Pós-Graduação em Ciência da Computação

DIOGO ABREU DE SIQUEIRA

Perfomance and Dependability Evaluation in a Convergent Network Service using BGP and BFD Protocols

Federal University of Pernambuco
posgraduacao@cin.ufpe.br
http://cin.ufpe.br/~posgraduacao

Recife
2020

DIOGO ABREU DE SIQUEIRA

Perfomance and Dependability Evaluation in a Convergent Network Service using BGP and BFD Protocols

A Master Dissertation presented to the Informatics Center of Federal University of Pernambuco in partial fulfillment of the requirements for the degree of Master in Computer Science.

Concentration Area: Performance and Dependability Evaluation
Advisor: Paulo Romero Martins Maciel

Recife
2020

Dissertação de mestrado apresentada por **DIOGO ABREU DE SIQUEIRA** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título "**Perfomance and Dependability Evaluation in a Convergent Network Service using BGP and BFD Protocols**", orientada pelo **Prof. Dr. Paulo Romero Martins Maciel** e aprovada pela banca examinadora formada pelos professores:

Prof. Dr. Jamilson Ramalho Dantas (Examinador Externo)
Departamento de Computação / UNIVASF

Prof.Dra. Edilayne Meneses Salgueiro (Examinador Externo)
Departamento de Computação / UFS

Prof. **Orientador:** Dr. Paulo Romero Martins Maciel
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 28 de fevereiro de 2020.

Prof. Ricardo Bastos Cavalcante Prudêncio
Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

I would like to dedicate this dissertation to God and my family who helped me to overcome great obstacles during these last years.

AGRADECIMENTOS

I am immensely grateful to Professor Paulo Maciel for accepting me in his research group. He gave me all the support I needed to achieve my goals. Paulo is an example of a professional, an inspiration as a scientist, as well as a great human being.

I would like to thank my co-author — Thiago Pinheiro — for support and collaboration. Thiago has a great talent for research and has given me valuable tips. He always was available to review my works and suggest improvements.

I want to thank my colleagues of Modcs group for helping me at some point during the previous two years.

Foremost, I would like to thank my wife - Mariana- and my daughter - Letícia - for supporting me throughout this period. They motivated me to stand firm when things got tough. I love you.

I would like to thank my parents for everything they have done for me to date. Thank you for always being by my side.

I am grateful to God for giving me health, strength, and patience to complete this journey. God had given me the strength to overcome the various obstacles that have appeared.

I am immeasurably grateful for everything that happened!

ABSTRACT

Critical network services require maximum availability. The design of a convergent network must be oriented so that downtime is as short as possible.It is important to highlight that the term convergent refers to recovery in case of failure and not to the integration of data and voice in the same channel. For this reason, a network device must be able to quickly detect any communication failure between adjacent devices, so that the upper layer protocol can fix that failure and prevent any interruptions on the provided services. The failure detection time is dependent on issues intrinsic to the protocol used. The association between BFD (Bidirectional Forwarding Detection) and BGP (Border Gateway Protocol) protocols allows BFD to quickly detect failures in the connections between BGP peers, implementing fast convergence of BGP routes. This dissertation proposes two approaches for supporting the estimation of convergent architectures' availability by considering different configurations and the related impact of each one on the performance. We have proposed a Continuous-Time Markov Chain (CTMC) model to represent network architectures using both protocols BGP and BFD. This CTMC represents architectures running critical systems and must be used to predict availability by considering a given configuration. Thus, our second approach is aimed at supporting an inferential performance evaluation by considering a paired before-and-after comparison. This approach is applied in order to obtain the best convergence condition for a corporate network and, as a consequence, to offer greater availability as possible. More specifically, this work proposes an approach to estimate the performance of the concentrating equipment, that supports the availability solution, by applying a model based on an M/M/1/K queue, which was changed in its original definition for supporting the analysis of BFD control packets. Sensitivity analyses were carried out by considering the mean arrival time (MAT) and discard rate of BFD control packets and the concentration equipment's CPU utilization. We also present a sensitivity analysis by considering the network's availability as a result of the fail-over time. In addition, we presented a closed-form equation to calculate the availability of critical warm-standby components for large network architectures. By combining all approaches, it is possible to evaluate the trade-off between availability and performance cost for implementing a solution and find out a configuration that offers the best solution by considering the project's requirements. Maybe, in some circumstances, our strategies need to be refined in order to support different characteristics of the network architecture to be evaluated. Our strategy may be refined in order to support different characteristics of the network architecture. We proposed two methodologies for supporting the applications of our strategies. Three case studies were performed to evaluate the effectiveness of our approaches. Our approaches have proven to be feasible, and they highlight the most appropriate scenarios, supporting network architects.

Keywords: Availability. Performance. Convergent Network. CTMC.

RESUMO

Serviços de rede críticos exigem disponibilidade máxima. O projeto de uma rede convergente deve ser orientado para que o tempo de downtime seja o menor possível. Importante destacar que o termo convergente refere-se à recuperação em caso de falha e não à integração de dados e voz em um mesmo canal. Por esse motivo, um dispositivo de rede deve ser capaz de detectar rapidamente falhas de comunicação entre dispositivos adjacentes, para que o protocolo da camada superior possa corrigi-la, impedindo interrupções nos serviços fornecidos. O tempo de detecção de falhas depende de questões intrínsecas ao protocolo usado e da configuração aplicada à rede. A associação entre os protocolos BFD (Bidirectional Forwarding Detection) e BGP (Border Gateway Protocol) permite que o protocolo BFD detecte rapidamente falhas nas conexões entre os peers BGP, implementando uma convergência rápida de rotas BGP. Esta dissertação propõe duas abordagens para apoiar a estimativa de disponibilidade de arquiteturas convergentes, considerando diferentes configurações e o impacto relacionado de cada uma no desempenho. Propusemos uma Continuous-Time Markov Chain (CTMC) para representar arquiteturas de rede usando os protocolos BGP e BFD. Esta CTMC representa arquiteturas executando sistemas críticos e deve ser usada para prever a disponibilidade considerando uma determinada configuração. Assim, nossa segunda abordagem visa apoiar uma avaliação inferencial de desempenho considerando uma comparação before-and-after emparelhada. Essa abordagem é aplicada para obter a melhor condição de convergência para uma rede corporativa e, como conseqüência, oferecer a maior disponibilidade possível. Mais especificamente, este trabalho propõe uma abordagem para estimar o desempenho do equipamento de concentração, que suporta a solução de disponibilidade, aplicando um modelo baseado em uma fila M/M/1/K, que foi alterada em sua definição original para apoiar a análise de pacotes de controle BFD. As análises de sensibilidade foram realizadas considerando o mean arrival time (MAT) e a taxa de descarte dos pacotes de controle BFD e a utilização da CPU do equipamento de concentração. Também apresentamos uma análise de sensibilidade considerando a disponibilidade da rede como resultado do tempo de failover. Além disso, apresentamos uma equação para calcular a disponibilidade de componentes críticos warm-standby para grandes arquiteturas de rede. Ao combinar todas as abordagens, é possível avaliar o trade-off entre disponibilidade e custo de desempenho para implementar uma solução e descobrir uma configuração que ofereça a melhor solução, considerando os requisitos do projeto. Nossa estratégia pode ser refinada para suportar diferentes características da arquitetura de rede a ser evaliada. Propusemos duas metodologias para apoiar as aplicações de nossas estratégias. Três estudos de caso foram realizados para avaliar a eficácia de nossas abordagens. Nossa solução provou ser viável e ela destaca os cenários mais apropriados, dando suporte aos arquitetos de rede.

Palavras-chaves: Disponibilidade. Desempenho. Redes Convergentes. CMTC.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

A	Availability
AS	Autonomous System
BFD	Bidirectional Forwarding Detection
BGP	Border Gateway Protocol
BOS	Balance of System
BSA	Basic Service Area
BU	Business Unit
CIDR	Classless Inter-Domain Routing
CPE	Customer-Provided Equipment
CPU	Central Processing Unit
CTMC	Continuous Time Markov Chains
DES	Discrete Event Systems
DTMC	Discrete Time Markov Chain
EBGP	External Border Gateway Protocol
EGP	Exterior Gateway Protocol
EIGRP	Enhanced Interior Gateway Routing Protocol
eNSP	Enterprise Network Simulator
FFD	Fast Failure Detection
GNS3	Graphical Network Simulator-3
GRE	Generic Routing Encapsulation
IANA	Internet Assigned Numbers Authority
IBGP	Internal Border Gateway Protocol
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IP	Internet Protocol
IPsec	IP Security Protocol
LPU	Line Processing Unit

MAT	Mean Arrival Time
MIB	Management Information Base
MIN	Multistage Interconnection Networks
MPLS	Multi Protocol Label Switching
MPPS	Million Packets per Second
MPU	Main Processing Unit
MST	Mean Service Time
MTBF	Mean Time Between Failure
MTTF	Mean Time to Failure
MTTR	Mean Time to Repair
NLRI	Network Layer Reachability Information
NMS	Network Management System
NP	Network Processor
NS-2	Network Simulator
OSPF	Open Shortest Path First
PB	Blocking Probability
QN	Queuing Network
QoS	Quality of Service
QR	Queuing Rate
RBD	Reliability Block Diagram
RFC	Request for Comments
RIB	Routing Information Base
RIPv2	Routing Information Protocol
RM	Redundancy Module
SDN	Software Defined Networking
SEN	Shuffle-Exchange Network
SFU	Switch and Fabric Unit
SLA	Service Level Agreement
SMP	System Management Plane
SNMP	Simple Network Management Protocol
SONET	Synchronous Optical Network

SPN	Stochastic Petri Net
SSP	Service Splitting Platform
TCP	Transmission Control Protocol
TP	Throughput
TRFC	TCP Friendly Rate Control
U	Utilization
VPN	Virtual Private Network
VRP	Versatile Routing Platform
WAN	Wide Area Network

SUMÁRIO

1 INTRODUCTION

Nowadays as showed in (DANTAS et al., 2015), some services being provided around the world need to be constantly available for their customers. However, there exists a class of services that does not tolerate to become unavailable for a long time because the consequences of this may be catastrophic. In this context, it may be catastrophic both by considering from a perspective of human lives as well as financial assets. The lost decurrent of any failure in this class of services may break a business and this loss may affect all actors interacting with the services that become unavailable. Being these actors people or entities that consume the services as well as that supply them. This class of services is classified as critical services.

Critical services require maximum availability and convergent networks have become an indispensable mechanism for making possible to offer these services.It should be noted that the convergent network mentioned here refers to the network with the ability to recover quickly from a failure and not the network that integrates data and voice services. Their application requires to apply all possible efforts in order to produce an improvement in the availability of services. For that aim, when any failure occurs in a communication link or in other infrastructure components, the fixing of this issue needs to be efficient and as fast as possible. The design of a convergent network must be oriented so that downtime is as short as possible. Speed in solving any issue is important in this context in order to maintain stable applications and services supported by the underlying infrastructure. Service Level Agreements (SLAs) take an important role regarding how these convergent networks need to be implemented in order to ensure that performance and availability requirements present on them are fulfilled.

Critical systems require that the underlying infrastructures for supporting them work uninterruptedly. Considering this, also a lower interruption on the service been providing may represent a relevant loss for the service provider using those infrastructures. On the other hand, an increase in availability is not always justifiable. There exist situations when even an increase does not justify the implementation of a new network arrangement. Thus, it is necessary to correlate the availability gains for implementing a new infrastructure arrangement with the resources required for making possible this implementation. Given the necessity to reduce interruption as many as possible and the ongoing need to reduce related costs, continuous improvements on the services should also incorporate efficiency aspects.

There are protocols that when associated on a network may help to reduce downtime. A network device must be able to quickly detect any communication failure between adjacent devices so that it becomes possible to prevent failures on their occurrences or at least reduce their negative impact on the services using the underlying infrastructure. The

failure detection time is dependent on issues intrinsic to the protocol being used, and the configuration applied to the network can lead to high downtime. Bidirectional Forwarding Detection (BFD) is a protocol that offers low duration and low overhead for detecting failures in a path between two forwarding mechanisms. Border Gateway Protocol (BGP) is an External Gateway Protocol (EGP) type protocol used for inter-AS communication, which is for communication between autonomous systems (ASs). By associating both protocols it allows BFD to quickly detect failures in the connections between BGP peers, implementing fast convergence of BGP routes. As the number of communication links grows, the greater may be the traffic of BFD control packets and CPU consumption on concentrator devices. This situation necessarily goes to a threshold, considering that resources are always finite. This threshold, which represents a bottleneck, must be avoided through adequate planning that can be carried out following our approaches, whose implementation includes a case study of this dissertation. The approaches developed by us consider the impacts of the association of the aforementioned protocols on the network's availability and on the performance of the network concentration equipment.

A corporate network backbone provides an elastic capability for supporting varying from both end-user demand patterns and due to control mechanisms such as monitoring through the BFD protocol. In this way, it is possible to design a network in order to comply with the performance requirements defined in SLAs. On the other hand, it is a difficult task to identify a configuration to deploy on the network in order to optimally meet both performance and availability requirements.

1.1 MOTIVATION

Companies, governments, service providers among other actors around the world are increasingly dependent on convergent network services. So that, they can offer their services without relevant interruptions on them. The design of a convergent network that needs to be always available and stable is essential in order to support any large businesses. In this context, aspects regarding failure handling are essential to ensure that performance requirements are properly reached.

Authors in (KIM; KIM, 2015) demonstrated that the need for real-time services is always increasing the necessity for improving the overall quality of the data communication networking services. Considering this context, issues regarding availability, reliability, and scalability of routers — that control data flow — and communication data links are well-known critical problems. Many methods have been studied aimed to improve reliability and availability in order to minimize the loss cost caused by any failure occurrence on a communication network routers.

The annual costs regarding paralyzation caused by failures occurrences in critical services are in order of billions of dollars. It occurs due to an increase in dependency of these systems by companies, in particular to convergent networks (JONES; RANDELL, 2004). Ac-

cording to the Gartner's report (GARTNER RESEARCH, 2014), based on industry surveys, the cost regarding downtime of networks is estimated at $5,600 per minute, which extrapolates to well over $300K per hour. Taking into account the aforementioned information, it is evident that it is worth investing time and resources in an attempt to promote gains in availability in order to minimize the downtime as maximum as possible. In general, a critical system has at least a redundant communication link for each point where the service is offered. However, depending on the applications' criticality and demand level supported by the network, a simple redundancy mechanism may not be enough in order to guarantee that there will be no losses due to any failure occurrence.

Authors in (SERMPEZIS; DIMITROPOULOS, 2017) present that routing between autonomous domains/systems is performed in a distributed manner through BGP and, despite its global adoption, BGP has several shortcomings, such as slow convergence after routing changes, which can cause packet loss and interrupt communication even for several minutes. The authors argue that while BGP is known to suffer from slow convergence, the deployment of other protocols or modified versions of BGP is difficult due to its widespread use and the political, technical, and economic challenges involved. Therefore, any advances and proposed solutions are excellent for using BGP. The authors (SERMPEZIS; DIMITROPOULOS, 2017) inspired us to propose an approach to overcome the intrinsic issues of the BGP protocol.

The availability metric corresponds to a relation regarding the mean time between failure (MTBF) and mean time to repair (MTTR). An increase in availability needs to come from an increase in MTBF or a decrease in MTTR or both. Taking into account the characteristics of these two metrics, the most sensitive variable to be considered by a network designer is MTTR, as there exists no configuration that may prevent failures by default. However, an optimized arrangement on the infrastructure can reduce the time spent to detect any failure occurrence, and consequently, reducing MTTR.

By considering that the BFD is aimed to establish a failure detection mechanism on network infrastructure, it is necessary to define a research line by associating the BFD and BGP protocols. Expecting that it will be possible to generate evidence that this new network arrangement would be able to produce better results regarding the availability metric.

Accordingly to preliminary research on the BFD protocol related to its backup mechanism, it is clear that depending on the size of the network, it is possible that a bottleneck may arise in the network's concentrating equipment. This motivated us to associate the availability and the impact on performance for reaching this availability.

1.2 SUMMARY OF THE MAIN RELATED WORKS

The convergence of networks as well as the association between BGP and BFD protocols have been studied in some works. However, there are still open questions regarding them

that motivate further researches.

Authors in (KIM; KIM, 2015) demonstrate the requirements necessary to establish a standby router. Besides that, they demonstrated how information from a Border Gateway Protocol (BGP) session established in an active router is controlled by a standby router. The redundancy model proposed by them operates in active and standby routers in a controlled and synchronized way, in which a TCP connection between the active-standby routers is established.

In turn, authors in (SERMPEZIS; DIMITROPOULOS, 2017) demonstrated that the existing methods of BGP reconnection have problems, such as an increase in the takeover time. Being this increase due to the time necessary to detect a session disconnection as well as to exchange connection information for reconnection. The authors presented an interesting comparative analysis regarding the performance of the BGP protocol, similar to what is was performed by us in this work.

Authors in (MASRUROH et al., 2017) argue that even though it is known that BGP presents slow convergence, the implementation of other protocols or modified versions of BGP protocol is difficult, due to generalized use of the BGP in its original version and the technical and economic challenges implied. Therefore, any advances and solutions proposed are considered favorable for using the BGP protocol in its original version. In this regard, the work analytically studies the effects of centralization on the performance of the routes between domains. Focusing on potential improvements for (slow) BGP convergence.

Authors in (TANYINGYONG et al., 2013) have tried to solve the problem of convergence time of the routing protocols. The authors have used the BFD primary mode (asynchronous mode) for all communications. In this mode, a BFD process (speaker process) negotiates with another BFD neighbor the interval value for *hello*. The purpose of this experiment was to measure and examine the mean failover time in some scenarios. The authors used an experimental configuration by creating two alternative ways. The same behavior was considered by the two alternatives. The experiment was performed by using ICMP packets from a gateway to another by simulating the interrupting of the link. The work presents an interesting methodology that is similar to the one presented in our work.

Some studies have proposed strategies based on M/M/1/K models for supporting evaluations on performance metrics such as buffer size, packets arriving, processing rates, and waiting time of the packets that have arrived, as noted in the work (MONDAL; MISRA; MAITY, 2018). No work has addressed a strategy similar to the M/M/1/K modeling adopted in our work. Aimed to specifically enabling the processing of BFD packets before the queuing process starts.

Although some studies focused on the use of the BFD protocol by associating it with software-defined networks (SDNs), in the recent literature few studies have addressed the association between BFD with the BGP protocol. It is important to highlight that, regarding the financial cost and high complexity to implement a real infrastructure considering

the aforementioned protocols, even more rarely studies have been produced carrying out testbeds in real environments like ours. Another issue that is not present in the literature is about studies regarding performance on BFD protocol when it is used in large scale infrastructures.

It is important to note that tradeoff evaluations between performance and availability on convergent network infrastructures have been ignored by these studies. Our work is the first one, to the best of our knowledge, that provides an integrated modeling approach by considering availability evaluations, based on CTMC modeling, and performance evaluations, based on an M/M/1/K queue model, taking into account workloads generated by BFD packets in a configuration associated with BGP protocol.

1.3 PROBLEM STATEMENT

By considering what was aforementioned, it is clear that there still exists a need to create mechanisms to properly identify how much availability is gained by associating BFD and BGP protocols when applying a convergent configuration on a critical network infrastructure, and, at the same time, to evaluate the processing cost for applying this configuration on the central network devices.

As we have addressed before, the high financial cost for implementing a real infrastructure for performing studies on it is an avoiding factor for performing evaluations on actual infrastructures. On the other hand, the existing simulation environments, although they serve to validate the configuration before the deployment, are not suitable for carrying out the tests. It is an established knowledge that the use of stochastic modeling is useful for supporting evaluations when there exists avoiding factors to perform evaluations on actual entities as is the case of the object considered in our study. In addition, by considering some assumptions, when applying stochastic models, it is possible to perform a myriad of evaluations by using only a model that had been validated before.

In this context, a question was raised and it has guided our work:

- How to evaluate the availability and performance of convergent network configurations that use BGP and BFD protocols by applying stochastic modeling?

1.4 OBJECTIVES

The main objective of this research is to propose approaches for supporting availability and performance evaluations by considering convergent network configurations that use BGP and BFD protocols applied on high-critical convergent networks.

Among the specific goals of this research, we can list:

- Develop a testbed that can validate the availability gains resulting from the association of the BGP and BFD protocols on a large scale.
- Develop a modeling strategy based on the CTMC formalism for evaluating the availability of convergent networks by considering the failover time.
- Develop a modeling strategy based on a Markovian M/M/1/K queue model for evaluating the performance of concentrator equipment by considering workloads generated by BFD control packets.
- Develop a model that considers the number of communication links available in large scale infrastructures for supporting the correlation evaluation between the gain in availability and the processing cost for applying a given configuration on a high-critical convergent network.
- Evaluate the proposed approaches, carrying out case studies based on experiments in a real corporate network.

1.5 PURPOSE OF RESEARCH

More specifically, we have proposed two approaches. They are aimed for supporting estimations about convergent architectures' availability by applying different configurations and the related impact for implementing each configuration on the performance of the failure avoidance mechanism. Our approaches are aimed to support the decision-making process about which configuration regarding the aforementioned protocols must be implemented on a real corporate network in order to reach the desired availability level. In addition, they enable us to carry out a comparative and objective analysis between different solutions available in order to let to evaluate the trade-off between availability and performance for each of them. Our approaches are supported by models that make it possible to predict values for a large number of scenarios.

This work proposes two models. The first model is a continuous-time Markov chains (CTMC) (BOLCH et al., 2006; TRIVEDI, 2008) to represent and calculate the availability of the network in question. By this model, it was possible to derive a closed-form equation for calculating availability by considering the number of communication links of large network infrastructures. The second model is a derivation of a Markovian queue M/M/1/K model. It represents the workload related to the BFD control packets that are processed by the network's concentrating equipment. The CTMC complemented by the derived M/M/1/K model allows for to correlate the gain in availability with the performance cost for meeting it. We have considered five performance metrics for supporting the decision-making process.

The performance metrics considered are as follows: *Mean Arrival Time* (MAT), *Mean Service Time* (MST), *Utilization* (U), *Queuing Rate* (QR), and *Throughput* (TP). *MAT*

corresponds to the mean arrival time of BFD control packets. *MST* is the amount of time that a BFD control packet requires to be handled by the concentrator equipment. *U* indicates the percentage of CPU usage on a concentrator device. Maybe, BFD packets can not be processed immediately after they arrive in a concentrator device. Thus, *QR* is the rate at which BFD packets are queued to be processed after when processing capacity is made available to handle them. Besides that metrics, we have evaluated throughput by considering two perspectives. From the first perspective, the *throughput* corresponds to the number of requests per unit of time made by each BFD control packet. Form the second one, it corresponds to the number of requests processed by a central device per unit of time. The performance evaluation is a key process that lets to perform the trade-off between the gain in availability and its impact on performance.

Also, we propose two methodologies for supporting the application of our approaches. The first methodology is aimed at supporting availability evaluations. The second one is aimed at supporting performance evaluation on the network concentrating equipment. The methodologies may be applied separately, that is, it is not a prerequisite that both must be applied for a given situation.

In order to validate our approaches, we have implemented a testbed environment for experimentation and collecting empirical data on a real corporate network. Changes at the BGP routing protocol level were applied to the testbed's configuration. Three case studies were carried out. The empirical data collected during the experiments were then analyzed, serving as entry parameters for our CTMC. Thus, making it possible to confront the results obtained on both sides and validate our approaches. A case study was performed in order to propose a mechanism that provides fast failover, without generating any false alarms and unwanted rerouting decisions. The mechanism was based on the allocation of the system's resources for processing BFD control messages. We demonstrated by analyzing the results a reduction in the average failover when using the BFD protocol. We have performed a sensitivity analysis of failure time in the evaluated scenarios. Our approaches demonstrated to be effective and by their application, it is possible to find out the most appropriate scenario to be implemented on an actual convergent networking infrastructure.

Some assumptions have to be taken into consideration when applying what is proposed here. Our scope is restricted to the data communication links, not including the availability of the network equipment or its internal components. It occurs because our research is focused on the reduction of the failover time by accelerating the switching process.

Our strategies can be applied in environments with different infrastructures regarding the one considered here. Different backbones are expected to have equipment with different processing and forwarding capacities. Even in a scenario where the implementation of the solution discussed in this work is not viable, our strategies can be used to guide acquisitions or hardware upgrades in a precise way. Our approaches facilitate the process of

choosing an ideal configuration as they make possible to predict in terms of availability and performance the impact for applying a given configuration on the convergent networking environment. Some results regarding this research have been already published (SIQUEIRA et al., 2019).

1.6 ORGANIZATION OF THE DOCUMENT

The remainder of the dissertation is organized as follows. Chapter 2 provides an overview of the main concepts about the theoretical foundations that have supported us during the development of the approaches presented here. More specifically, in Chapter 2we addressed concepts regarding BGP and BFD protocols, dependability and redundancy in high availability networks, continuous-time Markov chain, and, finally, queueing theory. Chapter 3 discusses and compares noteworthy works found in the literature that have some topics in common to those addressed in this dissertation. Chapter 4 presents detailed information about the methodologies proposed by us for supporting availability and performance evaluations in convergent networks that uses the BGP and BFD protocols. Chapter 5 describes the network architecture considered in this work. As described in our modeling strategy based on CTMC and M/M/1/K queue, which enable availability and performance evaluation. Chapter 6 details case studies considering the proposed approaches.

2 BACKGROUND

This chapter discusses the basic concepts needed to understand our work. More specifically, the concepts presented here should provide the necessary knowledge for a clear understanding of the chapters below, including aspects involving the proposed methodologies and subsequent case studies. The remainder of the chapter is organized as follows. Section 2.1 provides detailed information about the Border Gateway Protocol(BGP). Section 2.2 exposes essential fundamentals of the Bidirectional Forwarding Detection Protocol(BFD). Section 2.3 highlights the main concepts about dependability; Section 2.4 shows the main concepts about operational analysis; Section 2.5 presents concepts of queuing theory especially the Markovian M/M/1/K model; Section 2.6 discusses the basic concepts about continuous-time Markov chains (CTMC);

2.1 BGP PROTOCOL

An autonomous system corresponds to an organizational set of computers capable of operating in isolation from all other sets. The autonomous system (AS) is a set of networks that are under common management and identified by a number. This number is assigned by the IANA (Internet Assigned Numbers Authority) and is made up of 16 bis. The number of AS can range from 1 to 65535. The union of autonomous systems forms the internet. Within autonomous systems, routes are assigned either statically or dynamically. Dynamic routing depends on the use of internal and external protocols. Within the same AS the routing tables are distributed in such a way that a router member of the AS is able to build the path to be followed to another router within the same AS. This indicates that the routing protocols within an AS and outside it operate differently. The so-called IGPs (Interior Gateway Protocol) are used to exchange routing information within the same AS. On the other hand, routers with the role of communicating with another AS make use of a protocol of the type EGP (Exterior Gateway Protocol) (DOYLE; CARROLL, 2006).

IGPs and EGPs serve different purposes. Within an AS the main objective is to calculate the best route or to quickly update information about the status of the network, in case of failure in communication links or in another router. In turn, a router that is responsible for handling the EGP protocol has as its main task administrative, political, economic and security issues, which are manually configured and, therefore, do not make up the protocol directly. Due to these differences, the IGP and the EGP usually use different technologies (MENDONÇA; OLIVEIRA; LINS, 2012).

Autonomous systems use the Path Vector routing protocol to exchange routing information. The Path Vector routing protocol allows the route selection to follow a routing

policy that takes into account not only the distance or cost associated with the route but based on the state of routers and links that make up the path. Considering the need to connect different networks and autonomous systems, the internet makes use of EGPs protocols. One of the characteristics of the EGPs protocols is the route summary property of the Path Vector routing protocols which makes it possible for autonomous systems to be characterized within advertised routes, ensuring scalability and flexibility. For this type of routing, between autonomous systems, the BGP (Border Gateway Protocol) protocol, which was defined in RFC 4271, is used (MENDONÇA; OLIVEIRA; LINS, 2012).

According to RFC 4271 of the Internet Engineering Task Force (IETF) the Border Gateway Protocol (BGP) is "an inter-Autonomous System routing protocol. The primary function of a BGP speaking system is to exchange network reachability information with other BGP systems. This network reachability information includes information on the list of Autonomous Systems (ASes) that reachability information traverses. This information is sufficient for constructing a graph of AS connectivity for this reachability, from which routing loops may be pruned and, at the AS level, some policy decisions may be enforced" (REKHTER, 2006). BGP speaker is a router that implements BGP.

As presented in (Caesar; Rexford, 2005) Non-ISP businesses (enterprises) may also operate their own ASes to gain the additional routing flexibility that arises from participating in BGP.

In order to understand BGP it is essential to comprehend this decision process and the policies of ISPs that gave rise to its design.Understanding policies is also essential to solving BGP's problems, understanding measurement data from BGP, or determining which features to support when developing a new version of BGP.

The authors in (Caesar; Rexford, 2005) has listed policies into four general categories: business relationship policy arising from economic or political relationships an ISP has with its neighbor, traffic engineering policy arising from the need to control traffic flow within an ISP and across peering links to avoid congestion and provide good service quality, policies for scalability to reduce control traffic and avoid overloading routers, and security-related policies that are often used to protect an ISP against malicious or accidental attacks. An ISP implements its policies by modifying route attributes and changing the way routers react to advertisements with certain route atributes.An ISP implements its policies by applying configuration commands at routers. These configurations typically consist of a set of lists of preference, filtering, and tagging rules, one list for each session the router has with a neighboring BGP-speaking router.

The BGP-4 implements functionality to support Classless Inter-Domain Routing (CIDR) [RFC1518, RFC1519]. This makes possible to advertise a set of destinations as a IP prefix. In addition, the BGP-4 enables the route and AS path aggregation mechanism. In order to ensure reliability and reachability, BGP makes use of the TCP protocol (port 179). The Routing Information Base (RIB) on a BGP speaker has three different elements:

Adj-RIBs-In which contains unprocessed routing information that BGP peers have advertised to the BGP speaker. The Loc-RIB contains the routes that were chosen by the decision process of the local BGP speaker; and the Adj-RIBs-Out that manages the routes for advertising to specific peers (through Update messages) (REKHTER, 2006).

The BGP protocol does not require a periodic update of the routing table. The initial flow is due to the export policy (Adj-Ribs-Out). The routing table stores routes learned from BGP, routes learned IGP protocols, static routes and directly connected networks. As the routing tables change, incremental updates are sent. In order for policy changes to have the expected effect without the need for disconnection between BGP peers, the BGP speaker must maintain the current version of the routes announced to him by its peers or, alternatively, use the Route Refresh Extension process described in RFC 2918. To ensure that the connection is active, keepalive messages are sent periodically. If errors or special conditions occur, notification messages are sent in response and the connection is closed (REKHTER, 2006).

A BGP peer within the same AS is known as IBGP (Internal BGP). On the other hand, a peer found in another AS is called EBGP (External BGP). A high number of BGP speakers in an AS is transiting service with another AS, additional care must be taken in order to maintain the consistency of routing within the AS. The consistency of the AS's internal routing is provided by the IGP (REKHTER, 2006).

BGP deals with the route as the combination of information that unites destination data with path attributes to reach these destinations. Regarding destinations, BGP makes use of IP address prefixes in the NLRI(Network Layer Reachability Information) field of the standard update message. On the other hand, the attributes of the path are constant in another field of the same update message. The NRLI field of the update message can contain several prefixes. The routes that have the same path attribute can be announced in the same update message. It is through the update message that the BGP speakers announce their routes among themselves. It is possible for a BGP speaker to edit the path attributes of a received route, either by adding or modifying it before announcing to another BGP speaker. Another important feature is that BGP offers the mechanism for a BGP speaker to announce to its peers that a route previously advertised as active is now unavailable (REKHTER, 2006).

The maximum size of the BGP message is 4096 octets and the BGP header is 19 octets. The main BGP message codes are:

- OPEN
- UPDATE
- NOTIFICATION
- KEEPALIVE

The first message sent by each BGP peer is always an OPEN message whose format is shown in Figure 1. In addition to the fields shown there is also the BGP header with a defined size. This message is sent after the TCP connection is established between the peers. Once the OPEN message is accepted, a message KEEPALIVE that ratifies the receivement of the OPEN message is sent back (REKHTER, 2006).

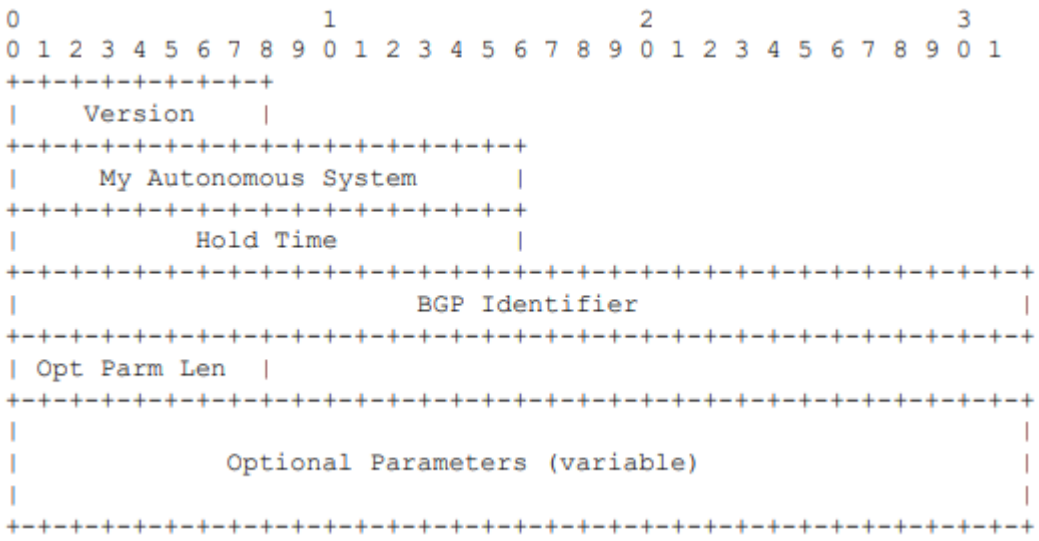

Figura 1 – BGP Open Message Format (REKHTER, 2006)

HOLD TIME is sent via the OPEN message by the BGP speaker sender to another BGP speaker as a proposal. Upon receiving the message, the receiving BGP calculates and selects the smallest of the values between that received through the BGP OPEN message and the value that was already configured. Considering that the field occupies two octets, its value can, in theory, vary from 0 to 65535. However, the lowest valid for HOLD TIME is 3 seconds. This hold time will define the time that the receiving BGP speaker will use as a limit for keepalives and updates with the sending peer (REKHTER, 2006).

Presently the Autonomous System number is encoded as a two-octet entity in BGP.To plan for the anticipated exhaustion of the two-octet AS numbers the document (VOHRA; CHEN, 2007) defines "a new BGP capability, Four-octet AS Number Capability, that can be used by a BGP speaker to indicate its support for the four-octet AS numbers. Two new attributes, AS4_PATH and AS4_AGGREGATOR, are introduced that can be used to propagate four-octet based AS path information across BGP speakers that do not support the four-octet AS numbers".

Our dissertation focus on overcoming intrinsic issues of the BGP protocol that is slow in the failure detection process. The source of the problem is in the HOLD TIME field shown in Figure 1. HOLD TIME set to 0 means that there are no keepalives. On the other hand, three seconds can be a very long time for critical applications.

Given the importance that this point has for our dissertation, we present in accordance

with RFC 4271 how the keepalives timing mechanism works and its recommended relationship with HOLD TIME: "BGP does not use any TCP-based, keep-alive mechanism to determine if peers are reachable. Instead, keepalive messages are exchanged between peers often enough not to cause the Hold Timer to expire. A reasonable maximum time between keepalive messages would be one third of the Hold Time interval. Keepalives messages MUST NOT be sent more frequently than one per second. An implementation may adjust the rate at which it sends keepalives messages as a function of the Hold Time interval" (REKHTER, 2006).

As showed in (HARES; LEE; VARLASHKIN, 2016) the convergence of BGP occurs at two levels: Routing Information Base (RIB) and FIB convergence. Convergence events or triggers are concepted as atypical occurrences in the network, which initiate route flapping in the network and hence forces the reconvergence of a steady state network. As presented in (BERKOWITZ et al., 2005) a soft reset does not clear the RIB or FIB tables. A hard reset clears BGP peer sessions, RIB tables, and FIB tables.

2.2 BFD PROTOCOL

Detecting failures in communication links between adjacent points quickly and efficiently is a mandatory necessity, just as it is imperative to establish an alternative route. When detection occurs at the hardware level, it presents high speed, which is what occurs with Synchronous Optical Network (SONET) alarms, for example. However, there are several types of media that are not able to identify certain types of failures, both in the data link and in a device (KATZ; WARD, 2010). In cases where there is no signaling from the hardware of a failure, the network protocols make use of the hello mechanism. This strategy, however, has slow convergence, making the detection times greater than 1 second among the existing routing protocols. This is an excessively long time for critical applications and can lead to significant data loss if gigabit transfer rates are considered.

According to RFC 5880 (KATZ; WARD, 2010) BFD is "a simple Hello protocol that, in many respects, is similar to the detection components of well-known routing protocols. A pair of systems transmit BFD packets periodically over each path between the two systems, and if a system stops receiving BFD packets for long enough, some component in that particular bidirectional path to the neighboring system is assumed to have failed".

The main objective BFD protocol is to provide detection of failures with low duration in a path between adjacent forwarding mechanisms and with low overhead. The detection method includes failures in the interface, in the communication link or even in the device itself. A secondary objective of the BFD protocol is to offer a mechanism capable of detecting a failure in any type of media, in any layer of the protocol and covering a wide variety of overhead and detection times. All features integrated into a single method.

The BFD protocol always operates in a point-to-point (unicast) topology. The BFD protocol can be encapsulated and transported as a payload in any network and link

protocol. This encapsulation interferes with the context of the operation of the BFD protocol. Considering the communication between two routing systems, the BFD operates on top of any data protocol (link layer, network layer, tunnels). Both to establish a BFD session and to disconnect it for some reason, the BFD protocol implements a three-way handshake. It is through this mechanism that the two systems are updated by the change. A path is only considered operational when bidirectional communication has been established between the forwarding mechanism. Adjacent systems can assess their ability to send and receive BFD packets and negotiate with each other the speed applied for failure detection. The initiation of a BFD session occurs with slow transmission of the BFD control packets (KATZ; WARD, 2010). If the session goes Down, the transmission of Control packets goes back to the slow rate. Once a session has been declared Down, it cannot come back up until the remote end first signals that it is down (by leaving the Up state), thus implementing a three-way handshake.

There are two modes of operation for the BFD protocol that can be adopted separately or even in an associated way. The primary mode is called asynchronous. In this mode, the systems periodically send BFD Control packets to one another, and if a specified number of packets in a sequence are not received by the other forwarding mechanism, the session is assumed to be inactive. The second mode is known as demand mode. In this method, each forwarding system adopts a BFD control packet forwarding strategy. In this mode, there is independence between the participants to define how connectivity is checked, and there may be an on-demand check (KATZ; WARD, 2010).

The Echo function acts as an option and is complementary to asynchronous and demand modes. Through the echo function, a loop mechanism is performed under a flow of packets sent from one forwarding system to another in order to return it to the origin. If a specific number of packages does not return to the origin, the session is assumed to be down. When the Echo function is enabled, the packet transmission rate is usually kept low. If the Echo function is not active, the transmission rate of control packets may be increased to a level necessary to achieve the Detection Time requirements for the session. If the Echo function is disabled, the transmission rate of the Control packets is raised to a level sufficient to meet the Detection Time requirements for the session (KATZ; WARD, 2010).

The BFD control packet has a standard non-changeable part that has the format that is shown by Figure 2.

Among the fields in the control packet format, the Diag field stands out, responsible for storing the cause of the last change in the session (fail), which can be one of the causes below:

- 0 −− No diagnostic
- 1 −− Control Detection Time Expired

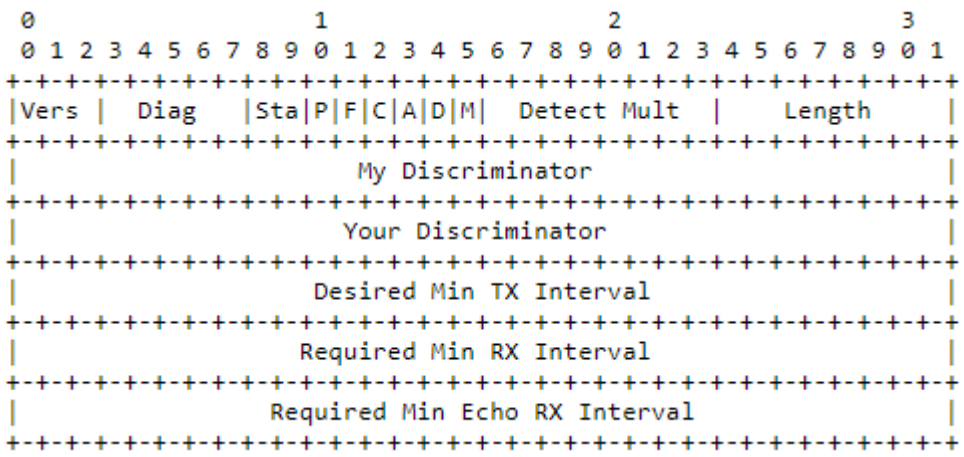

Figura 2 – BFD Control Packet Format (KATZ; WARD, 2010)

- 2 –– Echo Function Failed
- 3 –– Neighbor Signaled Session Down
- 4 –– Forwarding Plane Reset
- 5 –– Path Down
- 6 –– Concatenated Path Down
- 7 –– Administratively Down
- 8 –– Reverse Concatenated Path Down
- 9–31 –– Reserved for future use

Another relevant field is the Sta field, whose values can be:

- 0 –– AdminDown
- 1 –– Down
- 2 –– Init
- 3 –– Up

The values used as a reference for detection time and the transmission interval of BFD packets are negotiable. They are editable at any time for each direction. The proposed detection time and transmission interval are informed through the control packets. This feature allows each system to unilaterally define the shortest interval for receiving packets (maximum rate) in both directions. The BFD protocol is indicated for implementation in critical infrastructure. Mechanisms that may eventually block BFD control packets such as a firewall, priority queuing, traffic shaper or policy process can cause the BFD operation to

fail. Considering that the delivery of BFD packages is very time-sensitive due to the order of magnitude of the detection time. When implementing the BFD protocol, this must be taken into account. According to RFC 5880, the deployment of BFD on many hops implies the need to implement a congestion control mechanism for the traffic generated. This includes the computational cost that multiple BFD sessions with a short interval for sending control packets can represent. A control algorithm to avoid bottleneck situations must be implemented.Host-to-host or application-to-application deployment across the Internet will require the encapsulation of BFD within a transport that provides "TCP-friendly"TRFC behavior (FLOYD et al., 2008). The failure detection time increases with the adoption of a control mechanism that increases the transmission intervals of BFD control packets (KATZ; WARD, 2010).

2.3 DEPENDABILITY MEASURES AND REDUNDANCY IN HIGH AVAILABILITY NETWORKS

Dependability comprehends measures such as reliability, availability, and safety. Systems dependability can be explained as the capacity to deliver a specified functionality that can be reasonably trusted (AVIZIENIS et al., 2004). An interesting concept of dependability is presented in (AVIZIENIS et al., 2004): "the ability of a system to avoid failures that are more frequent or more severe, and outage durations that are longer than is acceptable to the user".

A failure in a large system can represent enormous losses. Critical systems require methods to detect, correct, prevent and tolerate failures. Availability is a highly relevant measure in the context of critical systems. A fundamental concept of the availability metric can be expressed as the probability that the system be found operational during a certain period of time or have been restored after a failure has occurred (O'CONNOR; KLEYNER, 2012a).

Availability can be presented as a percentage or as a real number between 0 and 1. Steady-state availability does not consider a defined time interval. It is obtained as a system average that does not take the time interval into account. It is a metric that focuses on long-term probability. The steady-state availability may be calculated from the mean time between failures (MTBF) and mean time to repair (MTTR) of the system.Consequently, with a constant failure rate $\lambda = 1/MTBF$ and a constant mean repair rate $\mu = 1/MTTR$, (O'CONNOR; KLEYNER, 2012b), the steady-state availability is equal to:

$$A = \frac{MTBF}{MTBF + MTTR} = \frac{\frac{1}{\mu}}{\frac{1}{\lambda} + \frac{1}{\mu}} = \frac{\mu}{\lambda + \mu} \tag{2.1}$$

Downtime represents the total time of unavailability in a given period of time. This is a typical reliability metric. In practical terms, downtime during 1 year (per year in minutes)

for a system is calculated as defined in Equation 2.2 (yearly downtime in minutes), where D is the downtime of the system and A is the availability of the system (SATHAYE; RAMANI; TRIVEDI, 2000).

$$D = (1 - A) \times 8760h \times 60min \tag{2.2}$$

2.4 OPERATIONAL ANALYSIS

Operational analysis is a set of basic quantitative relationships between performance metrics based on measured or known data on computing systems (DENNING; BUZEN, 1978).

Some usually accepted operational analysis notation is listed below:

- T: the observation period
- K: number of resources in the system
- B_i: total busy time of the resource i in the period T
- A_i: total number of service requests (i.e arrivals) to the resource i in the period T
- A_0: total number of service requests submitted to the system in the period T
- C_i: total number of service completions by the resource i in the period T
- C_0: total number of requests completed by the system in the observation period T

Derived values can be obtained from these operational variables as listed below:

- S_i: mean service time per request at the resource i; $S_i = \frac{B_i}{C_i}$
- U_i: resource's utilization i; $U_i = \frac{B_i}{T}$
- X_i: throughput (i.e., tasks processed per unit time) of the resource i; $X_i = \frac{C_i}{T}$
- λ_i: arrival rate (i.e., arrivals per unit time) at the resource i; $\lambda_i = \frac{A_i}{T}$
- X_0: system's throughput; $X_0 = \frac{C_0}{T}$
- X_i: mean number of visits (i.e., the visit count) per request to the resource i; $V_i = \frac{C_i}{C_0}$

The utilization law is defined by the equation presented below which relates the average time that the resource i was busy for each completion to its throughput (MENASCE et al., 2004).

$$U_i = S_i \times X_i \tag{2.3}$$

If $C_i = A_i$, then $X_i = \lambda_i$. It means that the number of arrivals during the analyzed time interval is equal to the number of completions by the resource i. Consequently, the utilization law can be calculated as follows:

$$U_i = S_i \times \lambda_i \tag{2.4}$$

According to (MENASCE et al., 2004) "The service demand, denoted as D_i, is defined as the total average time spent by a typical request of a given type obtaining service from resource i".

The service demand relates a resource to a set of tasks and it is essential for performance modeling. A request, while it lasts, can visit one or more devices multiple times. Service demand is the sum of all service times offered by a resource considering all visits to it. On the other hand, in the case of a scenario in which there are several requests using the same resource, the service demand is given by the average of all requests. The workload's intensity parameters plus the service demand are the input parameters for the queuing network (QN) models. This is one of the important characteristics of the service demand. It is possible to obtain the total time in which the resource was occupied by multiplying its utilization U_i by the observation time. Dividing this result by the total number of completed requests, C_0, allows finding the mean time that the resource has been occupied attending each request. This average value is the service demand. There may be difficulty in obtaining some individual values. This indicates that the service demand can get directly from the device's utilization and system's throughput. The following equation is named service demand law(MENASCE et al., 2004):

$$D_i = \frac{U_i \times T}{C_0} = \frac{U_i}{X_0} \tag{2.5}$$

There is a way to associate the system throughput, X_0, to the resource's throughput i, X_i which is named Forced Law.

$$X_i = V_i \times X_0 \tag{2.6}$$

2.5 QUEUING THEORY

Network administrators must strive to deliver the best performance and the lowest cost. When an analyst plans to compare configuration scenarios in order to find out which is the best, a performance evaluation is required. The performance evaluation is also applicable in the decision process between systems with similar characteristics, allowing to infer which is the best for a specific task. Through the performance evaluation, it is also possible to determine the performance of a system in specific tasks and if any improvements or upgrades are needed. In practical terms, assessing a system's performance consists of analyzing its behavior based on a defined set of metrics. The researcher must

adopt appropriate assessment techniques (for example analytical modeling, simulation, or measurement), perform a statistical analysis to highlight possible bottlenecks and design solutions for improvement. This dissertation applied a performance analysis based on analytical modeling with the Markovian queue model M/M/1/K.

In this dissertation, we have taken a more in-depth look at queuing systems. Queuing theory ranges from the study of simple single-server systems modeled as birthdeath chains to the analysis of arbitrarily complex networks of queues. The main purpose of queuing theory is to measure system performance under certain operating conditions, instead of selecting the operational plan to be implemented that presents the best possible performance. Many interesting queuing systems can be viewed as special cases of birth-death chains. Some extensions are necessary to deal with more complex situations involving event processes which do not satisfy the Markov property and with networks of queues connected in unsystematic ways (CASSANDRAS, 2008).

According to (CASSANDRAS, 2008) Queuing theory made some of the most important contributions to the analysis of stochastic Discrete Event Systems (DES), where resource containment problems predominate. Queuing systems form a very extensive and applicable class of DES, especially when dealing with resource sharing problems often encountered in the design and control of computers and communication systems. As already introduced, the big goal of queuing theory is to develop "descriptive" tools for studying queuing systems, rather than "prescriptive" tools to control and influence their behavior in an unpredictable and dynamic environment, that is constantly changing.

There are basic definitions and notation required to proper specify a queuing system model. Commonly, there are three components to the model specification process:

- Specification of stochastic models for the arrival and service processes;
- Specification of the structural aspects of the system such as the storage capacity of a queue, the number of servers, among other parameters;
- Specification of the operational policies used to impose conditions for acceptance of incoming customers, prioritization by some types of customer (differentiated treatment), among other policies.

An event is always associated with a clock sequence implicit in stochastic automata. In a queuing system, a stochastic sequence $\{Y_1, Y_2, ...\}$, where Y_k is the k-th time between arrivals is related to the events of arrival, that is, a random variable such that Y_k = time elapsed between the arrival of $(k-1)th$ and kth, k = 1,2, ... In summary, we always define Y_0, so Y_1 is the random variable that describes it in time for the first arrival. As already mentioned, in general, the stochastic behavior of this sequence requires the definition of the joint probability distribution of all events [$Y_k \leq$ t], k = 1, 2, .., and so on. According to (CASSANDRAS, 2008), in most queuing theories, the $\{Y_K\}$ stochastic

sequence is considered to be iid (i.e. Y_1, Y_2, ... are independent and distributed in an identical way. In this line, a single probability distribution is able to completely describe the time sequence between arrivals. In Equation 2.7 (CASSANDRAS, 2008), the random variable Y is usually seen as a generic time between arrivals, which do not need to be indexed by k.

$$A_{(t)} = P\left[Y \leq t\right] \tag{2.7}$$

The mean of the distribution function A(t), E [Y], is especially important and it is customary to use the notation to represent it. Thus, λ is the average arrival.

$$E\left[Y\right] = \frac{1}{\lambda} \tag{2.8}$$

Likewise, we associate an event with a stochastic sequence $\{Z_1, Z_2, ...\}$ where Z_k is the k-th service time, that is, a random variable such that Z_k = time needed for the k-th customer to be attended, k = 1, 2, ... If we consider that the stochastic sequence $\{Z_k\}$ is iid, we will define

$$B_{(t)} = P\left[Y \leq t\right] \tag{2.9}$$

where Z is generic service time. Alike to Equation 2.8(CASSANDRAS, 2008), we use the following notation for the mean of $B_{(t)}$:

so that μ is the average server service fee on our system.

In queuing theory, it is common to use a specific type of notation to briefly describe a system. This notation is as follows:

A/B/m/K

where

- A is the interarrival time distribution
- B is the service time distribution
- m is the number of servers present, m =1,2,...
- K is the storage capacity of the queue, K =1 ,2,...

Thus, the single server system with infinite queuing capacity is described by A/B/1/∞. We have listed some examples to illustrate the A/B/m/K notation:

- M/M/1 A single-server system with infinite storage capacity. The interarrival and service times are both exponentially distributed, that is $A_{(t)} = 1 - e^{-\lambda t}$, $B_{(t)} = 1 - e^{-\mu t}$, for some positive parameters λ and μ.

- M/M/1/K A single-server system with storage capacity equal to $K < \infty$.
- M/G/2 A system with two servers and infinite storage capacity. The interarrival times are exponentially distributed. The service times have an arbitrary (general) distribution (CASSANDRAS, 2008).

In the M/M/1/K model, the number of clients that can be attended in the system is limited to K. A denial occurs when a client arrives and finds the server with the resources fully in use. In this case, the customer is considered lost, in a phenomenon known as customer blocking. Otherwise, the model is alike to the M/M/1 case. It is possible to use the birth-death model to analyze this system, defining the arrival rate $\lambda_n = \lambda$ for all n=0,1,..,K-1 and $\lambda_n = 0$ for all $n \geq K$. In fact, the Poisson arrival process is terminated when the queue length is K and reactivated when it becomes $(K-1)$.

The strategy of ending arrivals only works with a Poisson process due to property without memory: when reactivation of the process occurs, the residual time interval between arrivals of the existing process has the same distribution as the entire time interval between arrivals, which results in a Markovian structure of the transition rate among states(CASSANDRAS, 2008).

The $\frac{\lambda}{\mu}$ ratio is fully applicable. It is possible to deduce that if $\lambda > \mu$, clients are lost, but the queue length remains limited by K and no instability (unlimited growth in queue length) can occur. Defining $\rho = \frac{\lambda}{\mu}$, it is evident that ρ does not represent the traffic intensity. This is because λ is the rate of customer arrivals, but not the rate of customers actually accommodated to the system, as some of them are blocked (CASSANDRAS, 2008).

One of the performance measures of interest to the M/M/1/K system is the Utilization, which can be calculated using the equation below (CASSANDRAS, 2008):

$$Utilization = 1 - \pi_0 = \frac{\rho^i \left(1-\rho\right)}{\left(1-\rho^{k+1}\right)} \tag{2.10}$$

The great difference between the M/M/1 model and the M/M/1/K model lies in the fact that the utilization is conditioned, in addition to λ and μ, to the structural parameter K. ρ functions as a variable that can float freely. It is possible to notice that when ρ tends to the infinity the Utilization approaches 1. Another point to be considered is that once $\rho < 1$, considering that $K \to \infty$ (the model approaches the stability of the M/M/1 model), consequently $(1 - \pi_0) \to \rho$, which is in fact the use of the M/M/1 system.(CASSANDRAS, 2008)

Throughput in the M/M/1/K system can be calculated from the arrival rate λ. It should be noted that throughput will always be less than the arrival rate as some customers do not have access to server resources (blocked)(CASSANDRAS, 2008).

$$Throughput = \mu \left(1 - \mu_0\right) = \lambda \frac{1-\rho^K}{1-\rho^{K+1}} \tag{2.11}$$

The probability of a client reaching a server and finding it full is equivalent to the probability that a client will get lost. This metric is one of the most relevant performance measures in the limited capacity of queuing systems. Accordingly to the text presented in (CASSANDRAS, 2008) the probability of finding the full queue, denoted as Blocking Probability (PB) is "the fraction of time the queue is observed full by an independent observer and not an arriving customer".

$$P_B = \pi_K = (1 - \rho) \frac{\rho^K}{1 - \rho^{K+1}} \tag{2.12}$$

Again, if the requirement of $\rho < 0$ is attended, when K tends to infinity the blocking probability P_B tends to 0. This occurs for an infinite capacity system in which there is total stability (M/M/1) (CASSANDRAS, 2008). One of the major challenges of the queuing system design is to define the storage capacity K so that the blocking probability remains below a predefined level. In the case of the Markovian model, it is possible to modulate the arrival and service rates λ and μ, considering the equation to ensure that there is a blocking probability below the maximum acceptable level.

2.6 CONTINUOUS TIME MARKOV CHAIN

By using Markov chains, it is possible to evaluate the performance and reliability of computers and communication systems. Modeling the interaction between different elements that make up infrastructure can be made by using Markov Chains. Markov chains can be defined as a state diagram associated with a Markov process (BOLCH et al., 2006). Markov chains are a stochastic process with memoryless property (CHUNG, 1954). The nonexistence of memory is a property formally known as a Markov property and it is present in some probability distributions. The basis for evaluating model-based systems is formed by the Markov chains and stochastic processes.

The evolution of a system over time T indexes a set of random variables X. This is the main characteristic of a stochastic process (CASSANDRAS, 2008). A random variable can be understood as a quantitative variable whose value covers different numerical values affected by random factors. The time of an activity or event in a ω sample space related to some probability distribution can be represented by a random variable.

A stochastic process can be considered as a set of X random variables defined as $X_t : t \in T$. In this context, a probability distribution function characterizes each instance X_t. The set of indexes T is associated predominantly with the increase of time. $Pr\{X_t\}$ represents the probability of a given event X_t occurring. In a deterministic process, as opposed to the stochastic process, times are not affected by random factors. The state-space of the stochastic process is represented by the set of all possible values of X_t ($\forall t \in T$) (BOLCH et al., 2006; CASSANDRAS, 2008), named here as S.

Discrete value implies that the state space of possible values in the Markov chain is finite or countable. A Markov chain is a discrete state Markov process. A Markov process is a stochastic process in which $Pr\{X_{t_{n+1}}\}$ depends only on the previous value of X_{t_n}, $\forall t \in T$ and $\forall s_i \in S$. A Markov process is generally called a memoryless process, in which the future state of the system is conditioned only to its current state (BOLCH et al., 2006; HAVERKORT, 2001). In summary, for evaluation purposes, it does not matter the path followed by a process from the initial to the current state. Additionally, in a memoryless process, the transition to the next state is not affected by the time spent in the current state, nor in previous states (BOLCH et al., 2006; CASSANDRAS, 2008). Due to the Markov property, the time associated with an activity must follow a distribution without memory. The probability distribution related to the times involved follows an exponential distribution for the case of Markov chains.

The times can be classified as discrete or continuous conditioned to the type of random variables associated with the activities of a process. Understanding the main difference between discrete and continuous Markov chains is of crucial importance in the decision-making process for the most appropriate abstraction to represent the process in hand. In order to represent the way in which transitions between states occur over time, some abstractions for Markov chains have been proposed. Discrete-time Markov chains (DTMC) are useful for representing systems that evolve in discrete time steps. Consequently, the system only alters at one of the discrete-time values in the set T, which $t \in T$ represents only non-negative integer values. In turn, systems in which the state changes may occur at any time along with a continuous interval (it means that $T \in R_+$) can be represented by continuous-time Markov chain (CTMC). Essentially, CTMC is similar to DTMC with the difference that transitions in CTMC may occur in any instant of time (BOLCH et al., 2006). The chosen abstraction regarding Markov chains directly interferes with the way in which metrics are calculated.

CTMCs, graphically, are represented by a directed graph. The directed arcs indicate how transitions between states occur while vertices represent states. The different conditions that a system may follow can be represented by Markov chains. Each arc has an associated transition rate or probability, signaling the way in which transitions happen from one state to another. Transitions between states represent the occurrence of events (SILVA et al., 2013).

Conceptually, CTMC is defined by a probability vector for the states π and a transition rate matrix Q, also called an infinitesimal generator matrix. Matrix Q has a dimension equal to the number of states in the S state space, so it is called a square matrix. The elements of the Q matrix represent the transition rates between the states of the chain. The q_{ij} elements (for $i \neq j$) represent the rate to reach state j departing from i. The main elements of the q_{ii} diagonal are always defined by $-q_{ii} = \sum_{j,j\neq i} q_{ij}$. The main diagonal elements are always negative while the elements outside the diagonal are always non-

negative. Once the model structure is defined so that the Q infinitesimal generating matrix is known. The sum of each row of the matrix is equal to zero. In turn, the probability vector for states π is a one-row vector that contains the transition probability from the current state to all states, including itself. The initial probability vector is defined as $\pi(0)$. Below, we presented an example of an initial probability vector $\pi(0)$ and matrix Q.

$$\pi(0) = \begin{pmatrix} 1.0 & 0.0 & 0.0 & 0.0 \end{pmatrix}$$

$$Q = \begin{pmatrix} q_{00} & q_{01} & q_{02} & q_{03} \\ q_{10} & q_{11} & q_{12} & q_{13} \\ q_{20} & q_{21} & q_{22} & q_{23} \\ q_{30} & q_{31} & q_{32} & q_{33} \end{pmatrix}$$

$$Q = \begin{pmatrix} -1.72914 & 1.72914 & 0.0 & 0.0 \\ 0.0 & -3.25260 & 3.25260 & 0.0 \\ 0.0 & 0.0 & -0.68932 & 0.68932 \\ 2.82939 & 0.0 & 0.0 & -2.82939 \end{pmatrix}$$

In (BOLCH et al., 2006), it is demonstrated that non-time-dependent metrics are obtained through stationary analysis. By Equation 2.13 (BOLCH et al., 2006), it is possible to calculate the state probability vector for steady-state scenarious. A probability distribution that converges to a vector of stationary probabilities, independent of the initial distribution $\pi(0)$ can be obtained by the resolution of a system of differential equations. It is possible to find out what happens in the long run through the obtained limiting probability distribution.

$$\pi \times Q = 0, \sum_{i \in S} \pi_i = 1 \tag{2.13}$$

By applying a transient evaluation, time-dependent metrics are obtained (BOLCH et al., 2006). By its application, it is possible to infer what happens in the system, based on a specific period of time. The behavior of the system before reaching a steady-state can be predicted by transient evaluations. The row vector $\pi(t) = [\pi_1(t), \pi_2(t),\ points, \pi_n(t)]$ represents the transient state probability vector of a CTMC at time t. The behavior of a CTMC can be described by the Kolmogorov Equation 2.14, given the initial probability vector $\pi(0)$. Equation 2.15 gives the total expected time that a CTMC spends in the state i during the interval [0, t]. From its π state probability vector and its Q matrix. Performance metrics can be derived by considering the characteristics of the evaluated system as Equations 2.14 and 2.15 (BOLCH et al., 2006).

$$\frac{d\pi(t)}{dt} = \pi(t) \times Q \tag{2.14}$$

$$L_i(t) = \int_0^t \pi_i(x)dx \tag{2.15}$$

Sometimes, it is important to consider and evaluate the amount of time required to conclude a process activity. The mean absorption time is used to calculate the mean time spent until absorption. The model must have at least one absorption state. A proper understanding of MTTA calculation involves conceptualizing the difference between transient states and absorbing ones. Transient states are defined as temporary states. That is, when the system leaves a transient state, there is a probability that the system will not reach it again. In turn, an absorbing state is a state that, once reached, there is no way out. The end of a task can be represented by the absorbing state. For MTTA evaluation, the state space S is divided into two sets, the set of absorbing states A and the set of non-absorbing states N. In this context, it is possible to create the initial probability vector for the transient states $\pi_N(0)$. A new Q_N matrix can be built from the Q matrix, restricting Q to transient states. The cardinality of the set of non-absorbing states N defines the dimension of the square matrix Q_N. The time spent before absorption can be calculated restricted to the states of the set of non-absorbing states (N) in $lim_{t\to\infty}L_N(t)$. Therefore, $L(t)$ satisfies Equation 2.16 where $\pi_N(0)$ is the vector $\pi(0)$ restricted to states in the set N. Q_N is the infinitesimal generator matrix restricted to non-absorbing states. At the end, Equation 2.17 may describe MTTA (BOLCH et al., 2006).

$$L_N(\infty)Q_N = -\pi_N(0) \tag{2.16}$$

$$MTTA = \sum_{i\in N} L_i(\infty) \tag{2.17}$$

3 RELATED WORK

This chapter will show an overview of related works, providing an association between our approach to existing works. Important concepts will also be presented that will be used in subsequent chapters of the work in order to demonstrate the state of the art of the technologies employed in the document .In this chapter, we will present a summary and comments on each work related to the research object.Subsequently, a table will be inserted synthesizing, comparing and relating in a synthetic way the present work with the other academic works selected as a reference. This table will also be commented on.The chapter ends by presenting points of contribution from our work considering the research area and similar academic works.

3.1 AVAILABILITY WORKS

DANTAS (DANTAS et al., 2012b; DANTAS et al., 2012a) proposed various models aimed at analysing the availability of the cloud computing platform Eucalyptus. In these works, SPNs, CTMCs and RBDs were proposed for establishing a basic architecture, besides evaluating the effects and impacts of the use of warm standby-type redundancy mechanisms for improving the availability of platforms and variations of the basic architecture, considering failure times and repair of components responsible for the functioning of the Eucalyptus platform for hierarchical and heterogeneous modelling, following an exponential distribution of status and blocks. A Markov Chain was used to calculate the availability of each component in the cloud.

Later, DANTAS *et al.* (DANTAS et al., 2015) carried out a study aimed at carrying out a cost analysis of energy consumption, proposing hierarchical models and determination of equations for assessing availability according to the implementation capacity of a public cloud using the Eucalyptus platform. However, different to what was performed by Dantas et al., the present work does not assess the scenario of a public cloud, but of a corporate network, also addressing warm standby-type redundancies, quantifying the availability of the network, promoting a combination of two variable architectures and assessing the cost x downtime ratio for implementing each architecture.

In turn, GUIMARAES (GUIMARAES et al., 2011) studied the availability of computer networks in four different scenarios, with two scenarios with cold standby redundancies, other two scenarios presenting no redundancies, and other a physical redundancy in the link. For investigating availability in these scenarios, a Stochastic Petri net was used as a modelling approach, which allowed an analytical assessment of complex scenarios. In addition, Reliability Importance was used for identifying the most important components in the system, according to the metric of interest. The scenarios presented in the work

generated traffic through simulation tools. The given work, similar to the present research, addresses the issue of switchover time and also establishes a comparison criterion between the scenarios presented. On the other hand, it only considers hardware requirements but not aspects of settings and protocols, which is the subject of this research. The present work uses warm standby redundancy, as well as using operational data of a production environment.

The work (KIM; KIM, 2015) shows that the needs of services in real time, as well as financial services, are increasing in terms of availability, with reliability emerging as a critical issue. In this context, availability, reliability and scalability of routers to control the flow of data and the data communication link are being recognised as critical problems. Several methods are being studied for improving reliability and availability, to minimise the cost of losses due to the failure of a communication network router. Therefore, the given work investigated a BGP session takeover method that supports high availability of a BGP session, a protocol that uses TCP, among various other routing protocols.

The authors still state that in the case of router failure, the TCP connection, which is part of the transport layer, is disconnected due to the protocol characteristics. Thus, routing protocols using TCP, such as BGP, can no longer maintain the connection. In this case, even if the failure is fixed and the operation resumes, the contents of the routing tables will need to be exchanged after the connection is re-established, resulting in long downtimes. Therefore, in order to provide seamless routing protocol services, such as BGP sessions, a non-stop active routing technology is necessary. The non-stop active technology is a functional element of the high-availability system, being a technology that enables the client and a peer to communicate with each other and to maintain sessions without being aware of communication disconnections, even if the communication disconnection has actually occurred between the client and the peer.

The work demonstrates the requirements for establishing a standby router and how the information of the BGP session established in the active router is controlled by the backup router. The paper still describes the mechanism in which the BGP Open Messages, which contain the destination address information and keepalive time information, as well as BGP update messages with updates of the session parameters, known by the standby router from the active router. The redundancy module (RM) proposed in the given work operates in active and standby routers in a controlled and synchronised manner, in which when a TCP connection between the active and standby router is complete, the active router will transmit the backup data generated at a designated time point for backup to the RM of the standby router.The BGP re-connection method, which exists without BGP backup data, detects the disconnection of the BGP session when the active router is interrupted and being re-connected in the standby router after the takeover time. When this takeover takes place, the standby router is reconfigured with new peers and, at this time, information such as routing table information is exchanged. The authors also carried

out a simulation in the NS-2 to compare the method proposed in this document and the method proposed in the study with the existing BGP re-connection method.

In our opinion, the work (KIM; KIM, 2015) demonstrated that the BGP re-connection methods have problems, such as the increase of control time due to the time taken to detect the session disconnection and the exchange of connection information for re-connection. The work carried out an interesting comparative analysis of performance in terms of BGP throughput, being similar to the work which is proposed in the present study. As a result of the simulation, it was noted that the BGP takeover method proposed in the given article ensured the recovery of throughput more quickly than the existing re-connection method. The simulation also showed that the existing re-connection method is much slower at re-connection than the method proposed by the authors, for the existing method of re-connection exchanges routing tables in the case of re-connection, while the BGP takeover method suggested uses a routing table, which was copied in advance by RM in case of takeover. The present work also suggests tackling the issue of reducing switchover time for takeover. A distinction between the article (KIM; KIM, 2015) and the present work includes the solution employed, which, in the given article, is focused on hardware failure, while the present work focuses on data link failures.

The work (SERMPEZIS; DIMITROPOULOS, 2017) states that intra-domain routing / autonomous systems is realised in a distributed way, over Border Gateway Protocol (BGP) and, despite its global BGP, has several shortcomings, such as slow convergence after routing changes, which can cause packet losses and interrupt communication for several minutes. The article is complementary to other works, approaching the acceleration of convergence and inter-domain centralisation approaches based on SDN (Software Defined Networking). The authors propose a probabilistic framework to analyse the effects of centralisation on the intra-domain routing performance. In addition, it defines bounds for the time needed to establish data plane connectivity between autonomous systems after a routing change, as well as predictions for the control-plane convergence time. The results obtained provide useful insights that can promote future research, like this present work, discussing the application of the results and demonstrating the gains of an autonomous system communication topology.

The authors argue that, despite the slow BGP convergence, the implementation of other protocols or modified BGP versions is difficult, due to its general use as well as the political, technical and economic challenges implied. Therefore, any advances or solutions proposed are excellent for the use of BGP. With this in mind, the work analytically discusses the effects of centralisation of intra-domain routing performance, concentrating on potential improvements to the (slow) convergence of BGP. The authors use scenarios considering two different assumptions. The first assumption is that BGP Update times are independent random variables drawn from an arbitrary distribution function. Thus, BGP update times are obtained by a renewal process. The model is very generic, as it

enables to use any valid fbgp(t) function and, thus, describe a wide range of scenarios with different parameters, which are shown in the work. Real measurements can be used to make a realistic selection for the fbgp(t) distribution.

In turn, the second assumption is that BGP Update times are independent random variables extracted from an exponential distribution with rate $\lambda = \frac{1}{\mu bgp}$ and mean value $E[Tbgp] = \mu bgp$. With this, the authors built a transient Markov chain for modelling the propagation of BGP updates, in which each state denotes the set of nodes that have updated the paths in their RIBs (Routing Information Base). However, analysing this Markov chain can still be quite complex, as the state space contain $2N1$ states, with transition rates still depending on the topology of the network, which is not exactly known. Finally, the authors consider at first a full-mesh network (a common approach in the literature), which can be described by a much simpler Markov chain, calculating convergence time in the control plane as a function of the size of the network.

3.2 PERFORMANCE WORKS

In turn, the work (MASRUROH et al., 2017) analyses the performance of a network using GNS3 based on the convergence parameters, throughput, jitter and packet loss when applying an internal routing protocol (RIPv2, OSPF, EIGRP) with an external routing protocol (BGP). The authors argue that each protocol has its own advantages and disadvantages, with the determination and selection of the routing protocol depending on various parameters that affect the quality of the network. Simulations were then used following the steps in the configuration as in the real implementation. The combinations used in this simulation were based on the combination of internal routing protocols (RIPv2, OSPF, EIGRP) with an external routing protocol (BGP). The work compares the performance of each IGP with BGP. The design of the network model divides the topology into three different routing protocol areas, enabling to separately analyse the performance in each scenario proposed.

Among the scenarios proposed, it is important to highlight the analysis of the convergence parameter, which is also a subject to be studied in this present work. The graph in this paper shows the comparison between the average value of convergence for each routing protocol combinations. The lower the value, the quicker the convergence of the routing protocol network for obtaining and updating the table. OSPF routing protocol, which is also used in the corporate network of this present work, has the average value of the best convergence.The work (MASRUROH et al., 2017) is highly related with the present work, as it also approaches the performance of protocols, measurements and the comparison of different scenarios. Similar to this work, it also aims at carrying out a performance evaluation through a peer before-and-after comparison for obtaining the best convergence condition and the best performance of a corporate network, consequently obtaining the best availability.

On the other hand, the work (TANYINGYONG et al., 2013) proposes a resilient communication through muti-homing for remote healthcare applications.The solution is based on Bidirectional Forwarding Detection (BFD) for fast failure detection and a customised re-routing operation. The authors investigate the challenge of establishing short reaction times for re-routing (fast failover) with low probability of false alarms, based on the allocation of system resources for processing BFD control messages and subsequent demonstrations. Accordingly, a simple mechanism is proposed, which provides fast failover, while maintaining a very low probability of generating false alarms and unwanted re-routing decisions. The work demonstrates, through empirical results, the reduction on average failover using BFD.

The work (TANYINGYONG et al., 2013) also attempts to tackle the issue of convergence time of routing protocols, as this convergence time is in the order of seconds, if not minutes. To tackle this issue, the research community has suggested various mechanisms, such as IP Fast Reroute Framework, as defined in RFC 5714, which provides protection against link or route failure. According to the RFC, there are several possible methods for FFD (Fast Failure Detection), such as 1) physical detection, such as loss of signal; 2) routing protocol detection with the use of fast hellos; 3) protocol detection that is routing protocol-independent, such as the BFD protocol. Thus, a resilient communication relies on quick response to failure. In the context of the study, the response time to the failure is the failover timer, which can be calculated according to the following equation: Failover Time = Failure Detection Time + Reaction Time, in which the failure detection time is the time required to recognise that the failure takes place, while reaction time is the time required to switch to an alternate path after the detection.

The authors use the BFD primary mode (the asynchronous mode) for all communications. With this, a BFD process (speaker) negotiates the value of the hello interval with a BFD neighbour. This experiment is carried out to assess the average failover time in the scenarios established. The authors used an experimental configuration, creating two alternative scenarios. Link 1 is mainly used in all communications, with the system alternating to Link 2 in case Link 1 fails. The same behaviour is seen in all alternatives considered. The experiment was performed using ICMP packet from one gateway to the other, forcing an interruption (in Link 1) and measuring failover time, using the period of time when R2 no longer obtains a response until the time it receives the first successful response. The same procedure was repeated according to statistical criteria.

In addition, the work (TANYINGYONG et al., 2013) generated traffic load for 1 minute in each test round. Taking into account that the BFD session is always active, the hypothesis that false alarms will occur as traffic load increased was studied. Thus, false alarm periods were measured, which is the duration, in percentage of the total experiment time, for which a BFD session is down. In addition, the authors measured the number of BFD status flappings (status changes from up to down). An experiment was then carried out, in which

two routers were configured with a BFD session established between them, generating an increasingly higher traffic load, aiming at analysing whether false alarms occurs as traffic load increases. The experiment showed that the BFD session is not affected when the load traffic is low. Nonetheless, false alarms occurred at 120 Kbps, where the CPU load reaches 96%.

The periods of false alarm increase as traffic load increases, as the system is saturated, and more BFD packets are discarded. The number of flappings as a function of traffic load in which an increase in flapping frequency is observed when the system is saturated. The work attempts to link the hello period with the period of false alarms, the saturation point. Thus, after the saturation point, the number of flappings decreases, having observed periods of false alarms and that the number of flappings decreases as the hello multiplier increases. However, this increases the failure detection time, with the system being consequently less sensitive to changes. The technical argument for this behaviour is the fact that there is a trade-off between the FFD and the stability of the system.

With this, it can be stated that the work (TANYINGYONG et al., 2013) presents an interesting methodology, which might be complementary to the present work. Nevertheless, it is necessary to optimise the unwanted failover with the operational circuit. The objective of the present study lies on a reliable system, with no false alarm periods nor flappings. This requires an extra mechanism that can solve the problem. In addition, there are many researchers already investigating this type of experiment, which require to be detailed, taking into account the effect of the hello variation and the aspects of resource processing. Although this work was carried out due to demands of the health sector, the solution can be considered to have a general application. Thus, it can be applied on different services in real time, with critical purposes in several applications, such as an automatic control and in train operations, rain traffic signalling systems, robots in assembly lines, etc.

In turn, the work (GHANNAMI; SHAO, 2016) focuses on failure recovery mechanism using OpenFlow. The proposed mechanism is divided into: (a) proactively computing the working paths between each source-destination and return a list of paths ordered by path latency, from the shortest to the longest path; (b) implement per-link Bidirectional Forward Detection (BFD) for failure detection, to enable fast detection time, thus, fast recovery time; (d) configure OpenFlow Fast Failover Group for restoration, such that the highest priority bucket is linked to the second shortest path and so on. In the case of failure, the switch will revert to the second fastest path; (e) to achieve high resource utilization, OpenFlow Select Group was used to split the flow among the working paths. The architecture proposed in this work provides high utilisation of network resources, as well as an efficient fast detection and recovery time. In addition, the Open Vswitch, an open source implementation of OpenFlow, was employed in the experiment of this work, which supports BFD.

The paper (GHANNAMI; SHAO, 2016) proposes an optimised and highly resource-

utilised fast recovery mechanism for OpenFlow networks. By using multipath routing, the controller computes the possible paths as a separate rooted tree between each source-destination pair. In the case of link failure, the returned ordered (by latency) list of paths for each source-destination pair will be used to configure Fast Failover groups, to ensure that in situations where a switch experiences failure, the switch will redirect the data of the failed path to the second fastest path available. The integration of BFD with the Fast Failover group enables a fast detection of failures. The chaining of Fast Failover and Select groups increases the throughput and improves the efficient use of the working paths. However, this work did not conduct any experiments to investigate link failure and network congestion possibilities.

On the other hand, the work (BISTOUNI; JAHANSHAHI, 2014) states that the reliability of a system cannot be precisely measured, regardless of the functionality of individual system components and their impact on the system's performance. On the other hand, reliability block diagrams (RBD) are the graphical representations of the components of a given system, with the relationships between them, which can be used to determine the overall reliability of the system, namely for complex and large-scale systems. However, for a broader analysis, it is necessary to examine all network reliability parameters (i.e. terminal reliability, broadcast reliability and network reliability), being used for a wider analysis. Thus, in this work, RBDs are widely used in detail, taking into account a structure that is as close as possible to reality from a reliability point of view. The complex series-parallel RBDs are used to determine the reliability of the fault-tolerant MINs (multistage interconnection networks). The reliability of one of the most common MINs, the shuffle-exchange network (SEN), was also assessed through the impact assessment of the increase of the number of switching stages. In addition, it was found that the reliability of SEN with an additional stage (SEN +) is greater than SEN alone or SEN with two additional stages (SEN +2), although the reliability of SEN is higher when compared to SEN with two additional stages (SEN +2).

The authors re-assessed the reliability of these networks, with the results of the reliability analysis of the terminal, transmission and network demonstrating that the reliability of SEN + and SEN +2 continuously surpass those of SEN, being both very similar in terms of this parameter.The results of reliability obtained in this work indeed show that SEN + and SEN 2+ have very similar reliability levels when compared to SEN. The authors then concluded that adding one stage is more efficient than adding two stages to the SEN in terms of reliability of the terminal. This is due to the fact that the reliability of the SEN terminal increases with the addition of an extra stage, while it does not change with the addition of another stage.

Therefore, the work (BISTOUNI; JAHANSHAHI, 2014) draws some similarity with the work proposed herein, as it establishes a comparative study, using modelling tools such as RBD. On the other hand, the work does not address the heterogeneity of switching

elements. That is, all different switching element in MINs are probably not of the same size, thus, each one has a different level of reliability that should be considered.

The paper(KITSUWAN; SRIKOON; NOPPANAKEEPONG, 2003) discusses finding an optimal buffer size in shared buffering by using raising rate of packet loss probability and mean delay, respectively denominated percentage of packet loss probability and approximate percentage relative delay. This can reduce the cost and complexity of the system, reaching an appropriate waiting time, while packet loss still remains adequate. The major problem that was found in packet switching is packet loss, which is caused by contention. The solution lies on using buffers to collect the contending packets that may be lost from this event. There are many buffering techniques in packet switching. Thus, this paper introduces the technique of using output buffers and shared buffer to work synchronously. If there is more than one packet to be sent to the same output at the same time (only one packet can reach out the output), some packets will be temporary stored in the output buffers and wait to be sent to the output in the next time slot. But, if the output buffer is full, the contending packet has to be sent to the shared buffer. The authors called this technique "Partially Shared Buffer".

The buffering technique certainly causes the delay, especially in the partially shared buffer, where packet loss and delay both depend on the output and shared buffer size. They also found that they do not have to use a high number of input and output ports, as the packet loss probability is usually the same in equivalent share buffer size. The analysis of optimal share buffer in switch with partially shared buffering carried out in this paper can find the appropriate shared buffer size to reduce the cost and the complexity of the system. Moreover, two measurements (mean delay and packet loss probability) were used as parameters for evaluation. The work carried out herein also uses packet loss probability as an evaluation metric and, similarly to the work pointed out, enables to infer on the optimal number of circuits configured, with the parameters packet discard rate used as an availability solution.

In the paper (MONDAL; MISRA; MAITY, 2018), the authors evaluated the optimum values of different performance metrics, such as buffer size, packet arrival and processing rates and packet waiting time, in the case of packet flow through an OpenFlow switch-based system. A queuing theory-based analytical scheme, named OPUS, based on the existing OpenFlow protocol was developed. This model depicts events such as the packet arriving at an ingress port, packets getting queued at ingress buffers and packets getting processed by an OpenFlow switch in OpenFlow systems. The authors performed a queuing theory analysis of the proposed model, OPUS. Based on the analysis, they commented on different events, such as the optimum buffer size, trafc intensity and packet waiting time. The authors also estimated the optimal value of buffer size in a simulated environment, as well as the packet arrival and processing rates of an OpenFlow switch-based system. In addition, they evaluated the maximum packet waiting time of an OpenFlow switch in

OpenFlow systems.

They proposed a queuing scheme for an OpenFlow system— $C - M/M/1/K/\infty$ queuing model. Furthermore, they theoretically calculated the minimum buffer size requirement of an OpenFlow switch. Simulation-based analysis showed a two-fold increase in packet processing rates, with the packet arrival rate being increased by 26.15–30.4%. The authors also infer that for an OpenFlow system, the minimum buffer size is of 0.75 million packets, with the maximum packet arrival and the minimum processing rate of 0.20–0.25 million packets per second (MPPS) and 0.03–0.35mpps, respectively, with a maximum packet waiting time of 0.173–0.249. This work is also similar to the present work, as it uses the queuing theory to analyse the performance of the system. It is important to point out that the calculations were carried out according to a set of equations, such as in the present work. One difference between the present work lies on the fact that the present work carried out experiments in a production environment, with the K of this work representing the Forwarding Performance parameter rather than buffer.

The works in the literature that analyse network convergence do not consider, in an integrated form, empirical data, nor aspects involving the allocation of processing resources to support the improvement of failover time. Besides, there are also no studies assessing these mechanisms under traffic stress conditions. In general, these works use simulators which, for issues intrinsic to the application itself and to the platform where they are developed, represent relevant differences when compared to a study carried out in corporate production networks. Nevertheless, the studies and models proposed in the literature are important, reflecting the lack of validation of real scenarios in large-scale and critical networks, in which minimum time differences are relevant.

Differently to the objectives of the works found in the literature review, the present work is based on the models CMTC and M/M/1/K queue to support both the selection of the best design of a network infrastructure, particularly establishing a comparative and objective analysis between the design solutions of the different converging network infrastructures, formally considering several technical aspects. The scope of the study carried out herein can be considered appropriate for real time and critical mission services, with several applications, such as automatic control and train operations, as well as rail signalling systems, robots in assembly lines, etc. Table 1 draws a comparison between the present work and the main works regarding network modelling, performance and convergence studies.

Tabela 1 – Related Work Comparison

Related Work	**RBD**	**CTMC**	**SPN**	**Availability**	**Computer Network**	**Convergence**	**Before and After Comparison**	**Performance Evaluation**	**BGP**	**BFD**	**M/M/1/K**	**Empirical Data**
(DANTAS et al., 2012b)	x	x	x	x			x					
(DANTAS et al., 2012a)	x	x	x	x			x					
(DANTAS et al., 2015)	x	x	x	x			x					
(GUIMARAES et al., 2011)	x	x	x	x	x	x						
(KIM; KIM, 2015)				x	x	x	x	x	x			
(SERMPEZIS; DIMITROPOULOS, 2017)		x		x	x	x		x	x			
(MASRUROH et al., 2017)				x	x	x	x	x	x			
(TANYINGYONG et al., 2013)				x	x	x		x		x		
(GHANNAMI; SHAO, 2016)				x	x	x				x		
(BISTOUNI; JAHANSHAHI, 2014)	x			x	x	x	x	x	x			
(KITSUWAN; SRIKOON; NOPPANAKEEPONG, 2003)								x			x	
(MONDAL; MISRA; MAITY, 2018)				x	x	x	x	x	x		x	
Our Work		**x**		**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**

4 METHODOLOGY

In this chapter, we present two methodologies that aim to evaluate enterprise backbones. The first methodology aims to support availability evaluations, making it possible to compare scenarios and evaluate gains. The second one aims to evaluate the performance of concentrating equipment in each scenario presented in the availability evaluation methodology, allowing to infer the impact of adopting the availability solution regarding the processing cost. As we will see, some steps are common between the two methodologies. In addition, we present concepts, resources, tools, techniques, and methods to support the evaluation processes. Section 4.1 describes in detail all the methodology's steps for comparatively evaluating the availability of the network. Section 4.2 describes in detail all the methodology's steps for supporting the performance evaluation of the availability solution adopted.

4.1 METHODOLOGY FOR SUPPORTING AVAILABILITY EVALUATIONS OF CONVERGENT NETWORKS

In this section, we present the methodology for supporting availability evaluations of convergent network services in order to obtain a scenario that can offer the lowest downtime. The methodology comprises a set of well-defined steps for supporting the evaluation process. The steps of the methodology are as follows: *Understanding the Convergent Network*, *Parameters Definition*, *Metrics Definition*, *Markovian Modeling*, *Measurement Planning*, *Data Collection*, *Data Analysis*, *Statistical Treatment*,*Data Storage*, *Configuration Generation*, *Evaluation Environment*, *Network Deployment Planning*, and finally *Deployment*.

This process supports the decision making process related to the configuration scenario that offers the lowest downtime for the evaluated system. Using the methodology, service providers are able to decide which scenario presents greater availability and which is the order of magnitude of the availability gain to be achieved. Figure 3 depicts the steps that analysts must follow in order to evaluate the availability of their backbone networks. *"Configuration Evaluation Process"* is a subprocess that is considered after the modeling strategy is validated.

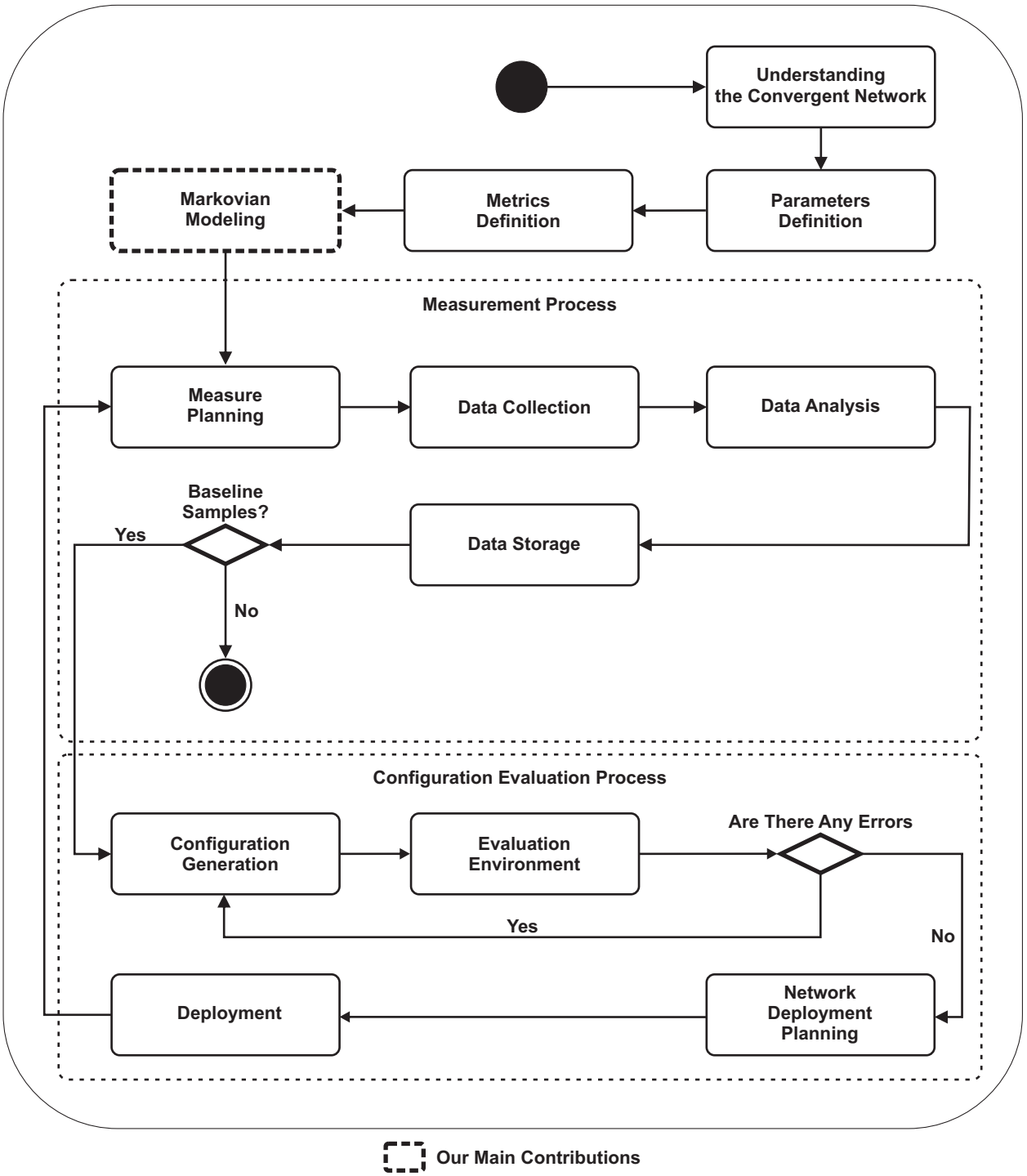

Figura 3 – Evaluating Availability of Convergent Networks

Understanding the Convergent Network:

Analysts should understand the network redundancy mechanism itself, its topology, and how the availability of the critical system can be planned when designing corporate networks. Understanding the network also includes understanding its architecture in order to find out how protocols can be associated.

The developed methodology can be applied in several convergent networks independent of the physical media of their data links, conditioned to the fact that the probability of failure in these media does not increase as time pass (memoryless property). A typical convergent network will always use media that is not subject to the relevant increment of failures over time. That means the times regarding failures always follow an exponential distribution.

Analysts should identify more intensive components in computational processing and understand how they interact with each other. Generally, networks have performance and availability requirements defined in SLAs. Thus, an analyst needs to identify which is the configuration arrangement most optimized considering the defined requirements in order to achieve the desired performance levels.

Let us now describe the availability evaluation process for convergent networks.

Analysts should limit the availability evaluation described in the methodology to the availability of the communication links. They should not take into consideration the availability of the hardware involved, as it does not vary depending on script implementation or protocols association. To do this, it is necessary to perform experiments in order to collect the failover time. Analysts should instrument the network equipment, specifically routers, so that they can simulate a link failure and register the time required for the critical network services to be available again.

Metrics such as CPU should be considered by analysts, but are part of the performance evaluation that will be detailed in the next section. Below, we describe how analysts may evaluate availability on convergent networks.

For availability evaluations, the most practical method that the analysts can adopt is measuring the downtime. The downtime is the total time that a system or infrastructure is in a down status over a given period of time. Empirical downtime data can be extracted from the evaluated equipment's logs, which records the total time it has been down. Another more dynamic and practical way is by sending packets of another directly connected network equipment and measuring from the packet sender how long the evaluated equipment was unresponsive to the sent packets.

Parameters Definition: Considering that the configuration depends on the redundancy topology, protocols used, among several other factors, we propose the model considering a warm standby redundancy strategy. Leaving the parameter of the mo-

del abstracted, so that the model metrics can be compared with and without setting the configuration parameters. Analysts can choose and evaluate their redundancy systems based on the result provided by the model through the availability equation generated. This methodology is valid for comparing scenarios in which there exists a configuration arrangement, and then a change on the configuration is performed in order to evaluate its reflection on the system's availability.

Metrics Definition: Metrics definition is an important step in our methodology. The metrics that are chosen in this step support analysts on the decision-making process to choose an appropriate scenario for the redundancy mechanism in the network. This work considers availability metrics. More specifically, the availability metric we consider is the fail-over time that corresponds to the time required for the network traffic to use the contingency communication link available in the network. Fail-over time is the most adequate metric that analysts may use to support the scenario for the redundancy mechanism available in the network. In this regard, the response time to failure occurrences is expressed by the failover time, with the failure detection time being the time necessary to identify the failure plus the time required to change for an alternative route after the failure detection.

Markovian Modeling: Our CTMC-based modeling strategies can represent the corporate network and the contingency activation (backup) process. Using our model, it is possible to estimate the availability and downtime of a network. The network modeling strategy can represent the network and the redundancy mechanism that is in place. In addition, it represents the traffic being switched to the contingency approach in case of failure occurrence. The configuration parameters are abstracted for the model. It is necessary to consider that the proposed Markovian model represents a part of the corporate network, that is, the data communication link that connects a business unit to the concentration environment and its respective contingency. From this model, considering that the part of the network represented is repeated N times it is possible to perform the calculation of the metrics through the derivation of binomial equation for the entire system (N communication links).

Measurement Planning: This activity defines the beginning of the measurement process. Measurement planning comprises the step when analysts should define how the determination of the fail-over time should be established. At this stage, based on the resources and tools available to the evaluators, analysts should decide one of the following options as a start point. That is, they need to decide whether they must to consider the logs of the network assets connected to the source link in the measurement, to use the data of any SNMP-based management tools, or to measure ICMP packets responses directly in the network. In the last case, the fail-over time

is being represented by the period of time from which the business unit router stops to respond to the pings fired from the concentration environment and it continues until detecting a first successful response. Measurement planning also defines how many measurements are taken and how data will be tabulated for further analysis. This is especially important to be taken into consideration the more the links that are concentrated on one node.

Considering that the methodology is intended for critical systems and that simulations of fails imply unavailability, analysts must perform measurements in scheduled maintenance windows when necessary in order to not affect the system's operations. Analysts need to document the measurement activities that will be performed by technicians. To do this, they must document the commands that need to be executed in an script in order to simulate the failure of the main link (simulating the fiber optic rupture mechanism), which takes place by configuring administratively the router interface (physical or virtual) with a down status and which are the commands to be executed to undo this change after measurement. Technicians after executing this command must register the time required for the communication to be restored via the redundancy link. After that, it is necessary to remove the down status associated with the physical or virtual interface and then repeat the procedure on another link.

Data Collection: In this step, following the documentation generated in the Measurement Planning activity, the technicians must perform the tasks exactly as defined before. It is important to highlight that repeatability should be considered by the technicians in order to obtain results with the lowest level of distortion. That is, successive measurements should be carried out with no changes in the measurement conditions in order to obtain results that offer statistical confidence. After the data collection, analysts should check whether the measurement steps defined before were followed as defined. A new data collection should be carried out in the case that any errors were found in the execution process.

Data Analysis: In this step, the data collected for the evaluated configuration scenario is analyzed and, in this process, it is possible to verify the degree of confidence of this data. Analysts should note whether the amount of data obtained is sufficient to ensure that they represent the actual system. Obviously, as we are evaluating a converged network, the fail-over time metric obtained by using the model may differ among links, as it is associated with some network parameters such as latency, which in turn depends on the physical distance between the related peers. Taking this into account, the validation of the collected data considers a confidence interval in relation to the metric evaluated. There are some statistical methods that may be used in the validation process. In this work, we propose to use in our metho-

dology a non-parametric test on the equality of continuous and one-dimensional probability distributions that can be used to compare a sample by considering a reference probability distribution, known as the Kolmogorov-Smirnov test (JR, 1951). In the special case of testing the normality of a given empirical distribution, samples are standardized and compared with a standard Gaussian distribution. In turn, the Kolmogorov-Smirnov test can be performed using some statistical tools such as EasyFit (MEHRANNIA; PAKGOHAR, 2014) or Statdisk (TRIOLA et al., 2005). By using these tools, it is possible to demonstrate whether the empirical data follows a normal distribution or not. Also, these tools make it possible to verify if the mean and standard deviation values are valid as measures of central tendency and dispersion, or, in a negative case, indicate that it is necessary to perform some statistical treatment in the evaluated data. The empirical distribution may be represented by histograms, being them drawn using the Statdisk tool (TRIOLA et al., 2005), for example. In this regard, the frequency in which the different failover times occur can be plotted, being considered as a baseline scenario for the experiment. Once the sample has been validated, the analyst can consider the fail-time time obtained in the baseline scenario as the default value to be reduced in a further improvement scenario that consists of changing in some configuration parameters.

Statistical Treatment: By considering that there were no failures in the measurement process, in the case that the empirical data does not pass in the Kolmogorov-Smirnov test, it is possible to apply the bootstrap method (EFRON, 1979) instead. The bootstrap can be an effective method to be adopted in many statistical inference problems. As an example, we can cite the construction of a confidence band in a non-parametric regression test, testing for the number of modes of a density, or the calibration of confidence limits.

Data Storage: This is the last activity of the measurement process. At this stage, analysts should tabulate data into a database, along with the history and registers of the measurement process. In the case that the measurements refer to data extracted from the configuration baseline, analysts should store the data for comparison with the data that will be obtained from the scenario by changing the configuration parameters that will be developed in the *Configuration Evaluation Process*. If the measurements refer to the data obtained in the scenario after changing the configuration parameter, the data is tabulated and compared with the data obtained in the configuration baseline. This comparison allows analysts, based on the results of the fail-over metric, to calculate the network's availability and to evaluate whether the availability gain obtained between the scenarios is relevant and to finish the evaluation of convergent network availability.

Configuration Generation: In this step, the environment settings, both concen-

tration and routers in the business units, are elaborated by the analysts. This work considers the association of BGP with BFD protocols in order to produce increase in the network's availability. This implementation in BGP networks should be evaluated individually given the specificity of each backbone, as it is linked to the intrinsic characteristics of the network and the services that it supports. Among the precautions that analysts responsible for implementing high availability projects should observe, we can cite two. One of them is the interoperability of the network equipment from different manufacturers. The other one is the capacity limits that may impact the deployment of a large scale solution. Analysts must follow the references of the BGP and BFD protocols in order to generate the configurations. These references are their RFCs and also the structure and syntax of the commands related to the implementation of the BFD protocol in the equipment platform installed in the company's backbone. These syntaxes can be obtained in the documentation provided by the equipment' manufacturers. The evaluations in this work were carried out by using Huawei devices, but recently all router manufacturers have equipment that supports the BFD protocol.

Evaluation Environment: The proposed methodology aims to be implemented in corporate networks that offer critical services. By taking this into account, analysts must first implement the generated scripts in an evaluation environment. So that the changes performed do not cause eventual unavailability in the production network in the case that there exist errors in the scripts. Today, the most known network device manufacturers offer simulation software that lets anyone emulate real network devices, by considering the device model, its operating system version, and its configuration files. Therefore, in order to validate the settings generated in the *Configuration Generation* activity, and also to ensure the effectiveness of the commands to be used in the real network, firstly analysts must run the necessary tests in a virtual environment. In the case that errors are identified, the *Configuration Generation* activity will be carried out again in order to that some adjustments must be executed. In this work, all applied configurations were initially implemented in a simulation environment using the eNSP(Enterprise Network Simulator) tool (HUAWEI TECHNOLOGIES CO, 2019a). The eNSP is developed by HUAWEI that is the manufacturer of the routers used in our production environment.

Network Deployment Planning: This step comprises the stage at which analysts should establish how to apply the configuration that offers higher availability (deployment) with the association of BGP and BFD protocols. Here, analysts plan to apply the configuration scripts to the same link samples that were defined in the *Measurement Planning* process for the baseline scenario. This will further allow a more consistent comparison between the fail-over times and the availability itself between

the baseline scenario and the scenario that offers higher availability. Again, analysts should consider the existence of need to do the changes in the network's configuration in the maintenance window of the evaluated network. They should be careful to document, besides the configuration, the commands required to be performed to validate the configuration in order to test whether the BGP and BFD sessions are active and functional. Again, reference to such commands can be obtained by consulting the documentation of the devices' manufacturer.

Deployment: This is the last activity of the *Configuration Evaluation* process. In this step, the analyst must perform the deployment by following the documentation generated in the *Configuration Generation* activity as well as the plan produced in the *Network Deployment Planning* activity. Analysts should perform the validation of the configuration by using the test commands defined in the documentation generated in the *Network Deployment Planning* activity. After that, the *Measurement Process* is started again so that the measurements are taken in the environment with the configuration that offers higher availability.

4.2 METHODOLOGY FOR EVALUATING PERFORMANCE OF CONVERGENT NETWORKS

In this section, we present the methodology for supporting performance evaluation of convergent networks in order find out the processing load associated with the higher availability solution presented in Section 4.1. This methodology comprises a set of well-defined steps for supporting the evaluation process. The steps of the methodology are as follows: *Understanding the Concentration Environment*, *Parameters Definition*, *Metrics Definition*, *M/M/1/K Modeling*, *Model Validation*, *Model-Experiment Refinement*, *Measurement Planning*, *Measurement Configuration Generation*, *Deployment*, *Data Collection*,*Data Storage and Calculation*, *Metrics Evaluation*, *Configuration Generation*, *Evaluation Environment*, *Network Deployment Planning*, and finally *Deployment*.

Figure 4 depicts the steps that analysts must follow in order to evaluate their networks. *"Configuration Evaluation Process"* is a subprocess that is considered after the modeling strategy is validated.

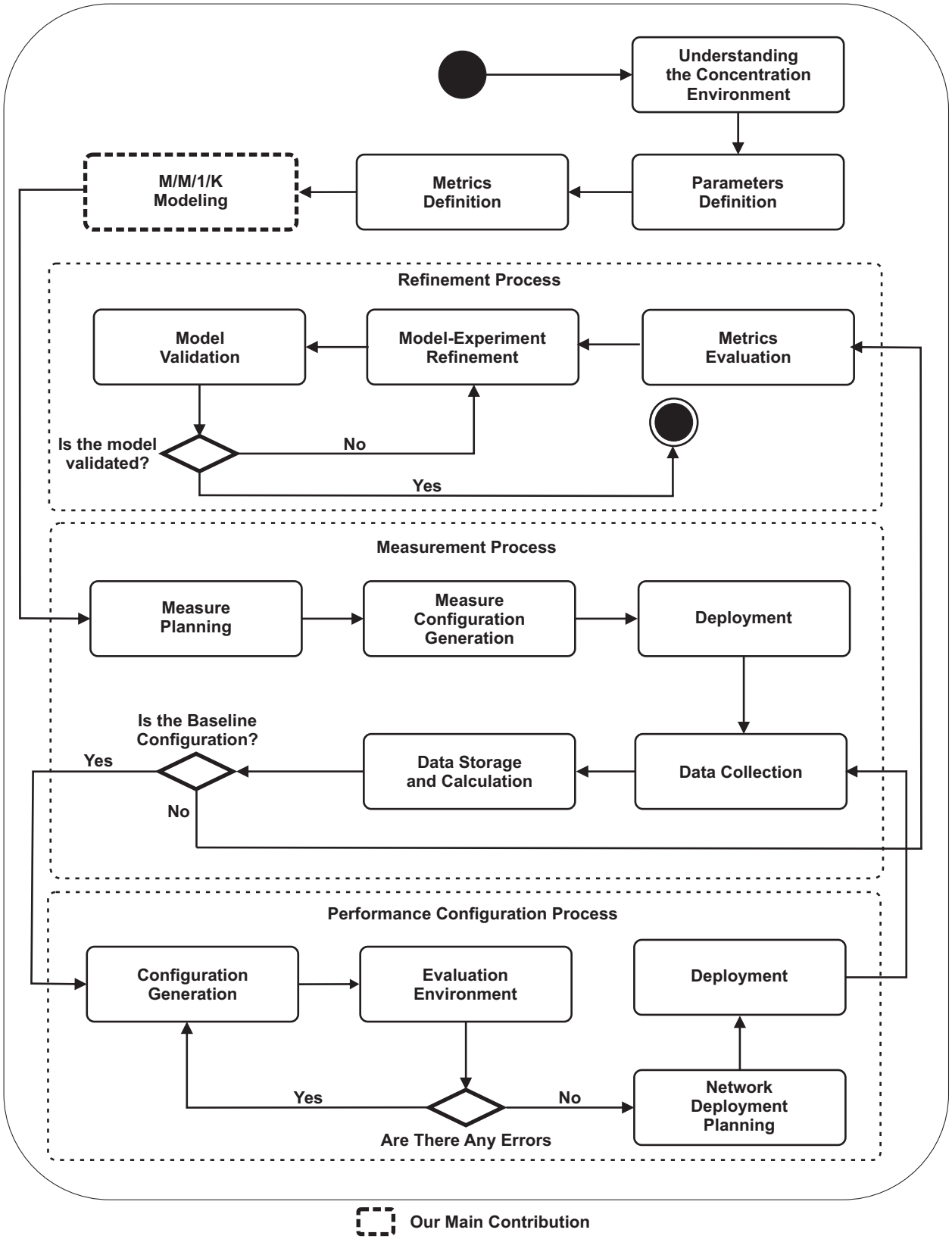

Figura 4 – Evaluating Performance of Convergent Networks

Understanding the Concentration Environment: Analysts should understand

how the network concentration environment works. Understanding the concentration environment means understanding the nature of the workload assigned to the concentrating asset. In order to understand the workload, analysts need to know the network topology and the configuration baseline. It is important that analysts understand the vital importance of the concentrating device for the internal routing functionality and interaction with external autonomous systems. Also, analysts must be aware that the high availability configuration by using the BFD protocol necessarily implies an increase in the computational processing for reasons intrinsic to the control mechanism. Is this increment in the computational processing related to the availability solution intended to be evaluated in this methodology. In addition, it is essential that the analyst studies in detail the processing functionality of the concentrating equipment. One of the most important inputs of this step is the manufacturer's documentation. By using it, the analyst can obtain both information related to the packet processing mechanism and the capacity limitations that must be considered in the queue model to be built.

Let us now describe the performance evaluation of converged networks. Analysts need to evaluate the performance of the concentration environment in each scenario, both the base scenario and the scenario that offers higher availability by associating the protocols BFD and BGP. This requires collecting the average CPU consumption of the concentrator device assigned to handle the common device routing tasks. One resource that analysts can use is an SNMP-based management tool that can read the Management Information Base (MIB) of the concentrator device, and collect the mean CPU consumption before any change is made in the configuration of the BFD protocol. In this work, we used a tool named Spectrum (CA TECHNOLOGIES, 2019). After performing the experiment to change the BFD configuration, collecting new data regarding the CPU usage allow analysts to infer what is the CPU consumption by using the protocol. In addition to the CPU metric, the methodology also provides for correlating the number of BFD packets processed during the observation period. This last one will serve as the basis for constructing the M/M/1/K queue model in order to establish the limit of BFD sessions in the concentration environment that will no cause bottleneck or fails in the redundancy mechanism.

The methodology considers performing the performance evaluation taking into account an M/M/1/K queue. A first premise that needs to be adopted in the methodology for performance evaluation is that in order to evaluate an availability solution, the analysts have to incorporate into your methods the characteristics of the chosen solution. Since the protocol BFD uses *keepalive* packets, these packets cannot be queued as this could generate false positives when the timed out associated with each echo packet is reached. Therefore, we recommend that analysts do not use the buffer as a resource limiter, as it would not make sense when using the availability

solution. Instead of the buffer parameter, we suggest to use the *forwarding performance* parameter, which represents the maximum packet rate that the equipment can process and forward before it starts to queue the packets. Analysts can obtain this parameter from the documentation provided by the concentrator devices' manufacturers. Another relevant point to ensure the validity of the methodology is to analyze the concentrating equipment directly in order to verify how the mechanism for packet processing works and to understand the logical architecture of the equipment. Another important starting point for the evaluation is to study in the documentation on how to enable statistical reporting of specific BFD sessions. In order that incoming packets in the concentration device are counted during the execution of the experiment, for example.

Parameters Definition: In this step, we define the parameters that will support the modeling process. Each parameter is a characteristic of the network or a configuration that affects its performance. This work considers as parameters the application of configurations associating the protocols BFD and BGP. Thus, the methodology for performance evaluation considers the effect of these configuration parameters on the network performance.

Metrics Definition: The metrics that are chosen in this step support analysts in the decision-making process to choose an appropriate configuration to deploy their convergent network, linking availability to a stable environment. We consider the performance metrics for an M/M/1/K queue according to the concepts described in Chapter 2 and which are tabulated in Table 2. Analysts can use the equations to evaluate the impact of BFD control packets on the environment. Obviously, the greater the number of network BFD sessions, the closer to a threshold the environment approaches. It is important to note that, in order to use the equations, it is necessary to perform in the device measurements regarding CPU consumption in order to find out the utilization percentage before and after the implementation of the chosen BFD solution. In other words, analysts need to find the difference between the mean CPU consumption before and after the change of the BFD configuration is applied. Only in this way it is possible to ensure that the defined metrics are associated with the BFD control packets in the environment.

Tabela 2 – Metrics for M/M/1/K BFD control packets

Metrics	**Values for $\rho \neq 0$**
Utilization	$\frac{\left(\rho\left(1-\rho^{k}\right)\right)}{(1-\rho^{k+1})}$
Discard Probability	$\frac{\rho^{i}(1-\rho)}{(1-\rho^{k+1})}$
Discard rate	$\lambda\Pi_{k}$
Mean Response Time	$\frac{\frac{\left(\rho\left(1-(k+1)\rho^{k}+k\rho^{k+1}\right)\right)}{(1-\rho)(1-\rho^{k+1})}}{\lambda(1-\Pi_{k})}$

M/M/1/K Modeling: In this step, analysts should use the M/M/1/K model to facilitate the representation of the proposed solution. This applies even to whether the solution is not a classic queue model, as we have explained before. It is important to note that analysts can use our modeling strategy to support resource consumption estimates in a scalable environment. To predict the behavior of the scalable BFD utilization, analysts should define the model K based on the *Forwarding Performance* parameter that is available in the device's documentation provided by its manufacturer.

Although the equations in Table 2 are sufficient to represent the model M/M/1/K, we have used the software Mercury (SILVA et al., 2015) to support us during the definitions and validations phases of our modeling strategies. Mercury is an integrated environment for supporting performance and dependability metrics evaluation of general systems and it allows users to define their models graphically.

Model Validation: Analysts should validate their models to make sure the metrics obtained by using them represent the results that the actual system would present by having the same parameters and processing the same workload. Obviously, since we are evaluating a stochastic system, the metric obtained from the model may differ from the metric of the actual system within a predefined limit. The validation process we suggest in this methodology is to establish a value comparison between one of the metrics defined, such as utilization. Thus, regarding the metric chosen, comparing the values obtained by measurement in the environment and calculated through the equation presented in Table 2.

Model-Experiment Refinement: When metric values differ, it means that it is necessary to refine the model to more accurately represent some system activities or, in some cases, refine the experiment executed on the actual system in order to collect more detailed information. After that, the experiment is executed again and the metric values are compared. This process is repeated until the evaluated metric is statistically equal, that is, within the confidence interval defined by the analyst. Validating a model allows an analyst to check the behavior of complex scenarios by using it. This provides confidence in the values obtained from it.

Measurement Planning: This activity defines the beginning of the *Measurement Process*. Measure planning comprises the stage at which analysts should define how the measurement of the mean CPU consumption of the concentrator device will be performed and what SNMP tool should be used in this process. As we aforementioned, we have used the tool Spectrum. Once properly configured, the management server communicates with the concentrating device every five minutes in order to read information from its MIB. Through it, analysts can know the mean CPU consumption in the baseline scenario as well as after to change to the solution with

BFD. It is also in this stage that analysts should research how to extract the statistical report from BFD packets arriving at the concentrating device. A plan and the observation time are the output of this step.

Measurement Configuration Generation: In this step, analysts must document operational procedures regarding CPU data collection, such as how to configure the counter of BFD packets in the concentrator.

Deployment: In this step, the measurement settings are deployed by considering the output of the previous activity. Obviously, at first, measurements must be performed in the baseline scenario. In this case, the BFD statistical settings will not have data to be measured.

Data Collection: In this step, analysts extract from the environment both the average CPU consumption by using the SNMP tool and, for the same observation period T, the total number of BFD control packets that have arrived at the concentration router for all active BFD sessions on the network. This last value will be zero in the case that the measurements should be performed on the baseline scenario.

Data Store and Calculation: In this step, the data regarding the mean CPU consumption, as well as the number of BFD packets that have arrived at the concentrating router must be stored in a database for further analysis. If this data is based on the configuration baseline scenario, only the data is stored and then the *Performance Configuration Process* is started. If data is already taken from the scenario using BFD, then a calculation is required to be performed. This calculation consists of subtracting the mean CPU consumption from the scenario that offers higher availability from the mean CPU consumption of the baseline configuration. It is required in order to make possible to associate a consumption delta considering the number of packets that have entered the concentrator during the observation period. Finally, this lets analysts move on to the next activity which is *Metrics Evaluation*.

Metrics Evaluation: In this last step of the *Measurement Process*, an analyst analyzes the metrics provided by the model. The performance metrics evaluated on the convergent network are *utilization*, *discard probability*, *discard rate*, and *mean response time*. These results guide the analysts in planning how to deploy the availability solution in the network. If the metrics obtained do not meet the level of performance and/or resource consumption expected by analysts, a possible upgrade of the concentration hardware should be evaluated. If the analysts choose to replace the concentrating device, they may analyze which device would have the appropriate capacity in order to meet the requirements of the solution that offers higher availability. The metrics are evaluated again in the case that replacement is performed.

Configuration Generation: The *Performance Configuration Process* is identical to the *Configuration Evaluation Process* defined in the methodology presented in Section 4.1. That is, both processes have the same activities as we can see by looking at the Figures 3 and 4. This occurs because, after the initial measurement phase is completed, analysts must configure the BFD in the environment and go back to the *Data Collection* activity. For further information about this process, see the step *Configuration Generation* in Section 4.1.

Evaluation Environment: For further information about this activity, see the step *Evaluation Environment* in Section 4.1.

Network Deployment Planning: For further information about this activity, see the step *Network Deployment Planning* in Section 4.1.

Deployment: For further information about this process, see the step *Deployment* in Section 4.1.

5 ARCHITECTURE AND MODELS

A convergent network that offers critical services should have the shortest possible downtime and it needs to keep itself stable. This chapter shows the convergent network architecture that we consider in this work. We dedicate the *Baseline Architecture* section to present the network architecture and its backup mechanism we have considered. Considering that the developed solution does not totally change the network architecture, we dedicate the *BFD Session Process* section, which presents the solution in detail. In this chapter, we present two modeling strategies: being the first strategy based on Markov chain for the availability evaluations and the second one based on the M/M/1/K Markovian queue model for supporting networking performance evaluations. Thus, the chapter also has two sections dedicated to the models, one section named *Availability Model* and another one named *Performance Model*.

5.1 BASELINE ARCHITECTURE

This section presents the corporate network that we used during the testbed experiment. In addition, we also present the baseline architecture that resulted in our availability model.

Firstly, in order to understand the baseline architecture, it is necessary to have the definition of the two main elements of the network: the concentration environment and business units (BUs). The concentration environment is responsible for two main activities: internal and external routing, and connecting BUs to internal application and service servers. In turn, the BU is where the end-user, who actually uses the network services, is located and who experiences the availability and performance issues of the network. It is in these distributed remote units that the company's business takes place. However, it is important to highlight that there is no computational processing in the BUs. All performed operations and transactions need to reach the network servers through the network backbone, what necessary implies that they need to pass through the concentration environment.

The corporate network used in this study is a highly critical environment with hundreds of communication circuits, each one connecting a business unit to the concentration environment. The baseline architecture of the main circuit is based on the connection between the local router and the regional concentration environment, using the metropolitan optical ring network provided by a telecommunications provider, as shown in Figure 5.

For each business unit router, a WAN link address (/30) is allocated, with the even side addressing the unit router interface (or sub-interface), and the odd side addressing the

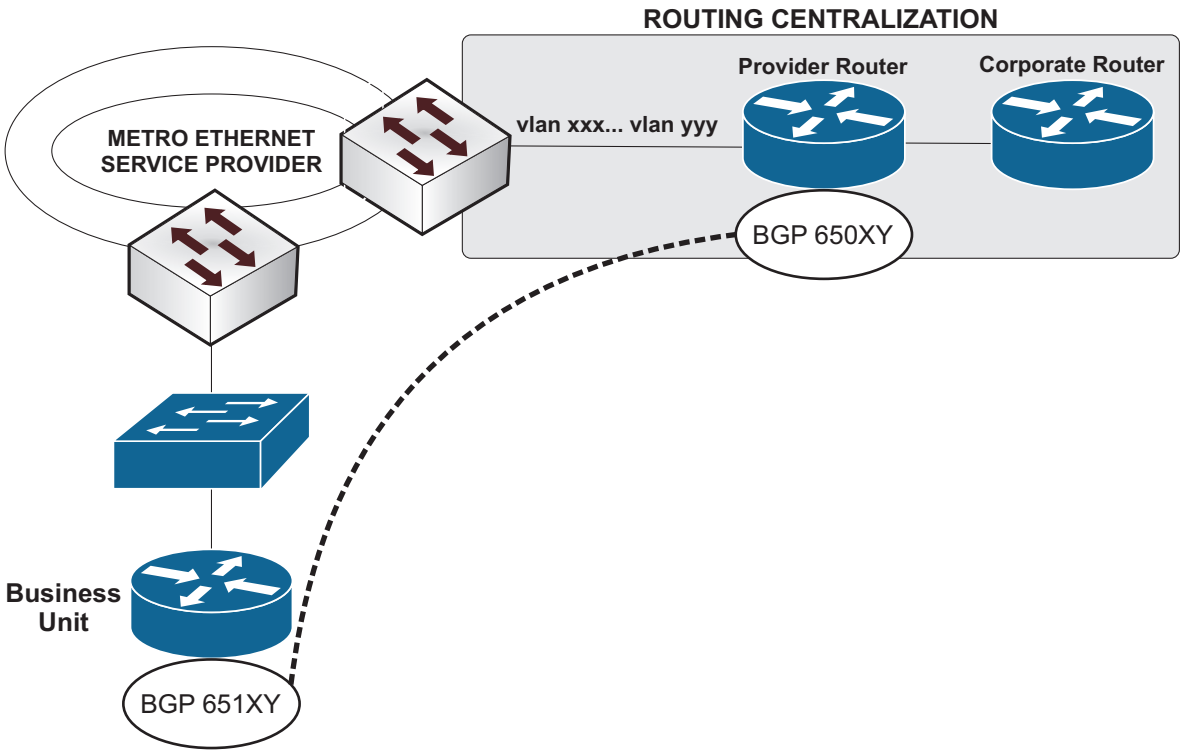

Figura 5 – Topology of the Main Link (SIQUEIRA et al., 2019)

sub-interface of the business unit concentrator router. Due to the issues of port number limitations, common to concentration environments, the concentrator uses interface in a trunking mode.

It is important to note that the network has a platform standardization, aiming to avoid interoperability problems between devices from different manufacturers whilst facilitating the management process of the configuration baseline. The current environment is composed of these following equipments:

- HUAWEI AR1220, as the platform of the BU; and
- HUAWEI NE40, as the platform of the concentration environment.

The baseline architecture establishes the configuration and activation of the BGP dynamic routing protocol between the corporate network's router and the telecommunications provider's router. BGP, also known as External Gateway Protocol (EGP), is a protocol used for inter-AS communication. That is, for communication between autonomous systems (ASs). By looking at the Figure 5, we can see the protocol in action in the interconnection between the AS. The main aim of the BGP in our baseline architecture is to provide a cross-domain routing system that ensures a loop-free exchange of routing information between ASs. The announcement of BGP routing information between routers (concentrator device and devices both presents in business units) that establish a neighborhood relationship, which we call here *peers*. For an adjacency establishment to

occur, the baseline architecture uses the peer-to-peer connection between the hub and BU router to ensure reachability.

For each BU is created an Autonomous System (AS) ID <BGP ID> and an ID for community BGP <ID COMMUNITY>. These IDs are used in the configuration of each BU router linked to the concentration environment. A BGP community is a group of destinations that share a common property. Community information is included as a path attribute in BGP update messages. This information identifies community members and enables analysts to perform actions on a group without having to elaborate upon each member. Analysts can use community and extended communities attributes to trigger routing decisions, such as acceptance, rejection, preference, or redistribution.

According to the baseline configuration, settings of the metropolitan network, with regards to BGP protocol, is established with EBGP route preference, IBGP route preference, and local routes preference.

A series of commands must be considered when configuring BGP in the concentration environment and in the BUs. By considering the side of the concentration environment, commands are used in order to include each neighbor in the concentrator router's BGP configuration as shown in Figure 7.

```
!
interface [GigabitEthernet | Eth-Trunk] X.<ID_VLAN>
 vlan-type dot1q <ID_VLAN>
 description Rede4 - <CCG> - <Nome PV>
 ip binding vpn-instance REDE4_PRINCIPAL
 ip address <End. Link WAN Principal - IMPAR> 255.255.255.252
!
```

Figura 6 – Configuration for the Concentrator Interface

```
!
bgp 65050
!
ipv4-family vpn-instance REDE4_PRINCIPAL
!
peer <End. Link WAN - PAR> group REDE4_PRINCIPAL
peer <End. Link WAN - PAR> description *** Hostname PV ***
peer <End. Link WAN - PAR> connect-interface [GigabitEthernet | Eth-Trunk]
X.<ID_VLAN>
!
```

Figura 7 – Configuration for the Concentrator BGP

On the other hand, in routers of distributed business unit, prefix-lists are created for advertising BU network addresses through the backbone.

Usually, there are a large number of routes in a BGP routing table. Transmitting a great number of routing information heavily burdens the devices. To address this issue, we configured routing policies (see Figure 9) in BU devices to advertise only the routes

```
!
ip ip-prefix REDES_OUT index 1 permit <Rede Bancária> 24
ip ip-prefix REDES_OUT index 2 permit <Rede Escritório> 24
ip ip-prefix REDES_OUT index 3 permit <Rede Penhor> 28
ip ip-prefix REDES_OUT index 4 permit <Rede Multicanal> <28|29>
ip ip-prefix REDES_OUT index 5 permit <Interface Loopback 0> 32
!
```

Figura 8 – IP prefix list

that they need to advertise, or routes that their peers require. Multiple routes to the same destination may exist and traverse different ASs. Routes to be advertised need to be filtered in order to direct routes to specific ASs.

```
!
route-policy REDE4_OUT permit node 10
 if-match ip-prefix REDES_OUT
 apply community <ID_COMMUNITY>
!
route-policy REDE4_IN permit node 10
 if-match community 100
!
```

Figura 9 – Routing Police

Establishing BGP adjacency between the concentration and BUs depends on the inclusion of the BGP peer in each of the remote routers

```
!
bgp <ID BGP>
 router-id <IP Loopback 0>
 import-route direct
 undo synchronization
 group REDE4 external
 peer REDE4 as-number 65050
 peer REDE4 route-policy REDE4_IN import
 peer REDE4 route-policy REDE4_OUT export
 peer REDE4 advertise-community
 peer REDE4 ebgp-max-hop 2
 peer <End. Link WAN Principal - IMPAR> group REDE4
 peer <End. Link WAN Principal - IMPAR> connect-interface [Fast|
Gigabit]EthernetX/Y.[25|ID_VLAN1]
!
```

Figura 10 – BGP Configuration for Business Unit

Figures 11 and 12 show information about the evaluated network environment.

The main circuit is deterministic, which is usually more expensive. The backup circuit is statistic, utilizing MPLS technology in diverse operator networks. According to the main characteristics of each network (bandwidth, latency, and availability), a priority matrix was determined, in which the metropolitan circuits have a preference over MPLS

```
# display bgp peer

BGP local router ID : 10.32.99.213

Local AS number : 65131
Total number of peers : 3              Peers in established state : 2


Peer              AS  MsgRcvd  MsgSent OutQ  PrefRcv   Up/Down  State
10.32.99.214    65002        0       0    0         0  0425h03m  Active
10.32.99.221    65050     5382    5230    0         1  0088h32m  Established
10.32.99.225    65050    25775   20780    0         1  0427h03m  Established
```

Figura 11 – BGP Communication Tests of the Business Unit (SIQUEIRA et al., 2019)

```
VPN-Instance REDE4_PRINCIPAL, Router ID 1.32.0.7:
Total number of peers : 23                 Peers in established state : 22


Peer           V    AS MsgRcvd  MsgSent OutQ   Up/Down        State PrefRcv
10.32.99.222   4 65131    5189     5389    0  0087h50m  Established       7
10.32.127.38   4 65131     867     1078    0  16:34:41  Established       5
10.32.127.158  4 65131    5325     5440    0  0088h41m  Established       4
10.32.142.222  4 65131    4516     5383    0  0087h44m  Established       6
10.32.145.222  4 65131    5050     5416    0  0088h17m  Established       8
```

Figura 12 – BGP Adjacencies in the Concentration Environment (SIQUEIRA et al., 2019)

circuits. In the present scenario, the networks use the BGP routing protocol and operate in an active-active system, i.e., all networks are simultaneously active, with the priority of each network being defined in the concentration by using the BGP parameter "*prepend*".

When is detected the break of the main circuit from the BU, a tunnel interface is activated, with the tunnel becoming the BU's access interface to the internal network and using the backup network, as illustrated in Figure 13. The baseline architecture for VPN configuration uses the operator's CPE router to close the IPsec tunnel, thereby ensuring the confidentiality and integrity of the traffic.

On the other hand, corporate network routing information remains on the BU router, it being transported through the GRE tunnel and OSPF dynamic routing process, and it is visible only to the corporate network analysts. The analysts have remote access to the CPE router by using SSH protocol through the loopback IP address provided by the telecommunication provider. The CPE router provided by the telecommunication provider is configured for making it possible to enable the IPSec connection on the VPN concentrator and performing the configuration of the tunnel on the GRE concentrator. This setting is indispensable for the operation of the backup communication link.

By applying static or dynamic routes that offer better costs, the formation of the tunnel is directed to the main network by blocking traffic of the concentration backup circuit through the access list. In this case, the formation of the tunnel is blocked and

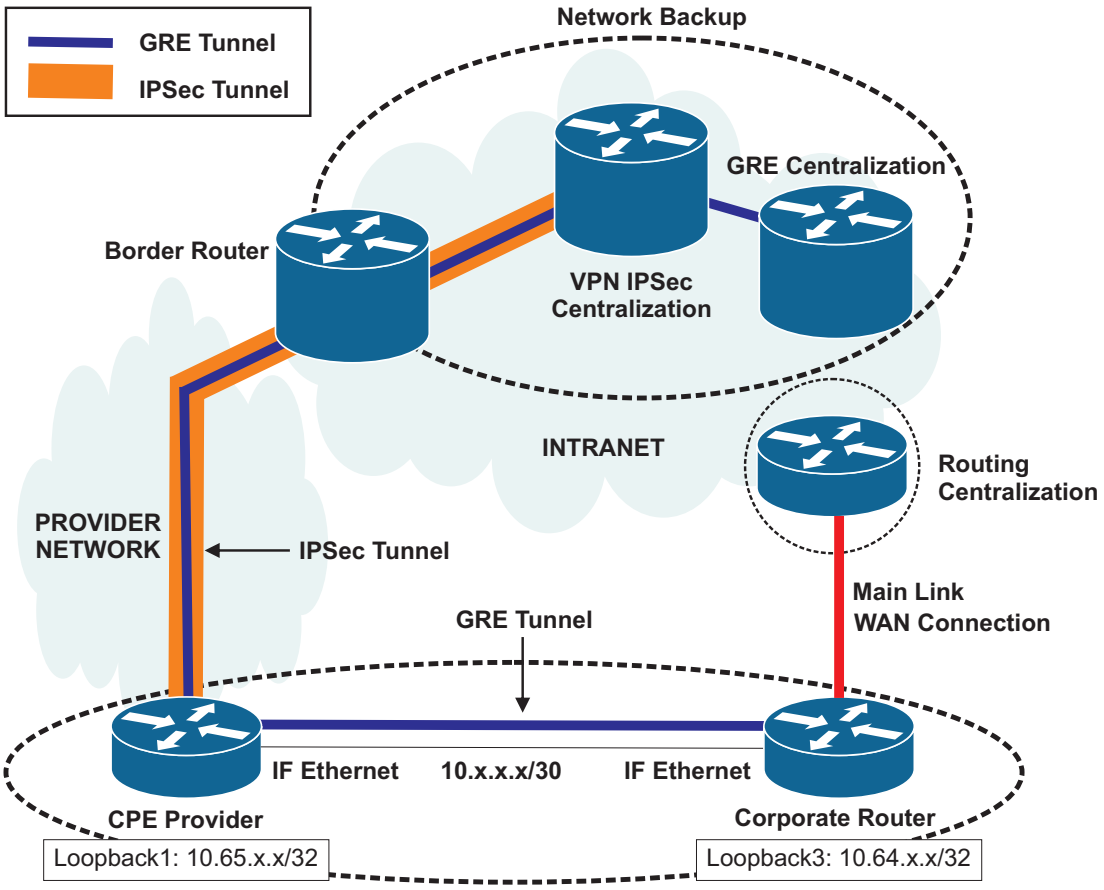

Figura 13 – Backup Network (SIQUEIRA et al., 2019)

the circuit remains in a down status. When the metropolitan network is unavailable, the only route remaining to form the tunnel of the backup network is the route pointing to the own circuit of these networks. In this case, the tunnel will be formed, as shown in Figure 15.

The problem experimented in this critical system is the convergence time, that is, when is high the unavailability time until the communication via the backup circuit is established. It is important to clarify that while baseline architecture is effective in producing a high availability network, slow convergence is still a point of attention that deserves the application of all effort necessary to overcome it, given that any interruption in the services implies a bad experience for the end-users and, as consequence, for the business incomes. It should be noted that the convergence problem observed in the baseline architecture is associated with the failure detection time. Furthermore, it is important to highlight that, eventually, a linkage can exhibit oscillations; thus, without a definite rupture, but with damage to the quality of the circuit, with a subsequent sequence of UP/DOWN alterations. These oscillations in a slow-convergence environment can lead to a bad experience to the end-users of the service.

In addition, it is important to note that we are considering a hierarchical network, with several remote units being physically and logically connected to a concentration

```
#
hostname <UF9999RAN>_OI
!
ip domain name <uf>.caixa
!
ip host <Hostname do Concentrador VPN> <End. IP do Concentrador VPN>
!
crypto keyring K_VPN
  pre-shared-key hostname <Hostname do Concentrador VPN> key <SENHA>
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
!
crypto isakmp profile PROF_VPN
   keyring K_VPN
   self-identity fqdn
   match identity host domain coresp.caixa
   initiate mode aggressive
!
!
crypto ipsec transform-set VPN esp-3des esp-sha-hmac
!
crypto map VPN_INTER_AG local-address Loopback1
!
crypto map VPN_INTER_AG 10 ipsec-isakmp
 set peer <End. IP do Concentrador VPN>
 set transform-set VPN
 set isakmp-profile PROF_VPN
 match address 133
!
```

Figura 14 – Configuration for the Telecommunication Provider's CPE

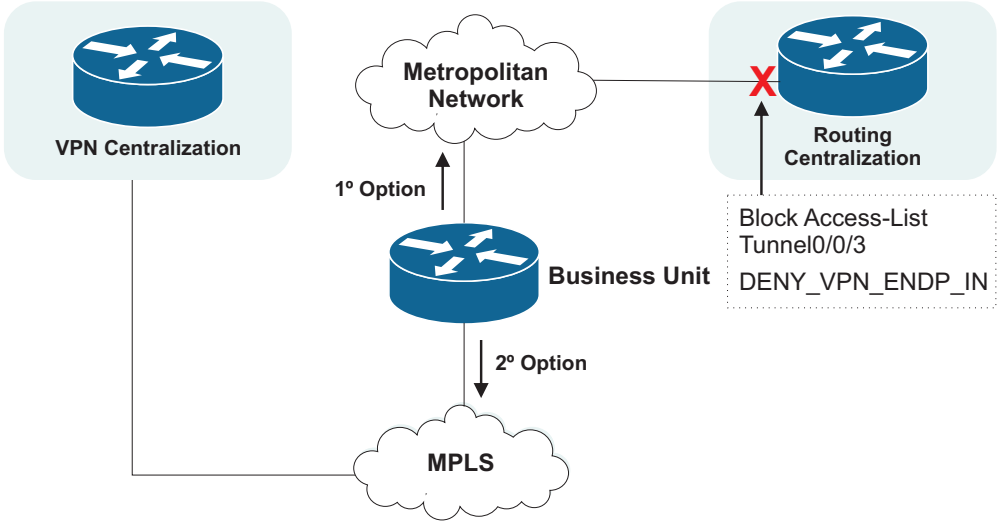

Figura 15 – Failure of the Main Link (SIQUEIRA et al., 2019)

environment, as shown in the schematic representation of part of the network presented in Figure 16.

The greater the number of communication links concentrated on the concentrating device, the greater the relevance of the convergence problem because it necessarily implies less availability for the critical system, considering the totality of communication links.

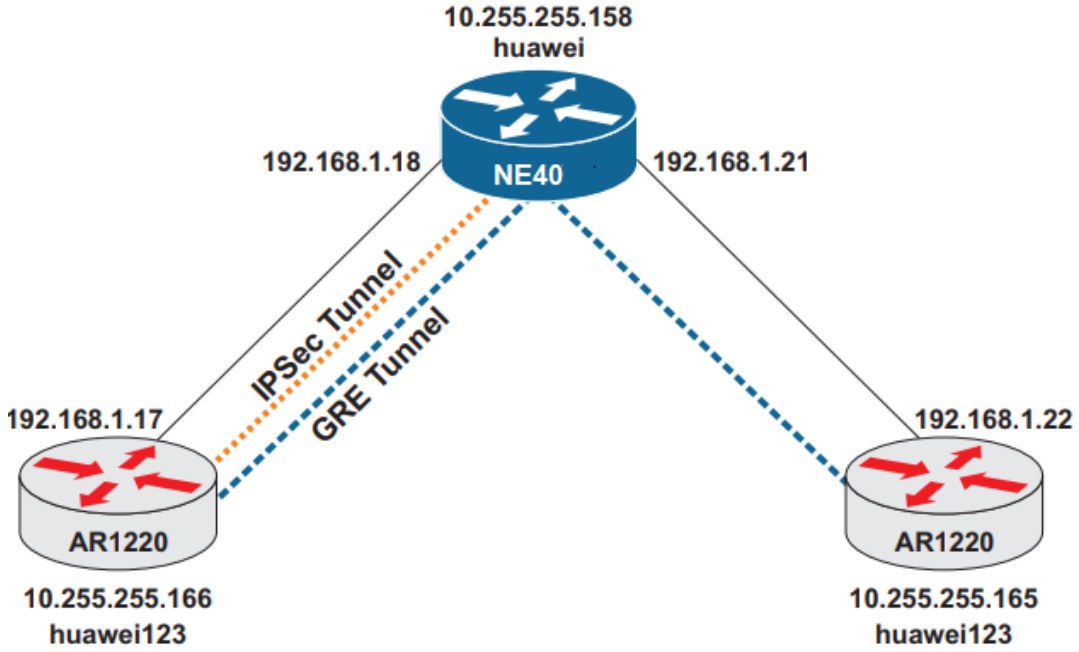

Figura 16 – Physical and Logical Concentration Environment (SIQUEIRA et al., 2019)

5.2 BFD SESSION PROCESS

This section is intended to introduce a new configuration and architecture scenario that is capable of overcoming the convergence problem reported in the *Baseline Architecture* section. Considering that the problem originates from the long failure detection time, the new architecture has been shaped to reduce this detection time, resulting in a reduction in both the fail-over time and network downtime and an increase in availability.

All BGP configurations applied in the baseline architecture, such as community-list, route policies, local prefix filters, BGP connection, and AS configuration are maintained in the latter scenario. Therefore, the results obtained considering changes in the configurations values in the latter scenario do not have a relation with the aforementioned parameters. Taking into account that their values are equals for both scenarios.

Due to the intrinsic characteristics of the BGP protocol discussed in the Chapter 2, networks use relatively slow "Hello"mechanisms. The time to detect failures ("Detection Times") available in the BGP protocol in the Baseline Architecture is too long for applications supported by the critical network and can represent a loss of data or transactions. Therefore, a more efficient failure detection mechanism must be deployed in the network by been associated with the BGP protocol.

In this context, in the new architecture, we chose to adopt the BGP protocol with the Bidirectional Forwarding Detection (BFD) protocol in order to provide low cost and short duration fault detection between adjacent BGP peers, one of the peers being the concentrating equipment and the other one being each router installed in the remote business units.

In the proposed new architecture, the BFD protocol was used in its asynchronous

mode (primary mode), where there is periodic sending of BFD control packets in both directions, from the concentration to the remote business unit's router and in the opposite direction. If the set number of consecutive packets are not received by the other device, the session is declared down.

The BFD session has the following statuses: Down, Init, Up, and AdminDown. The field State of a BFD control packet shows the session status. The system changes the session status based on the local session status and the received session status of the peer. The BFD state machine implements a three-way handshake for BFD session setup or deletion to ensure that the two systems detect the status changing. The following shows the BFD session establishment in our experiment to describe the state machine transition process.

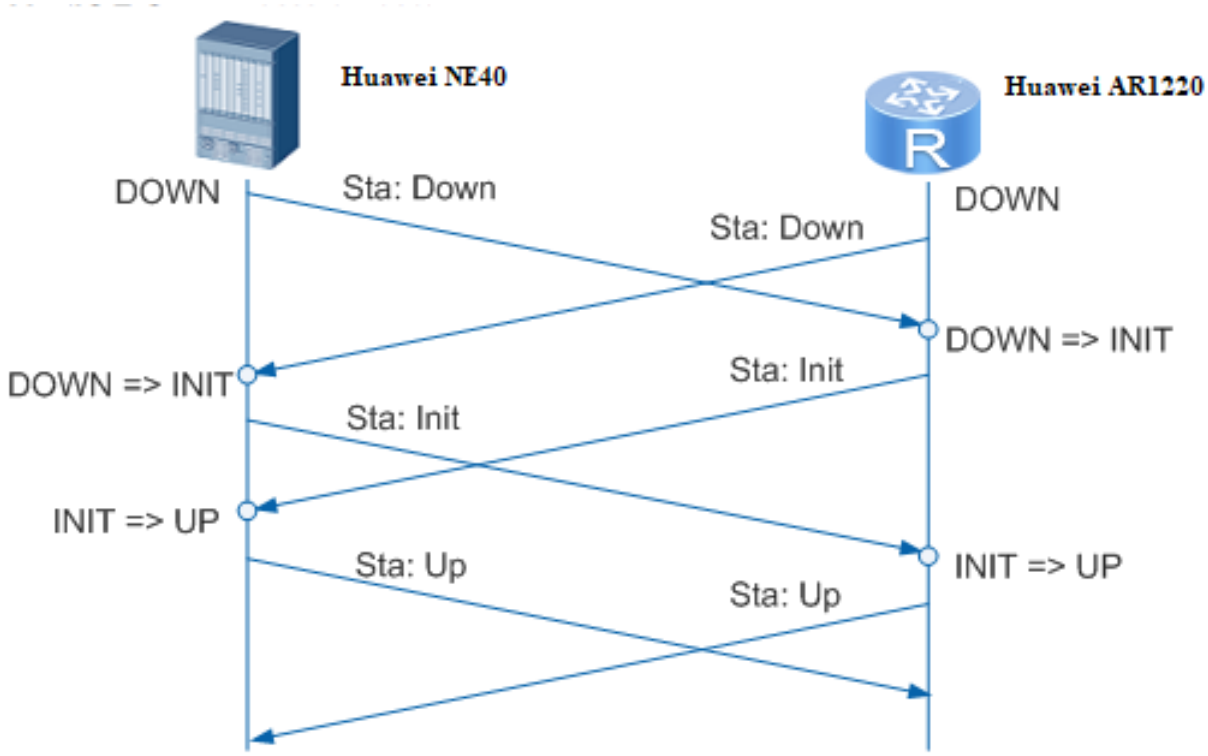

Figura 17 – BFD session setup

1. Huawei NE40 (Concentration Device) and HUAWEI AR1220 (Business Unit) each of them start BFD state machines. The initial status of BFD state machines is Down. Concentration Device and Business Unit send BFD control packets with the State field set to Down. BFD sessions are configured dynamically.

2. After receiving the BFD packet with the State field set to Down, the BU's router switches the session status to Init and sends a BFD packet with State field set to the value Init.

3. After the status of the local BFD session of BU's router changes to Init, the BU's router no longer processes the received BFD packets with the State field set to Down.

4. The change in the status of the BFD session on the Concentration Device is similar to the one on the BU.

5. After receiving the BFD packet with the State field set to the value Init, the BU's router changes the status to Up in the local BFD session.

Based on the implementation of the script shown below in Listing 5.1, it is possible to associate the BGP protocol with the BFD protocol and promote a significant reduction in the failover time. The configuration is applied in each business unit and concentration environment as shown in the script. The main parameters of the BFD architecture are presented in the configuration. They are the interval for sending BFD control packets and the number of unanswered packets that is sufficient to declare that the communication link is down. Thus, in that case, incurring in the switching to the contingency link. These parameters are declared through the command line.

```
### BGP SCRIPT CONFIGURATION ###


#### Concentration #######

bgp <AS_Number_Bank>

group BU_MPLS external
 peer BU_MPLS as-number <AS_Number_BU>
 peer BU_MPLS ebgp-max-hop 2
 peer BU_MPLS bfd enable
 peer <End_IP_BU> group BU_MPLS
 peer <End_IP_BU> connect-interface Tunnel0/0/10
 peer <End_IP_BU> bfd enable
 min-tx-interval 100 min-rx-interval 100
 detect-multiplier 4
 peer <End_IP_concentration> bfd enable


interface Tunnel0/0/10

 description Business Unit
 ip address <End_IP_Bank> 255.255.255.252
 tunnel-protocol gre
 keepalive period 10
 source LoopBack1111
 destination <End_IP_destination_BU>


interface LoopBack1111

 description EndPoint GRE
 ip address <End_IP_Loopback_Bank> 255.255.255.255
```

```
 target-board 4
 binding tunnel gre


######## Business Unit ##########

bgp <AS_Number_BU>

 group BANK_MPLS external
 peer BANK_MPLS as-number <AS_Number_Bank>
 peer BANK_MPLS ebgp-max-hop 2
 peer <End_IP_Bank> group BANK_MPLS
 peer <End_IP_Bank> connect-interface Tunnel0/0/104
 peer <End_IP_Bank> bfd enable
 peer <End_IP_Bank> min-tx-interval 100 min-rx-interval 100 detect-multiplier 4


interface Tunnel0/0/31

 description TUNNEL_GRE
 bandwidth 2048
 ip address <End_IP_BU_GRE> 255.255.255.252
 tunnel-protocol gre
 keepalive period 10
 source LoopBack13
 destination <End_destination_Bank>


interface LoopBack13

 description Interface Endpoint GRE
 ip address <End_IP_destination_BU> 255.255.255.255
```

Listing 5.1 – BFD scripts.

The multiple sessions initiated between the concentration environment and the multiple remote routers following the process described in the Figure 17 turns the concentration router (HUAWEI NE40) essentially into a processor for processing control BFD packets.

As can be seen by looking at Figure 18, the regional concentrator has several BGP sessions. It is important to highlight that in each Business Unit, two BFD sessions are established, which by definition is a two-way control mechanism. Being one of them from the hub towards the remote unit's router and the other one in the opposite direction. In this way, any peer that realizes through the BFD mechanism the fall in the communication of the main link is able to override the BGP protocol and force the establishment of communication through the contingency circuit. It is important to note that all control packets have a corresponding echo packet that returns to the source as a check for verifying whether the other peer is active and is also processed by the equipment that sends the BFD control packets.

With BFD sessions, a new and more efficient way to detect failures is deployed. The problem of slow convergence is resolved as the entire network reacts more quickly to

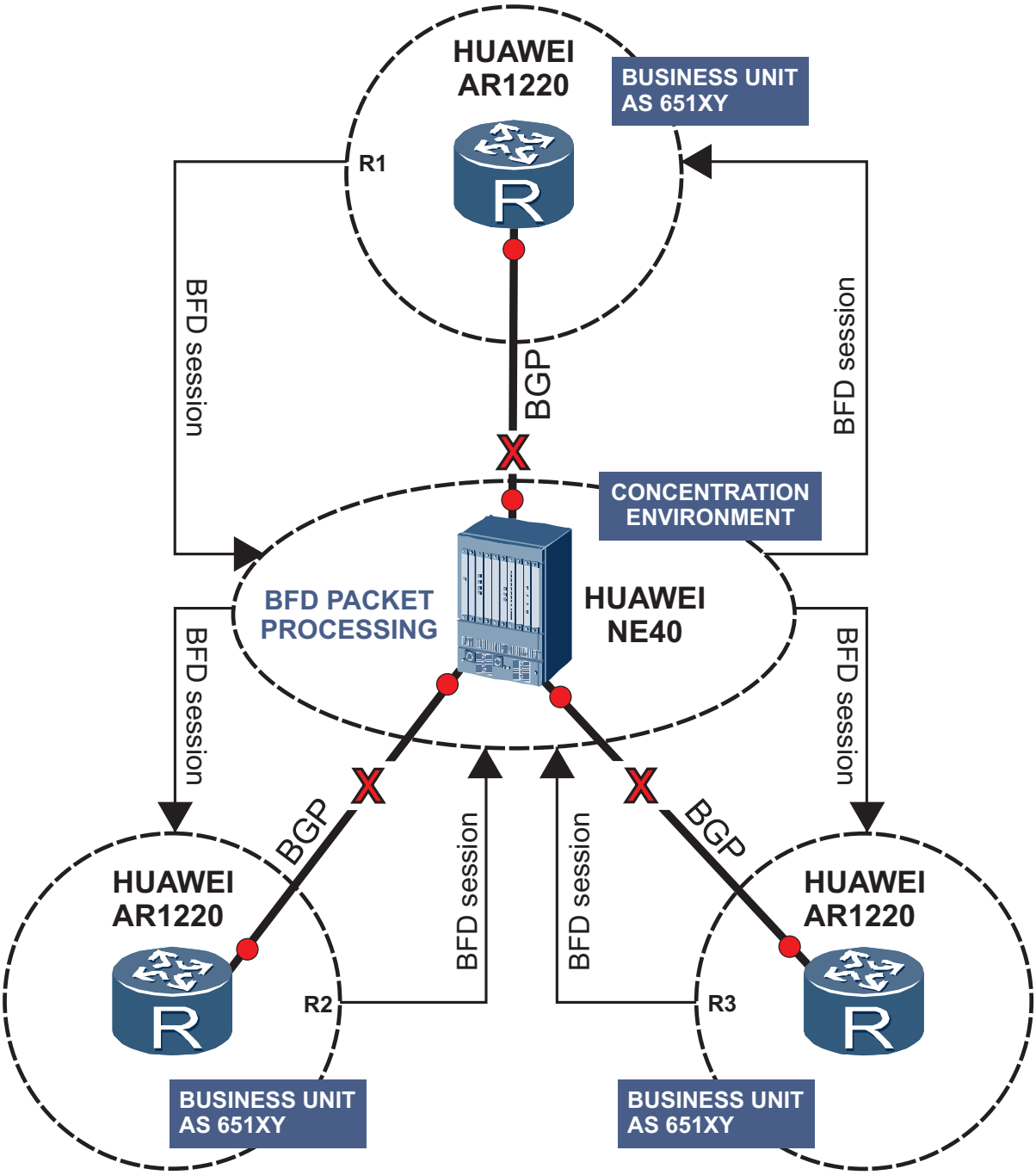

Figura 18 – Multiple BFD Sessions Processing

failures affecting the business units' communication links. Considering the availability gain inherent to the new architecture, it is essential to understand how the processing mechanism of these BFD packets occurs from the perspective of the concentrating device, as this will serve as a basis for the performance evaluation to be performed in this work.

Before understanding the device it is required a brief description of the equipment. The concentrator equipment HUAWEI NE40 executes the operational system Huawei Versatile

Routing Platform Software VRP (R) software in its version 8.181 (NE40E V800R011C00).

```
<HUAWEI> display version
Huawei Versatile Routing Platform Software
VRP (R) software, Version 8.181 (NE40E V800R011C00)
Copyright (C) 2012-2013 Huawei Technologies Co., Ltd.
HUAWEI NE40E-X16 uptime is 0 day, 1 hour, 2 minutes
Patch Version: V800R011C00SPH001
```

Figura 19 – Huawei Versatile Routing Platform Software

The NE40E uses a modular architecture. The physical architecture includes the following systems (HUAWEI TECHNOLOGIES CO, 2019b):

- Power Distribution System;
- Functional Host System;
- Heat Dissipation System; and
- Network Management System (NMS).

All systems except the NMS are located in an integrated cabinet. The functional host system is composed of the system backplane, main processing units (MPUs), line processing units (LPUs), and switch and fabric units (SFUs). It is connected to the NMS by NMS interfaces. The functional host system processes data and it monitors and manages the power distribution system as well as the heat dissipation system. Figure 20 shows the functional host system of the NE40E (HUAWEI TECHNOLOGIES CO, 2019b):

The logical architecture of the NE40E consists of three planes: data plane, control and management plane, and monitoring plane, as shown in Figure 21.

- "The data plane is responsible for high-speed processing and congestion-free switching of data packets. It encapsulates and decapsulates packets, forwards IPv4/IPv6/MPLS packets, performs QoS as well as scheduling and internal high-speed switching, and finally it collects statistics" (HUAWEI TECHNOLOGIES CO, 2019b).
- "The control and management plane provides all control and management functions for the system and is the core of the entire system. Control and management units process protocols and signals and configure, manage, report, and control system status" (HUAWEI TECHNOLOGIES CO, 2019b).

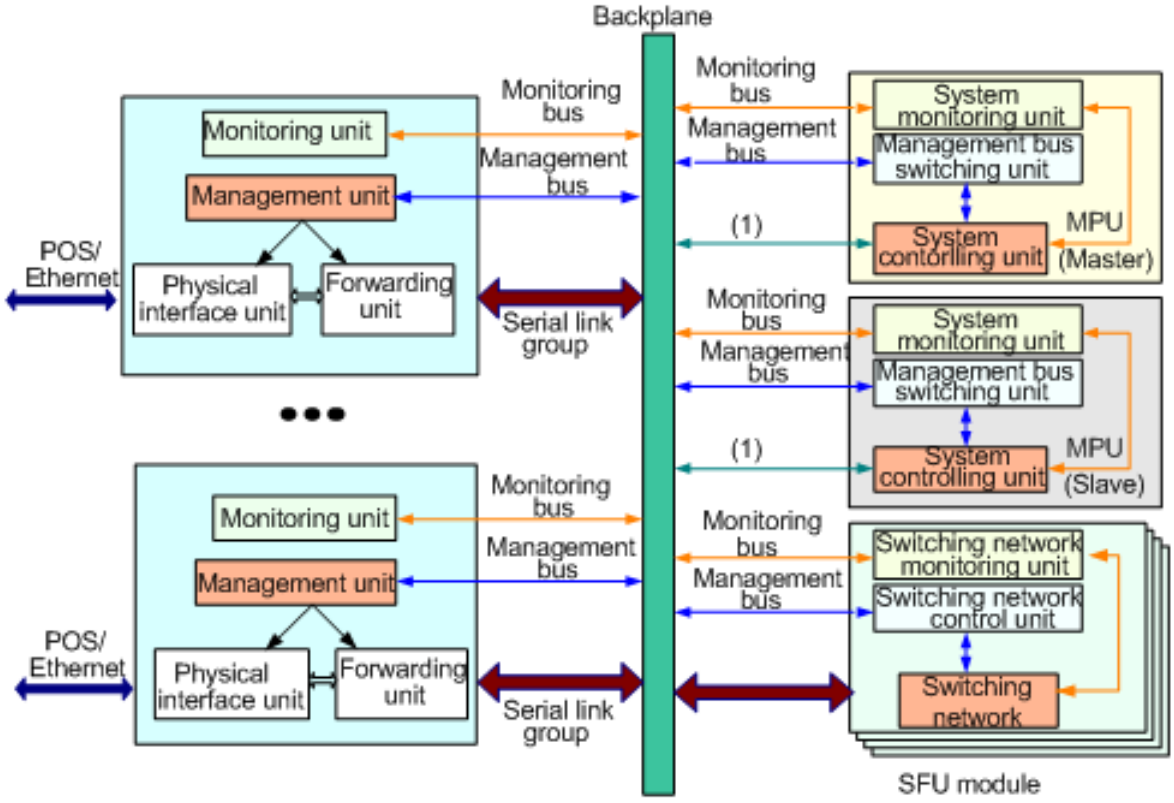

Figura 20 – Functional Huawei NE40 Host System (HUAWEI TECHNOLOGIES CO, 2019b)

- "The monitoring plane monitors the ambient environment to ensure the secure and stable operation of the system. It detects voltage levels, controls system power-on and power-off, monitors the temperature, and controls fan modules. If a unit fails, the monitoring plane isolates the faulty unit promptly so that the other units remain unaffected" (HUAWEI TECHNOLOGIES CO, 2019b).

The NE40E series routers provide a multi-process and full-service software architecture.

Involved acronyms in the figure and abbreviations are as follows:

- SSP: Service Splitting Platform;
- BOS: Balance of System;
- SMP: System Management Plane;
- NP: Network Processor; and
- BSA: Basic Service Area.

With the knowledge of the architecture exposed in detail, it is possible to notice the existence of a limiter in the concentration environment. In the new architecture, the capacity limit is defined by the Forwarding Performance parameter. In the present

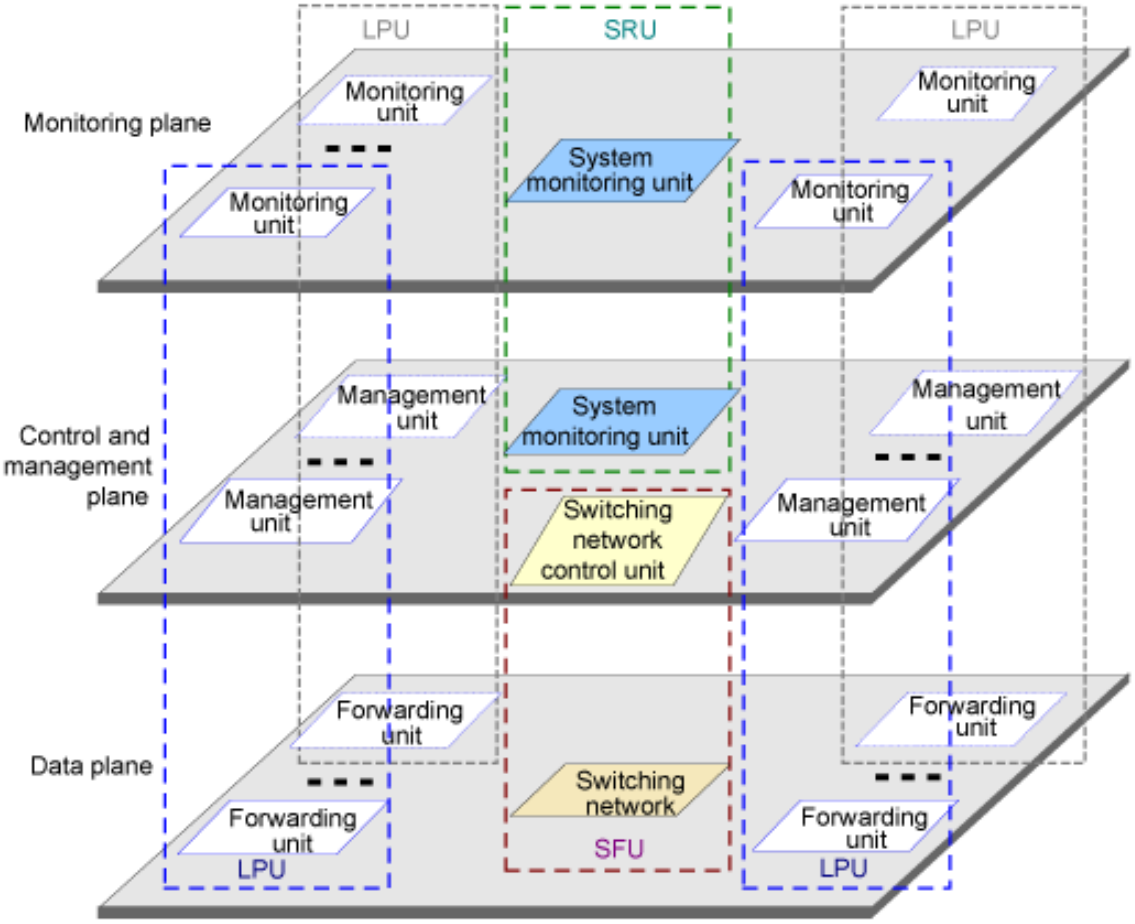

Figura 21 – Huawei NE4O's Logical Architecture (HUAWEI TECHNOLOGIES CO, 2019b)

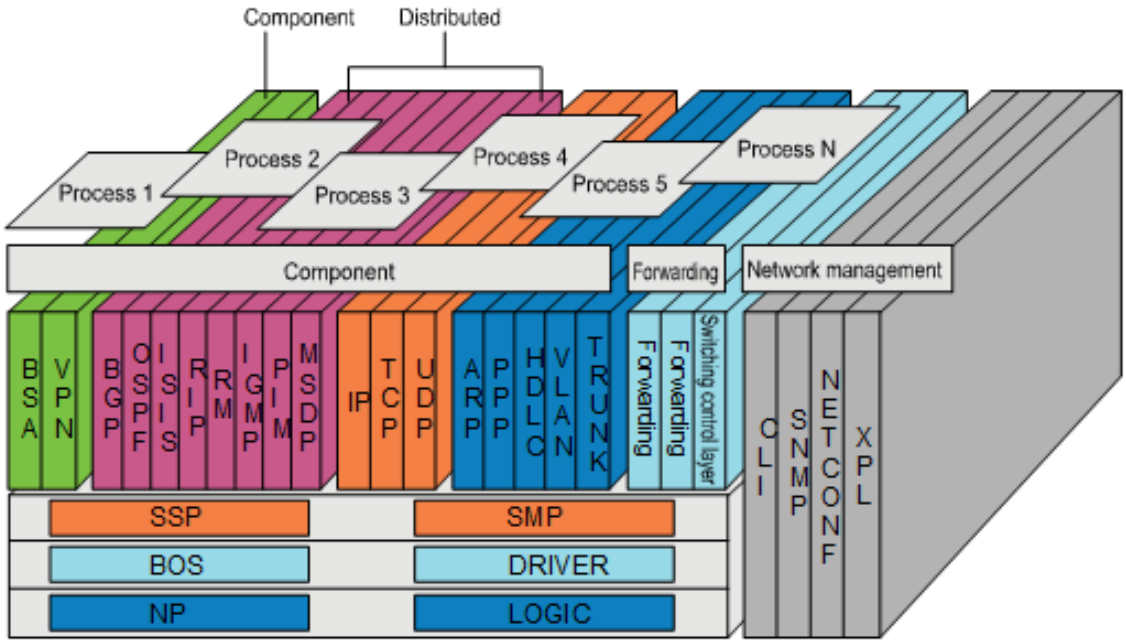

Figura 22 – Huawei NE4O's Software Architecture (HUAWEI TECHNOLOGIES CO, 2019b)

study case, considering that BFD packets are keepalives, it makes no sense to use the equipment buffer as a resource limit, since any queuing of these control packets can be interpreted by the concentration network asset as a false positive in the main link failure, causing it to switch to the backup circuit improperly. Therefore, we adopt as the packet limit the Forwarding Performance informed by the equipment's manufacturer as shown in Figure 23.

Item	NE40E-X3
Forwarding performance	• 105 Mpps (LPUF-51)

Figura 23 – HUAWEI NE40's Forwarding Performance Parameter

Forwarding Performance parameter is defined as the maximum rate at which the router can process and forward. In our communication system, the transmission capacity is (i.e., transmission speed) is C=105 Mpps (million packets per second) as stated in the manufacturer's documentation presented in Figure 23. If the Forward Performance resources are exhausted, it occurs the loss of the BFD control packet. Arriving messages to be transmitted by following a Poisson process. A message (or packet) consists of L bits, where L is a random variable (the length of the message). We assume that L is exponentially distributed with mean 100 bytes, because, according to the BFD protocol documentation (KATZ; WARD, 2010), the maximum length of each packet on the wire is in the order of 100 bytes, for a total of around 48 kilobits per second of bandwidth consumption in each direction.

The new problem here is determining the maximum arrival rate we can keep (in packets/sec) to ensure an average messaging waiting time of less than 100ms. It is important to note that according to the implemented script, if four BFD control packets have no response, the link is declared down and its traffic is switched to the redundancy link. Since these control packets are sent every 100 ms, the direct consequence is that the concentrator device's BFD control packet processing engine must be able to respond to packet arriving control in less than 400ms (4×100ms) to each remote business unit in which there is no improper switching to the backup link (false positive).

The number of arrivals in (t,Δt) in the concentrator device is independent of t and previous arrivals that contributed to the use of Performance Forwarding of the device. This makes possible to determine the service rate μ_s as linked to the Performance Forwarding because queuing for keepalive packets is not desirable. Of course, this performance evaluation will result in an optimal number of sessions and, as a consequence, an optimal number of remote units that should ideally connect to the concentration environment using BFD sessions.

Again, making use of our theoretical reference, it is pertinent to highlight that in order to reduce the usage of system resources when a BFD session is detected in *Down state*, the system adjusts the minimum interval for receiving BFD packets and the minimum interval for sending BFD packets to random values greater than 1000 ms. When the BFD session becomes Up, the configured intervals are restored.

5.3 AVAILABILITY MODEL

The type of redundancy used in this critical system is a warm standby, in which a detection and switch mechanism is employed to monitor the active and operational components. When the main component fails, the standby component is taken to an active and operational state through the switch mechanism. In the warm standby redundancy mechanism, both the main component and the standby components can fail (KUO; ZUO, 2003). Taking into account that the physical circuits do not age, the probability of failure does not increase with time.

Furthermore, the links have identical physical characteristics, only differing between network providers. For study purposes, a constant failure rate (λ_f) is considered. On the other hand, the repair of both the main circuit and the backup circuit is performed under the same SLA, adopting a repair rate (μ_r) for both circuits.

Based on these assumptions, Table 3 presents the states of the critical system with the characteristics of the case study.

Tabela 3 – States of the CTMC Model – Warm Standby

States	**Description**
UW	The system is active (UP)
DW	Inactivity state (DOWN), Main link failed, Activation process
DU	The system is active (UP), Main link failed
DD	Inactivity state (DOWN), Main and Backup links failed
UD	The system is active (UP), Backup link failed

The environment analyzed is a stochastic Markovian process in which the random variables do not depend on the previous history. Figure 24 represents two components, the main and replacement components, using a CTMC. In this model, labels U, D and W represent the activity, inactivity and standby states, respectively.

The model illustrated in Figure 24 was developed by using Mercury software (SILVA et al., 2015). Mercury was developed to support the analysis of the performance and

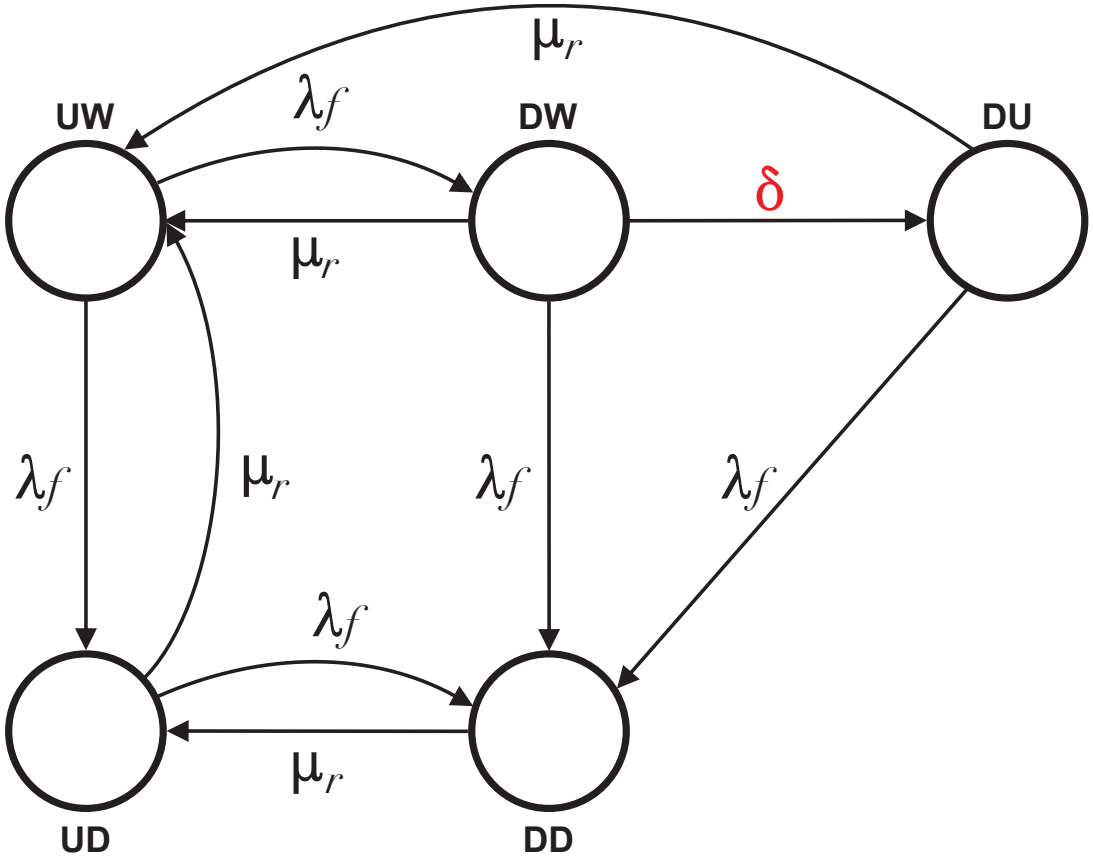

Figura 24 – CTMC for Availability Computation (SIQUEIRA et al., 2019)

dependability models, being sufficiently generic to also enable evaluations of systems in general.

As described in Table 3, this model is composed of five states, with states DD and DU representing the states in which the system is non-operational. When the main circuit fails and the backup circuit is up, the backup circuit receives all data traffic of the system. This activation process corresponds to the data traffic redirecting process and it has a delay that represents the failure detection time and the time required to the backup circuit makes the system operational again. In turn, the parameter MTTF (mean time to failure) of each main and backup circuit implies that these transitions have similar delays to the linkages, for the network in the study, use the same technology, despite being provided by distinct telecommunications providers. Besides that, the MTTR (mean time to repair) parameters are also considered identical, given that the providers present the same SLA to repair the links. The model in Figure 24 schematically represents part of the network. The entire network consists of hundreds of critical systems represented by the scheme depicted in Figure 16.

Figure 24 highlights the transition from state DW to state DU, which represents the time required for the circuit backup to become operational (δ). The study is mostly concentrated in the before-and-after comparison of this time, in order to improve the availability of the critical system. The availability (A) of the model is determined by

Equation 5.1, where P is the probability of being at a certain state.

$$A = P(UW) + P(DU) + P(UD) \tag{5.1}$$

Considering the sensitivity analysis of availability as a function of the parameter δ (inverse of fail-over time), we present Equation 5.2. It is important to note that, given the intrinsic nature of the case study, the values of both λ_f and μ_r, which correspond respectively to the MTTF and MTTR of the data circuits, are considered identical and follow an exponential distribution.

$$A = \frac{\mu_r \left(2\lambda_f \left(\delta + \lambda_f\right) + \left(\delta + 2\lambda_f\right)\mu_r + \mu_r^2\right)}{\left(\delta + \lambda_f + \mu_r\right)\left(2\lambda_f^2 + 2\lambda_f\mu_r + \mu_r^2\right)} \tag{5.2}$$

In order to establish the availability of the entire critical system, the representation of only one BU – concentration pair is considered. Thus, the availability of the entire critical system can be determined as a function of availability (A), initially calculated by Equation 5.2, by applying Equation 5.3 (k out of n) — binomial distribution formula.

$$A_{KooN} = \sum_{i=K}^{N} A^i \left(1 - A\right)^{N-i} \tag{5.3}$$

Equation 5.2 is important because it allows analysts to observe how much increment in the availability occurs as a result of adopting failure detection by applying the BFD. Equation 5.2 has three variables (λ_f; μ_r; δ), but between the original scenario and the scenario using the BFD protocol associated to BGP, the availability is sensitive only with the δ value. It occurs because the values of λ_f, which represents the failure rate, have no changing from one scenario to another because this failure rate is based on statistical data that considers one failure per month to occur on each link. On the other hand, the value of μ_r is based on a contractual standard for repair SLA.

In turn, the value of δ is related to the empirical data collected in each scenario as will be seen in Chapter 6. It occurs because the delta represented in the equation is an average of the inverse of the failover time extracted from actual measurements both in the scenario only with the configuration of the BGP protocol, as well as in the scenario where BFD is associated to BGP.

In order to use Equation 5.2 to calculate availability (A), it is necessary to define values for λ_f and μ_r according to the criteria used for MTTF and MTTR. These rates tabulated in Table 4 consider time measured in hours, being approximately 724.64 hours for failure and 4 hours for the repair.

Tabela 4 – Parameters Definition for λ_f and μ_r

	λ_f	μ_r
Value	$0.00138h^{-1}$	$0.25h^{-1}$

Considering that the availability model presented in Figure 24 represents the backup mechanism for a remote unit, the Equation 5.2 also refers to the availability of a remote unit. Equation 5.3 allows analysts to calculate the availability of the entire network both in the scenario only with the configuration of the BGP protocol, as well as the scenario where BFD is associated with BGP. Equation 5.3 has the advantage of making the total availability calculation a portable measure because it is able to associate the total availability value with the number of links (N) that make up the network.

5.4 PERFORMANCE MODEL

Considering the availability model presented in the previous section, that allows us to measure the availability gains, we have proceeded to build a model that is able to evaluate the performance of the network concentration environment. Based on the architecture presented in the section 5.3, where we described the BFD packet processing capacity limitation defined by the Forwarding Performance parameter. The proposed performance evaluation model is a Markovian queue M/M/1/K, as shown in Figure 25. This model was based on the theoretical framework presented in the Chapter 2, both the state of the art BFD protocols (Section 2.1) and the application of queueing theory principles (Section 2.5). In figure 25 λ_a represents the arrival rate of BFD packets in the concentrator equipment, in turn $\lambda_{\pi k}$ means the lost BFD packet rate. On the other hand, μ_s is the rate at BFD packets are processed.

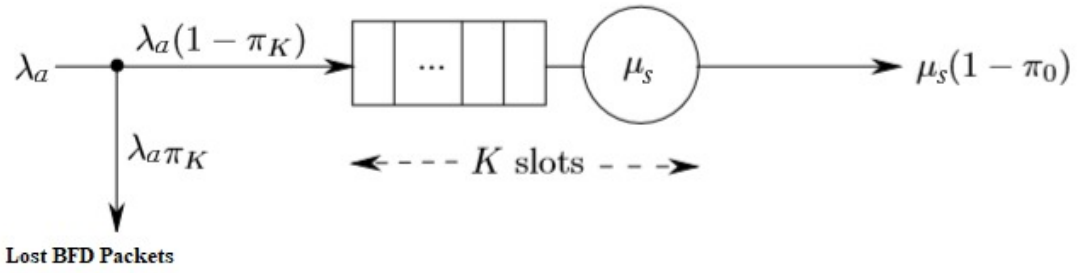

Figura 25 – The M/M/1/K BFD packets queuing system

As shown by the Figure 26, the boundary K of the queue model is given by the *forwarding performance* parameter. The value of *forwarding performance* was obtained considering the specification provided by the equipment manufacture (HUAWEI TECHNOLOGIES CO, 2019b). This document describes the features of the NE40E equipment.

The model is appropriated to be used for evaluating the maximum arrival rate and the mean processing rate, considering the *forwarding performance* parameter. Moreover, through the model, it is possible to estimate if there exists CPU bottlenecks with the increased use of the BFD solution for the entire network. We model packet flow through the concentration equipment as a Markovian process. Hence, we consider that the packet

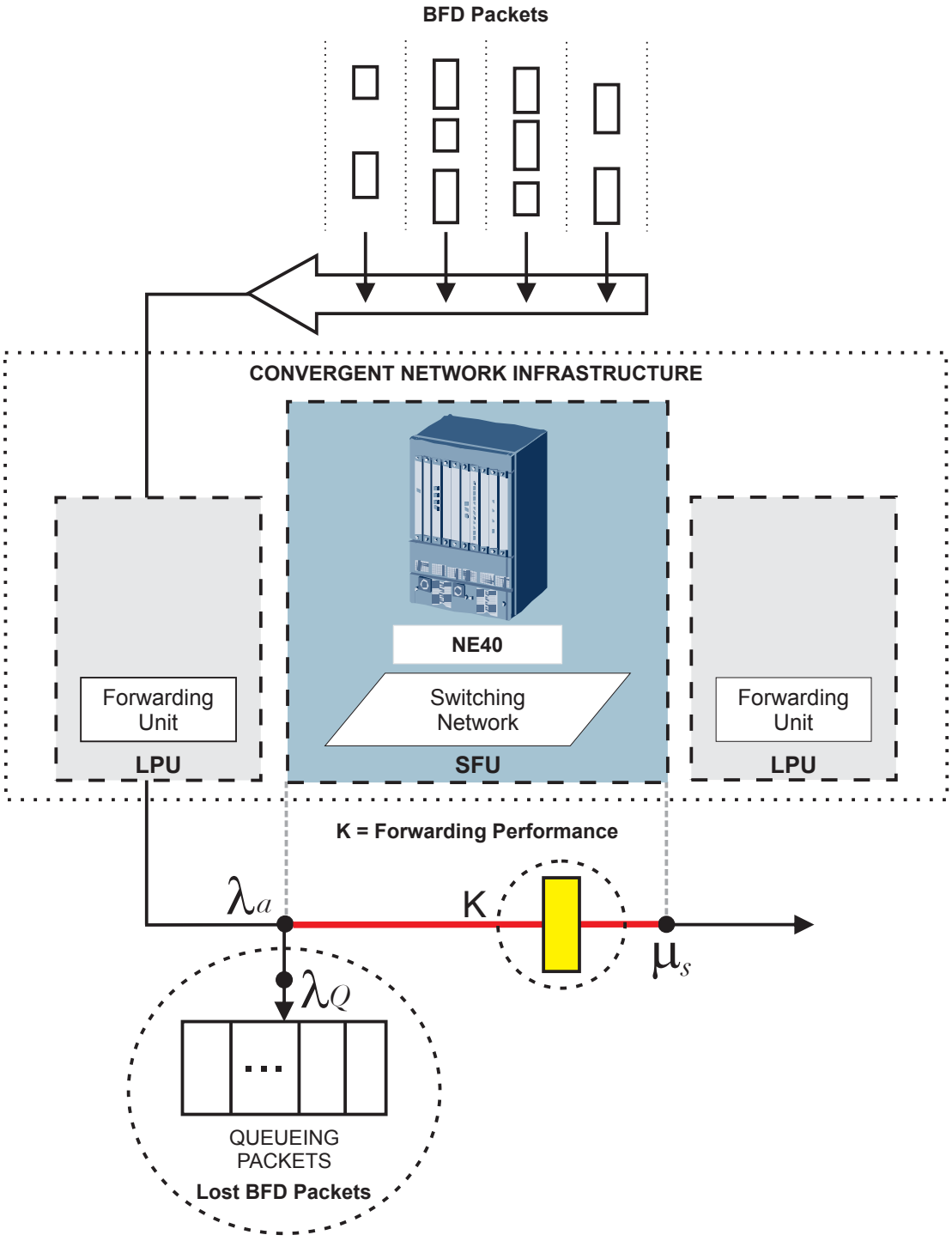

Figura 26 – Pre-queuing Process

arrival to the router as a Poisson arrival process because it is regardless of the service time distribution, as the arrival and service processes are independent. Additionally, we consider that the service time of each packet at the HUAWEI NE40 follows an exponential distribution (CASSANDRAS, 2008). The model captures the delay experienced by the packets that have to be processed by the concentration router as well as the probability

to queue packets in the case when the concentrator is under high load. In this case, a high load scenario means a load greater than the specified by the *forwarding performance* parameter. As previously described, considering that this is a monitoring process carried out in a very short time, packets that enter the buffer queue are considered lost. From this point down, BFD packets are queued, which is not desirable for the high availability solution because it tends to generate false positives alerts. Therefore, λ_Q shown in figure 26 represents both the queuing and loss rate.

As shown in Table 6, although we are using the M/M/1/K queue model, our objective is to keep the critical system at the stage prior to the queuing process. Taking this into account, equations and metrics of the M/M/1/K queue Markovian model were adapted by our model, transforming queued packets into discarded packets. Thus, for example, the traditional packet discard rate metric will be represented in our model as the queuing rate. That occurs because, in practice, queuing does not serve the purpose of the high availability monitoring system deployed with the BFD protocol.

On the other hand, it is important to note that the main objective of the model is to highlight the processing increment caused by the implementation of the BFD solution. Because of this, by applying queuing theory to the model implies to get the BFD control packet arrival statistics, which in turns shows the amount of BFD control packets (packets in) processed by the concentrator device within an observation period.

The BFD session statistics obtained, as shown by the table 5, provide the parameters needed for the application of queue theory. As we have pointed out in the Section 5.2, there exists a mechanism for processing incoming BFD packets. These incoming packets are associated with a finite *forwarding performance* parameter to process received packets. This limit (K) is set by the vendor documentation as presented in (HUAWEI TECHNOLOGIES CO, 2019b).

The proposed model must evaluate the CPU utilization of the concentrator equipment as well as the BFD packet arrival rate. In addition, it is necessary for the model to make it possible to perform sensitivity analysis between the BFD control packets discard rate, here considered as queuing rate, and arrival rate of BFD packets, considering the device's *forwarding performance* parameter.

In this case, the number of BFD control packets that can be processed in the system is limited to the value defined in K, as illustrated in Figure 25. An arriving BFD control packet that finds the system full is rejected and considered to be lost. We can analyze this system by setting the arrival rate λ_{an} for all n=0,1,...,K-1, and $\lambda_{an} = 0$ for all $n \geq k$. In fact, we turn the Poisson arrival process off when the arrival rate reaches the value defined by K and activate the process again when it becomes (K-1). Therefore,

$$\lambda_n = \begin{cases} \lambda & if\ 0 \leq\ n < K \\ 0 & if\ n \geq K \end{cases}$$

Tabela 5 – BFD statistics session

Parameter	Description	Value
all[1]	Displays statistics on all BFD sessions.	
static[1]	Displays statistics on all static BFD sessions.	
dynamic[2]	Displays statistics about dynamic BFD sessions and static BFD sessions with automatically negotiated discriminators.	
discriminator *discr-value*[2]	Displays statistics on a BFD session with a specified local discriminator.	The value is an integer that ranges from 1 to 16383.
peer-ip *default-ip*[2]	Display statistics about BFD sessions with the default multicast address.	
peer-ip *ip-address*[2]	Display statistics on BFD sessions bound to a specified IP address.	
for-ip[1]	Displays statistics on BFD sessions for detecting IP links.	

[1] **display bfd statistics session { all** | **static } { for-ip }**

[2] **display bfd statistics session { dynamic** | **discriminator** *discr-value* | **peer-ip default-ip** | **peer-ip** *ip-address* }

$$\mu_{sn} = \mu_s \ for\, all\, n = 1, 2, ..., K$$

We have studied the behavior of the router by using a Markovian model (KLEINROCK, 1975), as it follows the following Markov properties:

1. Each packet is processed individually and the behavior of the concentration equipment is memoryless.

2. Packet processing in a router may be described as a stochastic process which have Markov properties, that is, the conditional probability distribution of future states depends only on the current state. As we have pointed out, the past states of the evaluated process is not relevant for future state estimations.

The state transition rate diagram for this system is shown in 27. The process of stopping arrivals only works with a Poisson process because of the memoryless property. At the time we restart the process, the residual interarrival time of the actual process has the same distribution as the entire interarrival time, hence the Markovian structure of the state transition rate diagram in Fig 25 (in particular, the transition from (K-1) to K) is preserved.

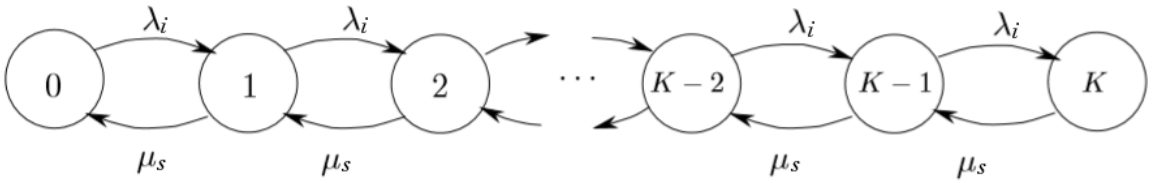

Figura 27 – Ingress of BFD/Forward Performance packets in a Concentration

To make use of the M/M/1/K model equations presented in the Table 6, it is necessary to obtain some empirical data. One data corresponds to the concentration CPU mean consumption what is extracted initially in the scenario that uses only the BGP protocol, considering the implementation of BFD associated to BGP in a representative sample, in order to collect the new mean concentration device's CPU consumption. Our aim is to obtain $\Delta\bar{U}_i$, which corresponds to the average CPU increment due to the adoption of BFD. Therefore,

$$\Delta\bar{U}_i = \bar{U}_{i-after} - \bar{U}_{i-before}$$

The other relevant empirical data is C_0 that represents the total number of requests completed by the system in the observation period (T). It corresponds to the total BFD packets processed by the concentration router whose statistical data are obtained by using the commands presented in Table 5.

Based on this empirical data, by considering the service demand law (MENASCE et al., 2004), it is possible to determine the value of D_i, which represents the mean service time (MST). MST corresponds to the time that a transaction demands of a resource. It is also possible to obtain the value of μ by derivation when having MST value. In this case, μ is the inverse of the MST value, as described here $\mu = \frac{1}{MST}$.

The ratio $\frac{\lambda}{\mu}$ is used as variable in equations of M/M/1/K queue systems. Intuitively, if $\lambda > \mu$, it means that BFD control packets are lost or queued, but the amount of packets remains bounded by K and no unbounded growth can arise.

We set $\rho = \frac{\lambda_i}{\mu_s}$, but emphasize that ρ does not represent the traffic intensity. This is because λ_i is the arrival rate of BFD control packets. In other words, λ_i is not the rate of packets actually processed in the concentration router, since some of them may be lost.

We also use a validation model that consists of obtaining empirical data directly in the network, by varying the input rate λ_i and comparing the collected empirical values by the value obtained by using the model equations. This validation is critical for using the model to predict bottlenecks and project the optimal number of units to migrate. In this context, taking the λ_i parameter as input, the measurements are tabulated in a table as presented as an example in the Table 7.

Tabela 6 – Equations for M/M/1/K BFD control packets

Equations	**Values for** $\rho \neq 0$
Utilization	$\frac{\left(\rho\left(1-\rho^{k}\right)\right)}{(1-\rho^{k+1})}$
Discard Probability	$\frac{\rho^{i}(1-\rho)}{(1-\rho^{k+1})}$
Discard Rate	$\lambda\Pi_{k}$
Mean Response Time	$\frac{\frac{\left(\rho\left(1-(k+1)\rho^{k}+k\rho^{k+1}\right)\right)}{(1-\rho)(1-\rho^{k+1})}}{\lambda(1-\Pi_{k})}$
Throughput	$\lambda\left(1-\Pi_{k}\right)$

Tabela 7 – Model Validation

	T-Paired	
	$\Delta\bar{U}_{imodel}$	$\Delta\bar{U}_{inetwork}$
λ $(packets/s)$	**Value**	**Value**
λ_1	model.value.1	network.value.1
λ_2	model.value.2	network.value.2
λ_3	model.value.3	network.value.3
λ_n	model.value.n	network.value.n

6 CASE STUDIES

This chapter presents three case studies to demonstrate the applicability of our approach. The first case study evaluates the availability of a real corporate network by using a before-and-after comparison. In this case study, we evaluate the network availability by considering two scenarios with distinct configuration parameters and network protocols that reflect in a distinct mean failover time.

The second case study is related to the first one, because it performs a performance evaluation on the concentration equipment for the same network whose availability was ever evaluated. In this second study, a before-and-after comparison is also performed taking into account the same changing in the protocols and configuration parameters ever performed in the first study.

The final study is directly linked to the second one. Based on the performance evaluation early performed, it is possible to infer the optimal number of communication links in order to maintain the corporate network with high availability, no processing bottlenecks and no occurrences of false positives alerts.

6.1 CASE STUDY ONE: AVAILABILITY EVALUATION

Initially, following the methodology for availability evaluation proposed in Section **??**, we will do a contextualization of the network used in this case study. The communication network represents an infrastructure designed to ensure that banking services can be made available nationwide. It is underpinned by high-performance critical equipment, making critical systems and applications accessible to end-users and customers. The network's backbone connects regional ring concentrators to application servers following a star topology. This enables business units to connect to hub devices that, through the backbone, lead to the application servers in a centralized data center. Connections between BUs and a specific regional hub are the object of the evaluation in order to find a design that provides a more efficient redundancy mechanism, increasing the overall quality of the provided service. In addition, the corporate network architecture follows the scope of what was exposed in Section 5.1, where the baseline architecture was introduced.

We defined as parameters for the evaluation process the environment configuration that presents two distinct contexts in each scenario. The first scenario uses BGP protocol, as described in the section *Baseline Architecture* (5.1), and the second one considers the association of BFD and BGP protocols, as detailed in the section *BFD Session Process* (see 5.2).

Based on parameters presented before, that are associated with the two scenarios to be compared, we defined the failover time as a metric to be estimated. This metric

corresponds to the failure detection time. More specifically, it is the time required for the system to identify the fault plus the time required to change to an alternate route after failure is detected. And as such, it is a type of metric to be extracted by performing experiments on the network.

Our CTMC model is detailed in Section 5.3 and it is capable of representing the contingency activation process (backup). More specifically, it represents the redundancy mechanism of the corporate network considering the connection of a BU and the concentrator router. Using this model, we were able to estimate the network availability and downtime. For this estimation purpose, we have derived from it some equations that make it possible to estimate availability for one communication link or for multiples ones (N) (See Equations 5.2 and 5.3). Three variables are required in order to calculate availability. The variable δ is extracted from the experiment performed for the availability evaluation. The other variables, δ and λ_f, are obtained from Table 4 following the characteristics already mentioned about failure and repair inherent in the evaluated network.

The corporate network considered in this case study does not allow us to perform long tests, as it is inserted in a critical production environment. Furthermore, the total amount of circuits sums up to some hundreds, with distinct physical distances that may influence the differences on the time obtained. The empirical data corresponding to the metrics considered in this experiment was obtained by applying a sampling technique. Thus, given that it is a technique that enables, through a random subset of the total amount of communications links, to draw conclusions about the parameters of the population, based on statistical summaries and error probabilities to obtain results with the lowest level of distortion, repeatability was used. That is, successive measurements were carried out with no changes in the measurement conditions.

Considering a population of 1,000 business units, tests could be carried out in a sample of 40 communication circuits. The experiment was performed utilizing ICMP packets from a gateway to another by simulating the interruption of the main link and measuring the failover time. The failover time being represented by the period of time from which the BU's router stops responding the pings fired from the concentration environment until detecting a first successful response. During the test, and in order to establish the failover time, the unavailability of the main circuit is triggered, simulating the fiber rupture mechanism.

Considering the Baseline Architecture presented, in which the convergence problem inherent to a backbone solution supported by the BGP protocol is highlighted. To overcome this problem we have developed a new configuration arrangement using the BFD protocol to more quickly detect failures between BGP peers. The association of the BFD protocol with the BGP protocol is presented in the script shown in listing 5.1. Through this configuration, it is possible to reduce failover time and bring availability (A) to a value very close to 1, as we will see below.

To generate the configurations, we have accurately consulted references of BGP and BFD protocols through the available RFCs. The structure and syntax of the commands related to the implementation of the BFD protocol was studied from the HUAWEI manufacturer's documentation for the two router models used in the network: Huawei AR1220, installed in the business units; and Huawei NE40, installed in the concentration environment.

Considering that the corporate network offers, we first implemented the scripts generated in the evaluation environment so that the changes do not cause eventual unavailability. Huawei offers a simulation software that lets us emulate the real devices that are installed in the network in production. It is important to highlight that all applied configurations were initially implemented in the evaluation environment using the Enterprise Network Simulator tool (HUAWEI TECHNOLOGIES CO, 2019a), developed by HUAWEI, manufacturer of the routers used in the evaluated production environment. Through the eNSP tool it is possible to manipulate the actual equipment configuration files and validate the generated configurations, as can be seen in the Figure 28. The failover data tests, however, are not efficient in the virtual environment, as this environment ignores some network parameters such as latency, directly associated to the physical distance between two devices. Considering this, the simulated environment is for preparation of deployment on real devices.

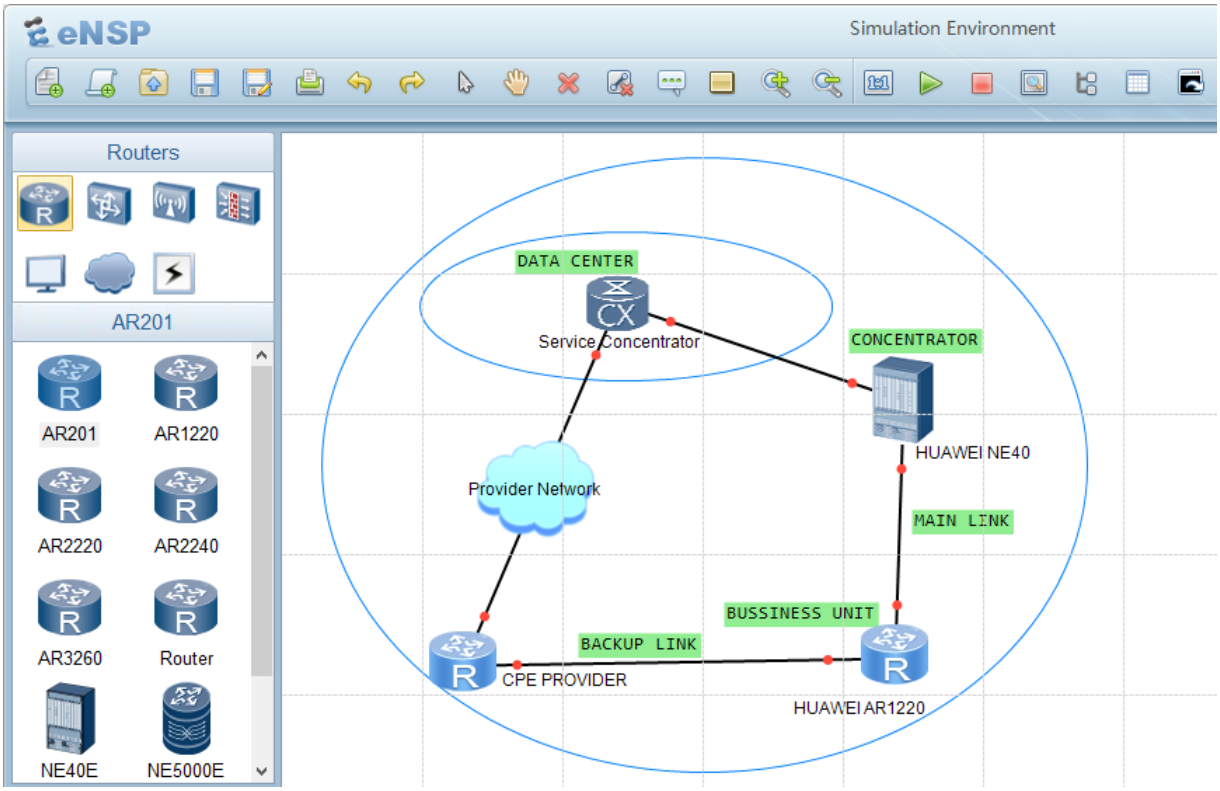

Figura 28 – Evaluation Environment with eNSP

In order to obtain the best convergence conditions, an inferential performance eva-

luation was carried out by using a before-and-after paired comparison. The empirical distribution is represented by the histogram drawn using Statdisk tool (TRIOLA et al., 2005). In this regard, the frequency in which the different failover times occur was plotted, being considered as a baseline scenario for the experiment (before), as illustrated in Figure 29.

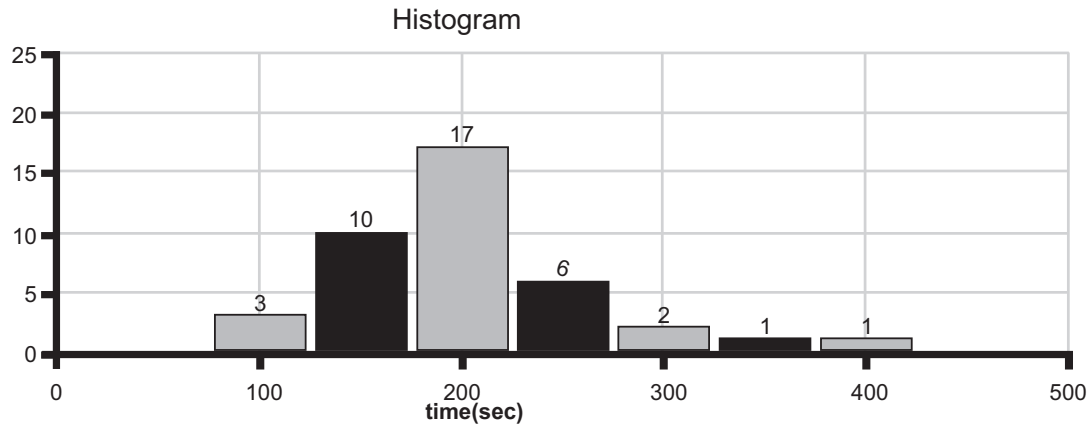

Figura 29 – Failover Time Distribution (Before) (SIQUEIRA et al., 2019)

In turn, the Kolmogorov-Smirnov test, carried out by using Easy-fit tool (MEHRANNIA; PAKGOHAR, 2014), demonstrates that the empirical data follow a normal distribution, as depicted in Figure 30.

Kolmogorov-Smirnov					
Sample Size Statistic P-Value Rank	40 0.1554 0.26066 36				
α	0.2	0.1	0.05	0.02	0.01
Critical Value	0.16547	0.18913	0.21012	0.23494	0.25205
Reject?	No	No	No	No	No

Figura 30 – Kolmogorov-Smirnov Test (Before) (SIQUEIRA et al., 2019)

According to Figure 31, the values of mean and standard deviation are valid as measures of central tendency and dispersion, with no statistical treatment of the data being necessary.

Following the change of configuration in the sample, as described in Chapter 2, a new measurement was carried out according to the methodology described in Section 4.1, aimed at evaluating the impact caused by the new arrangement and obtaining the results

Sample Size, n:	40
Mean:	202,025s
Median:	190s
Midrange:	240s
RMS:	211.2666s
Variance, s^2:	3917.41s²
Standard Deviation, s:	62.58921s
Mean Absolute Deviation:	44.63125s
Range:	300s
Coefficient of Variance:	30,98%
Minimum:	90s
1st Quartile:	166.5s
2nd Quartile:	190s
3rd Quartile:	223s
Maximum:	390s
Sum:	8081s
Sum of Squares:	1.785343e+6s²

95% CI for the Mean:
182,008s < mean < 222,042s

95% CI for the Standard Deviation
51,2706s < SD < 80,3667s

95% CI for the Variance:
2628,6763 s² < VAR < 6458,8139 s²

Figura 31 – Sample Analysis (Before) (SIQUEIRA et al., 2019)

of the after scenario. As presented in Figure 32, there was a reduction of the order of milliseconds in the failover time.

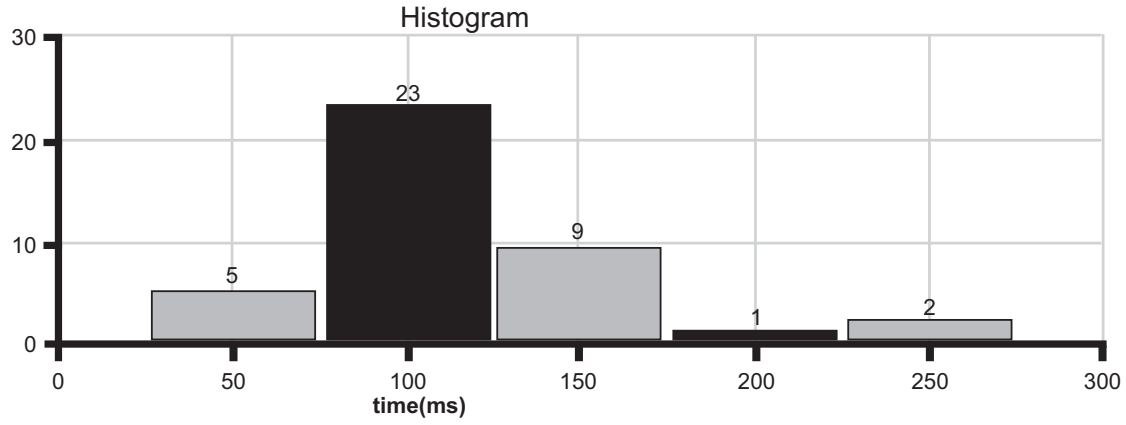

Figura 32 – Failover Time Distribution (After) (SIQUEIRA et al., 2019)

Therefore, it can be observed that the second measurement also follows a normal distribution, as seen in Figure 33. These results are subsequently shown in Figure 34.

Kolmogorov-Smirnov					
Sample Size Statistic P-Value Rank	40 0.14047 0.37397 40				
α	0.2	0.1	0.05	0.02	0.01
Critical Value	0.16547	0.18913	0.21012	0.23494	0.25205
Reject?	No	No	No	No	No

Figura 33 – Kolmogorov-Smirnov Test (After) (SIQUEIRA et al., 2019)

In order to use Equation 5.2 to calculate availability (A), it is necessary to define values for λ_f and μ_r according to the criteria used for MTTF and MTTR. These rates, tabulated in Table 4, consider time measured in hours, being approximately 724.64 hours for failure and 4 hours for repair.

Replacing the values obtained for average failover in each scenario (Figures 31 and 34) in the equation of the critical system, it is possible to calculate availability before and after the changes we have performed. Table 8 shows the values we have obtained.

Tabela 8 – Comparison of Availability (Before/After)

	Before	**After**
Availability	0.999893	0.999969

In turn, Figure 35 depicts the sensitivity analysis of the availability as a function of the parameter δ. It being represented in Equation 5.1 as the inverse of the failover time. This figure demonstrates that a reduction in the failover time represents a relevant increase in the availability of the critical system.

The application of the methodology allows a direct and strict comparison of the measurements. Because we applied the before-and-after comparison in the same environment and by considering the same statistical criteria. Isolating the environment variables that are not changed in the different scenarios.

An interesting conclusion that can be drawn by performing an analysis of the results and observing the curve of the graph of the sensitivity analysis (shown in Figure 35) is that: from a given value of δ, there is a little influence on the availability. There is a significant gain in the region of the graph worked on in the experiment, but from the gain

Sample Size, n:	40
Mean:	117.75 ms
Median:	114 ms
Midrange:	145 ms
RMS:	124.7504 ms
Variance, s^2:	1741.115 ms²
Standard Deviation, s:	41.72667 ms
Mean Absolute Deviation:	29.6875 ms
Range:	210 ms
Coefficient of Variance:	35,44%
Minimum:	40 ms
1st Quartile:	87.5 ms
2nd Quartile:	114 ms
3rd Quartile:	133.5 ms
Maximum:	250 ms
Sum:	4710 ms
Sum of Squares:	622506 ms²

95% CI for the Mean:
104,4052 ms < mean < 131,0948 ms

95% CI for the Standard Deviation
34,1808 ms < SD < 53,5785 ms

95% CI for the Variance:
1168,3304 ms² < VAR < 2870,6572 ms²

Figura 34 – Sample Analysis (After) (SIQUEIRA et al., 2019)

achieved with our testbed it is not convenient to spend more resources to further reduce the failover time. Since the availability increase is no longer significant from this point.

Another possible inference we may do is the following. Since the δ is the inverse of the failover time, that, in turn, is a function of both the time interval necessary to send BFD control packets and the number of packets lost before the communication link may be considered down. We have applied a configuration (parameter considered by the methodology) with 100ms interval between the BFD packets and a counting of four lost packets to the link be considered down. The new configuration decreased the failover time (metric defined in the methodology) to the optimized level. Thus, increasing the network's availability in a rational and modulated way.

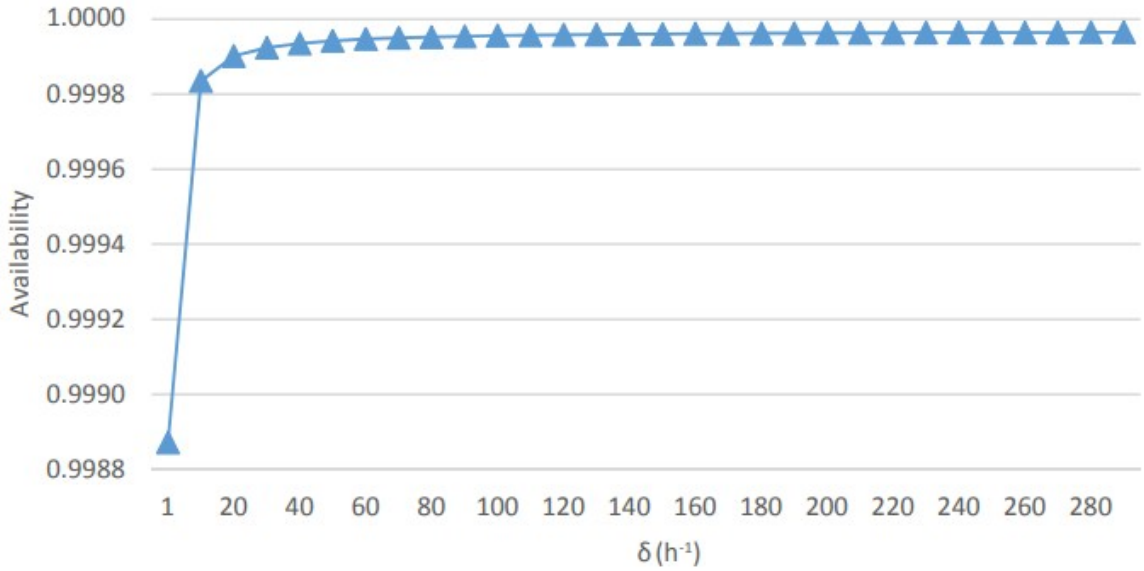

Figura 35 – Sensitivity Analysis -Availability (A) x δ (SIQUEIRA et al., 2019)

6.2 CASE STUDY TWO: PERFORMANCE EVALUATION

Following our methodology, the architecture presented in Section 5.2 allows the understanding of the corporate network's concentration environment, which is object of the current case study. The router model Huawei NE40 is the concentrating equipment presented in the evaluated network. There are a number of tasks assigned to this device, and it regionally concentrate all physical and logical connections. The primary workload attributed to the Huawei NE40 corresponds to the internal routing functionality and the interaction with external autonomous systems(ASs). These features are the essence of the network. Any configuration adaptation to improve availability must necessarily incorporate the essential features into the monitoring functions. It follows that the configuration that offers higher availability, by using the BFD protocol, necessarily implies a increased cost in processing for reasons intrinsic related to its control mechanism. It is exactly this increment on processing by using the availability solution that is intended to be evaluated here. In the Section 5.2, the processing mechanism of the Huawei NE40 has had its processing functionality studied in detail, including packets processing and capacity limitations.

We have evaluated the performance of the concentration environment for each scenario, with the association of BFD and BGP protocols. We considered both the base scenario as well as the scenario that offers higher availability. To do this, we collected the average CPU consumption of the concentrator device assigned to exclusively handle the common routing tasks. In this process, we used a tool named CA Spectrum r9.4 from CA Technologies (CA TECHNOLOGIES, 2019). It enables the management based on Simple Network Management Protocol (SNMP), through which it is possible to read the Management Information Base (MIB) from the concentrating device and collect the average

consumption before any changes had taken place to deploy the BFD protocol, as demonstrated in Figure 36. After performing the experiment to change the BFD configuration, by applying the script demonstrated in the Listing 5.1, the collected data regarding the new CPU usage allowed us to estimate the impact on the CPU due to the use of the BFD protocol. This impact is obtained by performing a simple calculation. The calculation performed corresponds to subtract the mean CPU consumption obtained in the scenario that offers higher availability, and the consumption of the original scenario.

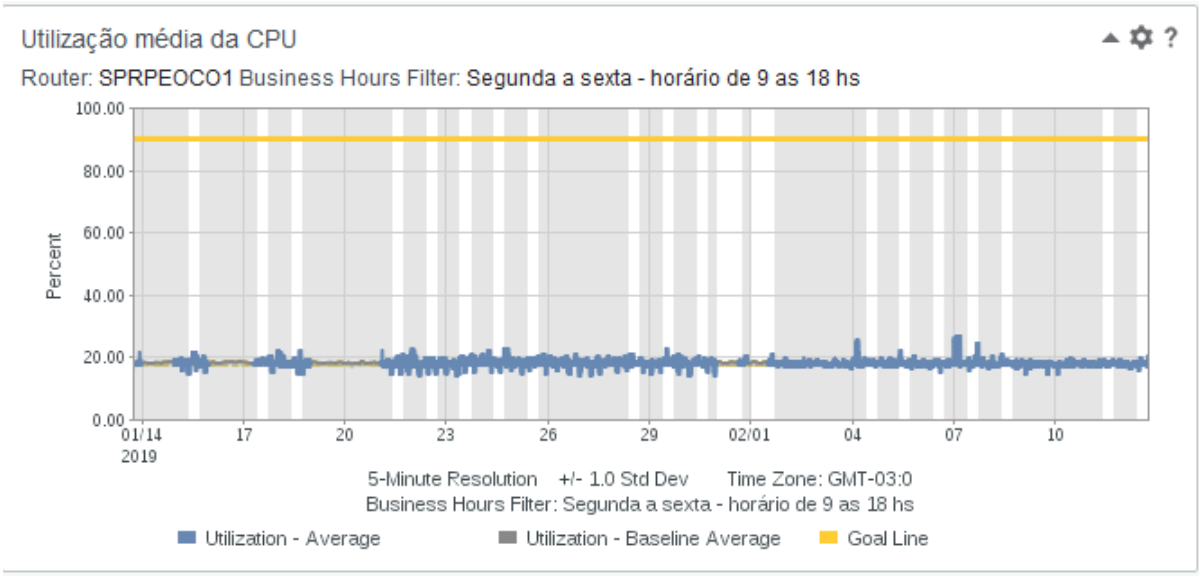

Figura 36 – CPU Usage by Spectrum

In this context, we defined the configuration applied in the network as a parameter for the evaluation. This configuration affects the performance. Thus, we consider the association and no association of BFD and BGP protocols as a parameter in the configuration. Also, it was considered the effect of these configuration parameters on the network's performance.

We carried out the performance evaluation by considering a queue M/M/1/K model. The evaluation was performed by taking into consideration the information presented in Section 5.2, and following the characteristics of the BFD protocol. We have concerned about to not generate false positives alerts whilst maintain the network's stability. In the M/M/1/K queue, we used the Forwarding Performance parameter as the K variable and its value is shown in Figure 23. As an initial requirement for starting the performance evaluation we have taken was to enable the statistical report for specific BFD sessions. So that, the received packets on the concentrator were counted during the experiment.

```
Router: SPRPEOCO1

The average value of the operational metric "CPU Utilization"

118 Samples (5-Minute Resolution),

Business Hour Status       Utilization - Average
11/18/2019 06:35 GMT-03:00     23 Percent
11/18/2019 06:40 GMT-03:00     23 Percent
11/18/2019 06:45 GMT-03:00     21 Percent
11/18/2019 06:50 GMT-03:00     23 Percent
11/18/2019 06:55 GMT-03:00     22 Percent
11/18/2019 07:00 GMT-03:00     22 Percent
11/18/2019 07:05 GMT-03:00     21 Percent
11/18/2019 07:10 GMT-03:00     22 Percent
11/18/2019 07:15 GMT-03:00     23 Percent
11/18/2019 07:20 GMT-03:00     22 Percent
11/18/2019 07:25 GMT-03:00     23 Percent
11/18/2019 07:30 GMT-03:00     21 Percent
11/18/2019 07:35 GMT-03:00     22 Percent
11/18/2019 07:40 GMT-03:00     23 Percent
11/18/2019 07:45 GMT-03:00     23 Percent
11/18/2019 07:50 GMT-03:00     23 Percent
11/18/2019 07:55 GMT-03:00     22 Percent
11/18/2019 08:00 GMT-03:00     23 Percent
11/18/2019 08:05 GMT-03:00     21 Percent
11/18/2019 08:10 GMT-03:00     23 Percent
```

Listing 6.1 – CPU Usage by Spectrum.

Considering what was described in Section 5.4, we use performance metrics (see Table 6) for an M/M/1/K queue based on the concepts described in Chapter 2. We have used the equations in order to evaluate the impact of the BFD control packets on the network. To make it possible to use the equations properly, we performed measurements on the device's CPU by using the Spectrum tool. We have found the percentage regarding the CPU utilization before and after the implementation of the BFD solution. In order to be able to properly associate the BFD control packets with the CPU consumption, we have confronted statistical data of the BFD sessions. Where the concentrator device itself records both the number of BFD packets processed in a given period and the increment in CPU usage that occurred within the same time slot. As can be seen by looking in Listing 6.1, the reading process is performed every five minutes to collect the CPU consumption data. The measurement of the mean CPU consumption considers the computational effort to fulfill the routing tasks related to the environment concentration role before any alteration takes place in the network.

Listing 6.1 depicts the searching performed at a time interval of T which coincides

with the time when BFD session statistics were enabled for the 200 communication links. In other words, what we did was to execute the configuration script (see Listing 5.1) on the 200 BUs and on the concentrator router, whilst we enabled the statistical reporting that records the amount of BFD packets received by the hub device (see Figure 38). It is important to highlight that the number of 200 communication links is related to the need to cause a minimally noticeable increase in the CPU consumption of the concentrator device. Figure 37 presents the information extracted from a particular BFD session. In turn, from the same particular BFD session, Figure 38 shows the data regarding the arrival of packets at the same time interval of T, in which the CPU samples were taken. It was possible to establish the total load of BFD packets addressed to the concentrator equipment, during the time interval T, by computing the number of BFD packets received by it from all 200 sessions. Thus, we could consistently consider the difference between the mean CPU consumption of the scenario with the BGP protocol, and the average consumption of the scenario that had the BFD and BGP protocols associated (see Table 9). We can conclude that the $\Delta\bar{U}_i$ value is directly related to the BFD monitoring process deployed in the environment during the experiment.

```
<HUAWEINE40>display bfd session peer-ip 10.206.247.70 verbose
------------------------------------------------------------------------------
Session MIndex : 1346089   (One Hop) State : Up        Name : dyn_12008
------------------------------------------------------------------------------
  Local Discriminator    : 12008            Remote Discriminator   : 2732
  Session Detect Mode    : Asynchronous Mode Without Echo Function
  BFD Bind Type          : Interface(GigabitEthernet6/0/5.819)
  Bind Session Type      : Dynamic
  Bind Peer IP Address   : 10.206.247.70
  NextHop Ip Address     : 10.206.247.70
  Bind Interface         : GigabitEthernet6/0/5.819
  Vpn Instance Name      : REDE4_PRINCIPAL
  FSM Board Id           : 6                TOS-EXP                : 7
  Min Tx Interval (ms)   : 100              Min Rx Interval (ms)   : 100
  Actual Tx Interval (ms): 100              Actual Rx Interval (ms): 100
  Local Detect Multi     : 4                Detect Interval (ms)   : 400
  Echo Passive           : Disable          Acl Number             : -
  Destination Port       : 3784             TTL                    : 255
  Proc Interface Status  : Disable          Process PST            : Disable
  WTR Interval (ms)      : -
  Active Multi           : 4                DSCP                   : -
  Last Local Diagnostic  : No Diagnostic
  Bind Application       : BGP
  Session TX TmrID       : -                Session Detect TmrID   : -
  Session Init TmrID     : -                Session WTR TmrID      : -
  Session Echo Tx TmrID  : -
  PDT Index              : FSM-14050009 | RCV-31D | IF-14050000 | TOKEN-0
  Session Description    : -
------------------------------------------------------------------------------
```

Figura 37 – Display BFD Statistics

```
<HUAWEINE40>display bfd statistics session peer-ip 10.206.247.70
-------------------------------------------------------------------------------
Session MIndex : 1346089   (One Hop) State : Up        Name : dyn_12008
-------------------------------------------------------------------------------
  Session Type                        : Dynamic
  Bind Type                           : IP
  Local/Remote Discriminator          : 12008/2732
  Vpn Instance Name                   : REDE4_PRINCIPAL
  Received Packets                    : 375691
  Send Packets                        : 355229
  Received Bad Packets                : 0
  Send Bad Packets                    : 0
  Down Count                          : 0
  ShortBreak Count                    : 0
  Send Lsp Ping Count                 : 0
  Dynamic Session Delete Count        : 0
  Create Time                         : 2019-11-18 06:34:19-03:00
  Last Down Time                      : -
  Total Time From Last DOWN           : ---D:--H:--M:--S
  Last Up Time                        : 2019-11-18 06:34:32-03:00
  Last Up Lasting Time                : 000D:10H:02M:21S
  Total Time From Create              : 000D:10H:02M:34S
-------------------------------------------------------------------------------
```

Figura 38 – BFD Packets Received

Tabela 9 – CPU Consumption Computation (Before/After)

$\bar{U}_{i-before}$	$\bar{U}_{i-after}$	$\Delta\bar{U}_i$
20.00%	**22.275%**	**2.275%**

Based on the empirical data obtained, we performed calculations based on the theoretical framework, as follows:

To start the evaluation, the first variable collected from the testbed was T, the length of time of the observation period. It is based on the data shown in Figure 38:

$$T = 10h02m21s = 36,141seg$$

Also, based in Figure 38, we determine A_0 as the total number of service requests (i.e., arrivals) to resource in the observation period T. In our case study, A_0 is the total number of packets that arrived at the concentrating device in the period T, considering the 200 sessions of the experiment.

$$A_0 = 200 \times 375,691\, packets = 75,138,200\, packets$$

$A_0 = C_0$ corresponds to the total number of requests completed by the system in the observation period T. We considered that all BFD packets received were processed by the concentrator. This assumption is taken in order to calculate the throughput (X_0).

$$C_0 = A_0 = 75,138,200\, packets$$

We have calculated the concentrator device's throughput for the BFD control packet traffic.

$$X_0 = \frac{C_0}{T} = \frac{75,138,200}{36,141} = 2,079.02\, packets/s$$

By performing these steps, we were able to define the λ_i value, which corresponds to the input rate of the BFD packets.

$$\lambda_i = \frac{A_0}{T} = \frac{75,138,200}{36,141} = 2,079.02\, packets/s$$

By using the λ_i, we calculated the MAT (mean arrive time)

$$MAT = \frac{1}{\lambda_i} = \frac{1}{2,079.02} = 0.00048099\, seconds$$

Based on the $\Delta\bar{U}_i$, obtained in the experiment and tabulated in Table 9, by applying the service demand law, it was possible to determine the value of D_i, which represents the mean service time (MST). MST is the amount of time that a transaction requires from a resource to be processed. In this case study, a transaction corresponds to the processing of one BFD control packet.

$$D_i = \frac{\Delta\bar{U}_i}{X_0} = \frac{0.02275}{2,079.02} = 0.00001094265\, seconds/request$$

As a result, we define the value for MST (mean service time).

$$MST = D_i = 0.00001094265\, seconds/request$$

In this case, as we can see, μ_s, that corresponds to the service rate, is the inverse of the MST:

$$\mu_s = \frac{1}{MST} = \frac{1}{0.00001094265} = 91,385.54\, requests/seconds$$

We calculate the value of K by considering the following items: the M/M/1/K model (defined in Section 5.4), the comprehension about how the BFD protocol (introduced in Section 5.2) and the Forwarding Parameter, required for the properly implementation of the availability solution, work. The threshold K represents the number of BFD packets that will be processed before that the queuing process occurs (see Figure 23). The Forwarding Performance parameter was obtained by looking at the manufacturer's documentation, and it is given in MPPS (Million Packets Per Second).

$$K = 105,000,000\, packets$$

As described in Section 5.4, the equations, and metrics for M/M/1/K systems are defined as a function of the ρ parameter. We used ρ to denote the traffic intensity. It is

possible to calculate the ρ value based on the empirical data (λ_i and μ_s) collected during the experiment. Then, by considering the definitions of λ_i and μ_s, we have

$$\rho = \frac{\lambda_i}{\mu_s} = \frac{2,079.02}{91,385.54} = 0.0227499$$

In order to produce a graphical sensitivity analysis to show the CPU utilization on the NE40 device as a function of the mean arrival time of the BFD control packets, we have used the equation for calculating utilization on M/M/1/K systems. However, an adaptation in the equation has been introduced. This adaption considers the CPU usage before the deployment of the BFD protocol, as the BFD workload is included in the CPU consumption of the baseline architecture. To make it possible to apply this solution for other case studies other than those presented in this dissertation, we considered as a variable the mean CPU consumption ($\bar{U}_{i-before}$) before the deployment of the BFD. In this case study, this consumption is set by the value presented in Table 9.

$$Utilization = \frac{\left(\rho\left(1-\rho^{k}\right)\right)}{(1-\rho^{k+1})} + \bar{U}_{i-before}$$

By using the equation, we drew the graph of CPU usage as a function of the MAT presented in Figure 39. The graph shows the MAT from which there will be an undesirable bottleneck in the equipment due to the control mechanism imposed by the BFD protocol. Another possible inference we may do is that the sample used, although statistically representative (200 communication links), caused small changing in processing (2.275%). It occurs due to the high processing capacity of the concentrator device. The MAT value, in which the bottleneck occurs, is very small. Also, it occurs due to the capacity of the Huawei Ne40 device.

We have collected 20% of average CPU consumption, being it measured before the deployment of the BFD protocol on the network. The evaluation certifies that the solution that offers higher availability, by associating the BFD and BGP protocols, although presenting a processing increase, is not enough to cause a bottleneck on the equipment. By considering from a cost-benefit perspective, this validates the evaluated solution.

$$mat_{bottleneck} = 0.0000127$$

As a next step in the performance evaluation process, we have used the equation for calculating discard rates on M/M/1/K systems for supporting us to produce a graphical sensitivity analysis that shows the queuing rate of the NE40 device as a function of the MAT generated from the BFD control packets. It is important to note that we considered the queuing process as loss, as explained in Section 5.2. This is the rate that an arriving BFD packet does not find the concentrator router idle, and, therefore, it is forced to wait in the queue. By looking at Figure 40, we denoted this rate by *Queuing Rate*.

$$Queuing\ Rate = \frac{\rho^i\,(1-\rho)}{(1-\rho^{k+1})} = 0.00000000753853173$$

Looking at the graph in Figure 40, we can see again that the equipment's capacity, especially, the Forwarding Performance parameter (K), does the packet queuing in a very low pace. What reduces the probability of false positives alerts occurrences significantly. The queuing rate for this experiment is very close to zero. Of course, for a very low MAT, the queuing rate increases, but even considering an actual high load, this rate does not reach relevant values.

After the evaluation, the model was validated by considering different BFD session loads (λ_i) and comparing the CPU measurements by using the SNMP Spectrum tool. In the same way, as described in the experiment, we have changed only the number of sessions, and, as a consequence, a distinct number of BFD packets (λ_i) arriving at the concentrator device on the period T. The values found in the T-paired comparison were very close, as can be seen in Table 10.

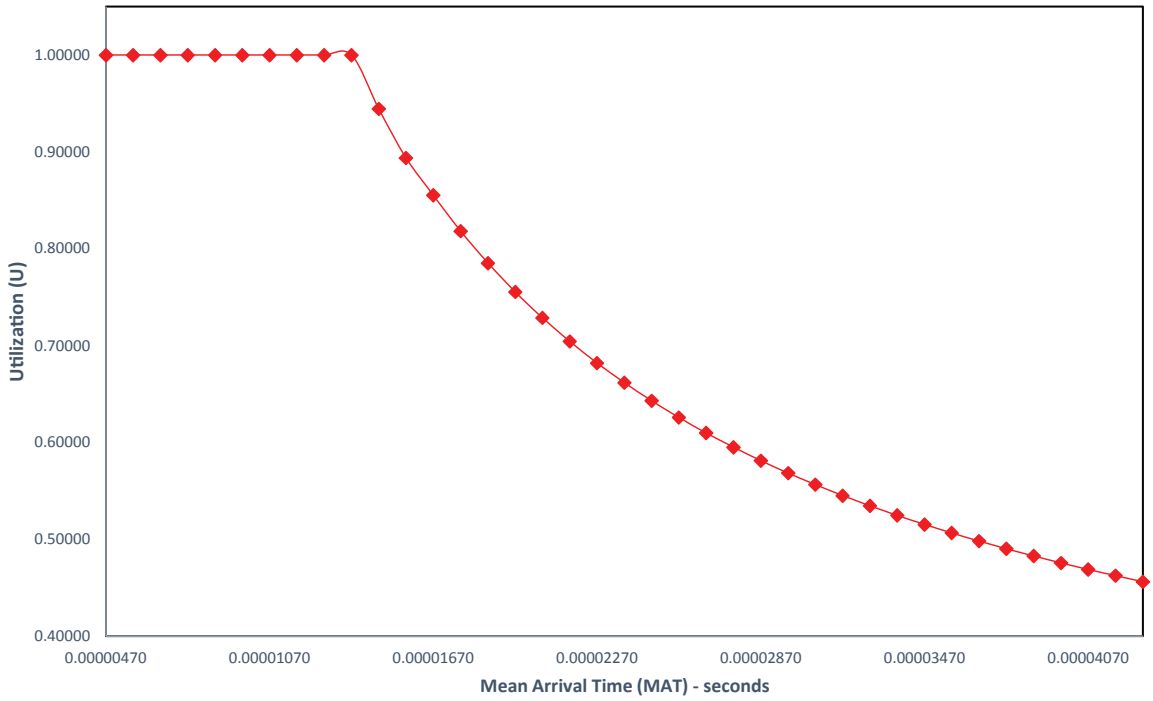

Figura 39 – Utilization as a Function of MAT

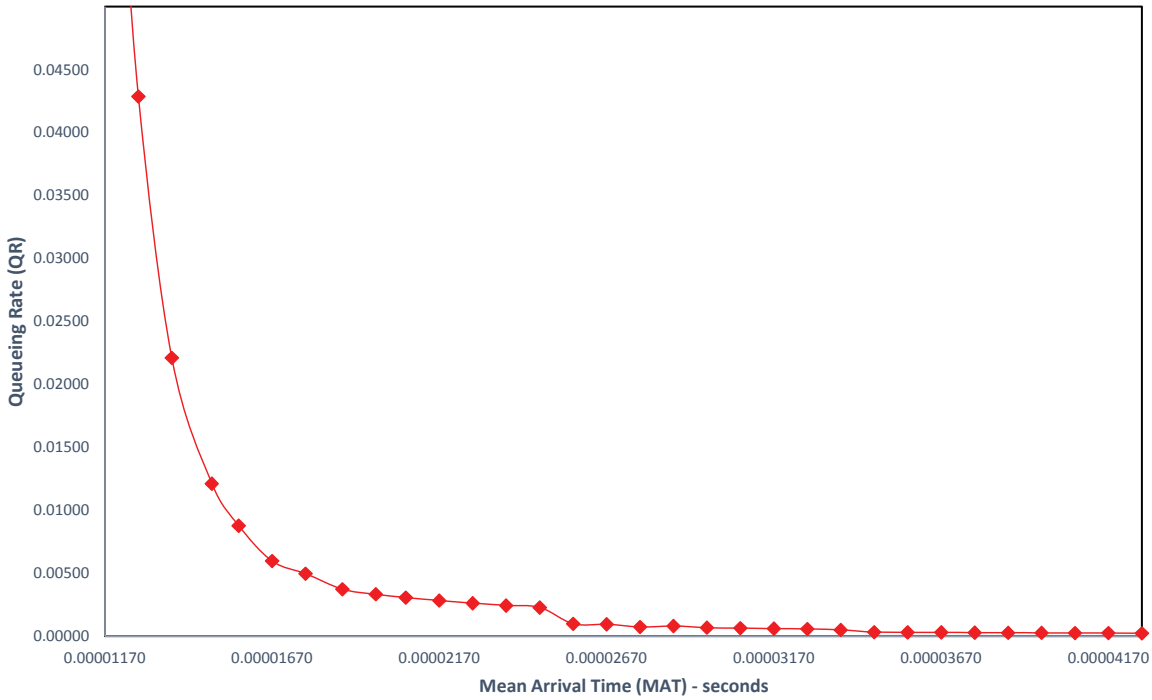

Figura 40 – Queuing Rate as a Function of MAT

Tabela 10 – Model Validation for the Case Study Two

	T-Paired	
	$\bar{U}_{imodel}$	$\bar{U}_{inetwork}$
λ_i $(packets/s)$	**Value**	**Value**
520	20.050%	20.050%
1040	21.140%	21.150%
1559	21.700%	21.750%
2079	22.275%	22.275%

6.3 CASE STUDY THREE: NUMERICAL ANALYSIS

In this last study, the performance evaluation performed by us investigated the correlation between the BFD monitoring packets and CPU consumption. It becomes clear that the greater the number of BFD sessions on the network, the closer the environment approaches to a threshold situation. We present an objective perspective, linking the analysis to the number of business units having the BFD protocol configuration deployed. From a practical point of view, decision-making for the solution that offers higher availability is always made considering the number (N) of BUs involved. Based on the empirical data obtained from experiments, we performed calculations considering the association between the number of BFD packets and number of BUs, as follows.

This case study uses the same value for T (observation period) which was also used in the case study two, as we used the same data obtained as an experiment.

$$T = 10h02m21s = 36,141\ seconds$$

Based on Figure 37, we collected the number of BFD packets arriving at the concentrator device during the period T. By considering that the number of packets arriving at the NE40 router is very close to being the same for each BU during the time interval T, this allows us to create a relationship between N — number of BUs — and A_0 — the total number of packets that have arrived at the concentrating device in period T for a single BU.

$$N = 1\ BU\ \rightarrow A_0 = 375,691\ packets$$

Once the relation was established, we could set other metrics used as a function of N. In the case of throughput, we have:

$$X_0 = \frac{A_0 N}{T} = \frac{375,691N}{36,141} = 10.39N\ packets/s$$

We were also able to set the BFD packet arrival rate λ_i as a function of N.

$$\lambda_i = X_0 = 10.39N\ packets/s$$

By considering the value for λ_i, we calculated the value of MAT (mean arrive time) as a function of N:

$$MAT = \frac{1}{\lambda_i} = \frac{1}{10.39N} = \frac{0.09624}{N}\ seconds$$

As we are taking the experiment of the case study two as reference, we could set the same value for the service rate μ_s.

Therefore, we have:

$$\mu_s = \frac{1}{MST} = \frac{1}{0.00001094265} = 91,385.54\ requests/seconds$$

For the same reasons, the value for K is also valid for the current case study. Therefore, we have:

$$K = 105,000,000\ packets$$

The traffic intensity value for ρ, according to the empirical data (λ_i and μ_s) collected on the experiment, can be calculated as a function of N. Then, by the definitions of λ_i and μ_s, we have:

$$\rho = \frac{\lambda_i}{\mu_s} = \frac{10.39N}{91,385.54} = 0.000113694N$$

We have used the equation for calculating utilization on M/M/1/K systems by performing some adaptations on it, in order to produce a graphical sensitivity analysis. The objective of the analysis was to show the CPU utilization on the Huawei NE40 device as a function of the number of BUs with actives BFD sessions (N). At first, we replaced the variable ρ for its representation as a function of N. At second, considering that the workload generated by the BFD protocol is added to the CPU consumption of the baseline architecture, we did the same adaptation in order to consider the mean consumption ($\bar{U}_{i-before}$) before the deployment of the BFD protocol (see Section 6.2).

$$Utilization = \frac{\left(0.000113694N\left(1-(0.000113694N)^{K}\right)\right)}{\left(1-(0.000113694N)^{K+1}\right)} + \bar{U}_{i-before}$$

By using this equation, we drew the graph presented in Figure 41. That graph correlates CPU usage as a function of the number of BUs (N). It shows N from which there will be an undesirable bottleneck in the equipment due to the control mechanism imposed by the BFD protocol. Another inference is that there exists a region in the graph that represents the ideal number of communication links in which the risk for bottleneck occurrence is low.

It is possible to see the number of links for which the device concentrator's CPU usage reaches its maximum point (N=7,578). It is important to note that given the high packet processing capacity of the Huawei NE40 router, the number of BUs used as a sample (N=200) is far away for generating bottleneck. We can infer the limiting number of communication links, and also it is possible to infer the expected behavior when the equipment is operating on its capacity limit.

$$N_{bottleneck} = 7,578\, business\ units$$

In order to expand the solution with the BFD protocol that offers higher availability, by taking into account good practices of capacity planning, it is necessary to define a maximum number of BUs with this protocol deployed, in which we can ensure that the concentrator device would have the smallest risk to reach its limiting capacity. This must be considered because, by considering the number of BUs, we have defined a way to estimate the usage of the device. However, average values do not eliminate the occurrence of occasional peaks. Thus, in order to avoid occasional performance fluctuations, we have limited the number of BUs to $N = 2,692$, which corresponds to an estimated utilization of 50%.

$$N_{maximum} = 2,692\, business\ units$$

For the corporate network used as a testbed, the maximum estimated number of BUs using the higher availability solution is more than enough for deployment across the entire network. And there is still a reserve for future expansions, as the existing regional hub

has less than 1,000 communication links. Figure 41 highlights the point at which we can infer the mean consumption after deploying this solution with BFD in the entire network.

$$\bar{U}_{After} = 0.31376$$

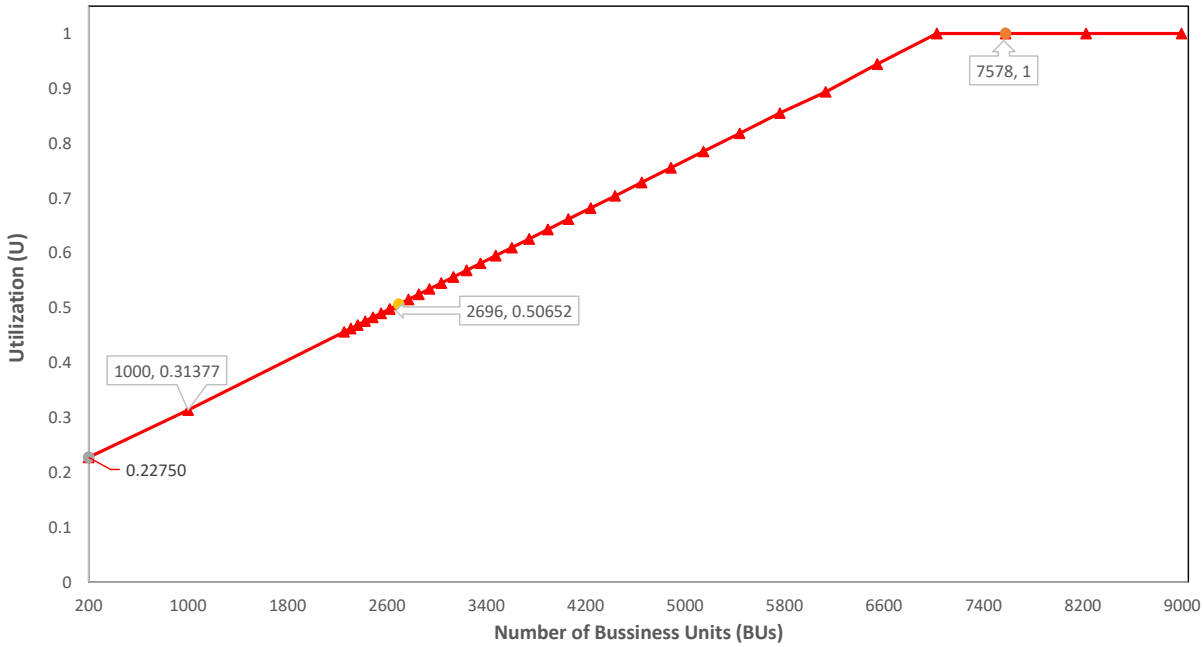

Figura 41 – Utilization as a Function of N

As we have done in the case study two, the proposed model for utilization computation as a function of the number of BUs was also validated. Measurements were taken and compared between the model and the empirical data. Table 11 shows the results.

Tabela 11 – Model Validation for the Case Study Three

	T-Paired	
	$\bar{U}_{imodel}$	$\bar{U}_{inetwork}$
N	**Value**	**Value**
50	20.050%	20.050%
100	21.140%	21.150%
150	21.700%	21.750%
200	22.275%	22.275%

By considering the numerical analysis presented in this section, another particularly relevant point is to correlate the results obtained in the availability evaluation (see Section 6.1) with the utilization collected in the performance evaluation (see Section 6.2). It means that the purpose of this case study is to compare the availability of the entire

network by considering the scenario in which the backup mechanism is supported only by BGP protocol and the inferred scenario in which the entire network would use BFD associated to protocol BGP, as the warm standby redundancy mechanism. In addition, associated with the comparison of availability of the evaluated network, and it being considered as a unique critical system, we demonstrate in Table 12 the concentrating device's CPU utilization in both scenarios, considering the business unit number N equals to 1,000.

Considering that all communication links that make the network up are independent and identical, so all of them have the same failure and repair distribution. Hence, the availability of the entire critical system can be calculated by applying Equation 5.3 presented in Section 5.3. For that aim, we have to define the value for the variable A according to the result obtained in the experiment demonstrated in Table 8. Thus, we have:

1. The critical system is 100% operational when 100% of the links are active. So the K and N parameters of the equation are equal to 1,000.

$$A_{KooN} = \sum_{i=K}^{N} A^i \left(1-A\right)^{N-i} = A_{1000oo1000} = \sum_{i=1000}^{1000} A^{1000} \left(1-A\right)^{1000-1000}$$

2. Looking at the Table 8, we can get the value of A for the before scenario. This value is the availability considered for only one communication link. So, we have inserted the value in variable A as an input to the Equation 5.3, as well as the values of K and N.

$$A_{Before} = A_{1000oo1000} = \sum_{i=1000}^{1000} 0.999893^{1000} \left(1-0.999893\right)^{1000-1000} = 0.898520$$

3. We repeated the calculation using as input to the equation the availability obtained in the experiment for only one communication link in the after scenario.

$$A_{After} = A_{1000oo1000} = \sum_{i=1000}^{1000} 0.999969^{1000} \left(1-0.999969\right)^{1000-1000} = 0.969470$$

.

Table 12 shows the correlation between the gain in availability achieved and the cost in processing generated to handle the workload in order to support the new configuration. It is noticed that the obtained gain in availability in the order of 7% is relevant when considering the entire network as a single critical system. On the other hand, the increase in the concentrating device's CPU consumption (11.3%) does not represent a risk for the occurrence of a bottleneck. This validates the effectiveness of our availability solution.

Tabela 12 – Correlation Between Availability and Performance

	Result (N=1,000)	
Scenario	**Availability**	**Utilization**
Before	0.89852	0.20
After	0.96947	0.31376

7 CONCLUSION

Nowadays, some services being provided around the world need to be constantly available for their customers. These services are classified as critical services. Critical services require maximum availability. Convergent networks have become an indispensable mechanism for making possible to offer these services. Their application requires to apply all possible efforts in order to produce an improvement in services' availability. The design of a convergent network must be oriented so that downtime is as minimum as possible. In this work, we have proposed two models for supporting availability and performance evaluations of convergent networking infrastructures. They are aimed at supporting predictions about convergent architectures' availability by applying different configurations and the related impact for implementing each one on the performance of the failure avoidance mechanism. Our approaches are aimed to support the decision-making process about which configuration regarding the BGP (Border Gateway Protocol) and BFD (Bidirectional Forwarding Detection) protocols must be implemented on a real corporate network in order to reach the desired availability level. In addition, they enable us to carry out a comparative and objective analysis between different solutions for evaluating the trade-off between availability and performance for each of them. We propose a model based on a continuous-time Markov chain(CTMC) for supporting availability evaluations. Also, we propose a Markovian M/M/1/K queuing model for performance evaluations. The former represents a real convergent infrastructure. The latter represents the workload related to the BFD control packets that are processed by the network's concentrating equipment. Besides that, it makes possible to represent the maximum packet processing capacity before queuing, as well as the rate at which queuing occurs. In addition, we have derived a closed-form equation of our CTMC for calculating the availability of large convergent networking infrastructures by considering the number of links. These models were used in a complementary way in order to represent a corporate network, each one been used by considering different perspectives. By using them, it is possible to evaluate availability and performance by considering a star topology with warm standby redundancy mechanisms. A correlation between availability gained by deploying a configuration and the CPU utilization regarding this one on the concentrator devices may be performed. Our approaches receive as input parameter the environment configuration by considering the association or not between BFD and BGP protocols. This parameter affected the environment in such a way that it generated considerable impact on the evaluated metrics. We have considered five performance metrics for supporting the decision-making process. The performance metrics considered are as follows: *Mean Arrival Time* (MAT), *Mean Service Time* (MST), *Utilization* (U), *Queuing Rate* (QR), and *Throughput* (TP). *MAT* corresponds to the mean arrival time of BFD control packets. *MST* is the amount of

time that a BFD control packet requires to be handled by the concentrator equipment. U indicates the percentage of CPU usage on a concentrator device. Maybe, BFD packets can not be processed immediately after they arrive in a concentrator device. Thus, QR is the rate at which BFD packets are queued to be processed after when processing capacity is made available to handle them. Besides that metrics, we have evaluated throughput by considering two perspectives. From the first perspective, it corresponds to the number of requests made by the BFD control packets. Form the second one, it corresponds to the number of requests processed by a central networking device. These metrics were associated with BFD control packets arriving at the concentrating device. A relationship was established between the control BFD packets and CPU consumption on the concentrating device. The failover time was the availability metric investigated. It conceptually is the time required to identify the failure plus the time required to switch to an alternate route upon detection. It was through the reduction of the failover time that we have obtained an increase in availability. This work was able to isolate the CPU consumption caused exclusively by the BFD control mechanism. We also evaluated the capacity for processing BFD packets of central equipment, allowing for inferring what is the expected behavior in the case that the maximum capacity has been reached. Regarding this, it is possible to predict the number of connections recommended for higher availability projects. We propose two methodologies for supporting the application of our approaches. The first methodology is aimed at supporting availability evaluations. The second one is aimed at supporting performance evaluation on the convergent network concentrating equipment. They may be applied separately, that is, it is not a prerequisite that both must be applied for a given situation. By using our methodologies and models, they make possible companies plan how to optimize their network infrastructures with minimal effort. They also let to see whether it is time to upgrade the concentrator devices deployed in the evaluated environment. The equations for calculating availability as a function of the network parameters represent a useful tool for inferring on the stability of the services. We have considered deployments on a platform composed of Huawei equipment. However, our configuration scripts may be adapted in order to be useful when they are applied to different platforms other than Huawei. Our solution can be applied in backbones similar to the one considered by us where critical services are offered, whenever the availability of the environment needs to be increased. Three case studies were performed to evaluate the effectiveness of our approaches. In the first study, we have focused on the availability evaluation. In the second study, we have focused on performance evaluation. In the third study, we carried out a performance evaluation aimed to investigate the correlation between the gain in availability achieved by deploying a configuration and the cost in processing generate by it. Before the deployment, the infrastructure's availability was 89.852% and the CPU utilization on the concentrator device was 20%. After, we have reached an availability correspondent to 96.947% and a CPU utilization of 31.37%. It represents an increase of

7.09% in the availability with a cost of processing correspondent to an increase of 11.37%. Our approaches have been validated successfully. They have proven to be feasible and they highlight the most appropriate scenarios, supporting network architects at design as well as production time.

7.1 CONTRIBUTIONS

Following, we list some contributions of this work:

- **A methodology for supporting availability evaluations of convergent networks.** The methodology for measuring failover detection time with a paired comparison can be used in its entirety or even adapted to the particularities of the evaluated environments.
- **A methodology for supporting performance evaluations of convergent networks.** By using this methodology, it is possible to evaluate the performance of central network devices in the context of convergent networking infrastructures.
- **Development of a CTMC model for calculating the availability of warm standby critical systems by considering large convergent networking infrastructures.** This CTMC makes it possible to predict the availability of a convergent network by considering specific network parameters and configurations. Through its application, it is possible to perform sensitivity analysis considering the trade-off between availability and failover time. This sensitivity analysis allows for adjusting the arrival time range of the BFD control packets via configuration in order to optimize the availability or to adequate to the level of demand of the critical network supported applications.
- **A closed-form equation derived from our CTMC for calculating the availability of warm standby critical systems by considering large convergent networking infrastructures.** In this work, the number of communication links — that is, the network size — is considered in one of our case studies. This equation is important because it lets us evaluate metrics for very large network infrastructure quickly. It demonstrates that the gains obtained by using our solution get even bigger, the bigger the network.
- **Our Markovian M/M/1/K queue model allows for performing performance evaluations on concentrator devices of convergent networking infrastructures.** Our model makes it possible to recognize whether the core network equipment has the capacity to perform monitoring tasks derived from BFD deployment, as well as to suggest an eventual upgrading. In addition, our model makes it possible to infer the behavior of the equipment when its capacity limit has been

reached. By using it, it is possible to perform sensitivity analysis considering the trade-off between the mean arrival time of BFD packets and the CPU utilization on the networking device. This sensitivity analysis can be applied to adjust the monitoring of the workload generated by the packets by considering the characteristics of the hardware deployed on the environment. So, it is possible to infer the ideal and maximum number of connections. Also, it is possible to perform sensitivity analysis considering the trade-off between the mean arrival time and the queuing rate of the BFD packets. The analysis lets to visualize the packet queuing rate, directly linked to the probability that the equipment will reach its capacity limit. This rate is important because it represents the occurrence of false positives.

- **A investigation of the state of the art regarding the use of the network protocols BFD and BGP in the context considered by our study.** For building this work, a deep packet-level analysis of the BGP and BFD protocols was performed, which makes this work useful as a basis for motivating further work with these protocols.

7.2 LIMITATIONS AND FUTURE WORKS

Following, we list some limitations and possible future works.

- **Our approaches consider deployment on a unique vendor platform.** Our approaches consider deployment on a platform composed of Huawei equipment. It means that an analyst responsible for a network infrastructure intended to apply our solutions to his infrastructure may need to adapt and test them when handling different devices. This restriction may lead to researches aimed to study the interoperability of our solution by performing testing on equipment of different manufacturers other than Huawei.
- **We have used simulation tools in which settings can be applied.** Our work is intended to be applied in production backbones on real networks, due to the intrinsic characteristics of the measured and evaluated parameters. Thus, it is not reasonable to perform these measurements in a simulation environment when the aforementioned option is in hand.
- **Performing long test on real convergent networking infrastructures**. Another limitation of our work was that one. As we have performing tests on real networking infrastructure, they were not long enough. Another issue regarding this one is about the possibility to force the concentrator to a bottleneck condition, as we have handled here powerful equipment with capabilities to handle a very large number of requests per unit of time. For this reason, we have adopted inference provided by our M/M/1/K model.

We understand that our work may be evolved to address some related topics to the one of this research and that deserve further investigation. Among the topics that justify more researches, it is possible to highlight the following:

- A research could investigate how BFD packets relate to firewalls or other policy processes. This research may analyze whether there exist failures in the BFD control mechanisms due to blockages. Through this work may be understood how packets controlling mechanism such as policers, traffic shapers, priority queuing, etc., can impact the solution with BFD.
- Another way to evolve this work is by investigating the behavior of BFD sessions in high traffic scenarios. It is possible to conduct a study associating traffic with the occurrence of false positives.
- Also, our work may evolve by evaluating how it works when associating BFD with IPsec and GRE security tunnels. Researchers may investigate whether BFD packets can cause instability in the operation of the aforementioned tunnels.

REFERÊNCIAS

AVIZIENIS, A.; LAPRIE, J.-C.; RANDELL, B.; LANDWEHR, C. Basic concepts and taxonomy of dependable and secure computing. *Dependable and Secure Computing, IEEE Transactions on*, IEEE, v. 1, n. 1, p. 11–33, 2004.

BERKOWITZ, H.; DAVIES, E.; HARES, S.; KRISHNASWAMY, P.; LEPP, M. Terminology for benchmarking bgp device convergence in the control plane. *Internet Requests for Comments, RFC Editor, RFC 4098*, 2005.

BISTOUNI, F.; JAHANSHAHI, M. Analyzing the reliability of shuffle-exchange networks using reliability block diagrams. *Reliability Engineering & System Safety*, Elsevier, v. 132, p. 97–106, 2014.

BOLCH, G.; GREINER, S.; MEER, H. de; TRIVEDI, K. S. *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*. New York, NY, USA: Wiley-Interscience, 2006. ISBN 0-471-56525-3.

CA TECHNOLOGIES. *Spectrum r9.4*. 2019. CA Technologies.

Caesar, M.; Rexford, J. Bgp routing policies in isp networks. *IEEE Network*, v. 19, n. 6, p. 5–11, 2005.

CASSANDRAS, C. *Introduction to discrete event systems*. New York, N.Y: Springer Science+Business Media, 2008. ISBN 978-0387333328. Available at: <https://www.springer.com/gp/book/9780387333328>.

CHUNG, K. L. Book review: Stochastic processes. *Bulletin of the American Mathematical Society*, American Mathematical Society (AMS), v. 60, n. 2, p. 190–202, mar 1954. Available at: <https://doi.org/10.1090/s0002-9904-1954-09801-4>.

DANTAS, J.; MATOS, R.; ARAUJO, J.; MACIEL, P. An availability model for eucalyptus platform: An analysis of warm-standy replication mechanism. In: IEEE. *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. [S.l.], 2012. p. 1664–1669.

DANTAS, J.; MATOS, R.; ARAUJO, J.; MACIEL, P. Models for dependability analysis of cloud computing architectures for eucalyptus platform. *International Transactions on Systems Science and Applications*, v. 8, n. 5, p. 13–25, 2012.

DANTAS, J.; MATOS, R.; ARAUJO, J.; MACIEL, P. Eucalyptus-based private clouds: availability modeling and comparison to the cost of a public cloud. *Computing*, Springer, v. 97, n. 11, p. 1121–1140, 2015.

DENNING, P. J.; BUZEN, J. P. The operational analysis of queueing network models. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 10, n. 3, p. 225–261, 1978. Available at: <https://doi.org/10.1145/356733.356735>.

DOYLE, J.; CARROLL, J. D. *Routing tcp/ip*. 2. ed. [S.l.]: Cisco Press, 2006. 160–162 p. ISBN 1-587505-202-4.

EFRON, B. Bootstrap methods: Another look at the jackknife. *Ann. Statist.*, The Institute of Mathematical Statistics, v. 7, n. 1, p. 1–26, 01 1979. Available at: <https://doi.org/10.1214/aos/1176344552>.

FLOYD, S.; HANDLEY, M.; PADHYE, J.; WIDMER, J. Tcp friendly rate control (tfrc): Protocol specification. RFC 5348 (Proposed Standard), 2008.

GARTNER RESEARCH. *Network Downtime*. 2014. <http://blogs.gartner.com/andrew-lerner/2014/07/16/the-cost-of-downtime>. Accessed: 2019-11-29.

GHANNAMI, A.; SHAO, C. Efficient fast recovery mechanism in software-defined networks: multipath routing approach. In: IEEE. *2016 11th International Conference for Internet Technology and Secured Transactions (ICITST)*. [S.l.], 2016. p. 432–435.

GUIMARAES, A. P.; OLIVEIRA, H. M. N.; BARROS, R.; MACIEL, P. R. Availability analysis of redundant computer networks: a strategy based on reliability importance. In: IEEE. *2011 IEEE 3rd International Conference on Communication Software and Networks*. [S.l.], 2011. p. 328–332.

HARES, S.; LEE, D.; VARLASHKIN, I. Internet engineering task force (ietf) r. papneja request for comments: 7747 huawei technologies category: Informational b. parise. 2016.

HAVERKORT, B. R. Markovian models for performance and dependability evaluation. In: _____. *Lectures on Formal Methods and Performance Analysis: First EEF/Euro Summer School on Trends in Computer Science Bergen Dal, The Netherlands, July 3–7, 2000 Revised Lectures*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001. p. 38–83. ISBN 978-3-540-44667-5. Available at: <https://doi.org/10.1007/3-540-44667-2_2>.

HUAWEI TECHNOLOGIES CO. *Enterprise Network Simulation Platform V100R003C00SPC100 (ENSP)*. 2019. HUAWEI. Available in: https://support.huawei.com/enterprise/br/network-management/ensp.

HUAWEI TECHNOLOGIES CO. *NE40E Series Product Documentation V800R011C00*. 2019. HUAWEI. Available in: https://support.huawei.com.

JONES, C.; RANDELL, B. *Dependable pervasive systems*. [S.l.]: University of Newcastle upon Tyne, Computing Science, 2004.

JR, F. J. M. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, Taylor & Francis, v. 46, n. 253, p. 68–78, 1951.

KATZ, D.; WARD, D. Rfc 5880 bidirectional forwarding detection (bfd). *Internet Engineering Task Force, Request For Comments (Standards Track)*, 2010.

KIM, H. S.; KIM, S. I. A bgp session takeover method for high availability. In: IEEE. *2015 Seventh International Conference on Ubiquitous and Future Networks*. [S.l.], 2015. p. 153–158.

KITSUWAN, N.; SRIKOON, C.; NOPPANAKEEPONG, S. The evaluation of optimal output buffer size and shared buffer size in partially shared buffering for packet switching architecture. In: IEEE. *Proceedings. Student Conference on Research and Development, 2003. SCORED 2003.* [S.l.], 2003. p. 123–126.

KLEINROCK, L. *Theory, Volume 1, Queueing Systems*. New York, NY, USA: Wiley-Interscience, 1975. ISBN 0471491101.

KUO, W.; ZUO, M. J. *Optimal reliability modeling: principles and applications*. [S.l.]: John Wiley & Sons, 2003. ISBN 0-471-39761-X.

MASRUROH, S. U.; FIADE, A.; IMAN, M. F. et al. Performance evaluation of routing protocol ripv2, ospf, eigrp with bgp. In: IEEE. *2017 International Conference on Innovative and Creative Information Technology (ICITech)*. [S.l.], 2017. p. 1–7.

MEHRANNIA, H.; PAKGOHAR, A. Using easy fit software for goodness-of-fit test and data generation. *International Journal of Mathematical Archive EISSN 2229-5046*, v. 5, n. 1, 2014.

MENASCE, D. A.; ALMEIDA, V. A.; DOWDY, L. W.; DOWDY, L. *Performance by design: computer capacity planning by example*. [S.l.]: Prentice Hall Professional, 2004. ISBN 0-13-090673-5.

MENDONÇA, R.; OLIVEIRA, J. M.; LINS, R. D. *Redes MPLS: fundamentos e aplicações*. [S.l.]: Brasport, 2012. ISBN 978-85-7452-539-6.

MONDAL, A.; MISRA, S.; MAITY, I. Buffer size evaluation of openflow systems in software-defined networks. *IEEE Systems Journal*, IEEE, v. 13, n. 2, p. 1359–1366, 2018.

O'CONNOR, P.; KLEYNER, A. *Practical reliability engineering*. [S.l.]: John Wiley & Sons, 2012.

O'CONNOR, P.; KLEYNER, A. *Practical reliability engineering*. 5. ed. [S.l.]: John Wiley & Sons, 2012. ISBN 978-0-470-97981-5.

REKHTER, Y. Rfc 4271: A border gateway protocol 4 (bgp-4). *http://www.ietf.org/rfc/rfc4271.txt*, IETF, 2006.

SATHAYE, A.; RAMANI, S.; TRIVEDI, K. S. Availability models in practice. In: *Proc. of Intl. Workshop on Fault-Tolerant Control and Computing (FTCC-1)*. [S.l.: s.n.], 2000.

SERMPEZIS, P.; DIMITROPOULOS, X. Can sdn accelerate bgp convergence?—a performance analysis of inter-domain routing centralization. In: IEEE. *2017 IFIP Networking Conference (IFIP Networking) and Workshops*. [S.l.], 2017. p. 1–9.

SILVA, B.; CALLOU, G.; TAVARES, E.; MACIEL, P.; FIGUEIREDO, J.; SOUSA, E.; ARAUJO, C.; MAGNANI, F.; NEVES, F. Astro: An integrated environment for dependability and sustainability evaluation. *Sustainable Computing: Informatics and Systems*, v. 3, n. 1, p. 1 – 17, 2013. ISSN 2210-5379.

SILVA, B.; MATOS, R.; CALLOU, G.; FIGUEIREDO, J.; OLIVEIRA, D.; FERREIRA, J.; DANTAS, J.; LOBO, A.; ALVES, V.; MACIEL, P. Mercury: An integrated environment for performance and dependability evaluation of general systems. In: *Proceedings of Industrial Track at 45th Dependable Systems and Networks Conference, DSN*. [S.l.: s.n.], 2015.

SIQUEIRA, D.; PINHEIRO, T.; DANTAS, J.; MACIEL, P. Dependability evaluation in a convergent network service using bgp and bfd protocols. In: *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. [S.l.: s.n.], 2019. p. 2378–2383. ISSN 1062-922X.

TANYINGYONG, V.; RATHORE, M. S.; HIDELL, M.; SJÖDIN, P. Resilient communication through multihoming for remote healthcare applications. In: IEEE. *2013 IEEE Global Communications Conference (GLOBECOM)*. [S.l.], 2013. p. 1335–1341.

TRIOLA, M.; GILMARTIN, W.; SOLBERG, E.; ABUEISAAD, T. *Statdisk 13.0.1 for Elementary Statistics*. [S.l.]: Upper Saddle River, NJ: Pearson Education, 2005.

TRIVEDI, K. S. *Probability & statistics with reliability, queuing and computer science applications*. [S.l.: s.n.], 2008. ISBN 81-203-0508-6.

VOHRA, Q.; CHEN, E. *BGP support for four-octet AS number space*. [S.l.], 2007.