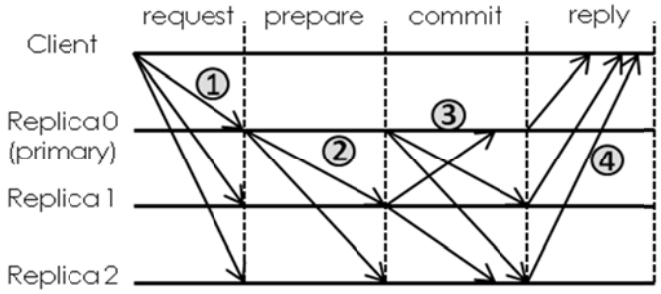


Figure 8 - MinBFT Normal Operation **[2]**.

The MinBFT has only two communication steps while PBFT has three. This is due to the fact that the primary uses the trusted counters to assign a sequence number to each request, for which the tamperproof component also produces a signed certificate that proves unequivocally that the number is assigned specifically to that message and that the counter was incremented [17]. This number is the counter value returned by the USIG service in the unique identifier UI.

1. ***Request***: A client sends a request to all servers;

2. ***Prepare:*** The primary assigns a sequence number to the request and sends it to all servers in a prepare message;

Chapter 3

GENESys System

This chapter provides a summary of the EDP Distribuição power grid management infrastructure. It first presents some context about the EDP Distribuição and the service it provides. Furthermore, the paradigm of the supervisory control and data acquisition (SCADA) system in the company is introduced as well its architectural and operation description. This system is presented as GENESys, a combination between the SCADA capabilities with the ones provided by a Distribution Management System (DMS). Moreover, some other auxiliary systems interconnected with GENESys are presented, as capability enhancers for the platform.

3.1 Context

EDP is the biggest company of Portugal. It is an electrical and gas company present in the Iberian peninsula, Brazil and US markets and is the only Iberian utility that has generation, distribution and supply activities. One of EDP's companies is EDP Distribuição that is responsible for the electric distribution in Portugal where the sales market is fully liberalized. It holds the concession for the exploration of the national distribution electrical power grid in Medium Voltage (MV) and High Voltage (HV), and municipal concessions of distribution of electricity in Low Voltage (LV).

The EDP Distribuição electrical power grid is managed by a structured information technology (IT) infrastructure. Within this infrastructure the GENESys system stands out since it is responsible for the near real-time monitoring and remote control capabilities over the power grid, providing global network supervision and the restoration time reduction based on remotely executed actions.

The GENESys system is composed by several components dispersed over three different layers:

- **Telemetry Site -** This is the lowest layer of the architecture. It corresponds to the electrical power grid facilities (e.g., substations, medium voltage secondary

substations and pole mounted auto-reclosers) which are endowed with automatic measurement, transmission and reception of data to and from the backend systems, known as telemetry. The telemetry capability is provided by the deployment of Remote Terminal Units (RTU) which interfaces between the physical electrical components and the backend systems.

- **Frontend Site -** It corresponds to the middle layer between the RTUs and the backend systems where the Frontend servers are deployed. The Frontend (FE) is a data collector and protocol translator. It is responsible for collecting and translating all the signals sent by its mapped RTUs, delivering them correctly to the backend systems, and, in reverse, collects and translates the signals received from the backend systems and deliver them to the correct RTU. It also includes a watchdog component to manage the Frontend redundancy at each site.

- **Systems Site -** It is the upper layer of the GENESys architecture. It contains the supervisory control and data acquisition (SCADA) system and the Distribution Management System (DMS). These systems are integrated in a unique platform, GENESys, which is operated by the power grid operators to monitor and remote control the power grid, i.e., the Telemetry Sites. We can also find a watchdog component at this layer to manage the SCADA and DMS redundancy. Furthermore, there are the Workstations which provide the applicational interface between the systems and the power grid operators. The GENESys application running at each workstation receives near real-time information from the SCADA and DMS servers.

The GENESys system is critical for a high quality of service in electricity distribution, however, with time, other systems were included to provide new capabilities, further robustness, more efficient operation and the registration of all events. All these systems together with GENESys shape the EDP Distribuição electrical power grid IT management infrastructure and are strictly connected to the GENESys private network. For security reasons, and considering the systems criticality, EDP Distribuição has separated this infrastructure from the corporate network.

We will provide further insight and description of the referred components in the next sections.

3.2 History

The first investments of EDP Distribuição concerning the automation and supervision of the power grid were done more than 25 years ago. At that time the company structure was very different from what it is today, it was divided into five different companies, representing the different geographical areas. At that time, each of the five companies had an independent and unstandardized basic SCADA system.

In 2000, all of the companies merged into what it is called today EDP Distribuição, which manages the whole Portuguese distribution power grid. Since the company was now as one, a new SCADA system was deployed as a standard, in architecture and operation. Despite the unification, there still remains until today a very clear separation between the north and the south region of the country. There are two twin SCADA systems, one for each region, totally separated. In spite of that, investments and upgrades had been applied for both systems guaranteeing that the evolution of the information technology attain them both, maintaining uniformity.

Nowadays, we have a far more complex system which is called Generation Network Information System (GENESys). The GENESys system interconnects the SCADA system functionalities with the more advanced Distribution Management System (DMS) features.

3.3 EDP Distribution System Architecture – GENESys

The GENESys system was born in 2001, when the EDP Distribuição decided to enhance the basic SCADA system, at the time, with distribution management capabilities. Since the SCADA system itself could not support the new features, the GENESys platform emerged as the combination of the existing SCADA system and the new DMS system. It is an application which is permanently in contact with the SCADA and DMS servers to be able to perform both systems features.

The SCADA infrastructure was maintained, and the DMS capable servers were added to the distributed system. As it was previously referred, the GENESys system architecture is divided in three different layers which are depicted below in Figure 9. In the next sections we explain each component in detail.

As we previously explained, the system is still separated into two geographical references, the North and the South. In EDP Distribuição there are two GENESys systems, one for each region.

3.3.1 Remote Terminal Unit

The Remote Terminal Unit (RTU) is a microprocessor-based device which works as an interface between the physical electrical components and the SCADA systems. It is located in the lower layer of the SCADA infrastructure, at the electrical facility and has two different purposes:

1. Collect data from sensors located at the electrical facility, convert them into the digital domain and send them to the upstream SCADA system.

2. Physically apply the remote controls received by the power grid operators.

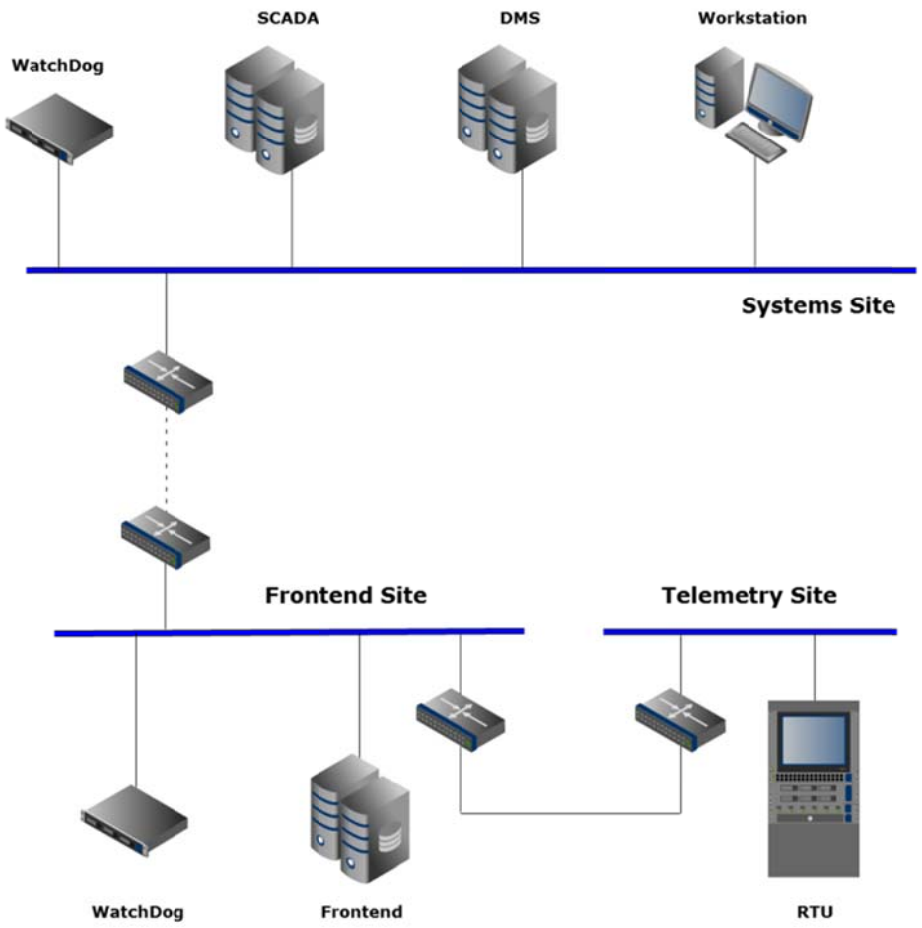


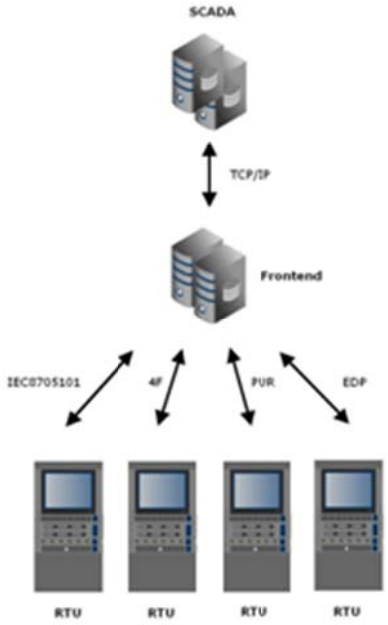
Figure 9 - GENESys architecture.

The RTUs are installed in different electrical facilities of EDP Distribuição such as substations, medium voltage secondary substations and pole mounted auto-reclosers [18]. The RTU consists in several units (Central Unit, Acquisition and Command Unit, Human-Machine Interface), which, taken together, provide the implementation of local automation and the supervision and control of the telemetry site, on site or remotely. The RTU must be constructed with materials capable of withstanding the mechanical, electrical, thermal stress and also the effects of humidity, which can be found in the operating conditions of the Telemetry Sites.

3.3.2 Frontend

The Frontend is a data collector and protocol translator. It is responsible for translating all the signals sent by its mapped RTUs, which are encapsulated in different communication protocols, to the protocol used to communicate with the backend SCADA system, and vice-

versa. The Frontend is considered as an intermediary between the RTUs and the SCADA system, as depicted in Figure 10.



ɔ)

Figure 10 - a) Frontend Communication Flow; b) RTU device.

3.3.3 SCADA System

The supervisory control and data acquisition (SCADA) systems are vital components of most critical infrastructures since they provide the required management capabilities with real-time data, for efficiency and availability. The SCADA system in EDP Distribuição is supported by a relational database where the whole portion of the automated part of the EDP electricity distribution power grid is reflected.

The remote controlled and monitored infrastructures, i.e., Telemetry Sites, correspond to the lower level of the chain of communications where Remote Terminal Units (RTUs) are installed. These components are responsible for monitoring the digital and analogical domain of the equipments installed, transmitting the information to communications Frontends, which in their turn will process all the data received in real-time, being responsible for its management.

The SCADA servers use software to acquire and oversee all controlled variables, being responsible for managing all the information received from the Frontends, ensuring a robust and accurate interface with the operator, making ongoing management and real-time database management, alarms, archiving and records of occurrences.

3.3.4 DMS System

As the complexity of the power grid grows, the SCADA system features become insufficient to answer the increasing demands over the quality of service and availability of the services provided by EDP Distribuição. It became imperative to integrate the SCADA system with a secondary system which could bring a real-time advantage for maintenance, planning and operation of the power grid, and that could also increase the efficiency of energy distribution. The distribution management system (DMS) was such system.

The most important features of the DMS system integration are:

- Graphical representation the entire HV and MV network;
- The graphical staining of the global network by empowerment, voltage level and feeder;
- Electrical tracing from the points of energy feeding until the end loads;
- Registration of the network components (Lines and Cables, Facilities, Cut Bodies, etc.);
- Geographical location of facilities, power lines and equipment.

The DMS servers require a high amount of data transfers since it requires a real-time mapping between the SCADA telemetry entities and the power system elements it has stored in its database. This information includes each component technical details, geographic location, and others. The DMS server also requires the near real-time [1] SCADA information to be able to perform the grid power flow analysis, which is the graphical presentation of the bulk transfer of electrical energy from the electrical sources to the electrical loads.

3.3.5 WatchDog

The WatchDog is a hardware component responsible to trigger a system reset and a server commutation due to an identified fault condition. We have two redundant SCADA and DMS servers running in parallel and synchronized. They operate in fail-over mode, where, at any given time, only one of the SCADA and DMS servers is online in the whole system, the other is in standby ready to replace the online server in the case of failure. The WatchDog is responsible for managing this procedure by monitoring the critical processes of both SCADA and DMS servers. In the case of a failure on the online server, the watchdog executes a mechanical and logical commutation of the control relay to ensure the takeover of the standby server. This mechanism is based on primary-backup replication [19].

3.3.6 Workstation

The workstation is the applicational interface of the power grid operators with the electrical power grid itself. They run the GENESys application which receives real-time information from the SCADA server and information upon demand from the DMS server. They provide all the SCADA and DMS capabilities integrated in the same user friendly interface to be used by the EDP Distribuição power grid operators, as in Figure 11.

Figure 11 - Grid operator managing GENESys workstation.

3.4 GENESys interconnection with other systems

The GENESys is the core system for the supervision and control of the EDP Distribuição power grid. It is a platform that includes both a SCADA and a DMS system to perform their most important features. However, the GENESys system also exchanges information with other parallel systems to acquire new capabilities, further robustness, more efficient operation and the registration of all power grid information and events. These systems are depicted in Figure 12 and will be presented in the next sections.

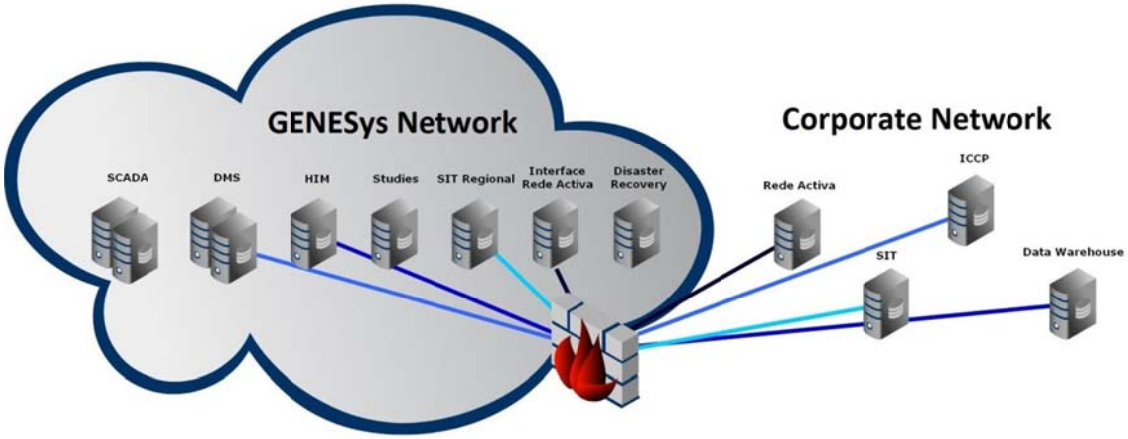


Figure 12 - EDP Distribuição Systems architecture.

3.4.1 Technical Information System

The Technical Information System (SIT) is located in the EDP's corporate network and provides detailed information related with infrastructures and equipments, together with its geo-reference.

The Regional SIT located in the GENESys network is a regional cache (there is one for the north and another for the south) of the unified corporate SIT but only with the essential information for the remote control system of each GENESys region. The information exchange is done through the inter-network firewall, on a daily bases, where the Regional SIT gets updated accordingly to the information residing in the Corporate SIT. Then, the DMS server by daily synchronization with the SIT regional server will have an update on the high and medium voltage network topology. Figure 13 illustrates the described architecture.

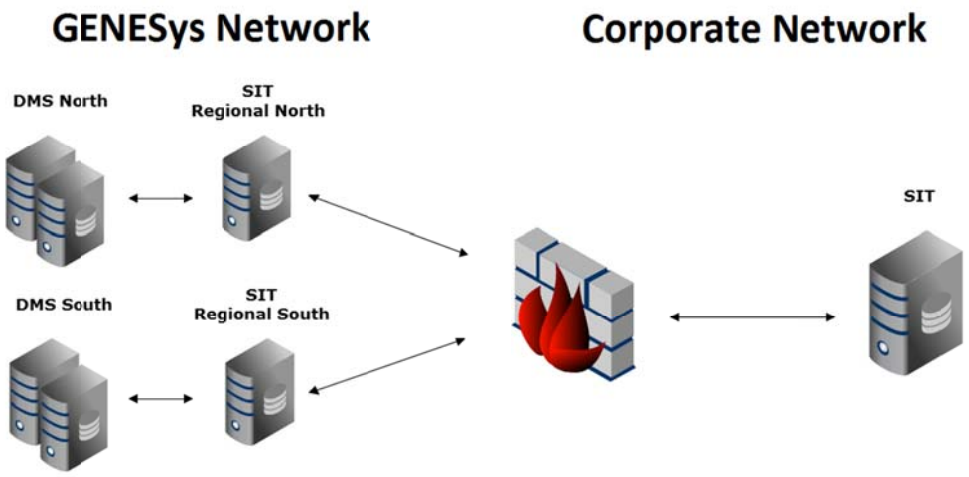


Figure 13 - SIT architecture.

With the integration of the DMS in the SCADA basic representation it is possible to represent the non-telemetered part of network and it also allow the operators to do the update of the operational states of the network in a centralized way.

3.4.2 Rede Activa

The Rede Activa is a system developed for managing, recording and analyzing incidents. It allows the automatic registry of incidents from events detected in the GENESys system, and also manual registry from the complaints collected in the call center.

The main advantage of this product is to concentrate and standardize the recording of incidents (LV, MV and HV) in the same application creating a logical hierarchy association between them. When there is a problem in a MV output from a substation, such a malfunction is detected by the RTU in the site which will then send that information to the SCADA system. Then, this information will be sent to the DMS server which will process the event and sent the required information to the Rede Activa Interface, so that an incident can be automatically opened on the Rede Activa server. Once a call center operator receives a call from a costumer indicating a problem with its service, while registering that incident manually in Rede Activa, it will automatically associate the incidents, considering that, the reason for the costumer call was the incident already automatically registered from the information provided by the system. The Rede Activa architecture is depicted in Figure 14.

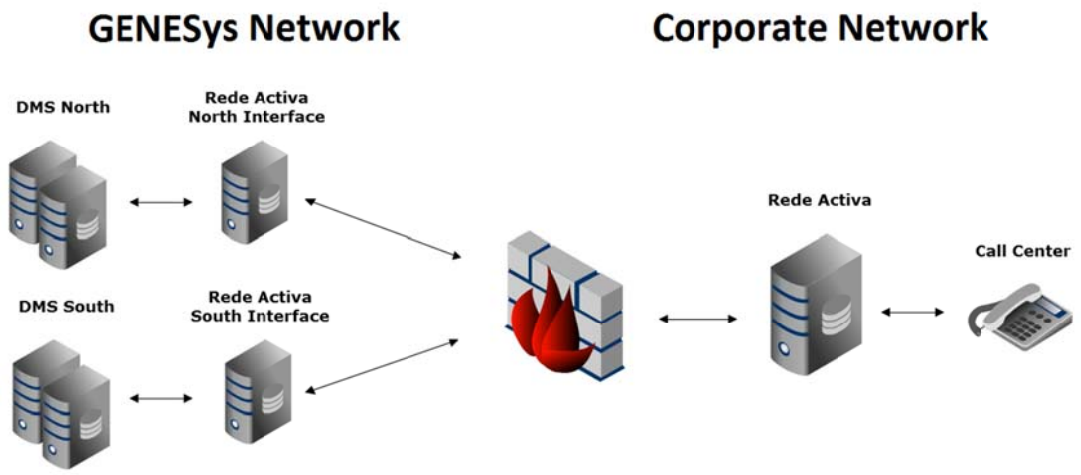


Figure 14 - Rede Activa architecture.

3.4.3 Historical Information Manager

The Historical Information Manager (HIM) server is responsible for storing the historical data of the system, which might be different types of events such as commands, state changes, measures, and others.

The events are initially registered in the SCADA server since it is the responsible for processing all the information first. Then, the HIM server is updated every fifteen minutes with the data acquired in the last fifteen minutes period by the SCADA server. That data loaded periodically to the HIM server remains stored in the SCADA system within the fifteen days following their occurrence. Afterwards, the data is purged from the SCADA system, and kept only the HIM server. The historical information is available through the GENESys application in the workstations, or from WebLists, i.e., web service directly connected to the HIM server and accessible in the GENESys network via web browser. The HIM server interactions are illustrated in Figure 15.

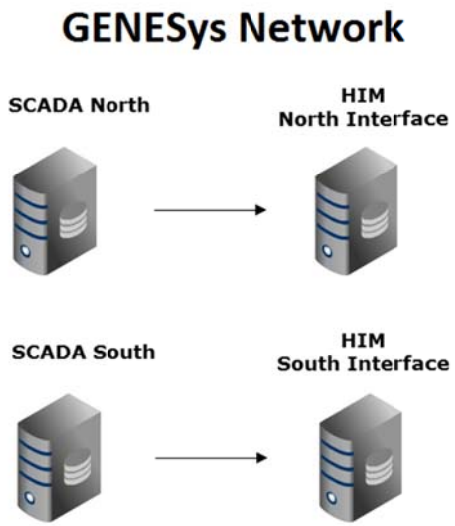


Figure 15 - HIM architecture.

3.4.4 Disaster Recovery

The Disaster Recovery (DR) is a system that ensures continuity of service and business, in the event of a failure or disaster on the most critical systems, with an intricate system of "*backups*". The system that EDP Distribuição had implemented includes two storage servers, one in each region, which will periodically exchange information with the most critical servers, and two DR servers responsible for assuming the role of any critical server that fails. The DR systems is depicted in Figure 16.

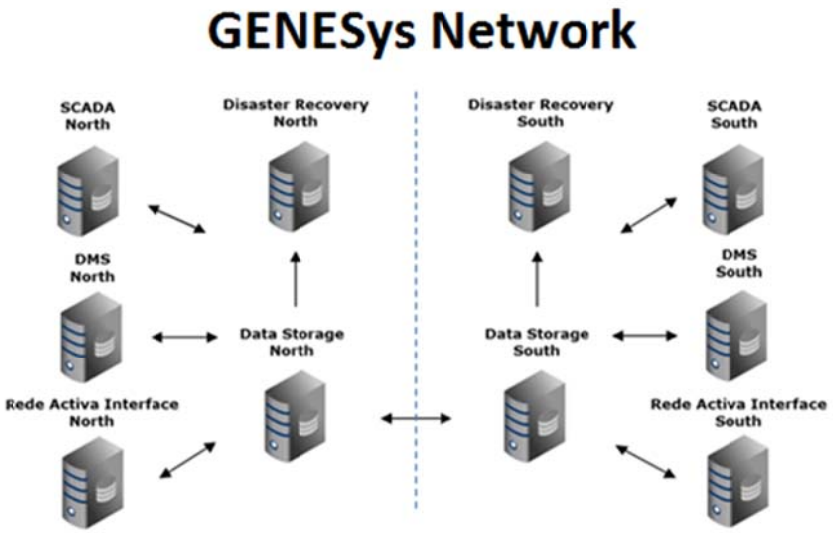


Figure 16 - Disaster Recovery architecture.

This way, if required, the DR system will be capable to take over the functions of any of the critical server as if it was the actual one. The disaster recovery solution will be thoroughly addressed in Section 3.8.

3.4.5 Data Warehouse and Business Intelligence

The Data Warehouse (DW) is a centralized repository for enterprise data used for reporting and analysis [20]. The data stored in the DW is uploaded from the SCADA servers, HIM and other external sources. The Business Intelligence (BI) application, available on the corporate network via web browser, is capable of generating intelligent queries over the data residing on the DW server. Its design allows simple and facilitated analysis, reports and key performance indicators that require the processing of a large amount of data [21]. The DW and BI service architecture is illustrated in Figure 17.

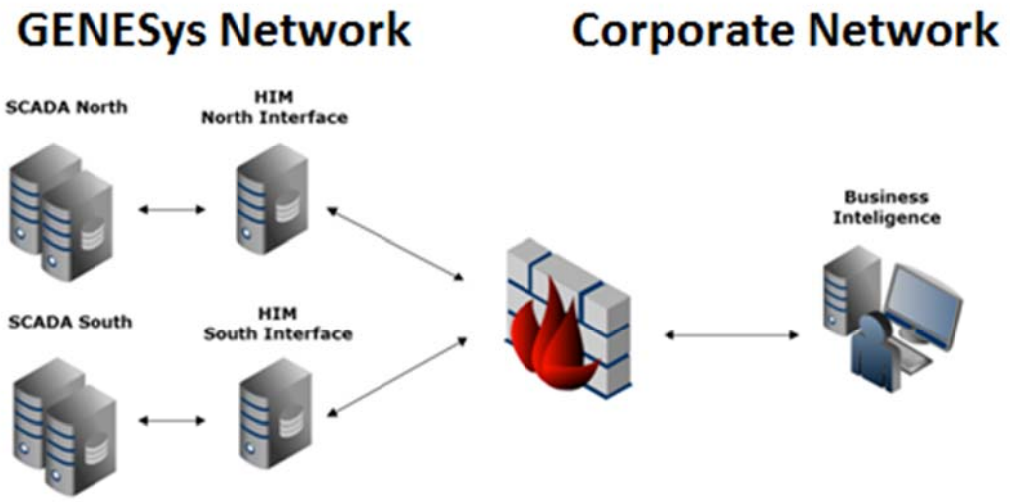


Figure 17 - Data Warehouse and Business Intelligence architecture.

3.4.6 ICCP Server

The ICCP is an Inter-Control Center Communications Protocol [22] that allows data exchange between different command centers. The ICCP server processes information of each of the control centers adapting it to the system and communication infrastructure of the other, as depicted in Figure 18. The EDP SCADA server receives information of the measures of buses and the status of circuit breakers, disconnectors and inter-bar panels of the Redes Energéticas Nacionais (REN) substations, so it can represent them in the GENESys system as if they were EDP Distribution infrastructure components.

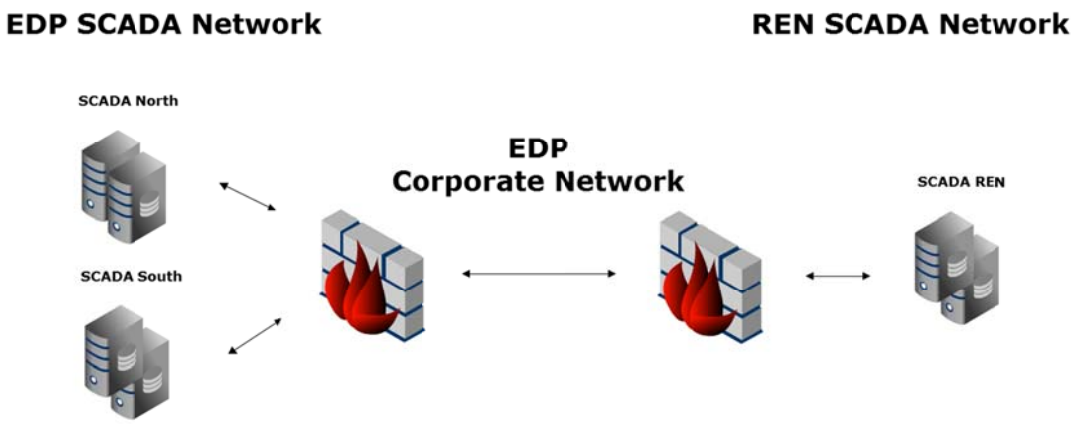


Figure 18 - ICCP architecture.

3.4.7 Studies

There is a learning environment that is identical to the real-time production environment, only differentiated by the background color which is blue instead of black, so that the operator can easily identify which world he is operating (production or study).

The main objective of Studies is to create a study environment where the operators and technicians can perform different procedures without affecting the production environment and potentially the power grid infrastructure. This experimental environment can be adapted from a snapshot of the network status and its measures in real-time or programmed for a specific historical date.

The study snapshots are based on the HIM and DMS servers information, therefore, its correct operation is dependent on the interactions with other servers, as shown in Figure 19.

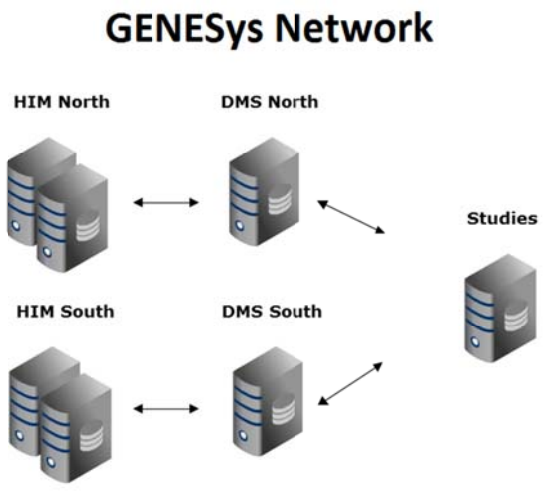


Figure 19 - Studies architecture

3.5 EDP Distribuição Infrastructure

After having introduced all the systems which are part of the EDP Distribuição infrastructure, we present the complete IT architecture in Figure 20. We intend on giving an overall architectural idea of how systems presented in the previous section are distributed and how they are interconnected.

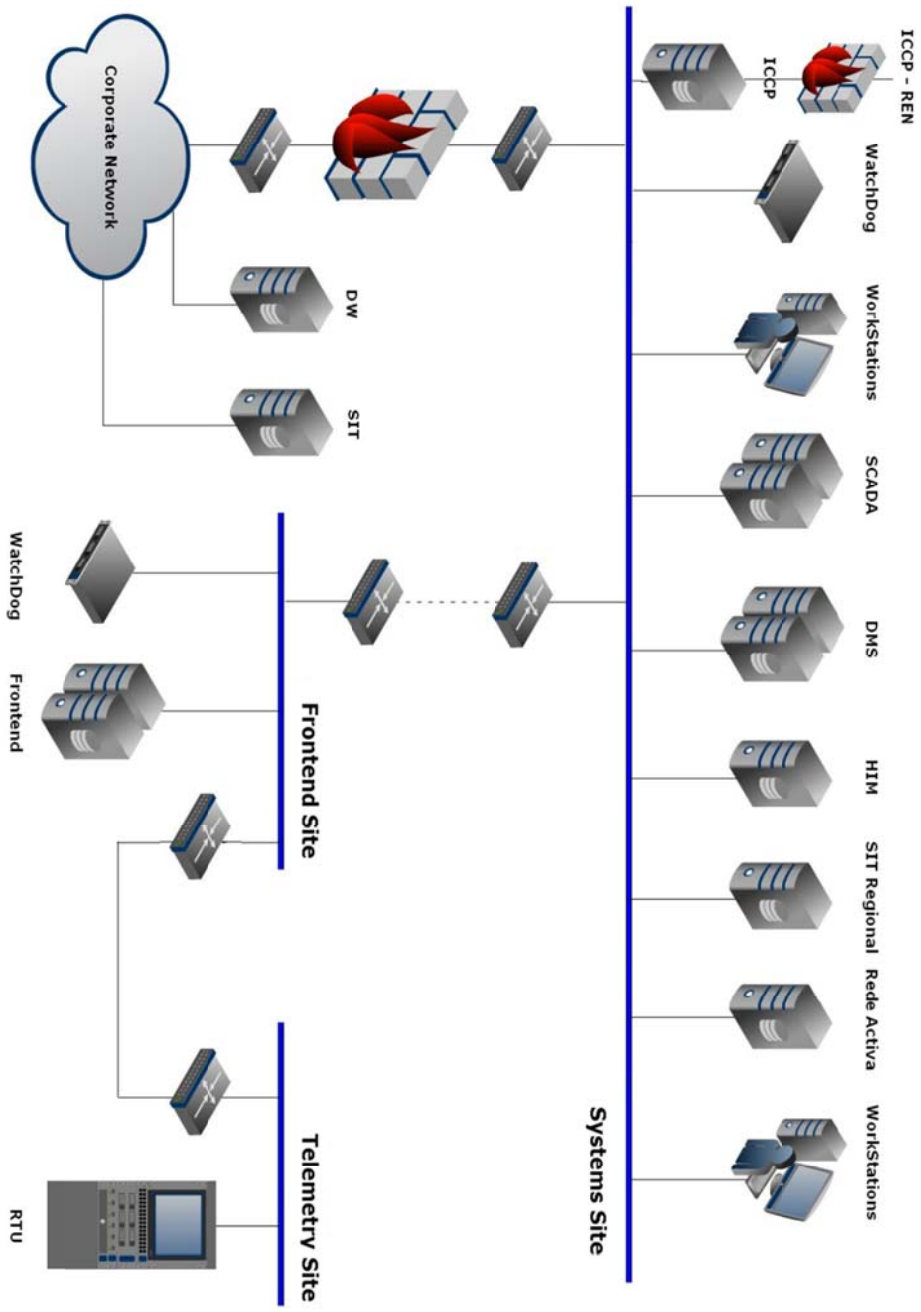


Figure 20 - EDP Distribuição Power Grid IT architecture.

3.6 GENESys Network

The GENESys network can be described as a private WAN network interconnecting all the GENESys sites. This WAN is composed by several LANs, one for each Frontend Site, following the WAN-of-LANs model currently used for critical infrastructure [23]. Every LAN communication is managed independently.

As it is possible to observe in Figure 21, it is a mesh network in which all sites have at least one redundant connection. The link between the backend systems regions, *Porto* and *Palhavã*, is stronger when compared to others, for the data transfer requirements of the Disaster Recovery system.

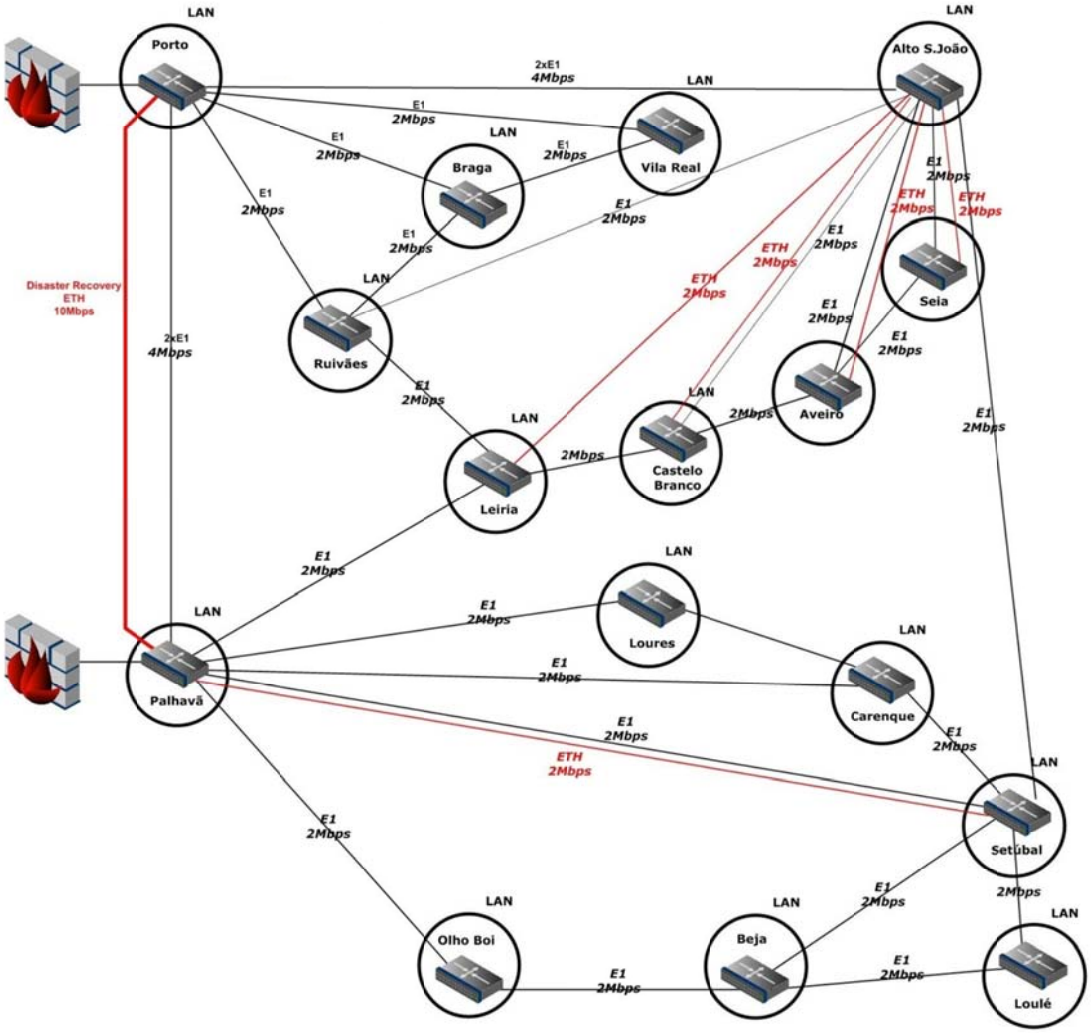


Figure 21 - GENESys Network.

3.7 System Data Flows

There are several communication flows in the system which have to be thoroughly comprehended so that they are guaranteed by any future proposed architecture. The correct operation of the management infrastructure is dependent on the set of interactions between the system components which have to be executed in time and correctly, not to jeopardize the monitoring and control of the EDP Distribuição power grid.

We will analyze the different data flows carried by the system, which will be divided in two types, the reactive and the proactive data flows.

The reactive data flows are related with data streams triggered by manual actions from the grid operators or by telemetry events in the power grid. The proactive data flows regard all scheduled tasks performed by the system, crucial to guarantee its trustworthiness, availability and reliability.

It is important to consider that all communication flows are validated by ACK messages confirming their correct reception. Otherwise, systems would not have a way of validating the success of the communications. If the ACK message is not received until the timeout, the data will be re-sent at most N times or until an ACK is received. When N is reached, a communication error will be displayed.

3.7.1 Proactive Data Flows

There are continuous, periodical and scheduled procedures that require a robust and dependable network to sustain the high quantity of traffic associated and to ensure the correct delivery of the information. These procedures are not caused by any power grid or operator activity, but scheduled for the correct functioning both of the GENESys system and its parallel support systems. The proactive data flows of the GENESys network are depicted in Figure 22, and these flows details are presented in the following:

1. The Watchdog is responsible to validate and police the SCADA redundant servers. The Watchdog sends periodic probe request messages to both SCADA servers to verify if they are working correctly. It will then wait for the "*I'm alive*" message from the servers. If the timeout is reached, the Watchdog will consider the server faulty. There are two different procedures for this failure, depending on the failed server: 1) if the online server fails on responding, the Watchdog automatically commutes to the standby server and sends an alarm to the GENESys system; 2) if the standby server fails, the Watchdog will only send the message to the GENESys system.

2. The DMS server provides the distribution management capabilities. Most of its functionalities depend on its connection and synchrony with the data in the Regional SIT machine. The DMS server will daily download specific technical and geographical information from the Regional SIT to its local database so it can later use it to provide the DMS functionalities. These functionalities are described in Section 3.3.4.

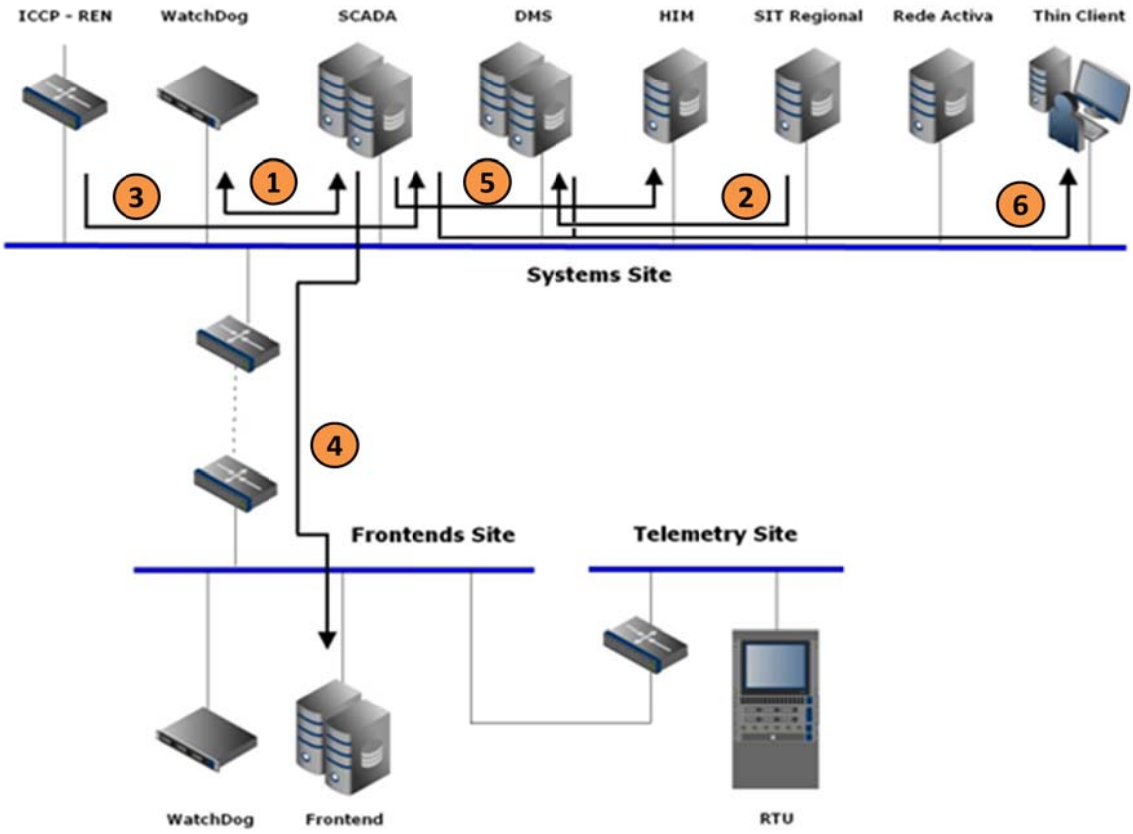


Figure 22 - Proactive data flow representation.

3. The ICCP Server is responsible for interconnecting different command centers. In EDP we are receiving information from the REN SCADA that we use for analysis and treatment of incidents. The telemetry data from the REN SCADA is being received in near real-time [1], limited to the entities corresponding to the interconnection of the REN power grid and the EDP power grid.

4. Concerning the GENESys access control there are three system profiles corresponding to the different roles and permissions one might have to the technical functionalities of the system: *user*, *operator* and *administrator*. The system Administrator profile allows the user to perform major system and database modifications which sometimes have to be replicated to the GENESys lower levels components to be correctly acknowledged and fully functional. These administrative operations are

done in the SCADA interface of the GENESys application and then replicated to the required components. One of the scenarios which mandate such a procedure is when someone with *administrator* permissions performs changes on the RTU information or the Frontend communication ports parameters. This information has to be delivered to the Frontend so that the whole system maintains its connectivity. The update is done automatically once the administrator validates the changes.

Another scenario is when telemetry entities are inserted, updated or deleted from the system. As explained, the GENESys maps entities between the SCADA and DMS servers. Therefore, whenever a change is made in such a context, the DMS machine will be automatically updated so it respects the SCADA server information and maintains the system consistency.

5. The HIM server is responsible to store all the historical information of the system. All the events, from system to communication events, to telemetry and commands, will be sent periodically to the HIM server by a SCADA gateway process in the SCADA server.

6. The workstations execute the GENESys application. They receive near real-time information from the SCADA server and information upon demand from the DMS server. Most of the DMS features are executed upon request of an operator, only then the application will request the DMS server the data it requires. The SCADA is constantly delivering information to the workstations since it regards critical information for a fully functional and reliable SCADA system.

3.7.2 Telemetry Data Flow

Telemetry describes the remote measurement and reporting of physical events in the electrical power grid. Therefore, the telemetry data flow corresponds to the procedure in which an RTU deployed in one of EDP Distribuição electrical facilities captures a physical change or measurement variation, and transforms it into the digital domain so it can be delivered to the backend systems. The telemetry data flow is presented in Figure 23 and explained in detail in the following:

1. The RTU, which is monitoring the physical components of the infrastructure, detects a change of state of a telemetry entity, or a measurement variation. This physical change will be interpreted and converted into digital data by the RTU. This data will include a unique identifier, associated with each telemetered entity, and the actual state which triggered the telemetry change. The digital information is then encapsulated in one of the adopted communication protocols.

2. A message with the telemetry data will be sent to the Frontend Site through the radio frequency (RF) or General Packet Radio Service (GPRS) backbones, for the MV

substations or reclosure sites, or optimal Synchronous Digital Hierarchy (SDH) [24] backbone for the HV substations, as described in flow 1.

The Frontend server will translate the protocol into TCP/IP and associate an RTU identifier with it, based on the communication port from where the packet was received. The association between the communication port and the RTU number is processed in the Frontend database.

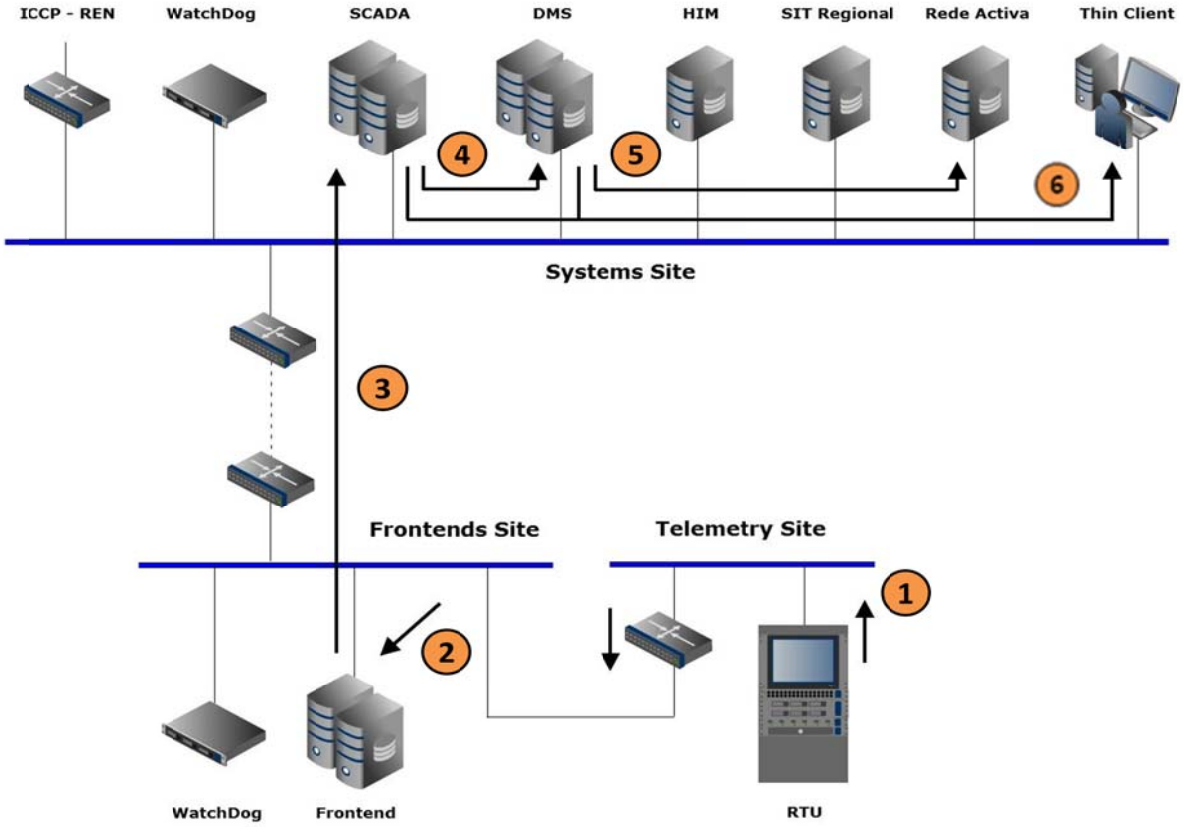


Figure 23 - Telemetry data flow representation.

3. Following, the message will be sent through the SDH backbone to the SCADA Online server. This SCADA server will then replicate the information received to the Standby server to guarantee synchronism between both redundant servers.

4. The information received gets processed in the SCADA server by a component called SCADA Gateway. This gateway triggers automatic procedures subsequently to the telemetry arrival:

   - It will generate the correspondent state update in the SCADA database;

- Verifies if the change of state is pre-configured to generate an alarm. If it is, it then generates it and delivers to the GENESys workstations;

- The change of state will be registered and stored in the event log of the SCADA servers;

- It sends a telemetry change of state to the DMS servers so it can reflect it in its database. Before sending the information to the DMS, the SCADA Gateway will verify if the change of state generating the alarm has not been corrected in a minimal time period, because if the change of state was quickly corrected, it will not be sent to avoid unnecessary processing for the servers (change of state validation).

5. The change of state validation in the previews step is important since the DMS will send that same information to the Rede Activa server so that the incident gets registered.

6. Finally, once the DMS server receives the change of state of an entity, it will deliver the representation changes over the GENESys workstations, so that the operator can acknowledge them visually in the grid representation, but also in the alarm list.

3.7.3 Control Data Flow

One of the most important functions of the GENESys system is to provide the remote control capabilities to respond speedily to any unexpected event in the power grid. Once the power grid operators identify one of these events, they will trigger a set of remote procedures on the grid in order to correct the problem. This flow is denominated control data flow and is depicted in Figure 24. In this section we discuss how this flow takes place.

1. When the operator executes a given command in the GENESys workstation, the control is sent to the DMS server since is responsible for processing all the logic associated with it. It will verify the current state of the equipment to be acted upon and based on the information in its database, it will validate the inhibit conditions that are configured for that command, i.e., list of conditions in which that remote control should never be executed. If any of these conditions is set, the control will not be delivered and an error message is sent to the workstation. Once the control gets validated, it will be sent to the online SCADA server.

2. The SCADA server will then send the control message to the Frontend using the TCP/IP protocol over the optical SDH backbone. Once the packet arrives the Frontend server it is processed so that it can identify to which RTU it is addressed and determine through which communication port it will be forwarded. Furthermore, before forwarding the

control packet, the Frontend will translate the packet into the protocol correspondent to the identified RTU. All this processing is done using the Frontend database information.

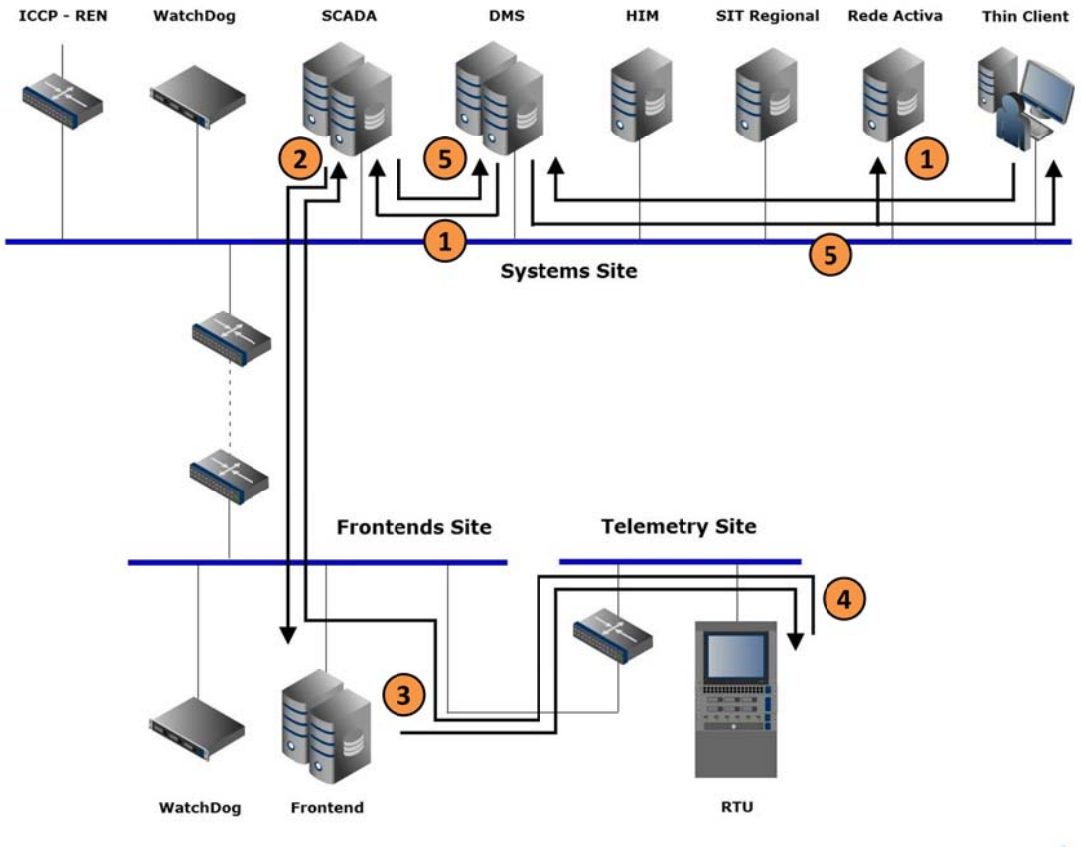


Figure 24 - Control data flow representation.

3. The control will then be sent through RF or GPRS backbones (for the MV substations or reclosure sites), or optimal SDH backbone (for the HV substations), to the local RTU unit which will be responsible for processing the control request and execute it.

4. There is one last important data flow to confirm the successful execution of the command. The SCADA server is waiting for the state change correspondent to the applied control. If the command is correctly executed it will generate a change of state that will be reported to the SCADA server as an event (through the data flow presented in the previous section).

5. Once the SCADA system validates the command it will send a confirmation message to the DMS and the correspondent change of state, which will then execute the procedures already described in Section 3.7.2, to update the workstations representation. Any state change, resulting from a remote control does not generate alarms.

3.8 Disaster Recovery System

Since it was first implemented, the SCADA system proved itself has as a very important asset for the power grid operators. The system provides real-time monitoring capabilities as well as the possibility of remote controlling the different SCADA components. Therefore, with time, the system became critical for the organization and over the last years many concerns have been raised towards their importance for the business continuity.

The Disaster Recovery (DR) system has exactly the objective of ensuring the continuity of service or preparing the recovery of systems that are part of a critical infrastructure, and which, in their absence, undermine the proper operation of the infrastructure as a whole [25].

The DR system follows a disaster recovery plan (DRP) which constitutes official corporation documents that define the resources, tasks and data that are required to manage the recovery procedure [26].

3.8.1 Disaster Recovery Solutions

We will in this section describe the different strategies that can be adopted while deciding a business continuity plan. It is important to balance the advantages and disadvantages of the several solutions and infer about which is the most correct for the system in question [27].

Hot Standby - This is the model of disaster recovery system adopted by EDP Distribuição. It is the fastest recovery model since it usually requires just a few minutes to be available after a disaster situation. This is only possible by the total replication of the critical system covered by the plan.

Warm Standby - This solution is not as fast as the previous one. It can take between 8 to 24 hours to be available since this DR solution is sold as a service, for several different clients. There is a remote recovery centre that, in case of a disaster situation, will deliver the disaster recovery service to the client. Unavailability is dependent on the complexity of the system, the location and the volume of data. It is the most common type of disaster recovery system employed by companies.

Cold Standby - This is the slowest type of disaster recovery solution. It is dependent on the provision of computer and people resources to be made available by the contractor to the service client, within a few hours of the incident.

3.8.2 EDP Disaster Recovery System

The Disaster Recovery system that EDP Distribuição implemented falls on the hot standby strategy. It consists on the deployment of two data storage servers and two disaster recovery

servers, one of each in each region (North and South). In case of failure, the disaster recovery server will be able to take over the functions of each of the covered systems, guaranteeing business continuity.

The whole process of daily synchronization is done automatically with no consequences for the infrastructure and the GENESys system operation. The process of activating the disaster recovery plan, in the case of a disaster, requires some manual configuration which will inevitably lead to a few minutes of system downtime. However, after the system activation the effectiveness of communications and operation is maintained.

3.8.3 Disaster Recovery System Scheduled Operation

The DR infrastructure is implemented in such a way that it can support several of the GENESys critical servers in the case of a disaster. Each GENESys system region has a DR server that can support the other region functions.

The data storage server installed in each region has several configured virtual machines, which are perfect clones of the critical servers systems the DR might cover. The technical configuration of all the virtual machines remains static throughout time, but still, the DR storage has to be updated whenever one of the real operation systems suffers an adjustment.

It is critical to guarantee that the databases are as updated and synchronized as possible and also in a confirmed safe mode. This is important to ensure that in the case of having to launch the DR plan we can ensure that the system will not only be synchronized and updated with the correspondent production system, but also that the DR virtual machines are secure and fully operational.

So that the disaster recovery platform can keep the system replicas in synchronized and correct operating conditions, thus ensuring that this alternative environment can be activated at any time in face of a disaster, the scheduled procedures are the following (see Figure 25):

1. The content of the covered servers gets transferred as individual database dumps to the data storage servers from the same region;

2. A synchronization process between the data storage servers is executed and the database dumps are exchanged between both system regions;

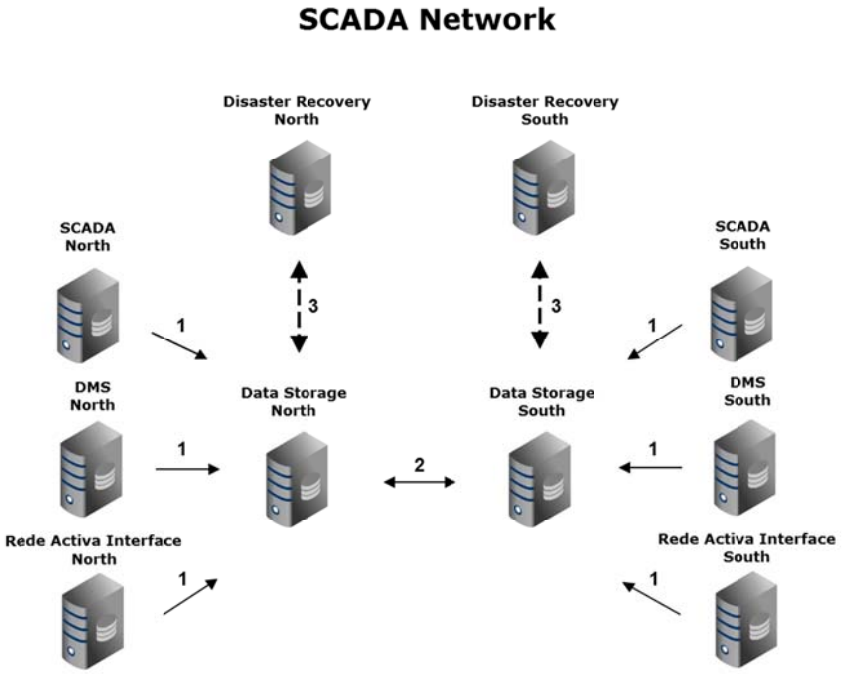


Figure 25 - Disaster Recovery scheduled operation.

3. The data storage servers are updated and synchronized with the complementary production servers and the disaster recovery servers can access their information under manual execution to take over any of the covered servers.

3.8.4 Disaster Recovery System Disaster Operation

Once there is a disaster affecting one or more of the critical GENESys servers, a manual procedure for the activation of the disaster recovery system is started. The procedure steps are presented next, and depicted in Figure 26.

1. A system failure is identified by the grid operators or the system administrators. The system administrators are responsible for triggering the pre-defined disaster recovery plan to ensure the shortest unavailability time;

2. A manual configuration is executed in the disaster recovery infrastructure so it can be prepared to receive the database dumps stored in the data storage servers;

3. When the database dumps are imported to the disaster recovery servers, the system is manually booted and the new servers, based on the disaster recovery virtual machines, are presented to the workstations. After this process, the system starts operating as usual and without any limitation.

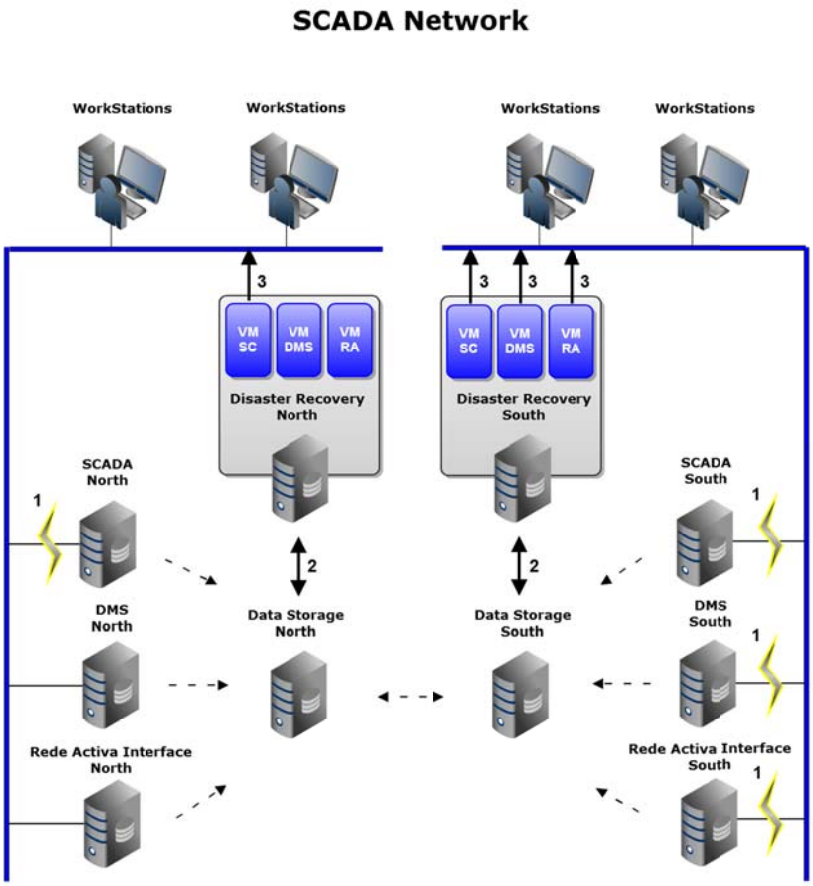


Figure 26 - Disaster Recovery disaster operation.

3.8.5 Disaster Recovery System Advantages

The disaster recovery system represents an investment on the security and robustness of the EDP Distribuição GENESys system. The fact that the disaster recovery virtual environment is able to completely takeover in the case of a critical system failure represents a major defense not only against ordinary system failures but also against cyber attacks, since a stored safe state can be re-established with the DR system support.

Therefore, the mains advantages of such a system can be presented as:

- **Business continuity** – It works as a recovery plan in the case of a system failure representing a faster response to the service discontinuity caused by the incident;
- **Data protection -** A backup solution for system data and also for its recovery;
- **System Reconfiguration -** In case of disaster it is possible to reconfigure different servers with different functionalities based on virtualization;

- **Data redundancy -** There are local and remote data storages with friendly and rapid access which represent further data protection by redundant storage.

3.8.6 Disaster Recovery System Weaknesses

The disaster recovery system works as a defensive shield in the case of any system failure. The system works as a secondary protection, since the existing dual redundancy of the production systems works first, in the case of any online server malfunction. However, when both the online and standby system servers fail, possibly due to a disaster scenario, the DR system comes into action, replacing the failed servers operation without any limitation.

Since we are addressing real-time critical systems the key word is availability. The DR system clearly answers the disaster scenarios and it increases the availability rate. However, it is not a perfect fault-tolerant implementation since it only performs reactive recovery, offering a manual commutation alternative in the case of an identified failure or intrusion. The system only performs its main function once the fault or intrusion is detected and, furthermore, only takes action after a manual execution of a configuration plan from a system administrator. The activation procedure requires experienced personnel to execute a set of tasks which are neither quick nor simple to perform.

Therefore, the system does not provide by itself the required real-time fault and intrusion tolerance that is required for such a critical infrastructure.

This weakness of the DR system and the clear perception of the importance of having a robust and available GENESys system motivates this study and, furthermore, the adoption of other quite different solutions that will result, in our expectations, in a far robust system.

There are some other disadvantages associated with the disaster recovery systems such as:

- Very costly solutions to develop and implement;
- It requires very experienced personnel to implement it properly and also to perform the manual execution;
- It is only a business advantage if designed and implemented properly;
- It requires periodical updates and maintenance, translating into a continuous investment.

Chapter 4

Fault- and Intrusion-Tolerant GENESys

The electrical distribution power grid of EDP Distribuição fulfills all the requirements to be considered a critical infrastructure. It is a complex and large scale operation system which has a high societary value, giving rise to high costs in the case of failure, which can range from economic to loss of lives [28]. Under these assumptions it is logical to include the electrical power grid under such classification which consequently elevates the challenge of security and dependability to the highest level.

In the last two decades EDP Distribuição has done a great deal of investment on remote supervising and controlling its electrical infrastructure. This type of transformation created a new paradigm for the whole infrastructure since it opened it to the networking world, and its associated threats. Before the SCADA implementation, all the equipments were locally controlled and the supervision was made 24 hours a day, 7 days a week, with no network connectivity. All the control operations were mechanic therefore the complex and critical infrastructure was safe from computer technology failures and cyber threats. Clearly, the advantages of adopting such a technological solution as SCADA surpasses the inconvenient of cyber threats, that can be prevented, even if not entirely, with a practical and robust security system.

At its beginning, under its most basic operation, the system was embedded in an air gap, i.e., a secure network which is completely physically, electrically, and electromagnetically isolated from insecure networks [29]. Over the years, with the technological evolution, new desirable solutions appeared with appellative and interesting features that could provide EDP Distribuição with a much more powerful and functional system and a more efficient and effective management, nowadays integrating the GENESys platform. Such an evolution came with a price, since it was impossible to give these steps without abandoning the current paradigm and integrating other functionalities in the system. By breaking the air gap, cyber security gained an even more relevant role in the reliability of GENESys. The infrastructure became heavily computerized and interconnected which made the power grid management infrastructure more exposed to accidental and malicious faults, of either physical or computerized origin.

With the change of paradigm, creating a dependable and secure supervision infrastructure for the power grid became one of the most important goals on the company.

There were initiatives to provide fault tolerance capabilities for the most critical services, mainly based on dual redundancy. Yet, there are many issues with the current implementation which still does not provide a full fault-masking solution, leading to casual system downtime. In addition, the enhancement of the GENESys platform security has been immediately addressed after a thorough external audit [30]. The security audit resulted in a series of security projects and investments aiming at the implementation of improvements in the systems infrastructure and network in order to increase the overall system security. One of them was the deployment of a Disaster Recovery (DR) System. However, the DR is not a perfect fault-tolerant solution since its reaction to system failures is manually induced, obliging to a period of unavailability (see Section 3.8.6).

The power grid operators are fully dependent on the GENESys platform to manage the power grid, for that reason, unavailability is not acceptable. It traduces in the incapacity of detecting and reacting to power grid incidents which probably lead to the disruption of service for clients. Furthermore, security incidents such as intrusions can lead to a widespread disruption of the power grid that could swiftly undermine governments, economies, and even endanger the health and public safety.

In this chapter we describe a new architecture based on fault and intrusion tolerance mechanisms. We will aim on a more dependable and secure system, as expected from a critical IT infrastructure such as GENESys. Our proposal will address some of the current implementation gaps and weaknesses, and will strengthen the infrastructure to prevent disruption events caused by accidental failures or catastrophic cyber attacks.

4.1 Overview

The intrusion-tolerant approach comes as a new paradigm to robust systems, designed for improving their security. In a conventional way, systems security was related with preventing faults and attacks from occurring, removing existing vulnerabilities and preventing attacks from resulting in intrusions. As signaled before, the traditional security approaches have been based in preventive mechanisms that, with time, proved insufficient to keep systems secure and dependable. Therefore, we believe that the fault tolerance approach would be an interesting upgrade over the GENESys system security planning with attractive operational results.

While deciding which components should be considered critical enough to be addressed, we took into account the ones which are strictly required for the basic power grid operation. Most of the systems we presented before represent new functionalities and management improvements over the management of the EDP Distribuição power grid. However, they are not fundamental for its main functional purpose, remote monitoring and control. The

components which are more important for GENESys to be able to perform those critical features are the *Remote Terminal Units*, the *Frontends* and the *SCADA* and *DMS servers*.

To understand how we will address each component, we have to analyze them to infer about the impact of a failure in each one of them. Our aim is to acknowledge which of these GENESys components has larger error propagation and consequently which different fault-tolerant mechanisms we should implement. Obviously, the components whose failures have the most impact deserve more attention and stronger security mechanisms. The analysis is summarized below:

- The RTUs are field sensing devices, therefore, their failure represent the incapacity of remote controlling and monitoring a specific Telemetry Site;

- The Frontends are middleware protocol translators which gather communication from several RTUs to afterward deliver it to the backend systems, or vice-versa. Therefore, their failure represents the incapacity of remote controlling and monitoring several Telemetry Sites;

- The SCADA server is responsible for the processing of the field information as well as for the remote control of the physical electrical components. Consequently, its failure is catastrophic and represents the incapacity of remote controlling and monitoring the whole power grid;

- The DMS server is responsible for the integration of the SCADA capabilities with distribution management capabilities required by the GENESys application, therefore, its failure is catastrophic and represents the incapacity of remote controlling and monitoring of all facilities as well as the DMS associated distribution management capabilities.

Since all these significant components have different levels of criticality, they will be differentiated regarding their system model assumptions. We will formulate stronger assumptions for the more important components, the SCADA and DMS servers, to ensure that the stronger mechanisms are employed to provide coverage over those assumptions.

The assumptions for the RTUs and Frontends are [31]:

- Systems remain to a certain extent vulnerable;

- Omissive faults will occur at a given time;

- Systems will fail at a given time;

- The overall system nevertheless remains secure and operational.

The assumptions for the SCADA and DMS servers are [31]:

- Systems remain to a certain extent vulnerable;
- **Omissive, Assertive and Byzantine faults will occur at a given time;**
- **Attacks on components/sub-systems can happen and some will be successful;**
- Systems will fail at a given time;
- The overall system nevertheless remains secure and operational.

The clean-slate architecture for the Fault- and Intrusion-Tolerant GENESys will have to take into consideration the presented analysis. The establishment of these assumptions is very important since they will reflect over the chosen solutions for each of the different GENESys layers to be addressed. The variety of mechanisms available are differentiated based on their coverage and were chosen accordingly to the established assumptions.

4.2 GENESys Architecture

In this chapter we introduce a clean-slate architecture that should be adopted in the GENESys system paradigm to address some of its vulnerabilities and weaknesses. The new architecture is composed by three different solutions aiming the different layers of the system, as depicted in Figure 27.

All the solutions can be individually implemented since they provide fault tolerance mechanisms independently, at the several layers of the GENESys architecture. However, even if they function correctly when isolated, to develop a strong defense against the higher threat level, the advice is to adopt an end-to-end implementation of the solutions presented, providing a security in depth approach [9]. If all the layers get strengthen then we guarantee there is a transversal degree of protection leaving no “low-hanging-fruit” in the chain of data processing.

Considering the assumptions we established for each of the components, we defined the more assertive mechanisms to be considered for the new GENESys architecture. First, we will introduce a dual redundancy mechanism for the RTUs and explore all the technical requirements for its implementation. Then, we address the Frontend layer by proposing a different approach on the dual redundancy already applied today. Given the assumptions established for the SCADA and DMS servers we considered that providing error masking by redundancy would not be a very effective solution. Therefore, we had to turn into a different and stronger approach that had the intrusion tolerance capabilities required. We implemented a Byzantine state machine replication protocol called MinBFT.

The fault detection model for the GENESys system is based in timeouts. This is possible since it is a synchronous system with a dedicated network. Nevertheless, there are other alternatives to ensure timely communication [32].

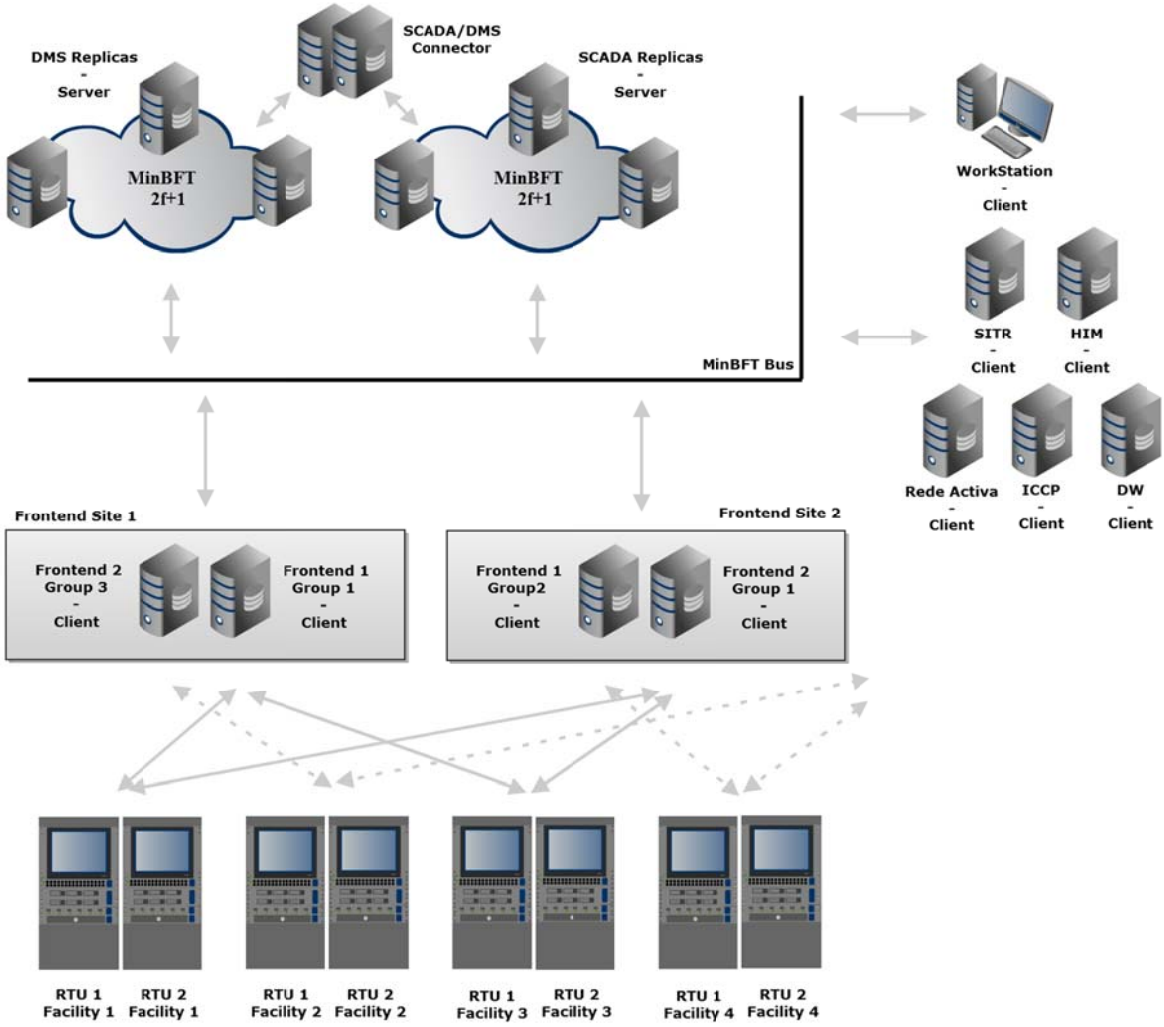


Figure 27 – Fault-Tolerant GENESys architecture.

In the following sections we will give more details about the fault and intrusion tolerance solutions proposed for the GENESys architecture.

4.3 Remote Terminal Unit

The Remote Terminal Units (RTUs) are field sensing devices responsible for translating the physical parameters of the electrical infrastructure into digital outputs. It is a standalone microprocessor-based data acquisition control unit that collects the field data with the incorporated sensors and sends it to the SCADA servers through the GENESys communication network.

As the RTUs represent the interface with the physical environment it is clear that without them the GENESys platform would be useless since there would be no field monitoring information and operators would not be able to control the electrical components. One can assume that the SCADA system's rationale and objective depend upon the proper functioning of a Remote Terminal Unit [33].

It is well know that RTUs operate in especially harsh environments where they are subject to intermittent and permanent failures [33]. In spite of their importance, the balance between its cost and its criticality dictated, a few years ago, that each telemetered electrical facility would have just one RTU installed.

We propose the introduction of a fault tolerance solution that provides a more robust architecture over this layer and that should be implemented at least in the most important Telemetry Sites. This will ensure a more dependable data acquisition and control on those electrical facilities.

The conventional RTU, presented in Figure 28, is composed by a central controller, a set of sensors, a memory module, a bus and the communication modem.

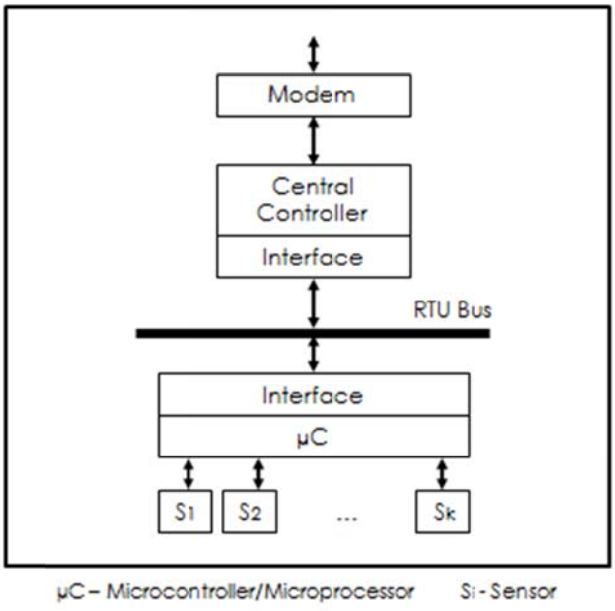


Figure 28 – Conventional RTU block diagram.

The sensors have a microprocessor responsible for initiating the data sampling and collecting the digital output. They translate the physical events from analogue to digital data and, after the conversion, the sensor interface will direct the digital output to the communication bus in some predefined communication protocol.

Due to the conditions to which RTUs are subject to, failures are common and there are two components that when faulty compromise the whole RTU operation, as well as the sensing capabilities.

1. The most critical failure of an RTU is related with a malfunction on the Central Controller Unit (CCU). The RTU becomes completely unavailable since the sensing data is inaccessible. In view of the fact that the central controller manages the sensor microprocessor, if it becomes faulty the RTU losts its whole operation.

2. The conventional RTU is composed by single processing elements for the sensor capabilities. Since all the sensing is processed by that single microprocessor, the RTU will be compromised and unavailable.

4.3.1 Redundant Remote Terminal Units

We propose the implementation of a Redundant Remote Terminal Unit (RRTU) composed by two redundant RTU components deployed together at the same electrical facility with an online/standby switchover mode, ensuring fail-over operation. The objective is to have two single equipments that are capable of executing all the sensing, control and communication tasks. One will be working in online mode while the other will be in standby mode. In the case of a failure of any kind in the online RTU, the other will be capable of ensuring all the tasks by itself. This type of solution is known as dual redundancy [34] where the system is provided with the ability to recover from a failure through the use of a backup or redundant hardware component. The proposed RRTU block diagram is presented in Figure 29.

The solution is not very demanding in terms of hardware implementation. On older facilities, an electrical splitter is required to distribute the analogue signal in two different sensors corresponding to both RTUs, which will then perform the sensing tasks independently. On the other hand, in the case of an Ethernet backend, the Electrical splitter will be a network switch connecting the sensing components to the RTUs. The communication infrastructure will be shared by both RTUs and no further implementations are required.

In terms of software development it is a little more complex. First, for the system to be able to communicate individually with each RTU, both of them have to be known by different addresses, otherwise, system messages or remote controls could not be directed to a specific RTU.

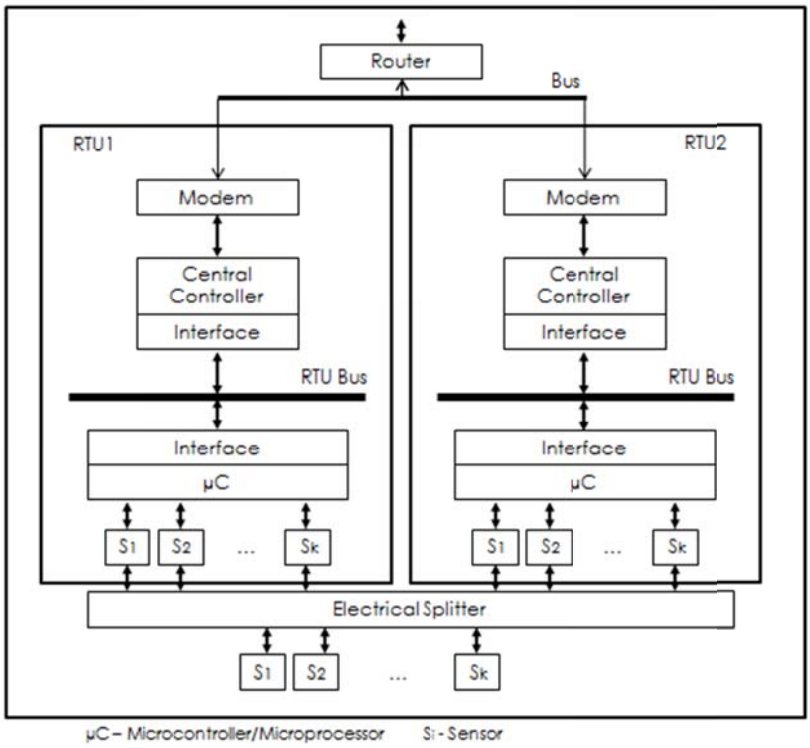


Figure 29 – RRTU block diagram.

Therefore, the SCADA database as well as the communication Frontend database should be added with a new table indicating which of the redundant equipments is in online mode, so that the whole system might know at any given time to which of the network addresses to communicate. This table, which is depicted in Table 2, must have a flag field indicating which of the RTUs is in online mode (the one with STATE=1) and the one in standby mode (with STATE=0).

RTU Redundancy Table

RTU ID	RTU N	FE N	SP N	STATE
500	1	32	11	0
500	2	32	11	1

Standby RTU

Online RTU

Table 2 - SCADA RTU Redundancy table.

This table should be managed by a new SCADA process responsible for updating the flag field as it receives an invalid RTU message. The SCADA servers are endowed with a mechanism that indicates if the RTU is active or invalid. Currently, what happens is that once the

communication Frontend stops receiving responses for the requests sent to the RTUs, they send an invalid RTU message to the SCADA server which is immediately reflected in alarms and telemetry information.

Before sending any message to a specific facility, the SCADA system will have to verify and validate the message address on that table and take into consideration the RTU state flag and the correspondent addressing fields, to be sure it will be delivered to the online RTU. Furthermore, the SCADA servers will only accept telemetry messages received by the RTUs flagged as online, other messages will be discarded.

The Frontend also requires a new table for mapping the information sent by the SCADA server to the correct RTU destination, represented in Table 3.

RTU Address Table

RTU ID	RTU N	SP N	RTU ADDRESS
500	1	11	12394
500	2	11	12395

Table 3 - Frontend RTU Address table

The most vital element for this implementation is to develop an effective procedure to perform a real-time verification of the correctness of both RTUs, as well as the required commutation in case of an Online RTU failure. Each RTU will be at one of two possible states:

- ***Online Mode*** **-** the online RTU in this state should operate as the conventional RTU installed nowadays at the EDP Distribuição remote facilities. The state can be described as active, where all the data acquisition and control in a specific facility is performed by it.

- ***Standby Mode*** **-** the standby RTU shall be inhibited from all telemetry and control operations. In spite of that, communication with the Frontend and SCADA servers is active so that it can receive system data such as “are you alive” or “activate” messages.

It is very important to manage correctly the RTUs states guaranteeing that the system knows exactly to which RTU to communicate and, furthermore, to perform activate/inhibit controls when required. When the online RTU becomes faulty its telemetry and control capabilities should be inhibited and, on the other hand, the standby RTU should be activated to perform all those tasks.

We propose two different mechanisms to be applied in parallel, or isolated, to guarantee that in case of any failure, whether processing or communication, an RTU can be corrected with the lowest downtime time possible.

1. A new watchdog process has to be included on each of the RTUs operation. For the online RTU this process should remain idle, but for the standby RTU the process will periodically send diagnose messages to its online counterpart. There should be a predefined timeout for the reception of the correspondent acknowledge message, that, if exceeded should trigger the commutation of the standby RTU to online state.

2. Taking advantage of the SCADA server mechanism that indicates if the RTU is active or invalid. Once the communication Frontend deliver an "invalid RTU" message, the referred SCADA process should update the flags on the RTU Redundancy table but also send an "activate" message to the standby RTU and a "inhibit" message to the online one. If communications with the previously online RTU are interrupted, the control message should be cached in the communication Frontend and delivered when communications are reestablished.

We suggest the implementation of only one of the mechanisms to guarantee operational consistency. We believe that the second mechanism is the best approach to implement since we might want to take advantage of the processing capacity and further resiliency of the upstream layers of the system.

We believe that RTU redundancy will improve the availability time and dependability of the RTU based communications. Even if software development is required at several layers of the GENESys architecture, the advantages of such a fault-tolerant architecture might be worthy, as it will be later presented, in Chapter 5.

4.4 Frontend

Considering the large electrical infrastructure on EDP Distribuição, the installation of RTUs on the electrical facilities have been performed throughout the years, while systems and communication technologies have been evolving with different and new standards to be applied for the specific types of service.

Accompanying this change without disrupting the legacy technologies has only been possible with the existence of the communication Frontends. They can be described as a communication middle system between the backend systems (SCADA and DMS servers) and the RTUs. They are responsible for the translation of the different communication protocols adopted for the different RTUs into the adequate format understood by the SCADA system. On the other hand, when the backend systems communicate with RTUs, the Frontend

performs the inverse translation, from the systems standard protocol to the one implemented on the specific RTU.

From the availability point of view, a failure on one of the Frontends invalidates all communication with RTUs mapped to it. It is very important to provide a dependable Frontend infrastructure since it covers a very wide range of facilities.

Nowadays, there are already some concerns with the Frontends and their critical operation. There is a dual redundancy mechanism based on primary-backup replication [19] applied at each of the Frontend Sites. There are duplicated Frontends composing a Frontend Group which operate based on online/standby switchover. The online Frontend covers all Frontend tasks while the standby waits for the first to fail. Once there is an online Frontend failure, the local watchdog which is monitoring their operation triggers a commutation which makes the standby Frontend the new online one, to perform all the required tasks. This is the only event that leads to the commutation between the two components.

The current solution has some weaknesses that need to be addressed. First, the redundancy is local since Frontend servers are located side by side, which, in the case of a disaster at the Frontend Site (i.e. flooding, earthquake, fire), will reflect in unavailability of the Frontend Group. Second, there is no communication redundancy between the RTUs and Frontends. The RTUs communicate to a network node at the Frontend Site and then communication is locally routed to both Frontend servers. Third, commutations are only effective in the case of a Frontend failure, no other component malfunction can be solved with it.

The solution we propose is quite different from the one already implemented. It has a different architecture which addresses the weaknesses of the current solution and intends to be more fault-tolerant covering a larger number of fault scenarios.

4.4.1 Fault-Tolerant Frontend Architecture

We propose a Fault-Tolerant Frontend (FTFE) architecture where the online/standby paradigm is discarded over an online/online approach. The Frontend redundancy is still implemented but each of the redundant Frontend servers composing the Frontend Group is placed in a different site to guarantee that in case of a disaster, both servers will not be affected. Furthermore, by placing them in different sites it oblige independent communication networks to the RTU guaranteeing that, at any time, RTUs have two different routes to the upstream systems, and single points of failure will be eliminated. This requires a new redistribution of the redundant Frontend servers, deploying them on separate locations. This redistribution has to guarantee that in case of a failure on one of the elements of a Frontend Group, there will not be a noticeable glitch for the GENESys operators since the other element, located in a different Frontend Site, will ensure a correct operation. We analyzed the several possibilities taking in consideration the proximity between different sites, since RF communication has distance constraints. The redistribution we consider the best solution is depicted in Figure 30, as opposed to the current Frontend distribution where

a Frontend Site includes both redundant Frontends, as described in Section 2.5. To ensure a well-built redundancy we rely on the following assumptions:

- Two matching redundant Frontends will never share a Frontend Site;
- Two Frontend Sites will share at-most two matching redundant Frontends.

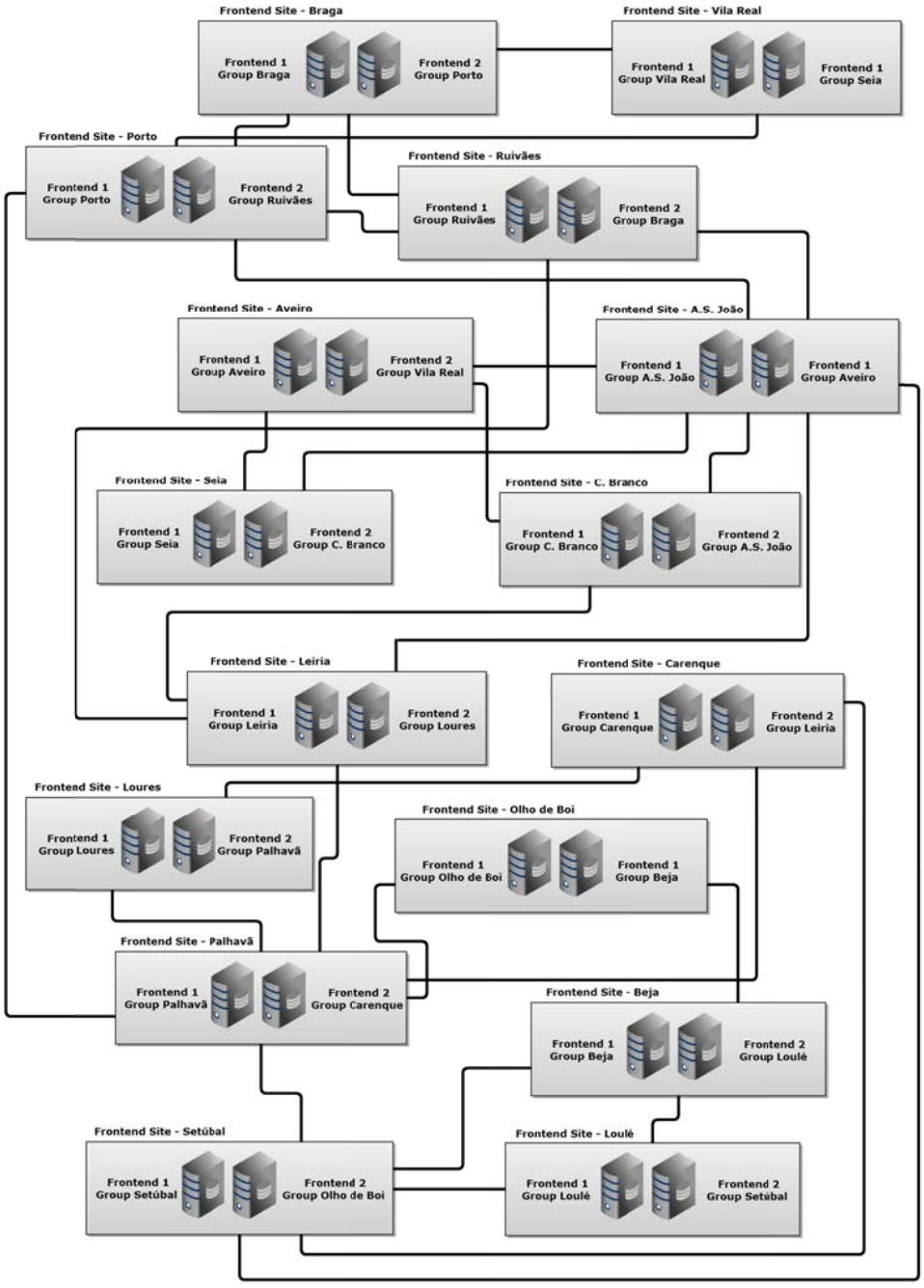


Figure 30 - Proposed Frontend Redistribution. This is our proposal for the separation of the existing Frontend groups through the already available Frontend Sites.

As previously referred, there is a change on the operational paradigm with respect to redundancy. Currently, only one of the Frontends is in online mode while the other waits for a commutation in standby mode. To understand how Frontends behave in both of these states:

- ***Online Mode* -** This Frontend communicates with all the RTUs mapped to its serial ports.

- ***Standby Mode* -** The standby Frontend remains inhibited from communicating with all the mapped downstream RTUs. In spite of that, communication with the SCADA servers is active so that it can receive system data such as “activate” or “are you alive” messages.

In the new approach both communication Frontends are in online mode. They are always active and communicating both with the RTUs and the backend system, unless they are faulty. The change of paradigm here is that currently the inhibition status covers the whole downstream communication of the standby Frontend, in other words, every RTU mapped into the redundant servers only communicates with the online Frontend since communications are being routed strictly to the online Frontend by the local watchdog. Only one of the redundant servers performs the Frontend tasks at a time, therefore, the redundancy only provides a fault-tolerant procedure in the case of a Frontend server failure.

In our proposal both Frontends are actively communicating with RTUs. Each RTU has redundant paths to the redundant Frontend servers and can communicate by both paths, although, only one at a time. It is worth to notice that this communication requirement will not require a big effort since the communication between RTUs and Frontends can by itself ensure traffic delivery on two different locations. The prevalent communication technologies used in the RTU layer are General Packet Radio Service (GPRS), Radio Frequency (RF) and Synchronous Digital Hierarchy (SDH). All these technologies have the capabilities of adapting to the proposed solution without requiring a duplicate network, one for each Frontend connection.

- **GPRS -** The GPRS modem installed in each facility includes a unique SIM card with which it will be able to initiate communication with the operator. Considering the scope of the infrastructure, EDP Distribuição has a private Access Point Name (APN) which is a specific configurable network to be used to connect to the EDP’s backend Systems. Each of the GPRS-capable facilities forwards their traffic through the EDP private APN which will then re-route it to the correct Frontend. Thus, the APN has the ability of configuring two different IP addresses to forward any RTU traffic, working on fail-over mode. There is a primary IP address pointing for one of the Frontends that, if failed, will trigger the re-routing of traffic to the secondary IP address, pointing for the other redundant Frontend in a different site [35]. The GPRS implementation is described in Figure 31.

- **RF -** The communication with facilities based on RF technology are based on transceivers (modems) which are capable of transmitting and receiving encapsulated

data via radio signal [36]. Since most of the RF based facilities are located far from Frontend Sites, EDP Distribuição has several radio repeaters distributed throughout the national territory. These repeaters are a combination of a radio receiver and a radio transmitter that receives a weak or low-level signal and retransmits it at a higher level or power, so that the signal can cover longer distances without degradation [37]. Since the ability to capture radio signals depends only on the knowledge of the transmission frequency, by having RF modems on both the redundant Frontends locations configured to receive the specific frequency allocated to a mapped RTU, we will ensure two separate and independent connections from the RTU site to both Frontend locations. To make this solution robust, two different repeaters should be used, one for each connection, to eliminate the repeater as single point of failure. The RF implementation is also described in Figure 31.

- **SDH -** The SDH architecture in EDP Distribuição is based in Bidirectional line-switched ring (BLSR) [38]. All facilities covered by this multiplexing protocol are guaranteed to have redundant communication, due to protection fibers of SDH, and direct ring connection to the Frontend Sites. Therefore, the traffic from the electrical facilities can be easily routed to any Frontend Site.

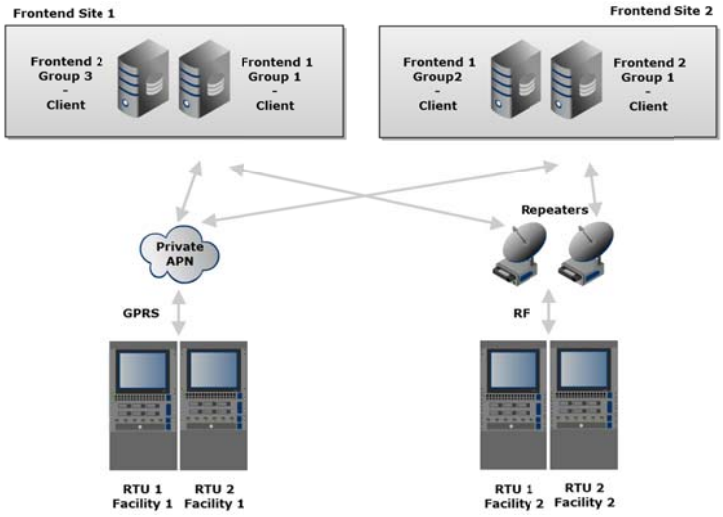


Figure 31 –Proposed Frontend architecture for GPRS- and RF-based RTUs.

The SCADA server will manage to which of the redundant Frontends any RTU is communicating to. The principle of operation is the same presented before for the RRTU.

First, the system needs to be able to know which Frontend to address when it wants to communicate with a specific RTU. Since they both have known different addresses, the SCADA system requires a mechanism to identify at any given time which is the active communication Frontend for a given facility. Therefore, the SCADA database requires a new table indicating which of the redundant Frontends is selected for each specific RTU, for the system to be able

to route downstream system messages and controls correctly. This table, presented in Table 4, must have a flag field indicating which is currently the Frontend active route (the one with FE state=1) and the inactive route (the one with FE state=0) for each RTU.

RTU Routing Table

RTU ID	FE N	SP N	FE State
500	12	15	1
500	13	15	0

Active Route

Inactive Route

Table 4 - SCADA RTU Routing table.

The registers in Table 4 should be managed by a SCADA process responsible for updating the *FE State* field as it receives an invalid RTU message from the Frontend. As we referred before, the SCADA system has a mechanism that indicates if the RTU is active or not when a Frontend stops receiving responses for requests sent to the RTUs. Before sending any message to a specific facility, the SCADA system has to verify and validate its active communication Frontend address on that table, so that it can route data through the correct Frontend.

On the other hand, the upstream messages from the RTU can be sent both ways, to each of the redundant Frontends. The fact is that only one of the Frontends is listening to the RTU messages, the other will just ignore them. This Frontend message validation will be done based on a database table, in Table 5, which will indicate for which RTUs is that specific Frontend currently in active mode.

Frontend 1 - RTU Comm_State Table

RTU ID	SP N	State
301	15	1
501	11	0
768	15	1

Active

Inactive

Active

Frontend 2 - RTU Comm_State Table

RTU ID	SP N	State
301	15	0
501	11	1
768	15	0

Inactive

Active

Inactive

Table 5 - Redundant Frontends RTU communication state table.

The table updates for both SCADA and Frontend servers will be performed by the same SCADA process we presented before. The process will send the update message to the Frontend at the same time it updates its own RTU routing table. This way, both the SCADA system and the communication Frontends are synchronized with respect to the active route. Thus, the Frontend will know it is the active route for each specific RTU and will be sensible to both downstream and upstream messages, delivering them to their correct destination.

The Frontend architecture proposed is capable of sustaining the presence of different faults. It is able to detect their possible causes and also masking them with integrated fault-tolerant mechanisms. In the following, we will present which are the possible faults that might occur in Frontends and how the proposed solution detects its origin and, when possible, masks them, as opposed with the current architecture. The fault-tolerant mechanisms are depicted in Figure 32.

Invalid RTU information at the SCADA systems - If one of the RTUs does not answer the Frontend "*are you alive*" request for a predefined period of time, it will be considered faulty and a message with this information is sent to the SCADA server. In this case, if communications are still possible, the SCADA operators will perform system level operations such as resets to recover the RTU, otherwise, a repair team is sent to the electrical facility to verify the malfunction.

This procedure does not detect which is the possible cause of the problem, since in the current architecture there are no mechanisms providing the required tools to do it. The possible causes for the presented failure are:

1. RTU malfunction;
2. Frontend serial port malfunction;
3. RTU serial port malfunction;
4. RTU-Frontend network failure.

There are the several possible faulty components originating the failure. The proposed architecture has the ability of changing the RTU communication flow to the other Frontend. This procedure might solve the problem by providing an automatic diagnose function detecting if the origin was a port malfunction at one of the ends, or an RTU to Frontend network failure. This way there is a first triage by the system in which the problem might be solved without any human intervention, simply based on the fault-tolerant architecture. On the other hand, if it is not solved, system operators will have the required information to identify which RTU component is responsible for the erroneous behavior.

Invalid Frontend information at the SCADA systems - When a Frontend fails there is a similar mechanism to the one we described for the RTU invalid state. The SCADA servers are continuously sending requests to the Frontend servers, when there is no response for a period of time there is a timeout. A communication monitor process triggers a system message indicating that the communication Frontend is failed. Furthermore, the local watchdog in the Frontend Site is also exchanging periodic "*are you alive*" messages with the servers and monitoring their processes to evaluate their state. If there is an abnormal situation on the online Frontend, the commutation is executed based on the watchdog order and the fault is masked.

On the proposed architecture the redundant Frontends are not located at the same site, so the watchdog component and the relay commutation are not required. The Frontend failure response will be executed only by the communication monitor process in the SCADA servers. As previously explained, both the redundant Frontends are in online mode having RTU communicating with one or another, depending on the context. In the presence of a fault, the SCADA needs to operate fast to reroute all RTU communications to the operational Frontend. Therefore, once the monitoring process identifies the failure it will not only trigger the invalid Frontend message but also update its RTU routing table and send a "*commute all*" message to the operational Frontend so that it also updates its RTU communication state table. The SCADA process is vital to manage the whole procedure where the Frontend is detected and given as failed. The SCADA table has to be updated to set the operational Frontend as the active route for all correspondent RTUs, and a command has to be sent to the operational Frontend to set itself as the active Frontend to all of the mapped RTUs. Based on this mechanism, the Frontend faults will be tolerated even in situations where the Frontend Site gets fully compromised, what does not happen with the current solution.

The proposed fault-tolerant architecture provides coverage for all the possible origins being analyzed except for the one in which the RTU component is damaged. Therefore, to increase the robustness of the architecture we propose the adoption of this proposal combined with RRTU solution. If the RTU is responsible for the invalid RTU state received by the backend systems, the fault-tolerant mechanisms of the RTU layer would provide a masking solution for the failure. The integrated solution would provide total coverage for the fault possibilities presented, but of course, under the assumptions that just one of the redundant components is faulty, either the RTUs, Frontends or the network connection.

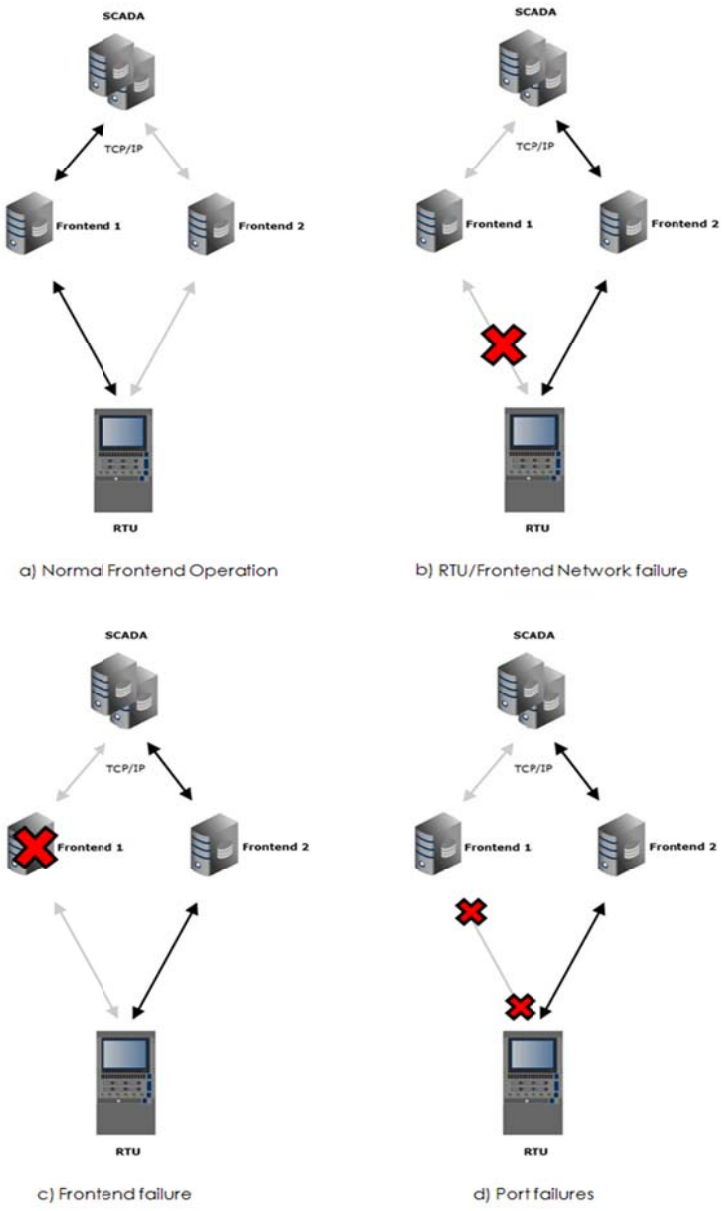


Figure 32 – Coverage over Fault models.

4.5 Core Systems: SCADA and DMS

After presenting different mechanisms to address fault tolerance in the lower layers of the GENESys chain of data, we will now address the most crucial components for the system operation. As introduced before, the failure of a RTU or a Frontend has serious operational consequences but, in fact, they only affect the remote operation of the facilities to which they are mapped, a single facility in the case of an RTU, and a group of facilities in the case of the Frontend. However, when we are referring to the redundant SCADA and DMS systems, their importance is in a totally different level, since when one of those fails, the GENESys platform

will simply ceases to function, meaning, that the whole electrical power grid is out of supervision and control capabilities.

Almost ten years ago, EDP Distribuição just had a conventional SCADA platform with an application interface. At that time, the redundant SCADA servers were solely responsible for the monitoring and control of the power grid, which made it the only critical server whose operation could not fail. More recently, the SCADA servers were connected with DMS servers to integrate distribution management capabilities to the more classical SCADA features. A new application was created to integrate both of the systems into a single and more powerful tool, GENESys. The new platform is dependent of both systems to operate correctly since most of its processes require constant communication with both SCADA and DMS systems.

Nowadays, both the SCADA and DMS servers have some fault tolerance capabilities. These capabilities are related with error masking based on dual redundancy (primary-backup replication [19]). The objective is to maintain a correct operation in the presence of a benign failure [39] in one of the servers, since in that case, the other will maintain the service.

The current mechanisms work perfectly under the assumptions of scarce benign faults on the system. However, considering their criticality, the considerations for these systems will be different than the ones presented for the lower layers. It is important to broaden the fault assumptions so that we can apply the right mechanisms for ensuring high coverage. By broadening we mean that we need to consider malicious faults which might result in arbitrary behavior. These faults are usually referred as Byzantine faults and might manifest as a failure to return a result, a return of an incorrect result, a return of a deliberately misleading result or a return of a different result to different parts of the system [40]. From this description it is easy to see that these faults might have catastrophic consequences for a system such as the one we are scoping in the thesis, the EDP Distribution critical infrastructure GENESys system.

Byzantine faults are usually associated with malicious intrusions. We require a different approach from the previously presented fault tolerance designs, which is the intrusion tolerance approach. The objective of this type of fault tolerance is to endure the systems with mechanisms that are capable of identifying intrusions and preventing them from leading into a system security failure, therefore, capable of tolerating intrusions.

Considering the Byzantine fault-tolerant algorithms presented in Chapter 2.2, we had to choose the better solution for the GENESys architecture. There are different designs providing different capabilities, from PBFT and Zyzzyva with a higher number of replicas but not requiring trusted components, and MinBFT where the reduced number of replicas represents a reduction on the cost of implementation and also a better performance since it relies on less machines and less communication steps to execute the algorithm correctly. Therefore, we had to validate all our available designs and balance each of its main strengths and weaknesses, taking into account the system model in which we want to apply them.

After our thorough analysis, we came to the conclusion that MinBFT [17] would be the most appropriate solution. It requires a reduced number of replicas, since it requires *2f+1* replicas to tolerate *f* faults, as opposed to the other solutions which tolerate *f* faults using at least *3f+1*

replicas. In addition, the algorithm has a lighter execution since it requires a reduced number of communication steps, reducing the time spend over protocol operations. The Zyzzyva algorithm is even faster than MinBFT, however, it has a major weakness for our intentions of integration in a near real-time [1] system. The problem is that since the protocol uses speculative execution, when a primary is identified as faulty, the system will perform a view change where the replicas roll-back into a safe state [16]. This is inappropriate in our scenario since in SCADA systems the state changes occur very frequently and are reflected physically, therefore, a roll-back is not always possible.

The factors presented were instrumental for our decision, since they reflect both a reduction of implementation costs and also, and more importantly, an increase in performance, which is critical for the GENESys system.

4.5.1 SCADA and DMS Intrusion-Tolerant Architecture

The MinBFT protocol will be used for replicating both the SCADA and DMS servers. A total of *2f+1* replicas for each system will be installed in the GENESys network. Considering the operation of the algorithm, these replicas will be the server components of the protocol, as opposed to other GENESys components (see Figure 27, in Chapter 4.2), which will be clients.

The SCADA and DMS replicas will still be responsible for storing critical information about telemetry, processing alarms, processing the remote control executions, processing the distribution management capabilities of the platform and deliver the correct response to each and every request from the clients. The clients, on the other hand, will be responsible for both delivering and requesting information to the replicas. Furthermore, considering the required data exchange between the SCADA and DMS replicas, we had to include a SCADA/DMS Connector, responsible for connecting both of their services, as depicted in Figure 33.

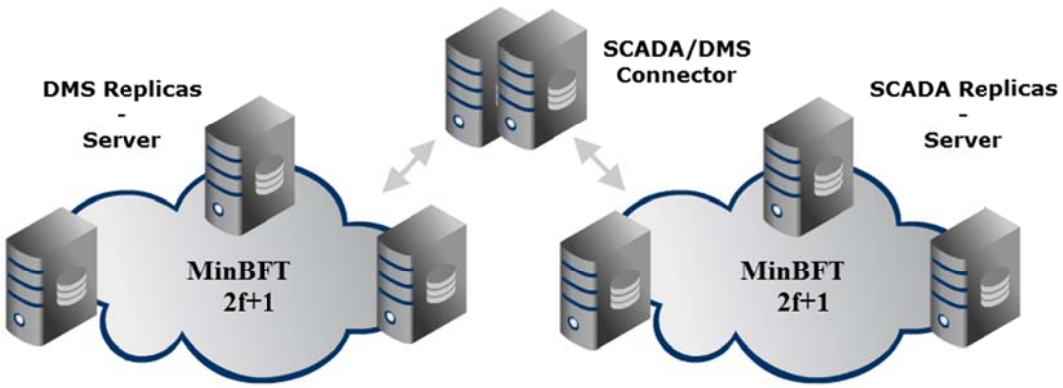


Figure 33 – Intrusion-Tolerant SCADA and DMS architecture.

4.5.2 MinBFT State Machine Replication Protocol

The GENESys network will not change in its structure, since the downstream layers will remain exactly the same. We will take advantage of the existing sites to deploy the SCADA and DMS replicas in diverse locations, making the system able to tolerate disasters and large-scale attacks like DDoS. As already presented, the MinBFT algorithm is able to tolerate *f* faults using *2f+1* number of replicas. For the case, we will choose three replicas which will allow both the SCADA and DMS system to tolerate one replica with Byzantine behavior. This choice is due to two considerations. First, reducing the costs on the number of replicas and all the implications for their implementation, and second, reducing the network load derived from an increased communication flow, since the higher the number of replicas, the higher the number of required messages exchanged between the intervenients.

While choosing the locations, we had to analyze the link performance of the different Frontend Sites (see Section 3.6). We measured the throughput and latency between all the different sites. The throughput capacity was known based on the communication media installed between each site, and the latency was evaluated by doing series of ICMP ping commands between the access routers of each site. From our analysis, we obtained the best three locations for replicas deployment, as presented in Figure 34.

4.5.3 System Components

The implementation of MinBFT requires changes over the architecture of the GENESys system as well as on its operation. We go from a distributed architecture in which all components have processes proactively capable of requesting and sending information to perform their operation, into a platform composed by servers and clients with well defined roles:

Clients - The clients are the only ones capable of proactively making requests. When required, they can request replicas for state information, modify any of its states or deleting state information. The replicas will then send the response back to the client which will wait for *f+1* matching replies for the request to complete its operation. The clients are responsible for triggering requests when needed and also for periodically planned requests necessary for many services provided by the system.

Servers - The servers are only capable of responding to clients requests, therefore, without being inquired they are unable to send any information to clients. The protocol requires a client request to trigger the operation on the servers. Once the replicas receive the clients request they wait for the primary to broadcast the execution order for that request so they can perform the commit phase of the protocol, execute the command and send their response back to the client, which delivers to the application the reply returned by at least f+1 replicas. From now own we will no longer refer to SCADA and DMS servers but to the abstraction provided by the MinBFT protocol set of replicas called SCADA and DMS service.

SCADA/DMS Connector - This component had to be included in the architecture to safeguard the required exchange of information between the SCADA and DMS. Considering that both services require each other's information, and they are both incapable of requesting it to one another, the role of the connector is to perform all the required requests to one service on behalf of the other, and then spread that information (i.e., it periodically requests telemetry information to the SCADA service, which responds back with all the non delivered data. Once the connector has it, it will make a request to the DMS service to deliver it). Since the correct operation of the GENESys platform is dependent on the connector, we propose a dual redundant solution in fail-over mode, installed in two different geographical locations.

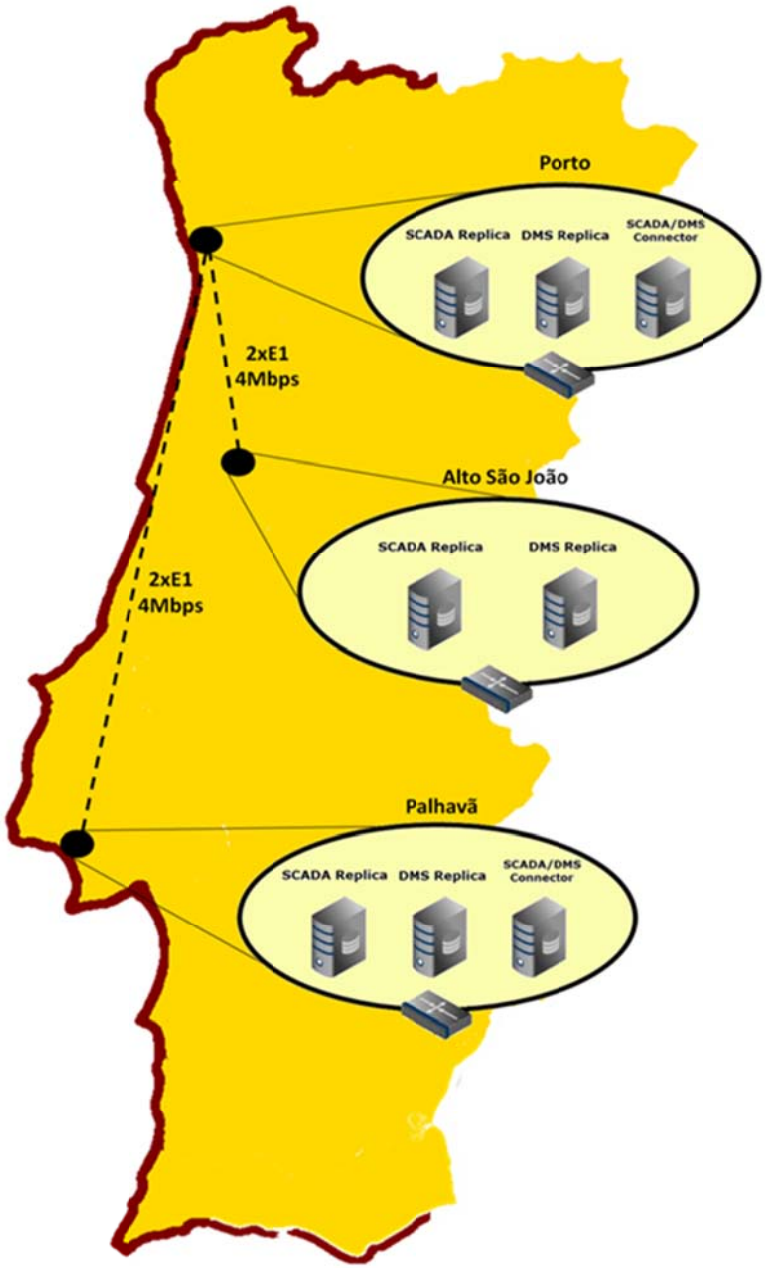


Figure 34 – Backend systems location (SCADA/DMS replicas, and Connectors).

To ensure that no information is corrupted during the transaction from one service to the other we propose the implementation of threshold cryptography [41] where the destination service will only accept data from the SCADA/DMS Connector if they are signed by *f+1* replicas from the source service.

4.6 System Data Flows

The modifications we are proposing require some alterations on the system operation, both at hardware and software level. However, such implementation should not jeopardize any of the system capabilities, so, the integration has to be carefully planned and tested.

We will next present how the GENESys reactive and proactive data flows should be implemented on the new fault-tolerant architecture, in line with what was previously presented in Section 3.7. We will present in detail the required interactions for the correct operation of GENESys and the auxiliary systems.

In this analysis we consider executions without failures to avoid complexity. Furthermore, the RTU and Frontend layers were reduced to only one component, for simplicity purposes.

4.6.1 Proactive Data Flows

These are scheduled procedures for the correct operation of both the GENESys system and all the parallel backup and support systems. We have previously presented how all the operations are currently performed (Section 3.7.1), and here we will describe the same procedures under the new architecture.

- The Watchdog is no longer necessary for the system operation. Since the protocol will manage a crash failure and a correct service will still be delivered by the correct replicas.

- The DMS replicas are updated by the SITR server, which will perform as a client. It should have a scheduler responsible for periodically triggering the exchange of information with the technical and geographical information from itself and the DMS service. Since the updates can be very large and complex, the replicas request should be based in a hash function produced over the total amount of data received, reflecting in a simpler validation at client-side. Figure 35 depicts this operation.

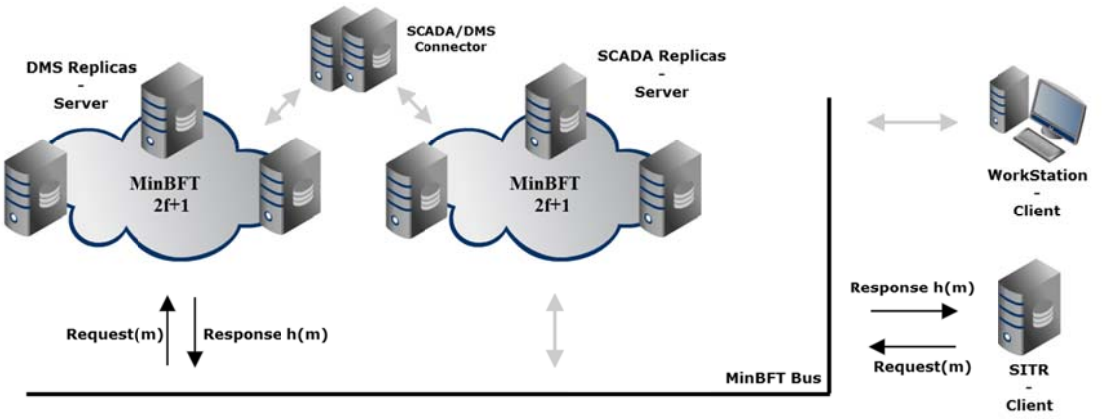


Figure 35 - SITR communication with the DMS replicas.

- The ICCP system will be a client of the SCADA service. This system requires very frequent data exchanges between both command centers, i.e., EDP Distribuição and REN, since it cannot deviate from the near real-time concept, even if data is coming from a different source. Therefore, the ICCP client should have a process that triggers at every second a procedure where it will updates telemetry information on the SCADA service based on the recently received information from the REN SCADA server, and, on the other hand, a request for the telemetry information from the last second regarding the SCADA service, to provide it after to the REN SCADA server. Figure 36 reflects the proposed operation.

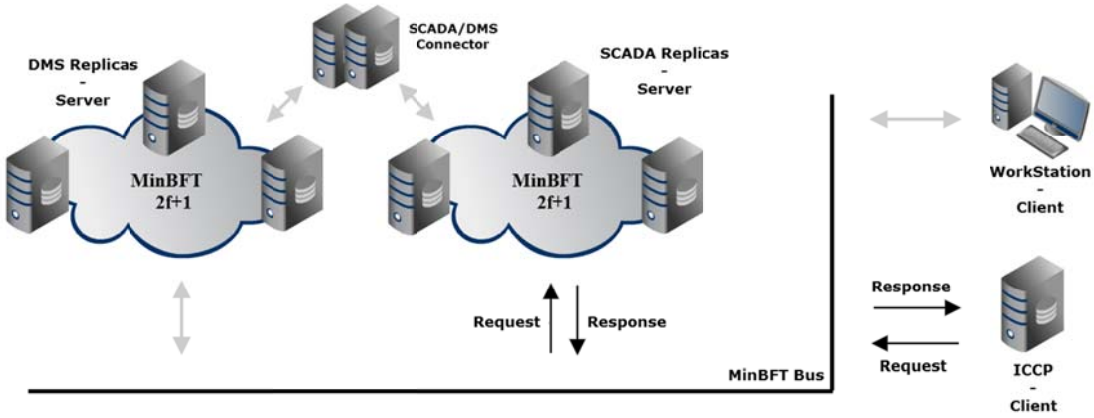


Figure 36 - ICCP communication with the SCADA replicas.

- Regarding administrative operations, in which the system administrator in a workstation modifies important operational configuration such as database relational data, component identifiers, communication parameters, among other, they should

be managed both by the workstation where the changes are performed and also by the other clients that require synchronization. It is critical that this information gets replicated and synchronized fast on the different GENESys components, to ensure the correct operation of the system. This should occur in two different steps, depicted in Figure 37.

1. The workstation process, responsible for the system operation, updates the SCADA service or/and DMS service. It waits for *f+1* matching responses for the request to validate the operation.

2. The client components which require synchronization with some administration operations should have a process sending periodical read-only [1] requests to the SCADA replicas asking for any recent modifications.

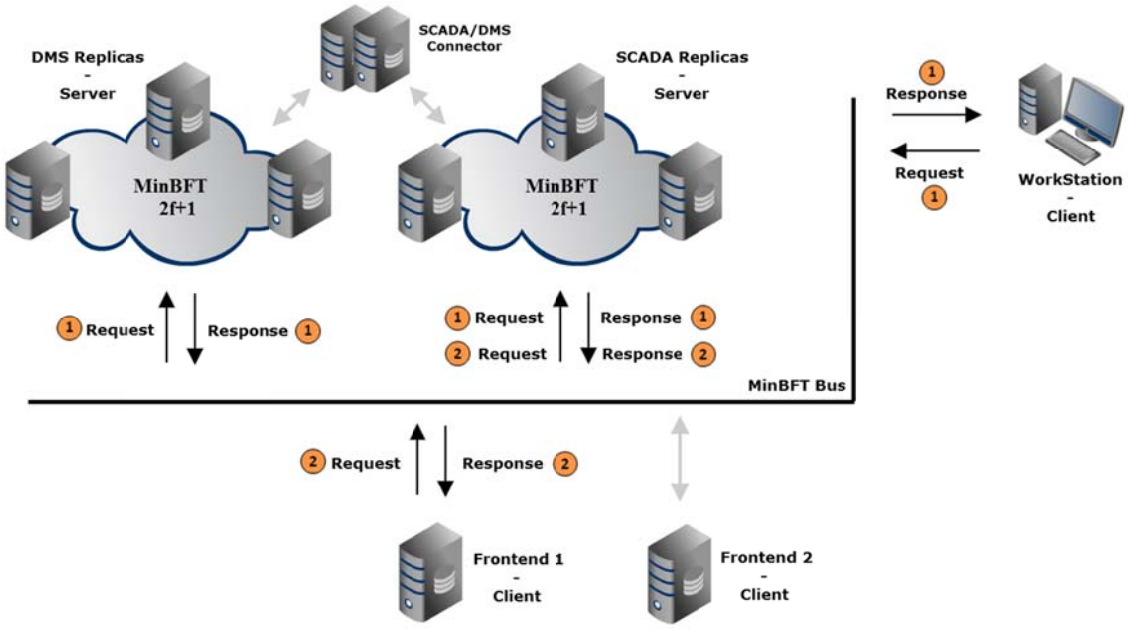


Figure 37 – Administrative operations communication.

- The system events are to be stored in the HIM server. Therefore, the HIM server works as a client which should perform periodic requests for recent events, to update its database. From time to time there should be a process sending a read-only request to the SCADA service which will respond with a message with the correspondent information. The procedure is depicted in Figure 38.

[1] Read-only requests are more efficient since they do not require consensus [12].

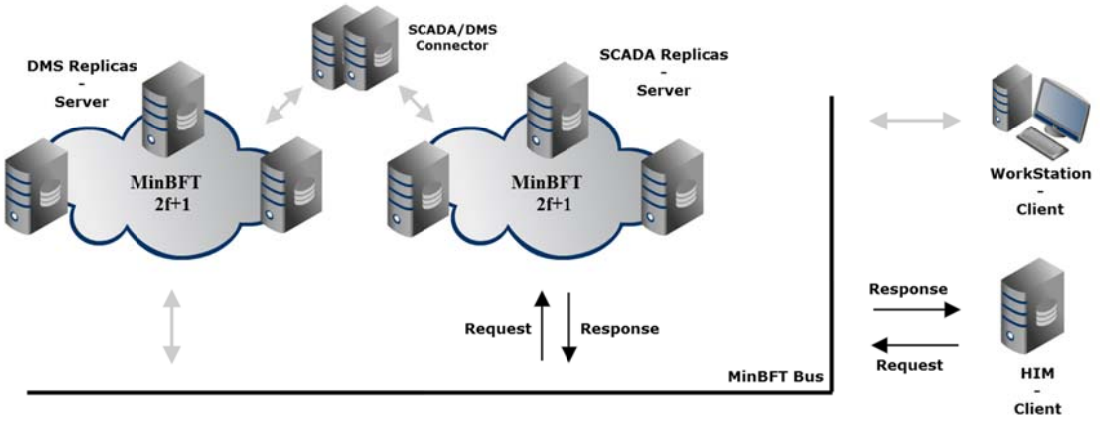


Figure 38 – HIM communication with the SCADA replicas.

- The real-time execution of the GENESys application requires information of both SCADA and DMS services. Each workstation will require several processes to manage requests for the required information for correct operation. These processes should periodically request recent information since the update of telemetry information and interface data has to be as close to real-time as possible. The procedure is illustrated in Figure 39.

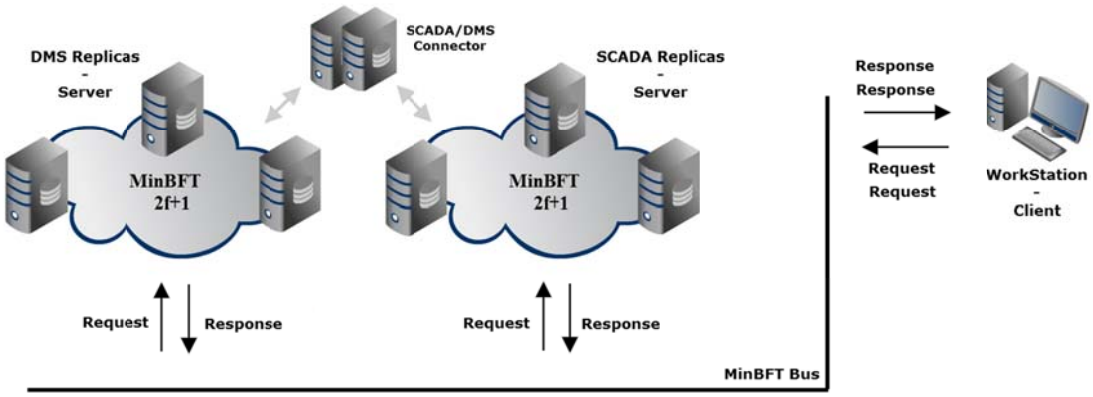


Figure 39 – Workstation communication with SCADA and DMS replicas.

4.6.2 Telemetry Data Flow

The telemetry process will be different because of the defined roles of clients and servers and their capabilities. Since servers are not able to perform requests, due to the MinBFT algorithm programming model, the data exchanges between the SCADA and DMS servers will be mediated by the SCADA/DMS Connector. In addition, with the exception of the Frontend and

SCADA service interaction, which will be triggered by a telemetry event, all requests for communication on the client side are based on periodical scheduling to guarantee that the information can get to the correct destinations in a timely manner. The whole telemetry process, from its detection on the RTU layer until its representation in the workstation, is described below and depicted in Figure 40.

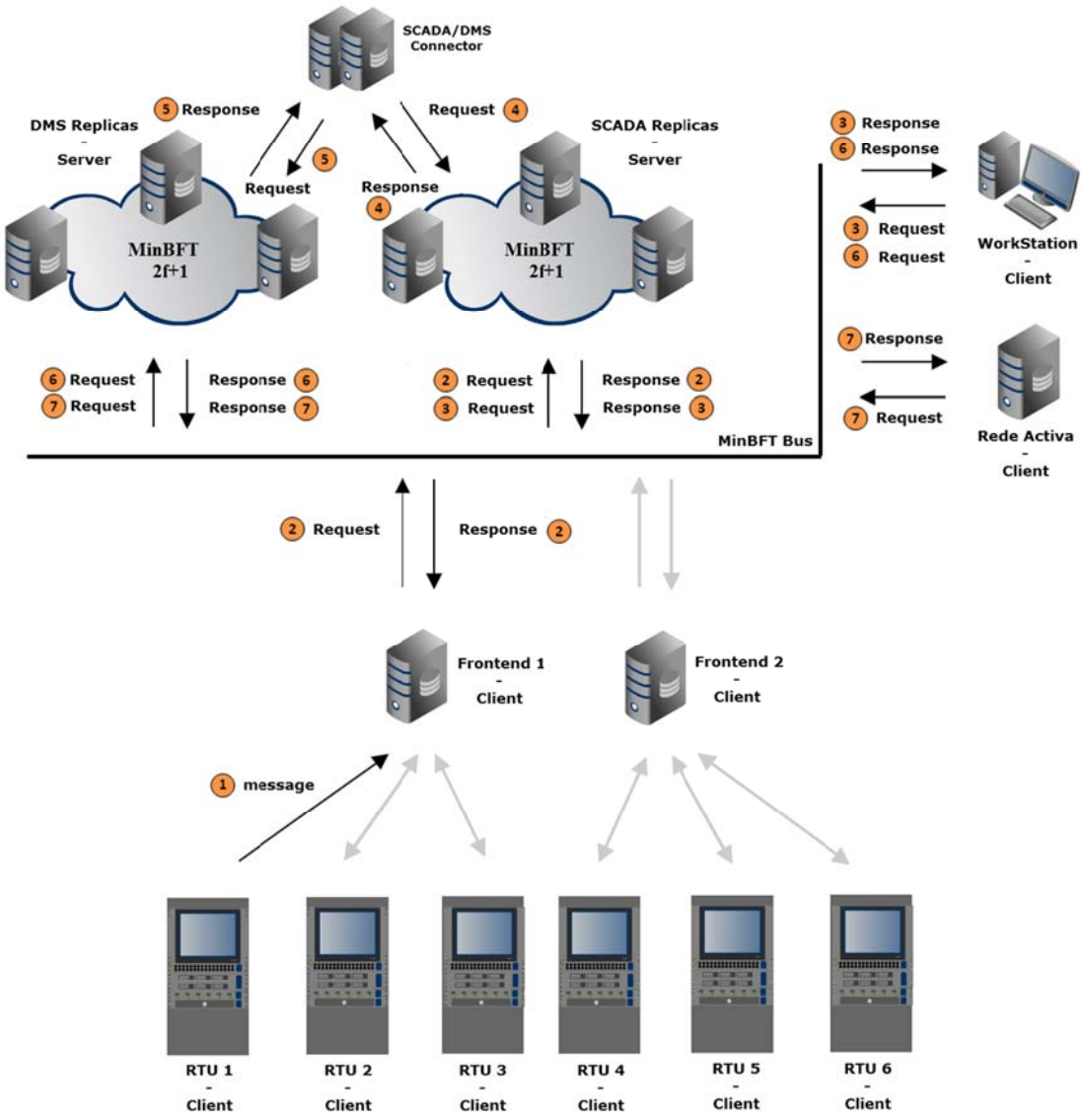


Figure 40 - Telemetry data flow representation.

1. The RTU, which is monitoring the physical components of the infrastructure, acknowledges a change in the state of a monitored entity or a measurement. This physical change will be interpreted and converted into a digital signal by the RTU where a signal unique identifier will be included, as well as the actual state which

triggered the telemetry change. The digital signal will be encapsulated in one of the communication protocols EDP uses for RTU to Frontend data transfers, and sent.

2. The Frontend component will translate the protocol into TCP/IP and associate an RTU identifier with it based on the communication port from where the packet was received. The Frontend performs a request to the SCADA service to send the telemetry information it has just received.

   The replicas will process the information received. They will individually update the register in their internal database with the change of state and identify the actions associated with that specific event (i.e., if it generates an alarm, which type of alarm and priority). The actions that need immediate transmission to the workstations should be stacked to be delivered upon request.

3. The telemetry and alarming information are the most important functionalities of the GENESys system since they enable operators to perceive in near real-time the state of the power grid. Therefore, since the workstations produce the graphical interface of the whole platform they have to include processes to request periodically all recent changes of state and alarms to the SCADA service in 2. Here, a process should periodically perform a request to the SCADA replicas for the recently cached alarms to be displayed in the application. The workstation client should wait for f+1 matching responses to validate the operations and present the alarming information to the operator.

4. There is important data that has to be exchanged from the SCADA replicas to the DMS replicas such as the telemetry change of state so that the DMS replicas can reflect that specific change in the synoptic representation and their databases. Since they cannot proactively perform these exchanges, the insertion of a Connector on the infrastructure as a middleware was inevitable, to mediate this process. Therefore, one of the SCADA/DMS Connector processes will request periodically the SCADA replicas for any recent data to be delivered to the DMS replicas.

5. The second part of the service-to-service exchange occurs when another process of the SCADA/DMS Connector requests, immediately after the previous operation, the DMS replicas to deliver the information it has previously requested to the SCADA replicas. This way, the required transfer of data between both server quorums is performed.

6. Another workstation process performs periodic requests to the DMS service for any change of state for the current graphical interfaces being executed in the application. However, if no graphical interface is opened at that time, the request for the real-time equipment state is not required.

7. The Rede Activa Interface Client will also have a process performing periodic requests for the more recent changes of state in the grid since it requires such data to create the incident information on the Rede Activa Server.

4.6.3 Control Data Flow

The remote control data flow will also be changed according to the algorithm specifications. The SCADA/DMS Connector will process data exchanges between the SCADA and DMS services and, excluding the initial interaction, between the workstation and DMS service, all other are based on scheduled requests. The whole process since the operator remote control at the workstation, until the control execution at the RTU layer, will be described next, and are depicted in Figure 41. The validation of a successful remote control at the backend systems is done based on the telemetry data flow previously presented. The remote control execution will cause a physical alteration in the electrical facility which will trigger the corresponding telemetry event.

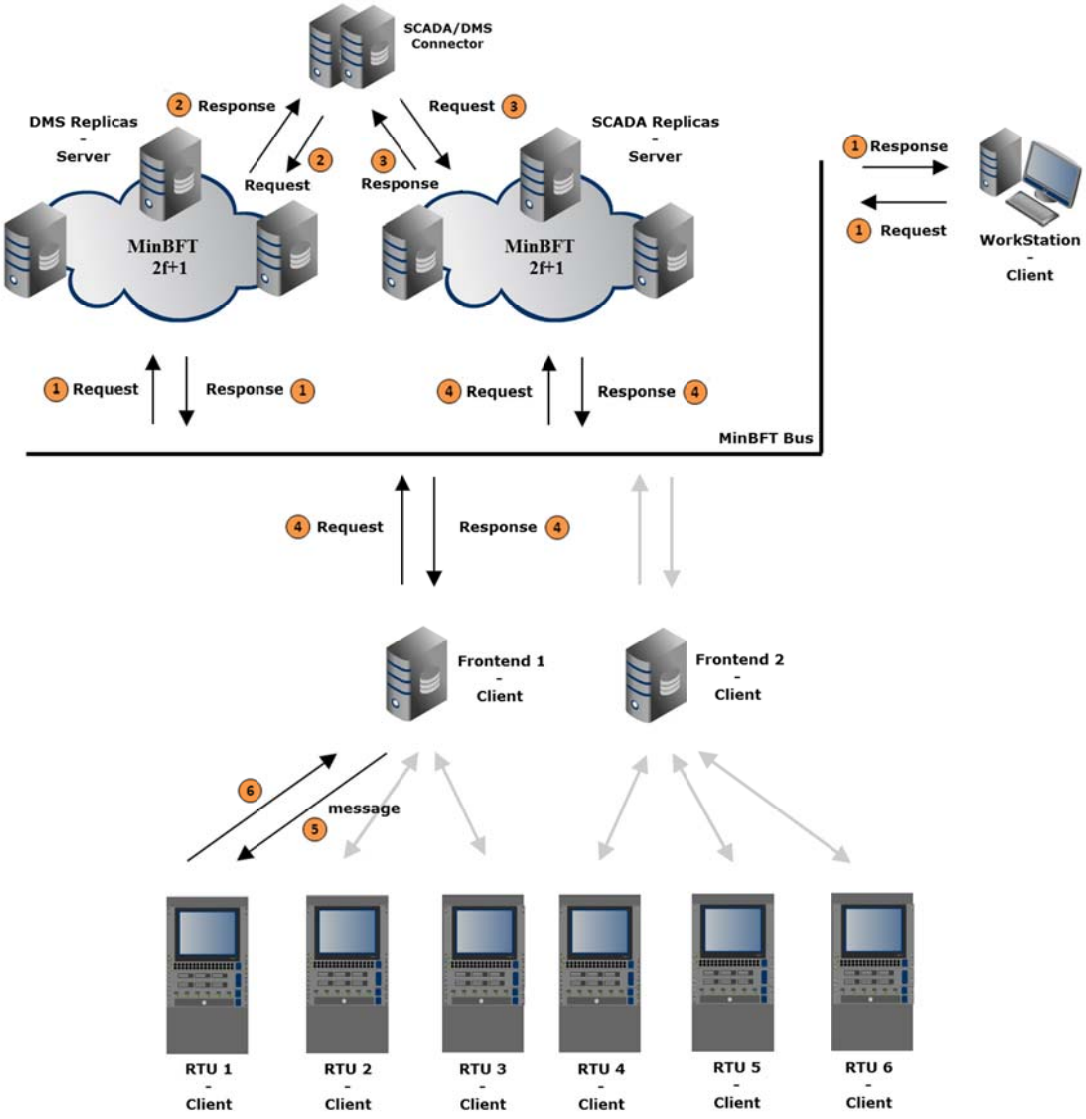


Figure 41 - Control data flow representation.

1. The workstation operator executes a remote control over one of the telemetered components of the electrical power grid. The workstation will post the command request on the DMS service. The DMS service is responsible for processing all the logic associated with the executed command, validating the current state of the equipment and the inhibit conditions that are configured for that command.

2. If the command can be executed, it will be sent to the SCADA service using the SCADA/DMS Connector, as previously explained. The Connector will periodically request the DMS service for any pending commands to be sent to the SCADA service.

3. The second part of the service-to-service exchange occurs when the Connector requests the SCADA service to deliver the information previously requested to the DMS replicas. This way, the required transfer of data between both replica sets is performed. The SCADA service then processes the control and place it in a stack with the pending remote controls.

4 The Frontends should also have a periodic task requesting the SCADA service for any new remote command for their mapped RTUs, in their pending commands stack. Needless to say that the control message gets processed and translated at the Frontend, so that it can be delivered in the correct format to the correct destination.

5 Immediately after, the remote control is delivered to the RTU, which will first verify if the control message corresponds to any of its components, and once validated, execute it in the physical component.

6 Since a successful remote command corresponds to a telemetry change, the rest of the process is similar to the one presented before, in the telemetry data flow representation. The main difference is that the change of state, whatsoever, will not produce an alarm since it was manually executed by a system operator. Once the DMS receives the change of state correspondent to the previously delivered remote control, it is confirmed as successful.

4.7 Conclusions

In this chapter we proposed a fault- and intrusion-tolerant architecture for the GENESys system. We addressed the three layers of the GENESys architecture with three dedicated solutions which we believe that should be combined with the view of better global results. For the two lower layers of the system, the Telemetry Sites and the Frontend Sites, we suggest fault-tolerant mechanisms based on dual redundancy with applicational management. For the backend systems layer we proposed the implementation of an

intrusion-tolerant replication protocol since both SCADA and DMS services are crucial for a dependable and secure management of the EDP Distribuição power grid.

In the next chapter we perform an analysis of our architecture regarding its fault and intrusion tolerance capabilities, and also its cost-benefit relationship.

Chapter 5

Analysis

After presenting the Fault- and Intrusion-Tolerant GENESys system we will now evaluate our proposal. Considering we have different solutions focusing on the different layers of the GENESys architecture, at each analysis we will separate the different components to have a better definition of how they are influenced by the proposed solution.

In this Chapter we perform two different analyses. The first analysis objective is to understand which are the fault tolerance capabilities introduced to the different layers of GENESys by our solutions. In the second, we perform a cost-benefit analysis to infer about the viability of our proposals by acknowledging both their costs and the technical and operational benefits.

5.1 Fault Tolerance Analysis

In this section we perform a thorough analysis focused on the fault tolerance capabilities of the several system components: *Remote Terminal Units*, *Frontends* and *SCADA/DMS servers*, before and after our proposed solutions.

5.1.1 Methodology

We will address each of the components before and after the implementation of the mechanisms previously proposed. The idea is to be able to acknowledge the direct benefits of each of the proposed solutions regarding the fault tolerance capabilities of the system. Since these capabilities describe the ability of each of the components to sustain and tolerate the presence of faults, we thought that it would be interesting to encompass the different types of faults in such an analysis.

The analysis will be divided in categories correspondent to the different types of fault:

- ***Omissive faults –*** Component not performing an interaction it was specified to:
    - *Crash faults* - An RTU or Frontend crashes stopping the interactions with the backend systems;
    - *Omission faults* - An RTU or Frontend is unable to communicate with backend systems;
    - *Timing faults* - The Frontend server delivers unorganized telemetry events to the backend systems;
- ***Assertive faults –*** Interactions not performed to specification:
    - *Syntactic faults* - An RTU sends a character in the state value field of the telemetry message;
    - *Semantic faults* - An RTU sends a non-existing state value on the telemetry message;
- ***Arbitrary faults –*** Any type or combination of faults caused by an intrusion in the SCADA or DMS replicas.

As referred, we will divide our analysis in the three different layers of the GENESys architecture. For each layer we will describe in which way the current implementation deals with the presence of the different faults presented (see *Capabilities* in Tables 6, 7 and 8). Subsequently, we describe the improvements that our proposals introduce at each of the layers (see *Improvements* in Tables 6, 7 and 8). Last, we discuss about the limitations of the proposed architectures regarding each of the fault scenarios presented. These limitations should be addressed in future work.

5.1.2 Redundant Remote Terminal Unit

In this specific solution we suggest the implementation of dual redundancy by installing a pair of RTUs at a given electrical facility. We also advise the implementation of independent communication links between each of the equipments and the correspondent Frontend to guarantee the minimum number of single points of failure. The fault tolerance analysis is presented in Table 6.

	Capabilities	Improvements	Limitations
Omissive Faults	Given the dual redundancy of the RTU equipment, this layer is capable of tolerating an omissive fault on one of the two redundant equipments. This capability is available due to the reactive commutation based on the hot-standby configuration.	The RTU site had no previous omissive fault tolerance capabilities. It is now able to tolerate omissive faults only with a reduced downtime correspondent to the communication timeout and the hardware commutation. Our proposed implementation aims on removing all single points of failure, with the exception of the site router and the power supply.	The RTU omissive faults tolerance is limited to one of the equipments at a time. Considering our suggested implementation, we removed several single points of failure, but not all. If an omissive fault occurs in both RTUs, the operator will not be able to remote control and supervise the electrical facility.
Assertive Faults	This implementation will not provide any assertive fault tolerance capabilities to the RTUs.	There are no improvements on the current state.	The system remains vulnerable to assertive faults. This can lead to the delivery of incorrect information to the systems. However, the range of error propagation is confined to the electric facility correspondent to the faulty RTU.
Byzantine Faults	This implementation will not provide any Byzantine fault tolerance capabilities to the RTUs.	There are no improvements on the current state.	The system remains vulnerable to Byzantine faults. This can lead to the inability for remote control and supervise the electrical facility, to the injection of incorrect information on the backend systems or the execution of unauthorized controls over the electrical facility. However, the range of error propagation is confined to the electric facility correspondent to the faulty RTU.

Table 6 - Fault tolerance capabilities of Redundant RTU (RRTU).

5.1.3 Fault-Tolerant Frontend architecture

In the Fault-Tolerant Frontend (FTFE) solution we propose a different architecture and implementation for the current dual redundant Frontend Sites. There are two main differences: first, the Frontend operation is based on online-online configuration where both Frontends, if correct, will be delivering service; second, the redundant Frontends will be deployed in different geographical locations, to tolerate disasters, large-scale attacks (e.g., DDoS) that may lead to site failures. The fault tolerance analysis of the FTFE is presented in Table 7.

	Capabilities	Improvements	Limitations
Omissive Faults	Given the dual spacial redundancy of the Frontend equipment, this layer is capable of tolerating an omissive fault on one of the two redundant equipments and any of the communication nodes between the RTU and the Frontend server itself.	The proposed architecture eliminates the single points of failure of the dual local redundancy, on the current architecture. It is an important improvement for the dependability of the Frontends since they might cover hundreds of electrical facilities. Furthermore, the link separation between RTUs and Frontends allows a deeper diagnosis on RTU failures, based on the communication port based commutation, rather than the conventional Frontend server commutation, which provides no response for an RTU failure.	The Frontends omissive faults tolerance is limited to one of the equipments at a time. In spite of relieving the redundant equipments from single points of failure and all the common location deployment vulnerabilities, it might happen that both fail at the same time. If an omissive fault occurs in both Frontends, the operator will not be able to remote control and supervise the electrical facilities mapped under that set of Frontends.
Assertive Faults	This implementation will not provide any assertive fault tolerance capabilities to the Frontends.	There are no improvements on the current state.	The system remains vulnerable to assertive faults. This can lead to the delivery of incorrect information about all the Frontend mapped RTUs to the backend systems. However, the range of error propagation is confined to the electric facilities which are mapped to the correspondent faulty Frontend.
Byzantine Faults	This implementation will not provide any Byzantine fault tolerance capabilities to the Frontends.	There are no improvements on the current state.	The system remains vulnerable to Byzantine faults. This can lead to the inability for remote control and supervise the electrical facilities mapped to the Frontend, to the injection of incorrect information on the backend systems or the execution of unauthorized controls over the electrical facilities mapped to the Frontend; however, the range of error propagation is confined to the electric facilities which are mapped to the correspondent faulty Frontend.

Table 7 - Fault tolerance capabilities of FTFE.

5.1.4 SCADA and DMS Intrusion-Tolerant Architecture

In this specific solution we suggest an intrusion-tolerant architecture for both the SCADA and DMS servers. The proposal is based on Byzantine quorum protocols for state machine

replication. Our implementation includes the deployment of three SCADA and DMS server replicas at different locations guaranteeing not only redundancy but also validity on the operations performed. Since we propose an intrusion-tolerant architecture we will focus our analysis on Byzantine faults only which will encompass all existing types of fault, as explained in Section 2.1.1. The fault tolerance analysis is presented in Table 8.

	Capabilities	Improvements	Limitations
Byzantine Faults	The proposed solution provides the ability of tolerating one faulty replica with Byzantine behavior, from a set of 3 replicas. In such case, the two remaining replicas will be able to deliver a correct service without any noticeable glitch for the system operators.	The SCADA and DMS servers are currently deployed with dual local redundancy and based on a hot-standby operation. This type of implementation is limited to omissive fault tolerance capabilities (when only one component is faulty) and vulnerable to single points of failure and all the common location deployment vulnerabilities. The proposed solution addresses such limitations and provides the capability of tolerating intrusions and faults of any type on at most one of the deployed replicas. Considering the criticality of the GENESys system, such a paradigm comes as a great improvement on the system dependability.	The SCADA and DMS replicas are only tolerant to at most one faulty component. If two or more replicas are compromised, the system will not be able to guarantee a correct operation, which may be catastrophic since the attacker might have full control over the electrical distribution power grid. However, it is possible to extend the system to tolerate *f>1* faults, if there are enough resources available.

Table 8 - Fault tolerance capabilities of Intrusion-Tolerant SCADA and DMS services.

5.1.5 Discussion

In this section we analyzed and compared the fault and intrusion tolerance capabilities of the current GENESys solution and the proposed Fault- and Intrusion-Tolerant GENESys architecture. We divided the analysis for the three different layers of the architecture and our objective was to identify which were the main improvements regarding their fault coverage with the potential deployment of our solutions.

The current RTU site implementation has no fault tolerance capabilities. Our Redundant RTU (RRTU) proposal tolerates omissive faults due to redundancy. This capability will reflect in the reduction of the RTU site downtime.

The current Frontend Site layer already adopts fault tolerance mechanisms. The improvements are related with the elimination of communication infrastructure single points

of failure due to the geographical separation of the Frontend servers. From the backend systems point of view this will result in the reduction of omission failures from the Frontend Group.

In the backend systems the improvements are more significant. The SCADA and DMS servers are currently capable of masking omission faults due to dual redundancy. However, with our proposed BFT state machine replication approach both critical services will be capable of tolerating byzantine faults, e.g., malicious intrusions.

5.2 Cost-Benefit Analysis

In this section we will perform a cost-benefit analysis of the fault-tolerant solutions presented in Chapter 4. The objective here is to present the main operational and technical advantages of the several proposals, trying to highlight the costs and benefits of its implementation.

Our cost analysis will be mainly focused on the increase of the investment value of the proposed solutions, when compared with the one currently deployed taking into account hardware, software and connectivity requirements. The benefit analysis will be based on the collection of downtime of the several GENESys components, and the study of how our various proposals will affect these times. We will also identify some benefits that are not measurable but that for sure would have very serious economic and operational advantages.

During our analysis we will perform statistical computations based on information related with the year 2010. To simplify our exposition we will only use average values for each type of component particulars, for a matter of clarification and simplicity. However, the impact will still be perceived.

It is important to refer that since there is no current applicational off-the-shelf solution for our proposals, there are research and development (R&D) costs associated with each of the implementations which are not measurable at the time being. They would have to be subject to a technical and financial analysis by the SCADA system vendors. For that reason we will depreciate this value since it is considered a start point investment, which will not affect each of the future deployments. A high scale deployment of our proposed solutions will dilute the initial R&D investment.

As already discussed, the GENESys system is the supervision and remote control system for the EDP Distribuição electrical facilities. When there is a physical event in one of the telemetered equipments, the GENESys system will react to that event in a way it was pre-configured to do. However, if there is an incident on the GENESys system other than a remote control, there will be no direct consequence to the physical infrastructure. Meaning that, if any layer of the GENESys fails in a non Byzantine way, the power grid itself will not be affected, just its supervision and remote control capabilities. Therefore, whenever we refer in the following analysis to the GENESys downtime, we refer just the inability of monitoring the power grid, and not downtime of the grid itself.

5.2.1 Redundant Remote Terminal Unit

From the backend systems point of view an RTU failure is an abstraction of a telemetry site communication failure. When the system observes an RTU failure there is no way of knowing exactly what lead to the disruption of communications. Such failure leads to a facility telemetry downtime which has no direct relation with its physical operation. However, in the case of an erroneous event in the physical facility, the GENESys operator would be unable to detect and correct it.

There are several fault scenarios which could lead to the RTU communication failure and our RRTU proposal aims on solving one of them, which is, when the RTU (or any of its internal components) fails. We will analyze the costs associated with the implementation of our proposed solution and quantify its benefits.

Cost. The cost analysis will be based on the cost specification of a standard EDP Distribuição HV/MV substation. We analyzed the system architecture and identified the required add-ons for our proposed solutions (see Appendix A).

The cost analysis is represented in Table 9. The cost of installing a telemetry system in a remote substation is around 223.000 €. From this total cost, only 18.000 € correspond to the RTU equipment itself. After our previous architecture analysis we retained that the implications of our proposed solution would be the addition of a new RTU equipment, a partial increase of 25% on the overall commissioning, and the required sight acceptance tests (SAT) of the new RTU, to validate the database configuration and the network connectivity. Apart from that, the current site system architecture would ensure all other requirements.

Pos.	Designation	Quantity	Price
1	**Equipment**		190.000 €
1.1	RTU	1	18.000 €
1.2	Other	1	172.000 €
2	**Development**		8.000 €
3	**Testing**		10.000 €
3.1	Factory Acceptance Test	1	5.000 €
3.2	Sight Acceptance Test	1	5.000 €
4	**Comissioning**		15.000 €
	Total		223.000 €

Pos.	Designation	Quantity	Price
1	**Equipment**		208.000 €
1.1	RTU	2	36.000 €
1.2	Other	1	172.000 €
2	**Development**		8.000 €
3	**Testing**		10.000 €
3.1	Factory Acceptance Test	1	5.000 €
3.2	Sight Acceptance Test	2	5.000 €
4	**Comissioning**		18.750 €
	Total		244.750 €

a) Cost analysis of a standard RTU architecture. b) Cost analysis of a RRTU architecture.

Table 9 - Cost comparison between a standard and a Redundant RTU architecture.

The results are very encouraging. The integration of the proposed RRTU in the current telemetry system would generate an increase of 21.750€ on its total cost which represents

only an increase of 9,75% on the total cost of installing a telemetry system in a remote substation.

Benefit. The RTUs are subject to harsh environments which might sometimes lead to the damage of one of its critical components. However, the failures on any of the RTU components do not represent the most common causes for the RTU downtime. Beyond that, the RTU downtime can also occur due to a failure on the power supply or a failure on any network equipment between the RTU equipment and the Frontend server (i.e., RF transceivers, IP routers or communication channels). After an analytical survey made by EDP Distribuição we got the estimates, in a non-exact way, about the probabilistic details for the different RTU downtime origins. In any case, as we do not have any other more reliable source we will use it for the analysis. The downtime causes are listed in Table 10.

RTU Downtime Causes	Probability
RTU Motherboard	4%
RTU Communication Board	2%
Other RTU components	1%
Network	92%
Other	1%

Table 10 – Probability of the several RTU downtime causes.

As it is possible to observe, only 7% of the GENESys' RTUs downtime is directly connected with the RTU equipment itself. All other unavailability events are related with the network (e.g., network media, network equipments) connectivity between the RTU component and the Frontend Site, or other minor occurrences such as power supply failure. Notice that the majority of faults are related with the communication infrastructure, these faults are partially addressed by our FTFE architecture (see next section).

The Redundant Remote Terminal Unit (RRTU) is a capable solution for mitigating those 7% failures since it provides an extra RTU. We will next explore how that percentage is reflected in the GENESys system, based on RTU downtime information regarding the year of 2010. To infer about the RTU downtime benefits, we will use the average of all the 411 EDP Distribuição HV/MV substations downtimes to perform our analysis. The specific telemetry downtimes of the several substations, used for the average values calculations, were based on the 2010 GENESys Key Performance Indicators Report [42].

As previously referred, we will use the average values and apply them on a representative RTU. The average RTU downtime details are presented in Table 11**.**

Telemetry Site Solution	Downtime (hours)	Uptime (%)
Single RTU	16:50:00	99,7338
Redundant RTU	15:20:00	99,7525

Table 11 – RTU Site communication details for the conventional and the proposed RRTU architectures. We provide the communication downtime and uptime percentage.

Considering that the proposed solution would result in the immediate resolution of 7% of the RTU related failures causing the telemetry downtime, the improvements of the RTU downtime indicators, also described in Figure 48, would be approximately 1:30hs per RTU.

Discussion. The integration of the proposed RRTU in the current telemetry system would generate an increase of 21.750€ on total cost per RTU. Meaning that any new substation would have an increase of 9,75% on its total cost, with the benefit of reducing system downtime in 7%. Our proposal requires an investment of 21.750€ for Telemetry Site to guarantee an average of 1 hour and 30 minutes increase on the RTU uptime, each year. These facts are important to understand if the benefits cover the investment.

5.2.2 Fault-Tolerant Frontend Architecture

A Frontend downtime is related with any failure associated with the Frontend Group and the communication link between the Frontend Site and the GENESys backend systems. The current Frontend architecture already offers error masking capabilities due to the fail-over redundancy adopted. However, both redundant servers (i.e., Frontend Group) are located in the same Frontend Site which has some disadvantages, leading to a large number of Frontend Site failures. When this happens, we have a Frontend Site downtime and consequently the downtime of all of its mapped RTUs. These failures occur possibly due to a single point of failure (i.e., a failure which affects both Frontend servers). The objective of our Frontend architecture proposal is exactly to mitigate these kinds of failures which have a relevant effect on the GENESys operation, considering that a large number of facilities lose their supervision and remote control capabilities.

In the rest of the section we will analyze the potential costs of the proposed solution and in which way it affects the current downtime indicators and the GENESys operation.

Cost. The current implementation already includes redundant Frontend servers, for error masking. We will not need to introduce any infrastructure or equipment in the GENESys architecture, what we will require is the reconfiguration of the Frontend layer so it can be

adapted to our proposal. This procedure will not include any new components, just the logistics costs of a new implementation and integration of the current servers. These costs are associated with the disassembly of the Frontend servers, their transportation, the reassemble on the new Frontend Site, and last but not least, their configuration and testing. By comparing these requirements with some previous projects we had on EDP Distribuição, we estimate the total cost of 2.500€ per Frontend server and consequently 5.000€ per Frontend Site. Once again, we consider the development costs negligible and diluted in the final product.

Since we currently have 16 Frontend Sites we will require a total investment of 95.000€ to implement our proposed Fault-Tolerant Frontend (FTFE) architecture.

Benefit. In our proposed FTFE architecture there is a change of paradigm on the Frontend operation mode and in their geographical distribution. The redundant Frontends will start operating in an online-online mode in which both servers are actively communicating with RTUs. Furthermore, they will be located in separate sites to guarantee the mitigation of Frontend Group failures, since each RTU will now be able to communicate with both redundant servers located at different sites.

The current Frontend architecture is already capable of handling Frontend server failures, since it has dual redundancy which is triggered whenever the online Frontend crashes. In these cases, the downtimes of the Frontend Site and the correspondent mapped RTUs are minimal. However, there are other failure scenarios which lead to critical downtime on the Frontend Site and consequently the GENESys operation itself. As it was referred, these failures are usually associated with network components or the power supply, which are common for both redundant Frontends. We analyzed the Frontend server failures for different locations during 2010. Since most of the failures are immediately solved by redundancy, our aim was to identify the cases in which both Frontend servers failed, since this situation is exactly what our proposal intends to address. In Table 12, we present the Frontend Sites downtime details for the 16 Frontend Sites of the GENESys system.

North				South			
Frontend Site	**Downtime (min.)**	**Uptime (%)**	**Mapped RTUs**	**Frontend Site**	**Downtime (min.)**	**Uptime (%)**	**Mapped RTUs**
AVEIRO	00:55:00	99,9893	202	BEJA	04:10:00	99,9512	181
BRAGA	02:17:00	99,9733	212	CARENQUE	01:50:00	99,9785	160
CASTELO BRANCO	00:00:00	100,0000	34	LEIRIA	11:50:00	99,8615	293
COIMBRA	00:31:00	99,9940	303	LOULE	02:09:00	99,9748	263
PORTO	00:11:00	99,9979	278	LOURES	04:26:00	99,9481	157
SEIA	00:55:00	99,9893	287	OLHO BOI	08:17:00	99,9031	156
VILA REAL	01:09:00	99,9865	186	PALHAVA	00:10:00	99,9980	58
RUIVAES	01:02:00	99,9879	154	SETUBAL	01:55:00	99,9698	384

Table 12 - Frontend Sites communication details in 2010. The analysis is divided in the two regions in which the system is separated. For each Frontend Site we provide the communication downtime, the uptime percentage and the number of mapped RTUs.

The table shows that the total downtime of the Frontend Sites is far from negligible. We were able to identify a total of 17 hours and 47 minutes of Frontend Sites downtime, which represents not only the total downtime of the two redundant servers but also the downtime of all the RTUs mapped to them. Considering that each Frontend Site collects information from a large number of electrical facilities, the results suggest that the GENESys service can be seriously affected in the event of a Frontend Site failure.

Our proposed FTFE architecture aims on reducing the downtime values collected in Table 12. The main benefit of our solution is that by separating the redundant Frontends we will eliminate all single points of failure which are responsible for the crash of Frontend Group and reflect in the values presented above.

Furthermore, we will also be able to reduce the RTU downtime expressed in Table 11, considering that most RTU downtime are due to network failures, as presented in Table 10. The FTFE architecture will imply having separate network paths for the redundant Frontends, therefore, some of the network faults which lead to the RTU downtime can be overcome by the link redundancy.

Discussion. The proposed architecture does not require a great investment to be put into practice. In spite of requiring some software development for the SCADA system to be able to perform all the involved operations, we did not incorporate these costs, associated with the vendors R&D. It does not require any new components, however, it requires a redistribution of the Frontend servers that comes with some logistics costs

The benefit of the FTFE architecture proposed in this thesis is the expected elimination of the total downtime of the Frontend groups. It translates in uptime gain of 17 hours and 47 minute. This is achieved due to the geographical separation of the redundant Frontends and, consequently, the mitigation of an important single point of failure (the Frontend Site). Furthermore, by combining this solution with the RRTU it is possible to reduce the RTU communication downtime since we will have redundant communication to the upstream system.

5.2.3 SCADA and DMS Intrusion-Tolerant Architecture

The SCADA and DMS servers are the most critical components of the GENESys architecture. They are responsible for the processing of all telemetry information, for the processing of every workstation interface and for the correct integration with the other GENESys systems. Furthermore, there is the GENESys application running in each workstation which relies on the communication with both systems to be able to deliver a correct service to the system operators. The existence of correct SCADA and DMS services is indispensable for the supervision and remote control of the whole EDP Distribuição power grid.

Taking this fact into consideration, a few years ago, the fault tolerance capabilities of the servers were addressed by implementing a dual redundancy mechanism for each of the servers. However, this redundancy covers only faults on the server component but not on the communication network responsible for the inter-connection of these servers with the applicational interface, since they are located in the same site. The total downtime of the system is primarily associated with this fact. Our solution will address this type of fault scenarios since each SCADA and DMS service replica will be deployed in different sites, to guarantee rule out the existence of any single points of failure.

The electrical power grid is very attractive target for Cyberterrorism since it is considered a critical infrastructure and because of the impact it would cause. For the last few years SCADA systems attacks have become very common, and in many cases very dangerous [43]. In this thesis we propose a solution that will also address intrusions and will be able to tolerate them, under a set of assumptions. The MinBFT protocol [17] we are applying will provide an extra layer of security to the SCADA and DMS servers since the state machine replication algorithm will guarantee that if an attacker compromises one of the system replicas he will not be able to control and jeopardize the power grid operation.

We will next perform a cost-benefit analysis of the proposed SCADA and DMS architecture where we will estimate the solution costs and acknowledge about its measurable benefits.

Cost. In the current GENESys implementation there are two SCADA and DMS servers in each region (north and south) which makes a total of eight servers.

Our proposed architecture requires the deployment of a SCADA replica, a DMS replica and a SCADA/DMS Connector at two of the three chosen sites and on the third site just one replica for each service (see Figure 34). We were able to gather the information on the acquisition of the latest SCADA and DMS servers to estimate the price of the six required replicas, their cost is about 5.000€ each. The two SCADA/DMS connectors also have to be acquired but considering their simpler operation they might be less performant machines, therefore, their estimate price is 2.500€. The values include the hardware, software and the GENESys licensing. The machines deployment, configuration and SAT costs about 2.500€ for each machine. There are other costs to acknowledge related with the management of diversity[2] for each replica of the BFT services (SCADA and DMS). However, recent studies show that it is possible to achieve substantial gains using diverse but similar configurations of database management systems and operating systems [44] [45], thus reducing the management costs to values close to what EDP Distribuição have with the current system. The total cost per site will correspond to the acquisition and deployment of the required components, as presented in Table 13. The total cost of the proposed Byzantine fault-tolerant solution will be of

[2] each component uses different software to perform the same functions, with the expectation that the differences will reduce the occurrence of common vulnerabilities [44].

42.500€. Once again, the research and development costs of the GENESys platform based in BFT state machine replication are not included.

Pos.	Designation	Porto		Alto de S. João		Palhavã	
		Quantity	Cost	Quantity	Cost	Quantity	Cost
1	**Equipment**		12.500 €		10.000 €		12.500 €
1.1	SCADA replica	1	5.000 €	1	5.000 €	1	5.000 €
1.2	DMS replica	1	5.000 €	1	5.000 €	1	5.000 €
1.3	SCADA/DMS connector	1	2.500 €			1	2.500 €
2	**Deployment**		2.500 €		2.500 €		2.500 €
	Total		15.000 €		12.500 €		15.000 €

Table 13 – Cost of the proposed backend systems architecture (per site).

It is important to refer that the proposed backend systems implementation requires the same number of servers that are already deployed in the current architecture, however, with two potential advantages. First, the two servers required for the SCADA/DMS Connector can be a lot simpler than the others, thus, cheaper. Secondly, since we propose the deployment of the service replicas in different sites we potentially would no longer need the Disaster Recovery system.

Benefit. Our proposal results in a total change of the backend system architecture and operation. The inclusion of Byzantine state machine replication protocols will allow the GENESys system to tolerate the presence of a replica with arbitrary behavior, which can even be caused by an intrusion. Obviously, the ability to tolerate intrusions is one of the main advantages of our proposal, considering the constant and far-fetched threats of the cyber world.

The other advantage is related with the system downtime, in the same line to what we previously did for the other system layers. As it was referred, the SCADA or DMS server's failure is the most critical threat for power grid operators. When they lose the GENESys interface they are unable to operate the entire grid and react to any equipment failure. Although the system is already embedded with the server redundancy, there are some failures which are reflected in the total downtime of the GENESys system, as it is reflected in Table 14. These failures are commonly caused by with the network equipments which connect the servers with the remaining GENESys infrastructure (shared by both replicas). Our proposal will solve the problems related with these existing single points of failure since we propose the SCADA and DMS service replicas deployment in different locations, where the mesh SDH network will guarantee inter-connectivity and link redundancy.

Site	Downtime (min.)													Uptime (%)
	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	2010	
North	207	0	107	89	163	52	116	99	40	110	0	6	989	99.81
South	5	70	100	0	21	23	73	60	0	72	45	290	759	99.85
Total	212	70	207	89	184	75	189	159	40	182	45	296	1748	99.66

Table 14 – SCADA/DMS downtimes (in minutes) per month in 2010.

We believe that by applying our SCADA and DMS layer proposal we will still be able to tolerate one replica crashes, as it currently happens with the dual redundancy. Moreover, and perhaps even more importantly, we will guarantee an environment without any single point of failure, where the expected probability of total system downtime will only depend on the probability of two SCADA and DMS servers be failed at the same time. This will potentially reflect on the mitigation of the downtime values collected, which may be critical in the event of a full scale disaster in the electrical power grid.

Discussion. The implementation of the proposed architecture for the SCADA and DMS services will require a change of paradigm for the whole infrastructure. It will require a development effort especially regarding all the system interactions which will now be dependent on the MinBFT protocol constraints. Once those operational details are specified and the intrusion-tolerant SCADA and DMS systems are fully developed and stabilized, the implementation of the solution in EDP Distribuição will only require an investment effort of 40.000€, for the acquisition of the required machines and the installation and reconfiguration of the entire SCADA and DMS service replicas plus the SCADA/DMS connectors.

The benefits of such an implementation are clear and vital for the best possible management of the entire EDP Distribuição power grid. Furthermore, taking into account the risks of the cyber world, providing the critical systems with intrusion tolerance capabilities will only recognize EDP Distribuição as a company which is able to predict and respond to modern challenges and concerned on providing the best and most secure service possible.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The GENESys platform was introduced as an indispensable tool for the operation of the EDP Distribuição power grid. The grid operators depend on GENESys to monitor the electrical infrastructure and to be able to perform a fast and effective response to any unexpected event. However, the system is not perfect. The amount of interconnected nodes subject to different communication layers sometimes results in a system downtime. This downtime does not reflect any behavior on the grid itself, however, in the absence of GENESys the grid operator is unable to detect and respond to any incident.

The system is mainly composed by three different layers, the electrical facility layer which as an RTU for telemetry functions, the Frontend layer with Frontend servers that collect telemetry information by a large number of RTUs, and the system layer with all the backend system, including the SCADA and DMS servers. In their own way, all layers have some weaknesses which might affect the delivery of correct service of the GENESys system itself.

One of the objectives of this master thesis was the identification of the main weaknesses of the GENESys platform. These weaknesses were associated with the incapacity to tolerate some faults. In spite of having already some fault-tolerant capabilities (mostly based on primary-backup replication [19]), there is much to be done regarding the fault tolerance capabilities of such a critical system.

We have proposed a new GENESys architecture where we address the three different layers of the system. We made different fault assumptions for each of the layers considering the criticality of their failures and the costs of the solutions, and identified in which way we could better address those weaknesses. We presented fault-tolerant architectures for the RTU and Frontend layers, based on redundancy and applicational management, and an intrusion-tolerant architecture for the SCADA and DMS backend systems whose correct service is imperative for the GENESys operation. The decision on choosing an intrusion-tolerant architecture for the backend systems was not only aiming on improving their robustness but

also their security, because if these systems are attacked and compromised, the consequences on the EDP Distribuição power grid can be disastrous.

We performed a fault-tolerant analysis of the current and the proposed GENESys architecture where we identified how the system currently deals with different fault scenarios, and how it would deal with them under our proposed solution.

Furthermore, we performed a cost-benefit analysis where we identified the costs of each of the three separated solutions and presented their main operational and technical advantages. We reason about the costs of its implementation in which way they correspond to any technical, operational and financial gain.

We believe that throughout our work we presented a logically chained reasoning regarding the dependability and security of the GENESys infrastructure. The reasoning is that the existing system lacks on robustness and resiliency and that there other available fault-tolerant mechanisms that can be applied to address those critical requirements.

It is worth to notice that our proposal does not solve all the resilience and security issues for such complex system. However, we believe it is a clear demonstration that there is much to be done to build a truly dependable SCADA system for EDP Distribuição, and more importantly, we showed that intrusion tolerance is a practical paradigm to do it.

6.2 Future Work

As future work we propose:

- To carry out a performance evaluation of the Fault- and Intrusion-Tolerant GENESys system. The objective would be to acknowledge about the most important communication indicators, such as the throughput and latency, and compare them with the ones regarding the current architecture.

- The implementation of a prototype could be interesting to have a better perception of the overall operation of the proposal and also to be able to produce better results on the evaluation mentioned above;

- To improve the quality of the data collected regarding the probabilistic causes for the components failures. This would lead to relatively more intelligible results on the benefit analysis of our solution.

- As referred in Section 5.1 there are some limitations on the fault and intrusion tolerance mechanisms we are proposing. These limitations should be further addressed in future work.

Bibliography

[1] Wikipedia. Near real-time. Wikipedia, 2011. [En ligne] http://en.wikipedia.org/wiki/Near_real-time.

[2] *G. S. Veronese, M. Correia, A. Neves Bessani, L. Lung, P. Veríssimo. Efficient Byzantine Fault Tolerance. IEEE Transactions on Computers. Accepted for publication. To appear.*

[3] *P. Veríssimo and L. Rodrigues. Distributed Systems for System Architects. Kluwer Academic Publishers, 2001.*

[4] *J. Fraga and D. Powell (1985). A Fault and Intrusion-Tolerant File System, in IFIP 3rd Int. Conf. on Computer Security, (J. B. Grimson and H.J. Kugler, Eds.), Dublin, Ireland, Computer Security, pp.203-218.*

[5] *Guy Dewsbury and John Dobson. Responsibility and Dependable Systems. Springer, pages 2-5, 2007.*

[6] A. Avizienis, J.C Laprie, B. Randell and C. Landwehr. Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Transactions on Dependable and Secure Computing, Vol.1, N.1, Pages: 11-33, 2004.

[7] A. Haeberlen, P. Kouznetsov, P. Druschel. The Case for Byzantine Fault Detection. In Proceeding of HOTDEP'06, Vol.2, 2006.

[8] P. P. Pal, F. Webber, R. E. Schantz and J. P. Loyall. Intrusion Tolerant Systems. 2000.

[9] A. Adelsbach, D. Alessandri, C. Cachin, S. Creese, Y. Deswarte, K. Kursawe, J. C. Laprie, D. Powell, B. Randell, J. Riordan, P. Ryan, W. Simmonds, R. Stroud, P. Veríssimo, M. Waidner, and A. Wespi. *Conceptual Model and Architecture of MAFTIA. Project MAFTIA deliverable D21, January 2002.*

[10] Paulo Veríssimo, Nuno Ferreira Neves, and Miguel Pupo Correia. P. Veríssimo, N. F. Neves and M. Correia. Intrusion-tolerant architectures: Concepts and design. In R. Lemos, C. Gacek, and A. Romanovsky, editors, Architecting Dependable Systems, volume 2677 of Lecture Notes in Computer Science. Springer-Verlag, 2003.

[11] *F. Schneider. Implementing Fault-Tolerant Services Using the State Machine Approach: A Tutorial. ACM Computing Surveys 22, Vol.4, 1990.*

[12] *M. Castro and B. Liskov. Practical Byzantine fault tolerance. In Proceedings of OSDI'99, pages 173-186, 1999.*

[13] Wikipedia. Byzantine fault tolerance. Wikipedia, 2011. *Wikipedia.* [En ligne] http://en.wikipedia.org/wiki/Byzantine_fault_tolerance.

[14] The Encyclopedia of Science. Determinist System. The Encyclopedia of Science. 2011. [En ligne] http://www.daviddarling.info/encyclopedia/D/deterministic_system.html.

[15] F. Schneider. Decomposing Properties into Safety and Liveness using Predicate Logic. N. 87-874, Oct. 1987. October, 1987.

[16] *R. Kotla, L. Alvisi, M. Dahlin, A. Clement, and E. Wong. Zyzzyva: speculative Byzantine fault tolerance. In Proceedings of the 21st Symposium on Operating Systems Principles, Oct. 2007.*

[17] G. Veronese, M. Correia, A. Bessani, L. C. Lung, P. Veríssimo. Minimal Byzantine Fault Tolerance: Algorithm and Evaluation. DI-FCUL TR-2009-15, June 2009.

[18] Siemens Energy. Auto-Reclose and Check Synchronising. Siemens Energy, 2011. [En ligne] http://www.energy.siemens.com/hq/en/automation/power-transmission-distribution/protection/reyrolle/auto-reclose/.

[19] *N. Budhiraja, K. Marzullo, F. Schneider and S. Toueg. Distributed systems, The primary-backup approach. ACM Press/Addison-Wesley Publishing Co. New York, NY, USA, 1993.*

[20] Principle Partners, Inc.. Data Warehouse Concepts And Architecture. Principle Partners, Inc., 2011. [En ligne] http://principlepartners.com/presentations/DataWarehouseConceptsAndArchitecture.pdf.

[21] *N. Solomon. Business Intelligence. Communications of the Association for Information Systems, Vol.13, 177-195, 2004.*

[22] *International Standard - IEC 60870-6-802.* IEC. s.l. : IEC, 2005.

[23] A. Bessani, P. Sousa, M. Correia, N. Neves, P. Veríssimo. The CRUTIAL Way of Critical Infrastructure Protection. IEEE Security & Privacy, Vol.6, N.6, Nov/Dec. 2008.

[24] *ITU-T recommendation G.803: Architecture of Transport Networks Based on the Synchronous Digital Hierarchy (SDH).*

[25] Magazine, Director's and Boards. Business Continuity and Disaster Recovery. *Boardroom Briefing.* Spring, 2006.

[26] EDP Distribuição. Disaster Recovery Plan. EDP Distribuição Internal Document, 2009.

[27] Ezine Articles. Paul E. Moore, What Is the Difference Between Hot, Warm and Cold Disaster Recovery?. Ezine Articles, 2011. [En ligne] http://ezinearticles.com/?What-Is-the-Difference-Between-Hot,-Warm-and-Cold-Disaster-Recovery?&id=5735955.

[28] *J. Moteff and P. Parfomak. Critical Infrastructure and Key Assets: Definition and Identification. CRS Report for Congress, 2004.*

[29] Air Gap. *Packet Nexus.* [En ligne] http://www.packetnexus.com/kb/greyarts/docs/981158999:10606.html.

[30] EDP Distribuição. GENESys System Security Audit. EDP Distribuição Internal Document, 2008.

[31] F. Cristian, Understanding Fault-Tolerant Distributed Systems. Communications of the ACM, Vol. 34, N. 2, pages 56-78, 1991.

[32] *W. S. Dantas, A. Bessani, M. Correia. Not Quickly, Just in Time: Improving the Timeliness and Reliability of Control Traffic in Utility Networks. HotDep'09: Workshop on Hot Topics in System Dependability. Lisbon, Portugal. June 2009.*

[33] *S.Misbahuddin. Fault Tolerant Remote Terminal Units (RTUs) in SCADA Systems. In Proceedings of the International Symposium on the Collaborative Technologies and Systems (CTS), 2010.*

[34] *M. Chartrand. Dual Redundant Controller Systems. Control Microsystems White Paper, 2001.*

[35] *L. Ekeroth, and P.-M. Hedstrom. GPRS support nodes. Ericsson, 2000.*

[36] *C. Coleman. An Introduction to Radio Frequency Engineering. Cambridge, pages 293-307, 2011.*

[37] Wikipedia. Radio Repeater. Wikipedia, 2011. [En ligne] http://en.wikipedia.org/wiki/Radio_repeater.

[38] *H. G. Perros. Connection-Oriented Networks: SONET/SDH, ATM, MPLS and Optical Networks. Wiley, 2005.*

[39] B. Charron-Bost, A. Schiper. The Heard-Of model: computing in distributed systems with benign faults. Distributed Computing 22(1): 49-71, 2009.

[40] *L. Lamport, R. Shostak, and M. Pease. The Byzantine Generals Problem. ACM Transactions on Programming Languages and Systems, 4(3), 1982.*

[41] *A. Lysyanskaya and C. Peikert. Adaptive security in the threshold setting: From cryptosystems to signature schemes. ASIACRYPT, 2001.*

[42] *EDP Distribuição. 2010 GENESys Key Performance Indicators Report. EDP Distribuição Internal Document, 2010.*

[43] Rose Tsang. Cyberthreats, Vulnerabilities and Attacks on SCADA Networks, 2009.

[44] *M. Garcia, A. Bessani, I. Gashi, N. Neves, R. Obelheiro. OS Diversity for Intrusion Tolerance: Myth or Reality? DSN'11: International Conference on Dependable Systems and Networks. Hong Kong, China, June 2011.*

[45] *Ilir Gashi, Peter Popov, Lorenzo Strigini. Fault tolerance via diversity for off-the-shelf products: a study with SQL database servers. IEEE Transactions on Dependable and Secure Computing, IEEE Computer Society Press, 4(4), pages. 280-294, 2007.*

[46] *S. A. Boyer. SCADA: Supervisory Control And Data Acquisition. ISA, pages 9-14, 2009.*

Appendix A

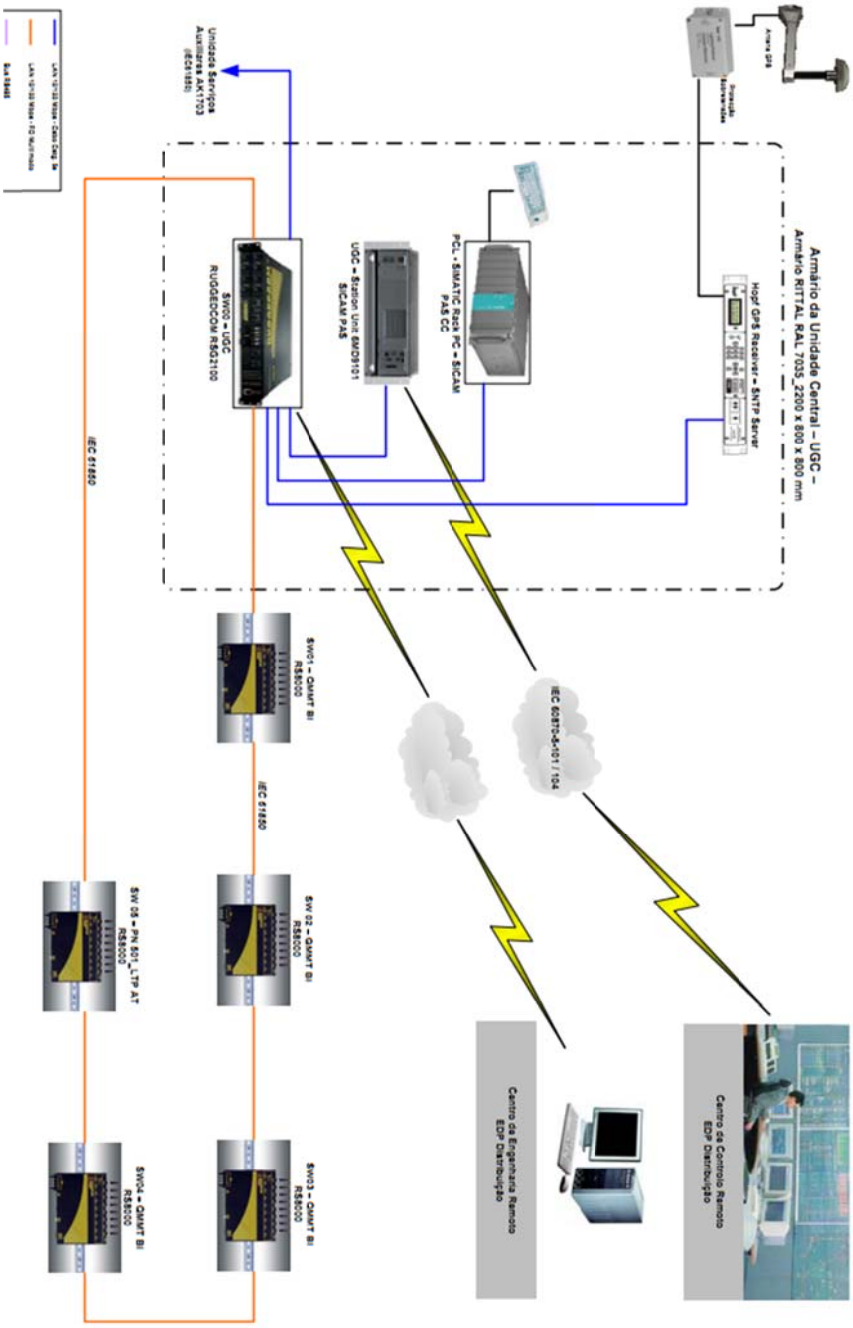


Figure 1 – Conventional architecture of HV/MV substation.

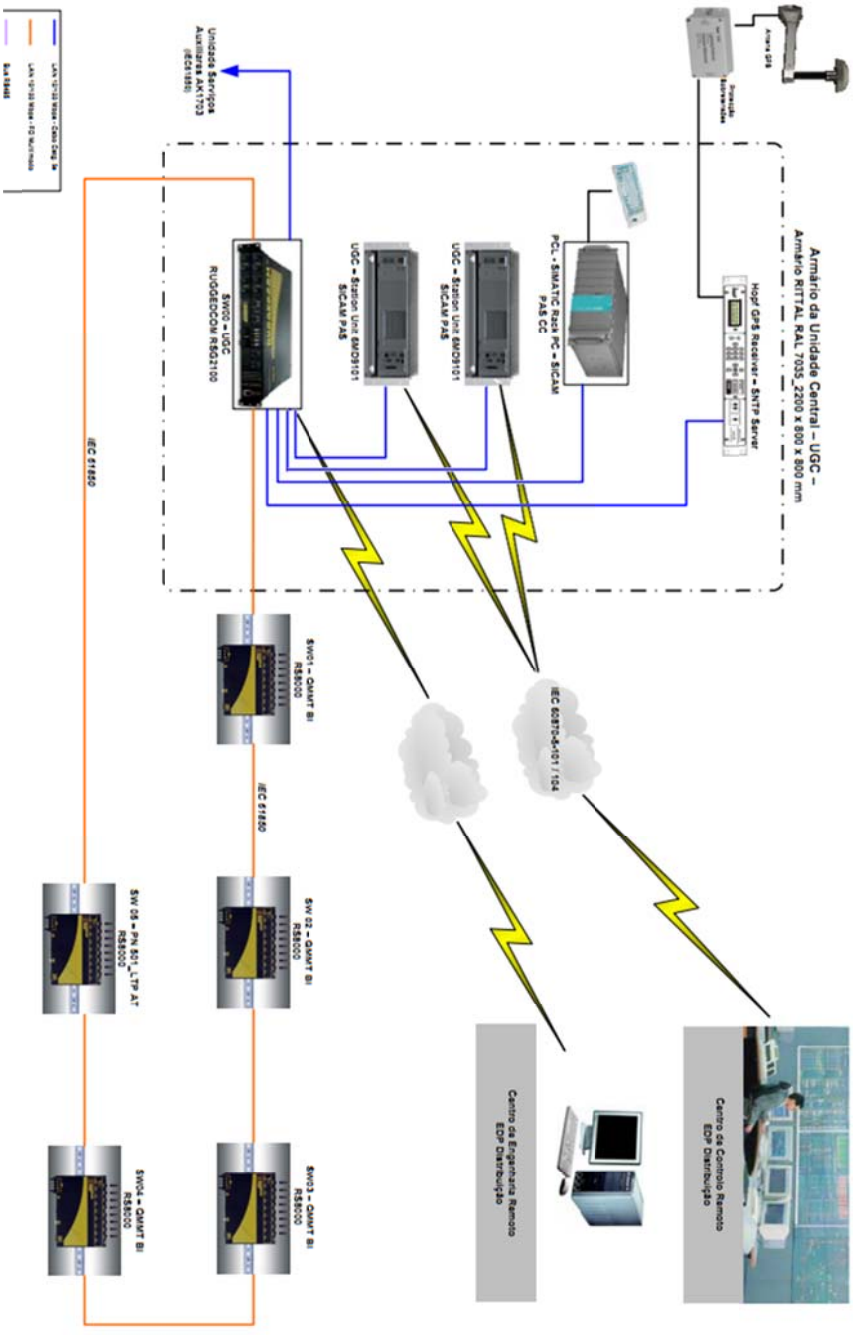


Figure 42 – RRTU architecture of HV/MV substation.